

ITADINFO 2024

Secondo Convegno Italiano
sulla Didattica dell'Informatica

a cura di
Sara Capecchi
Barbara Catania

ITADINFO 2024

**Secondo Convegno Italiano
sulla Didattica dell'Informatica**

a cura di
Sara Capecchi
Barbara Catania

Atti del convegno
ITADINFO 2024
Secondo Convegno Italiano sulla Didattica dell'Informatica
18-20 ottobre 2024
Laboratorio CINI Informatica e Scuola,
Università degli Studi di Genova
e APS Programma il Futuro

Comitato scientifico del convegno

Comitato scientifico

Sara Capecchi, *Università di Torino (Presidente)*

Agnese Addone, *Università di Salerno*

Claudia Canali, *Università di Modena e Reggio Emilia*

Antonella Carbonaro, *Università di Bologna*

Giorgio Delzanno, *Università di Genova*

Valentina Di Noi, *Università di Torino*

Luca Forlizzi, *Università dell'Aquila*

Ilenia Fronza, *Università di Bolzano*

Enrica Gentile, *Università di Bari*

Giovanna Guerrini, *Università di Genova*

Michael Lodi, *Università di Bologna*

Violetta Lonati, *Università di Milano*

Sabrina Mantaci, *Università di Palermo*

Simone Martini, *Università di Bologna*

Giovanna Melideo, *Università dell'Aquila*

Claudio Mirolo, *Università di Udine*

Mattia Monga, *Università di Milano*

Alberto Montresor, *Università di Trento*

Anna Morpurgo, *Università di Milano*

Enrico Nardelli, *Università di Roma "Tor Vergata"*

Maria Angela Pellegrino, *Università di Salerno*

Veronica Rossano, *Università di Bari*

Marcello Sarini, *Università di Milano – Bicocca*

Ugo Solitro, *Università di Verona*

Tullio Vardanega, *Università di Padova*

Atti del convegno

Sara Capecchi, *Università di Torino*

Barbara Catania, *Università di Genova*

Comitato organizzatore del convegno

Presidenza

Enrico Nardelli, *Università di Roma “Tor Vergata” e direttore del Laboratorio CINI*

Giorgio Delzanno, *Università di Genova*

Giovanna Guerrini, *Università di Genova*

Sponsorizzazioni

Tullio Vardanega, *Università di Padova*

Relazioni istituzionali

Enrico Nardelli, *Università di Roma “Tor Vergata”*

Ufficio stampa e comunicazione

Reputation Agency

Segreteria organizzativa e sito web

Francesco Lacchia, *APS Programma il Futuro*

Organizzazione locale

Barbara Catania, *Università di Genova*

Giorgio Delzanno, *Università di Genova*

Lorenzo Gerini, *Università di Genova*

Giovanna Guerrini, *Università di Genova*

Davide Ponzini, *Università di Genova*

Daniele Traversaro, *Università di Genova*

Convegno sponsorizzato da:

Sponsor espositori



Sponsor



Convegno patrocinato da:



I contributi qui pubblicati sono stati revisionati dal Comitato Scientifico del Convegno con single blind peer-review.

Impaginazione, editing e revisione del presente volume a cura di Sara Capecchi e Barbara Catania.

© 2024 GUP

I contenuti del presente volume sono pubblicati con la licenza
Creative commons 4.0 International Attribution-NonCommercial-ShareAlike.



Alcuni diritti sono riservati

e-ISBN (pdf): 978-88-3618-287-9

Pubblicato a ottobre 2024

Realizzazione Editoriale
GENOVA UNIVERSITY PRESS
Via Balbi 5, 16126 Genova
Tel. 010 20951558
e-mail: gup@unige.it
<https://gup.unige.it>

INDICE

PREFAZIONE	11
<i>Sara Capecchi, Barbara Catania, Giorgio Delzanno, Giovanna Guerrini, Mattia Monga, Enrico Nardelli, Maria Angela Pellegrino, Veronica Rossano e Tullio Vardanega</i>	
VIDEO DIVULGATIVI	
Un concetto dell'informatica in un video di 2 minuti: grafi	13
<i>Martina Palazzolo</i>	
RACCONTI DI ESPERIENZE SUL CAMPO E RELAZIONI SCIENTIFICHE PER TUTTE LE ETÀ	
Conoscenza dell'informatica e professionalità docente: spunti dalla letteratura	16
<i>Agnese Del Zozzo, Luca Lamanna, Violetta Lonati e Alberto Montresor</i>	
Coding, pensiero computazionale e informatica: quale evoluzione possibile?	20
<i>Michael Lodi e Simone Martini</i>	
Laboratorio e.DO Learning Center: una collaborazione fruttuosa tra scuola e aziende	25
<i>Paola Greco, Marcello Missiroli, Salvatore Parlato, Zakaria Touati e Leonardo Ciocca</i>	
Il coding nell'educazione inclusiva	33
<i>Lorenzo Notaro, Roberta Cavalli, Enrica Gentile e Veronica Rossano</i>	
Una metodologia per l'integrazione del coding nell'educazione scolastica	40
<i>Alessandro Marcelletti, Sara Pettinari, Barbara Re e Beatrice Strappa</i>	
RACCONTI DI ESPERIENZE SUL CAMPO E RELAZIONI SCIENTIFICHE PER L'INFANZIA, LA PRIMARIA E LA SECONDARIA DI PRIMO GRADO	
Coding, robotica educativa e competenze visuo-spaziali nella scuola dell'infanzia	46
<i>Francesca Craviotto</i>	
Percorsi per l'introduzione degli automi nel contesto didattico scolastico	52
<i>Luca Forlizzi, Giovanna Melideo e Alessia Sebastiano</i>	
“Leonardo a scuola con noi”: coding e STEAM con Arduino in una scuola secondaria di primo grado	58
<i>Alessandro Guffanti</i>	
Desiderio di sfida nell'apprendimento del cifrario di Cesare	64
<i>Daniele Scala</i>	
RACCONTI DI ESPERIENZE SUL CAMPO E RELAZIONI SCIENTIFICHE PER LA SECONDARIA DI SECONDO GRADO E L'UNIVERSITÀ	
Insegnare l'informatica all'intelligenza artificiale: learning by teaching attraverso tecniche di prompt engineering	70
<i>Giovanni Pedroncelli</i>	

Lezioni “AI-aumentate”: come trasformare una lezione in una performance multi-attore <i>Michele Iacobellis</i>	78
Esercizi di programmazione non banali con ChatGPT nelle superiori: un esperimento <i>Davide Ponzini, Giovanna Guerrini, Flaviano Cipollone e Giorgio Delzanno</i>	85
Coding in ambienti di Extended Reality <i>Lorenzo Gerini, Manuela Chessa, Giorgio Delzanno, Giovanna Guerrini e Fabio Solari</i>	92
Esperienze di apprendimento non-convenzionali: insegnare informatica uscendo dagli schemi <i>Ilenia Fronza, Gennaro Iaccarino e Veronica Rossano</i>	97
Formazione universitaria per l’insegnamento dell’informatica e casi studio con Arduino <i>Alessandra De Vitis e Guglielmo Abbruzzese</i>	103
LabOrinto: progettazione e realizzazione di stage di orientamento verso l’informatica <i>Giorgio Delzanno, Giovanna Guerrini e Davide Ponzini</i>	109
“Laboratorio di informatica” ai Licei Faes di Milano, parte seconda: il livello dei dati <i>Fabio Sartori, Elisabetta Zanichelli e Miriam Nobile</i>	113
Due progetti PNRR di approfondimento informatico: VideoGame Academy e Cybersecurity <i>Salvatore Imparato, Silvia Stranieri, Vincenzo Schiano Di Cola, Giulia Sellitto e Filomena Ferrucci</i>	117
Come nasce un videogioco: aspetti di progettazione multimediale audio-video ed interfacce <i>Adriano Mancini, Leonardo Gabrielli, Maura Mengoni, Gionata Massi e Federico Robuffo</i>	124
Insegnare informatica con Unreal Engine: l’esperienza di Ragazze Digitali <i>Diletta Maniglia, Lisa Fregni e Claudia Canali</i>	130
Diagrammi di flusso a supporto dell’apprendimento della programmazione <i>Gabriele Pozzan e Tullio Vardanega</i>	135
Insegnare SQL a chi non ha mai programmato: analisi delle misconcezioni <i>Daniele Traversaro</i>	141
Indagine su approccio cognitivo e risultati conseguiti dagli studenti delle scuole superiori in problemi di programmazione concorrente <i>Emanuele Scapin e Nicola Dalla Pozza</i>	145
Le analogie visive nel processo di insegnamento e apprendimento di sistemi e reti <i>Virginia Beato e Luca Pinet</i>	152
LABORATORI PER LA PRIMARIA	
Mostri, pesci e virus: un laboratorio per la comprensione dell’apprendimento automatico <i>Matteo Baldoni, Cristina Baroglio, Monica Bucciarelli, Sara Capecchi, Elena Gandolfi, Francesco Iani, Elisa Marengo e Roberto Micalizio</i>	158
Ilognatter Angesid: un laboratorio sulla rappresentazione delle informazioni tramite codifiche digitali <i>Carlo Bellettini, Marta Crippa, Sofia Gavanelli, Luca Lamanna, Violetta Lonati, Lucia Mellini, Mattia Monga e Anna Morpurgo</i>	159

Le grandi idee della crittografia alla scuola primaria <i>Maria Cristina Carrisi, Michael Lodi e Simone Martini</i>	160
LABORATORI PER LA SECONDARIA DI PRIMO GRADO	
Trovare il lupo: cassetta degli attrezzi per insegnare il debugging a scuola <i>Gabriele Pozzan e Tullio Vardanega</i>	162
Laboratorio di intelligenza artificiale con la programmazione a blocchi e la robotica educativa <i>Ersilia Pagano</i>	163
Laboratorio formativo per insegnanti “Welcome to NIMROD” <i>Claudio Mirolò</i>	164
LABORATORI PER LA SECONDARIA DI SECONDO GRADO	
Ascoltiamo gli errori più comuni nella programmazione concorrente con Sonic Pi <i>Giorgio Delzanno, Giovanna Guerrini e Daniele Traversaro</i>	166
Pyth – Un ponte tra Scratch e Python <i>Sara Fiori, Ben North e Glenn Strong</i>	167
Introduzione alla realizzazione di ambienti tridimensionali in realtà virtuale e aumentata <i>Manuela Chessa, Giorgio Delzanno e Lorenzo Gerini</i>	168
Ordinamento distribuito nel mondo reale <i>Matteo Dell’Amico</i>	169
ALTRI CONTRIBUTI	
Reti unplugged: l’apprendimento cinestetico per imparare i concetti di incapsulamento, instradamento e indirizzamento attraverso la gamification <i>Virginia Beato e Luca Pinet</i>	171
First Lego League: la robotica educativa per avvicinare gli studenti all’informatica <i>Giorgio Gasparini, Maria Elena Lai, Elisabetta Pezzati e Marco Sorrentino</i>	177
Creazione di videogiochi VR per la didattica dell’informatica nella scuola superiore <i>Alessandro Forgiarini, Sonia Guerci, Eva Venturini e Fabio Buttussi</i>	181
Progetto App me Up: creatività, autonomia e peer tutoring <i>Nicoletta Rossi</i>	186
“L’intelligenza artificiale in aiuto allo studio”: un breve corso Moodle <i>Giuliana Barberis</i>	190
Intelligenza artificiale generativa nella didattica: un’analisi dei dati raccolti nelle attività formative <i>Andrea Canesi</i>	198
Living syllabus per l’informatica per la cittadinanza digitale e per le digital humanities: un approccio di sviluppo collaborativo <i>Giovanni Adorni</i>	204

Coding e realtà virtuale: un'esperienza laboratoriale interattiva nella scuola primaria <i>Manuela Chessa, Giorgio Delzanno, Lorenzo Gerini, Giovanna Guerrini, Marco Barbagelata e Barbara Gaggero</i>	208
Un curriculum verticale per comprendere l'IA: dalle reti neurali alla citizen science <i>Laura Cesaro e Giovanni Dodero</i>	213
PepperCare un umanoide a servizio della didattica <i>Luca Spandre, Laura Camplani e Celestina Zandonai</i>	221
“Siete connessi?” Educazione alla cittadinanza digitale consapevole <i>Claudia Incerti</i>	226
#IA: GiochiAmo e #Scenari Futuri con VR, AR e IoT: gare on line per imparare giocando <i>Luca Basteris e Maria Cristina Daperno</i>	232
Virtual coding tour <i>Giovanni Silvestro e Marta Sanz Manzanedo</i>	239
Un primo contatto con gli algoritmi di intelligenza artificiale: verso un'opinione informata <i>Matteo Moro, Nicoletta Noceti, Francesca Odone e Silvia Villa</i>	246
Esperienze di apprendimento attivo e collaborativo nel progetto “Secoli Bui? Il Medioevo tra reale e virtuale” <i>Maria Concetta Brocato, Nicola Zangrando, Biagio Tomasetig, Letizia Donnini, Marzia Toso e Fabio Buttussi</i>	250
Il coding in classe <i>Giovanna Caracciolo e Giacomo Cassalia</i>	257
Co(de)spaces: a long voyage. Insegnare il pensiero computazionale con la metodologia CLIL <i>Chiara Storace</i>	263

PREFAZIONE

Il convegno italiano sulla Didattica dell'Informatica (ITADINFO) si propone di creare sinergia tra docenti di scuola di ogni ordine e grado e il mondo della ricerca per discutere metodi e tecniche per l'insegnamento dell'Informatica come scienza nella scuola. ITADINFO è giunto alla sua seconda edizione.

La [prima edizione](#) di ITADINFO, svoltasi a Bari a ottobre scorso, ha riscosso notevole successo in termini di contributi e partecipazione. Le sessioni di presentazione di esperienze, progetti e risultati di ricerca, e i laboratori formativi hanno visto nascere discussioni partecipate e costruttive. A margine di ciò, è emersa da parte del corpo insegnante la richiesta di formazione mirata sull'insegnamento dell'Informatica, e di luoghi di scambio e condivisione su approcci e metodi. La domanda di formazione sui temi dell'Informatica, pur se tardiva, è imponente. Nei prossimi due anni le scuole sono chiamate a organizzare circa 40.000 percorsi su tematiche che hanno l'Informatica come disciplina fondante: dirigenti e docenti dovranno mettere in atto tali percorsi scegliendo tra una grande varietà di possibilità e proposte.

In questo particolare contesto i contributi selezionati per questi atti costituiscono un prezioso catalogo di attività ideate da esperti del settore, sperimentate sul campo e valutate dalla comunità scolastica e accademica.

ITADINFO 2024 ha previsto sessioni dedicate alla discussione di esperienze sul campo realizzate da insegnanti, alla presentazione di risultati di ricerca, alla formazione laboratoriale interattiva e una tavola rotonda conclusiva sulle sfide e opportunità poste dall'Intelligenza Artificiale generativa nella Didattica dell'Informatica.

*Sara Capecchi
Barbara Catania
Giorgio Delzanno
Giovanna Guerrini
Mattia Monga
Enrico Nardelli
Maria Angela Pellegrino
Veronica Rossano
Tullio Vardanega*

VIDEO DIVULGATIVI

Un concetto dell'informatica in un video di 2 minuti GRAFI

Martina Palazzolo

Istituto Comprensivo Ilaria Alpi di Milano

martina.palazzolo@icilariaalpi.edu.it

Quest'anno insieme a due colleghe, Rita Signorini e Daniela Viganò, abbiamo realizzato un breve video [1] su un concetto dell'informatica da divulgare ai ragazzi. Il titolo del video, GRAFI, ne sintetizza l'argomento.

Abbiamo dedicato la prima parte del video a definire cos'è un grafo, ovvero una rappresentazione della realtà estremamente sintetica nella quale un punto, detto vertice o nodo, viene utilizzato per rappresentare un qualunque oggetto o elemento della realtà che si vuole prendere in considerazione. Un punto può rappresentare oggetti anche molto differenti tra loro come una casa oppure un animale. Due punti vengono uniti tramite una linea, detta arco o spigolo, se intercorre tra loro una particolare relazione. Nell'esempio del video i punti rappresentano due case e l'arco la strada che le collega. Nel secondo esempio i punti vengono utilizzati per rappresentare animali e gli archi indicano la relazione preda-predatore. In questo caso gli archi sono orientati tramite una freccia che dal predatore arriva alla preda.

Abbiamo intenzionalmente scelto questi due esempi (due case e la strada che le collega e la relazione preda – predatore che intercorre tra animali) perché sintetizzano una delle caratteristiche fondamentali dei grafi: essi permettono di modellare aspetti della realtà molto distanti tra loro e che non sembrano avere nulla in comune attraverso una stessa struttura dati. La teoria dei grafi in ambito informatico permette quindi di interrogare aspetti della realtà molto differenti tra loro sfruttando gli stessi algoritmi.

Nella seconda parte del video utilizziamo un esempio tratto dai giochi Bebras dell'informatica, *Animali nel recinto* dell'edizione kilo e mega del 2019, che riprende la relazione preda-predatore tra animali.

La scelta dei grafi come argomento per il video nasce dalle osservazioni fatte in classe durante le attività curriculari di laboratorio scientifico che hanno sfruttato come materiale di lavoro i giochi dell'informatica Bebras [2]. Tali giochi hanno l'obiettivo di divulgare, attraverso una gara non competitiva, i principi dell'informatica e il pensiero computazionale [3, 4, 5].

Nelle varie edizioni dei giochi una media di 2 quesiti sul totale di quelli proposti (circa 12) è dedicata a problemi che riguardano i grafi. I ragazzi hanno dimostrato un grande interesse per tali quesiti che hanno spesso scelto per i loro progetti durante l'attività denominata *Allenatori Bebras* [6]. L'attività *Allenatori Bebras* richiede ai ragazzi di scegliere e lavorare su un quesito Bebras di interesse per realizzare un gioco cartaceo da usare per allenare i compagni più giovani alla gara dell'anno successivo. L'anno scorso, per esempio ben tre gruppi distinti hanno scelto di lavorare partendo da uno stesso quesito, *La tana di Mary*, che invita i ragazzi a ragionare sfruttando le informazioni che possono trarre da una rappresentazione a grafo dei dati [7].

I quesiti dei giochi Bebras insieme all'attività *Allenatori Bebras* permettono ai ragazzi di avvicinarsi in modo concreto e con rielaborazione personale a diversi aspetti del mondo dell'informatica. Con il video vorremmo far soffermare i ragazzi sul concetto di grafo in quanto fornisce uno strumento per modellare dati e informazioni utile ai ragazzi per ragionare e trovare strategie risolutive in vari ambiti e problemi.

Bibliografia

- [1] <https://drive.google.com/file/d/1ZdxzQXZdGBXs8EgUEMQEIPYc846msaqI/view?usp=sharing>
- [2] www.bebas.it
- [3] A. Calcagni, V. Lonati, D. Malchiodi, M. Monga, A. Morpurgo. Promoting Computational Thinking Skills: Would You Use this Bebras Task? *ISSEP 2017* pagg. 102 – 113 LNCS 10696, Springer 2019
- [4] V. Lonati, D. Malchiodi, M. Monga, A. Morpurgo. Bebras as a Teaching Resource: Classifying the Tasks Corpus Using Computational Thinking Skills. *ItiCSE '17*, July 3 – 5, 2017, Bologna, Italy
- [5] V. Dagiené and G. Stupuriene. Informatics education based on solving attractive tasks through a contest. *IFIP-KEYCIT 2014*, p. 51-62, 2014
- [6] C. Bellettini, V. Lonati, M. Monga, A. Morpurgo, M. Palazzolo. Situated Learning with Bebras Tasklets. *ISSEP 2019* pagg. 225 – 239 LNCS 11913, Springer 2019
- [7] M. Palazzolo, R. Signorini, D. Viganò. I giochi bebras dell'informatica nel curriculum della scuola del primo ciclo. Atti del convegno ITADINFO Bari 2023
<https://www.itadinfo.it/2023/attidelconvegno/ATTI-ITADINFO-2023.pdf>

**RACCONTI DI ESPERIENZE SUL
CAMPO E RELAZIONI SCIENTIFICHE
PER TUTTE LE ETÀ**

Conoscenza dell'informatica e professionalità docente: spunti dalla letteratura

Agnese Del Zozzo¹, Luca Lamanna², Violetta Lonati³, and Alberto Montresor⁴

¹ Università di Trento

`agnese.delzozzo@unitn.it`

² Università degli Studi di Milano

`luca.lamanna@unimi.it`

³ Università degli Studi di Milano

`lonati@di.unimi.it`

⁴ Università di Trento

`alberto.montresor@unitn.it`

Abstract

La formazione dei docenti è un elemento chiave per la diffusione di una cultura informatica nelle scuole, soprattutto nel contesto italiano del primo ciclo d'istruzione, in cui l'informatica non è prevista come materia curricolare distinta e i docenti non hanno in genere una preparazione specifica in questa materia. Per progettare efficacemente iniziative e percorsi di sviluppo professionale per insegnanti, è innanzitutto importante poter rilevare quale conoscenza gli insegnanti abbiano, sia dei contenuti e delle pratiche dell'informatica come disciplina scientifica, sia di come insegnarla. In questo contributo presentiamo un lavoro di analisi sistematica della letteratura di ricerca sulla didattica dell'informatica, volto a individuare strumenti e approcci utilizzati per rilevare appunto le conoscenze informatiche degli insegnanti. Presentiamo brevemente la metodologia utilizzata e i primi risultati trovati, illustrandoli con alcuni esempi.

1 Introduzione

L'insegnamento dell'informatica si è recentemente affermato in molti Paesi, fin dai primi livelli di istruzione, e sta attirando attenzione crescente da parte di educatori, enti formativi e decisori politici. Emerge la necessità di integrare l'insegnamento dell'informatica nel curriculum scolastico ma, allo stesso tempo, ci si scontra con una carenza di insegnanti qualificati e, quindi, con l'urgente bisogno di formarli [12].

Per sviluppare proposte adeguate allo sviluppo professionale è essenziale partire da una valutazione diagnostica iniziale delle conoscenze informatiche pregresse degli insegnanti da formare, partendo dall'assunto socio-costruttivista che ciascuno costruisce la propria conoscenza partendo da ciò che già sa e capisce [1]. Tale valutazione è particolarmente complessa quando si considerano i livelli scolastici inferiori, dove la familiarità degli insegnanti con l'informatica è generalmente piuttosto bassa. Ad esempio, in Italia, per diventare insegnante di scuola primaria è necessario laurearsi in Scienze della Formazione Primaria tramite percorsi in cui, nella maggior parte dei casi, non vengono offerti corsi specifici sull'informatica o sul suo insegnamento. Quindi, al di là di corsi di sviluppo professionale isolati, la maggior parte degli insegnanti di primaria in Italia non ha ricevuto alcuna formazione formale né in informatica né nei metodi di insegnamento dell'informatica [7]. Anche a livello di scuola secondaria superiore, nonostante la presenza di una specifica classe di insegnamento, i titoli che abilitano all'insegnamento di materie informatiche sono piuttosto variegati e non sempre adeguati [4].

Il concetto di *Pedagogical Content Knowledge* (PCK), proposto da Shulman [13], propone una prospettiva articolata in cui la professionalità docente si manifesta non soltanto tramite la

conoscenza disciplinare (“cosa insegnare”) e quella pedagogico-didattica (“come insegnare”), ma attraverso una conoscenza specifica, che riguarda “come insegnare quella cosa”. A partire dal lavoro di Shulman, sono stati proposti in letteratura diverse concettualizzazioni della conoscenza degli insegnanti, anche in riferimento a specifiche materie. In relazione all’insegnamento dell’informatica si è diffuso il modello chiamato *Technological PCK* (TPACK) [8], che tuttavia si concentra soprattutto sul rapporto tra conoscenza pedagogica ed uso delle tecnologie e non tanto sulla didattica dell’informatica.

Basandosi sul concetto di PCK, Ball et al. hanno esplorato la natura della conoscenza disciplinare volta all’insegnamento nel caso della matematica [2]. Gli autori distinguono in particolare tra “conoscenza *della* disciplina” - che si riferisce alla conoscenza di concetti e procedure corretti, principi e significati che li supportano, e la connessione tra idee disciplinari - e “conoscenza *sulla* disciplina” - che si riferisce alla visione sulla natura della disciplina stessa. Applicando tale distinzione all’informatica, ci siamo concentrati in questo lavoro sulla “conoscenza dell’informatica”, ovvero sulla conoscenza disciplinare di base degli insegnanti, nel contesto del loro ruolo professionale.

La sfida da affrontare, quindi, è: su cosa ci si può basare per indagare la “conoscenza dell’informatica” degli insegnanti dei diversi gradi scolastici? Dal momento che gli insegnanti spesso partecipano attivamente e svolgono ruoli di rilievo in attività di ricerca (ad esempio, come esperti intervistati, *informants* in gruppi di discussione, istruttori in interventi didattici), risulta interessante identificare quali degli strumenti già utilizzati in questi contesti possano essere impiegati per rilevare le loro conoscenze informatiche.

Abbiamo dunque scelto di affrontare la questione conducendo un lavoro di analisi sistematica della letteratura, nell’ottica di individuare quali siano gli elementi e strumenti già a disposizione della comunità. Nella prossima sezione descriveremo brevemente la metodologia usata per condurre l’analisi e ne presenteremo, con degli esempi, i primi risultati emersi.

Benché i lavori che trattano esplicitamente questo tema risultino essere in numero ridotto, dall’analisi si possono ricavare sia risorse da utilizzare operativamente nell’ambito della formazione docenti sia spunti metodologici per lo sviluppo di nuovi studi (come, ad esempio, è stato fatto nel caso della TPACK in [15]). Riteniamo inoltre che questo contributo possa fornire ai docenti degli spunti di riflessione e strumenti attraverso i quali è possibile procedere con un processo di autoanalisi della propria pratica didattica da diversi punti di vista.

2 Metodologia e cenni di risultati

Il nostro lavoro di analisi della letteratura si è focalizzato, oltre che su articoli di ricerca, sull’analisi di resoconti di esperienze il cui oggetto di ricerca fossero gli insegnanti e la loro conoscenza (sia teorica che posta in uso nelle attività didattiche). Sfruttando i principali database scientifici di pubblicazioni in ambito informatico (p.e., *ACM Digital Library*), abbiamo innanzitutto definito una *query* di ricerca e formulato dei criteri per la selezione delle pubblicazioni da analizzare. Non essendo il nostro focus di interesse già identificato da una terminologia standard nella letteratura in didattica dell’informatica, né avendo una struttura codificata a priori, non è stato possibile formulare una *query* concisa. Si è invece costruita pezzotta-pezzotta, come verrà dettagliato nella presentazione, una *query* ampia ma che centrasse i lavori di ricerca interessanti per il nostro studio. All’insieme totale di articoli ottenuto ($n = 2774$) abbiamo poi applicato alcuni criteri di inclusione/esclusione, che hanno consentito di individuare un sottoinsieme di articoli particolarmente rilevanti, da analizzare con maggior dettaglio.

Il numero di tali articoli risulta decisamente ridotto, rispetto al numero di articoli totali restituiti dalla *query*; un esempio è dato da [9] in cui si analizzano gli effetti (sulla conoscenza

informatica, l'auto-efficacia e gli atteggiamenti nei confronti della disciplina) di un corso di formazione per docenti di scuola primaria. In particolare, per valutare la conoscenza informatica degli insegnanti vengono usati due strumenti validati (in origine progettati per la valutazione degli apprendimenti degli studenti): il *Nebraska Assessment of Computing Knowledge* [10] che si concentra su concetti base (es: selezione, funzioni, ordinamento) e il *CTCAST* sul pensiero computazionale [11].

Nonostante il numero ridotto di articoli che si occupano specificatamente del tema di nostro interesse, dall'analisi della letteratura esistente è possibile individuare risorse operative che possono essere sfruttate nell'ambito della formazione docenti o nello sviluppo di nuove direzioni di ricerca.

Ad esempio, in [16] gli autori mettono a punto uno strumento di valutazione per rivelare la PCK di futuri insegnanti di informatica alla scuola superiore nell'ambito della programmazione. Tale strumento, che gli autori mettono interamente a disposizione del lettore, è composto da una serie di vignette/descrizioni di comportamenti di studenti progettati in modo da esemplificare le più comuni misconcezioni relative alla programmazione. Agli insegnanti viene proposto di commentare/prendere posizione rispetto alla situazione proposta. In tal senso, l'esemplificazione di situazioni reali da commentare rappresenta quindi uno strumento metodologico di ausilio sia in contesto didattico, sia di formazione, sia di ricerca.

In altri lavori vengono proposti degli strumenti metodologici utili a rivelare conoscenze e convinzioni rispetto a un certo contenuto/situazione. Ad esempio, in [6] vengono condotte interviste semi-strutturate e in [18] si propongono task in cui si chiede agli insegnanti di immaginare una situazione specifica nella loro classe di informatica e di disegnarla, raffigurando l'insegnante, gli studenti, l'ambiente e altri elementi rilevanti. Altri lavori, come [3, 17, 14, 5], pur non avendo la conoscenza degli insegnanti come argomento principale, contengono considerazioni pertinenti il tema e citano strumenti o metodi che possono rivelarsi utili nel suo studio.

In generale, la ricerca condotta permette di delineare una panoramica degli strumenti metodologici a disposizione di ricercatori e formatori per lavorare nell'ambito della conoscenza informatica degli insegnanti e, allo stesso tempo, definendo uno spaccato di quello che è stato studiato fino ad ora in questo ambito, si ricavano spunti di riflessione fruibili da tutti gli attori coinvolti nel processo (formatori, ricercatori e insegnanti stessi) assieme ad ulteriori, nuove, direzioni di ricerca.

References

- [1] D. Ausubel, J. Novak, and H. Hanesian. *Educational Psychology: A Cognitive View*. Holt, Rinehart and Winston, 1978.
- [2] D. L. Ball, M. H. Thames, and G. Phelps. Content knowledge for teaching: What makes it special? *Journal of Teacher Education*, 59(5):389–407, 2008.
- [3] S. Basu, D. Rutstein, C. Tate, A. Rachmatullah, and H. Yang. Standards-aligned instructional supports to promote computer science teachers' pedagogical content knowledge. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1, SIGCSE '22*, page 404–410, Providence, RI, 2022. ACM.
- [4] C. Bellettini, V. Lonati, D. Malchiodi, M. Monga, A. Morpurgo, M. Torelli, and L. Zecca. Informatics education in italian secondary schools. *ACM Trans. Comput. Educ.*, 14(2), jun 2014.
- [5] R. Faherty, K. Quille, R. Vivian, M. M. McGill, B. A. Becker, and K. Nolan. Comparing programming self-esteem of upper secondary school teachers to cs1 students. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, ITiCSE '21*, page 554–560, Online, Germany, 2021. Association for Computing Machinery.

- [6] M. M. McGill, L. Thigpen, and A. Mabie. Emerging practices for integrating computer science into existing k-5 subjects in the united states. In *Proceedings of the 18th WiPSCE Conference on Primary and Secondary Computing Education Research*, WiPSCE '23, Cambridge, UK, 2023. ACM.
- [7] L. Michael. *Introducing Computational Thinking in K-12 Education: Historical, Epistemological, Pedagogical, Cognitive, and Affective Aspects*. PhD thesis, University of Bologna, Italy, 2020.
- [8] P. Mishra and M. Koehler. Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6):1017–1054, 2006.
- [9] G. Nugent, K. Chen, L.-K. Soh, D. Choi, G. Trainin, and W. Smith. Developing k-8 computer science teachers' content knowledge, self-efficacy, and attitudes through evidence-based professional development. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, ITiCSE '22, page 540–546, Dublin, Ireland, 2022. Association for Computing Machinery.
- [10] M. S. Peteranetz and A. D. Albano. Development and evaluation of the nebraska assessment of computing knowledge. *Frontiers in Computer Science*, 2:11, 2020.
- [11] M. S. Peteranetz, P. M. Morrow, and L.-K. Soh. Development and validation of the computational thinking concepts and skills test. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 926–932, Portland, OR, 2020. ACM.
- [12] Royal Society. After the reboot: Computing education in uk schools, 2017. Accessed: 2024-05-07.
- [13] L. S. Shulman. Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(2):4–31, 1986.
- [14] E. Skuratowicz, M. Vanderberg, E. E. Hung, G. Krause, D. Bradley, and J. P. Wilson. I felt like we were actually going somewhere: Adapting summer professional development for elementary teachers to a virtual experience during covid-19. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, page 739–745, Online, 2021. ACM.
- [15] J. Voogt, P. Fisser, N. Pareja Roblin, J. Tondeur, and J. Van Braak. Technological pedagogical content knowledge – a review of the literature. *Journal of Computer Assisted Learning*, 29(2):109–121, 2013.
- [16] A. Yadav and M. Berges. Computer science pedagogical content knowledge: Characterizing teacher performance. *Transactions on Computing Education*, 19(3):1–24, 2019.
- [17] S. Yeni, N. Grgurina, F. Hermans, J. Tolboom, and E. Barendsen. Exploring teachers' pck for computational thinking in context. In *Proceedings of the 16th Workshop in Primary and Secondary Computing Education*, WiPSCE '21, Online, 2021. ACM.
- [18] P. Zaugg and A. Gumpert. Imagine yourself as a media and computer science teacher. In *Proceedings of the 17th Workshop in Primary and Secondary Computing Education*, WiPSCE '22, Morschach, Switzerland, 2022. ACM.

Coding, pensiero computazionale e informatica: quale evoluzione possibile?

Michael Lodi and Simone Martini

Dipartimento di Informatica - Scienza e Ingegneria
Alma Mater Studiorum - Università di Bologna
Bologna, Italia
Laboratorio CINI Informatica e Scuola
{michael.lodi, simone.martini}@unibo.it

Abstract

Analizziamo, dal punto di vista dell'informatica e della didattica dell'informatica, l'espressione "coding, pensiero computazionale e informatica" utilizzata in recenti documenti ministeriali e più in generale per parlare dell'introduzione dell'informatica a scuola.

1 Introduzione

"Coding, pensiero computazionale e informatica: quale evoluzione possibile?", così recita il titolo dell'ultimo paragrafo delle "Linee Guida STEM" [1] emanate a ottobre 2023, che mirano, anche nel contesto del PNRR, a integrare nell'offerta formativa delle scuole azioni per potenziare le competenze matematico-scientifico-tecnologiche e digitali.

Negli ultimi anni queste espressioni sono entrate a far parte, oltre che del linguaggio della scuola e della formazione docenti, anche di alcune leggi e decreti. Tuttavia sono spesso usate senza essere definite e possono essere fraintese dai non addetti ai lavori. Inoltre, esistono interpretazioni diverse anche tra gli esperti.

In questo articolo, forniamo il nostro punto di vista da informatici e ricercatori in didattica dell'informatica, indicando come naturale evoluzione il pieno riconoscimento dell'informatica come disciplina autonoma e la sua introduzione nei curricula scolastici, con la stessa dignità delle altre discipline scientifiche.

Le nostre riflessioni potranno essere utili nel valutare quanto un'attività didattica possa essere considerata di "coding, pensiero computazionale e informatica".

2 Informatica

Chiedendo a gruppi di studenti di diverse età che cos'è la biologia, moltissimi risponderanno correttamente che è la scienza che studia la vita. Lo stesso con l'astronomia: la scienza che studia i corpi celesti e il cosmo. Per l'informatica, qualcuno dirà che studia i computer. Nessuno dirà che è una scienza. Ma, come suggerito dall'etimologia (*informat(ion)* e (*automat)ique*) e semplificando la definizione di Denning [9], potremmo dire che l'informatica è la scienza che studia l'elaborazione (descrizione, memorizzazione, trasmissione...) automatica dell'informazione. Si tratta certo di una scienza, e di una disciplina, giovane, ma riconosciuta e autonoma (di informatica troviamo dipartimenti universitari, corsi di laurea, materie scolastiche, titoli di professori universitari e docenti).

3 Coding

Nel mondo delle professioni informatiche, *coding* viene sempre più usato come sinonimo di *programmazione* (es. si trovano annunci di lavoro che parlano di *coders* e *coding languages* invece di *programmers* e *programming languages*).

Nell'ingegneria del software *coding* indica solo una delle varie fasi dello sviluppo software (insieme ad analisi dei requisiti, design, test, ...): la più noiosa e meccanica, quella in cui si "pigiano" materialmente i tasti sulla tastiera per scrivere codice e quindi implementare il programma.

Il termine viene usato, più o meno formalmente, anche in altre aree dell'informatica: nel senso di "codifica" dell'informazione (più propriamente *encoding*) o cifratura (più propriamente *encrypting*).

Tuttavia, di recente è emersa una accezione didattica: *coding* è la "descrizione più giocosa e non intimidatoria della programmazione per principianti" [18, traduzione nostra]. In Italia, in particolare, si è diffuso l'uso secondo cui "il coding (in quest'ambito) è l'applicazione intuitiva di principi, metodi e strumenti di programmazione al fine di sviluppare il pensiero computazionale" [5], spesso inteso come metodologia didattica trasversale.

Questa recente accezione didattica spiega anche espressioni che possono apparire ossimoriche, come "coding unplugged". Nella didattica dell'informatica si parla di attività unplugged come di "attività cinesastiche e divertenti per insegnare i concetti di informatica senza computer" [6, 7, nostra traduzione]), cioè che fanno uso di materiali facilmente reperibili in classe o di giochi in giardino. Tradizionalmente, queste attività non si concentrano sulla programmazione, ma su altri concetti informatici quali algoritmi notevoli, strutture dati, codifica (*encoding*) dell'informazione, e così via. Se si mantiene l'accezione informatica (e non quella didattica) di *coding* come sottoinsieme della – o equivalente alla – programmazione, non è pensabile che sia svolto *unplugged*. L'espressione "coding unplugged" si riferisce dunque in realtà ad attività unplugged per introdurre concetti propedeutici alla programmazione, quali introduzione agli algoritmi, strutture di controllo espresse in pseudocodice, e così via.

L'espressione "programmazione informatica (coding)" è anche presente nella legge 159/19 e nel Decreto Interministeriale 1612 del 28/10/21, che l'hanno resa parte della formazione degli insegnanti di ogni ordine, grado e disciplina. Nella legge 233/21 (PNRR), art. 24-bis, si indica inoltre che la "programmazione informatica (coding)" dovrà essere introdotta nell'ambito degli insegnamenti già esistenti nelle scuole di ogni ordine e grado, anche tramite l'aggiornamento di Indicazioni Nazionali e Linee Guida. Sebbene la formulazione suggerisca che si stia usando *coding* come sinonimo di programmazione informatica, questo non viene mai esplicitato. Ma leggendo il dibattito che ha portato a diverse mozioni approvate l'11 marzo 2019 [2], che hanno impegnato il governo proprio a emanare le leggi di cui sopra, si può chiaramente dedurre l'uso di *coding* con accezione di metodologia didattica trasversale.

Le linee guida STEM utilizzano "programmazione informatica (coding)" quando si riferiscono ai documenti precedentemente citati, introducono l'espressione *coding unplugged* già discussa, oppure usano inscindibilmente l'espressione "coding, pensiero computazionale e informatica".

Tuttavia, l'Accademia della Crusca sconsiglia l'uso di questa terminologia per supportare l'introduzione dell'informatica a scuola. Discutendo della traduzione di *coding* suggerisce che

Codifica risulterebbe quindi un traduttore più legittimo. In realtà, se tale traduttore è adatto in ambito informatico, non sembra possibile applicarlo all'insegnamento della materia scolastica. Quando si parla di *coding* a scuola, infatti, si fa riferimento alle attività di introduzione all'informatica. [...] In ambito scolastico, sarebbe dunque più adeguato [...] favorire un sostantivo italiano che è sicuramente noto, trasparente e facilmente applicabile alle diverse situazioni: l'insegnamento dell'*informatica*. [11, p. 100, corsivo originale]

e poi conclude che

sembra più opportuno seguire la strada intrapresa dagli istituti superiori ed estendere tale proposta alle scuole di ogni ordine e grado; si potrebbe quindi chiamare la materia con una delle seguenti denominazioni: *principi di informatica*, *basi di informatica* o *fondamenti di informatica*. [11, p. 101, corsivo originale]

4 Pensiero Computazionale

Come già sottolineato, l'espressione *pensiero computazionale* è utilizzata negli ultimi anni in abbinamento a *coding* per parlare dell'introduzione dell'informatica nella scuola. Sebbene l'espressione sia stata utilizzata già da Papert nel 1980 (per una prospettiva storica si vedano [16, 17]), il pensiero computazionale è stato definito da Wing prima informalmente nel 2006 come “pensare come un informatico per risolvere problemi” [19] e poi come “il processo mentale coinvolto nella formulazione dei problemi e delle loro soluzioni, di modo che tali soluzioni siano rappresentate in una forma che possa essere eseguita in modo effettivo da un agente che processa informazioni” [20, traduzione nostra].

Ancora una volta l'accademia della Crusca conferma questa interpretazione:

Potremmo quindi considerare il *pensiero computazionale* come una modalità di pensiero che coinvolge particolari facoltà cognitive umane e che prevede uno specifico approccio alla formulazione e risoluzione di problemi complessi, utile alla comprensione della società contemporanea, consentendo di analizzare ed elaborare dati con metodi e strumenti che si acquisiscono con lo studio dell'informatica. [11, p. 95, corsivo originale]

L'espressione è già presente (non definita) nella legge 107/15 (La “Buona Scuola”), e nel D.M. 741/17 sull'Esame di Stato conclusivo del primo ciclo di istruzione. La definizione presente nel documento “Indicazioni Nazionali e Nuovi scenari” [3], che va ad integrare le Indicazioni Nazionali per il curriculum della scuola dell'infanzia e del primo ciclo di istruzione, definisce *pensiero computazionale* come “un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici pianificando una strategia”. Il documento omette il collegamento diretto con la disciplina scientifica sottostante (i metodi e gli strumenti sono specifici *dell'informatica*) e colloca il pensiero computazionale tra le competenze trasversali alle altre discipline. In tempi recenti inoltre l'interpretazione restrittiva del termine *computazionale* come “legato al calcolo” [13, pp. 2-3] e l'inclusione del *computational thinking* tra le competenze di matematica valutate nell'OCSE PISA [4] ha avvicinato il concetto alla matematica, allontanandolo dall'informatica.

In altre discipline è normale parlare di “pensiero disciplinare” (es. *pensiero matematico*, *ragionamento matematico*, *problem solving matematico*). Come riconosciuto da Li e colleghi [14], i “pensieri disciplinari” contengono sia elementi specifici della disciplina sia elementi più trasversali.

Purtroppo notiamo [15] la tendenza di insegnanti e policymaker a focalizzarsi solo sugli aspetti trasversali (come creatività, collaborazione, perseveranza e tolleranza per l'ambiguità) o su aspetti più specifici ma letti in chiave di problem solving generale (scomposizione di problemi, astrazione, approcci iterativi...). Sebbene è ovvio che questi aspetti, facenti parte delle complesse abilità di problem solving umane, siano condivisi tra diverse discipline, non dobbiamo dimenticare che l'informatica, come ogni altra disciplina, apporta contributi specifici che istanziano queste skill (es. l'astrazione è molto diversa in informatica, in matematica e in fisica [10, p. 21]) e formano il suo modo di pensare, il “pensare informatico”.

Uno tra tutti: il problem solving di cui parla Wing è il cosiddetto problem solving *computazionale* (da intendersi non come “legato al calcolo” ma “legato all'elaborazione automatica dell'informazione” – *problem solving informatico*). Uno degli elementi caratterizzanti è la presenza di un interprete (esterno, automatico) con un insieme limitato di istruzioni che comprende e che può eseguire effettivamente, per risolvere problemi espressi in un linguaggio non ambiguo [12, 8].

5 Evoluzione possibile

Da quanto detto dovrebbe risultare evidente la nostra posizione: esiste la disciplina dell'informatica, fondamentale per comprendere ed essere attori protagonisti nella società digitale in cui viviamo, e quindi è questa disciplina che dovrebbe essere imparata e insegnata a scuola, a tutti i livelli. Lo studio dell'informatica (così come fanno le altre discipline) permette di acquisire un particolare modo di pensare disciplinare, il “pensiero computazionale”, che permette di risolvere problemi di elaborazione automatica dell'informazione. Il linguaggio della scienza informatica è la programmazione (volgarmente detta coding), e per questo – sebbene non sia l'unico aspetto – introdurla è fondamentale per lo studio di tale disciplina.

Chiamare “coding” o “informatica” attività che non includano, insieme alle cruciali componenti di competenze trasversali, anche gli aspetti disciplinari fondamentali – quale quello di interprete – danneggia la battaglia per il riconoscimento dell'informatica come disciplina da insegnare a scuola e preclude agli studenti la possibilità di impararne i principi fondamentali.

Fondi

Il lavoro di M. Lodi è supportato dallo Spoke 1 “FutureHPC & BigData” del Centro Nazionale di Ricerca in “High Performance Computing, Big Data and Quantum Computing” (ICSC) finanziato da MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S (M4C2-19) - Next Generation EU (NGEU).

Il lavoro di S. Martini è parzialmente finanziato da: (a) SERICS (PE00000014) under the Italian MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU; (b) Italian PRIN 2020 2022BP7K3 Learning informatics; (c) INdAM GNSAGA.

Bibliografia

- [1] URL: <https://www.miur.gov.it/documents/20182/0/Linee+guida+STEM.pdf/2aa0b11f-7609-66ac-3fd8-2c6a03c80f77?version=1.0&t=1698173043586> (visitato il giorno 13/07/2024).
- [2] URL: <https://www.cs.unibo.it/~michael.lodi2/mozioneencoding-annotata.pdf> (visitato il giorno 13/07/2024).
- [3] URL: <https://www.miur.gov.it/documents/20182/0/Indicazioni+nazionali+e+nuovi+scenari/> (visitato il giorno 13/07/2024).
- [4] URL: <https://pisa2022-maths.oecd.org/ca/index.html#Interpret> (visitato il giorno 13/07/2024).
- [5] Alessandro Bogliolo. *Un modello per il coding a scuola*. 2016. URL: <https://codemooc.org/un-modello-per-il-coding-a-scuola/> (visitato il giorno 13/07/2024).
- [6] Tim Bell e Michael Lodi. «Constructing Computational Thinking Without Using Computers». In: *Constructivist Foundations* 14.3 (2019), pp. 342–351. URL: <https://constructivist.info/14/3/342.bell>.
- [7] Tim Bell et al. «Computer Science Unplugged: school students doing real computing without computers». English. In: *New Zealand Journal of Applied Computing and Information Technology* 13.1 (2009), pp. 20–29. ISSN: 1174-0175.

- [8] Carlo Bellettini et al. «To Be Or Not To Be . . . An Algorithm: The Notion According to Students and Teachers». In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 102–108. ISBN: 9798400704239. DOI: [10.1145/3626252.3630950](https://doi.org/10.1145/3626252.3630950). URL: <https://doi.org/10.1145/3626252.3630950>.
- [9] Peter J. Denning et al. «Computing as a discipline». In: *Communications of the ACM* 32.1 (feb. 1989), pp. 9–23. DOI: [10.1145/63238.63239](https://doi.org/10.1145/63238.63239).
- [10] Andrea A. diSessa. «Computational Literacy and “The Big Picture” Concerning Computers in Mathematics Education». In: *Mathematical Thinking and Learning* 20.1 (2018), pp. 3–31. DOI: [10.1080/10986065.2018.1403544](https://doi.org/10.1080/10986065.2018.1403544). URL: <https://doi.org/10.1080/10986065.2018.1403544>.
- [11] Lucia Francalanci. «Coding». In: *Italiano digitale* 12.1 (mar. 2020), pp. 95–101. ISSN: 2532-9006. DOI: [10.35948/2532-9006/2020.3311](https://doi.org/10.35948/2532-9006/2020.3311). URL: <http://dx.doi.org/10.35948/2532-9006/2020.3311>.
- [12] Robert R. Korfhage. «Algorithm». In: *Encyclopedia of Computer Science*. GBR: John Wiley e Sons Ltd., 2003, pp. 36–38. ISBN: 0470864125.
- [13] Yeping Li et al. «Computational Thinking Is More about Thinking than Computing». In: *Journal for STEM Education Research* 3.1 (apr. 2020), pp. 1–18. ISSN: 2520-8713. DOI: [10.1007/s41979-020-00030-2](https://doi.org/10.1007/s41979-020-00030-2). URL: <http://dx.doi.org/10.1007/s41979-020-00030-2>.
- [14] Yeping Li et al. «On Thinking and STEM Education». In: *Journal for STEM Education Research* 2.1 (feb. 2019), pp. 1–13. ISSN: 2520-8713. DOI: [10.1007/s41979-019-00014-x](https://doi.org/10.1007/s41979-019-00014-x). URL: <http://dx.doi.org/10.1007/s41979-019-00014-x>.
- [15] Michael Lodi. «Informatical Thinking». In: *OLYMPIADS IN INFORMATICS* (dic. 2020), pp. 113–132. ISSN: 1822-7732. DOI: [10.15388/ioi.2020.09](https://doi.org/10.15388/ioi.2020.09). URL: <http://dx.doi.org/10.15388/ioi.2020.09>.
- [16] Michael Lodi. «Pensiero Computazionale: dalle “scuole di samba della computazione” ai CoderDojo». In: *Atti del convegno DIDAMATICA 2018*. In Italian. Cesena, Italy: AICA, apr. 2018. URL: https://www.aicanet.it/documents/10776/2101882/didamatica2018_paper_57.pdf.
- [17] Michael Lodi e Simone Martini. «Computational Thinking, Between Papert and Wing». In: *Science & Education* 30.4 (apr. 2021), pp. 883–908. ISSN: 1573-1901. DOI: [10.1007/s11191-021-00202-5](https://doi.org/10.1007/s11191-021-00202-5). URL: <http://dx.doi.org/10.1007/s11191-021-00202-5>.
- [18] Kiki Prottzman. *Coding vs. Programming - Battle of the Terms!* 2015. URL: https://www.huffpost.com/entry/coding-vs-programming-bat_b_7042816 (visitato il giorno 13/07/2024).
- [19] Jeannette M. Wing. «Computational thinking». In: *Communications of the ACM* 49.3 (mar. 2006), p. 33. DOI: [10.1145/1118178.1118215](https://doi.org/10.1145/1118178.1118215).
- [20] Jeannette M. Wing. «Research Notebook: Computational Thinking—What and Why?». In: *The Link Magazine* (2011). URL: <https://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf>.

Laboratorio e.DO Learning Center: una collaborazione fruttuosa tra scuola e aziende

Paola Greco¹, Marcello Missiroli¹, Salvatore Parlato¹, Zakaria Touati², and Leonardo Ciocca²

¹ IIS Corni Liceo e Tecnico, Modena

m.missiroli@istitutocorni.it, s.parlato@istitutocorni.it, p.greco@istitutocorni.it

² Comau

zakaria.touati@external.comau.com, leonardo.ciocca@comau.com

Sommario

Presentiamo il Laboratorio e.DO Learning Center di Modena, realizzato in collaborazione tra l'istituto Fermo Corni, Ferrari S.p.A. e COMAU S.p.A. Il progetto prevede lo sviluppo di competenze coding attraverso l'utilizzo di e.DO Robot, un braccio robotico a sei assi prodotto da COMAU che permette ai partecipanti di sperimentare e apprendere diversi linguaggi di programmazione attraverso un'interfaccia dedicata e semplice da utilizzare.

1 Introduzione

La Digital Transformation comporta un radicale cambiamento nello scenario sociale e organizzativo in cui viviamo, di conseguenza diventa necessario sviluppare delle competenze dedicate per fronteggiare questo scenario e le sfide che comporta.

Gli istituti tecnici italiani costituiscono un *unicum* nell'ambito dell'istruzione per la loro capacità di formare persone pronte all'inserimento nel mondo del lavoro, fornendo loro conoscenze e competenze sia teoriche che pratiche di alto livello.

L'Istituto Corni di Modena non fa eccezione, e ha anzi sempre cercato di tenersi all'avanguardia sia nell'ambito didattico, attraverso modelli innovativi, sia in quello tecnologico, tramite uno stretto contatto con le aziende del territorio.

La realizzazione di e.DO Learning Center si inserisce nell'ambito della didattica innovativa in una prospettiva focalizzata verso l'apprendimento esperienziale e cooperativo. La trasmissione dei contenuti di coding, di robotica e delle discipline STEM punta a potenziare la capacità del territorio emiliano e di un Istituto Tecnico come il Fermo Corni di formare professionisti competenti ed efficaci nel gestire le trasformazioni in atto nel mondo lavorativo odierno e futuro.

2 Contesto

e.DO Learning Center è il frutto di un lavoro di ecosistema tra istituzioni, aziende e territorio ed è il risultato della collaborazione tra tre realtà di eccellenza: l'Istituto Corni, Ferrari e Comau. L'impegno messo in atto si declina in un'ottica di ecosistema sociale, puntando a favorire l'incontro tra scuola, formazione e lavoro in uno degli ambienti maggiormente produttivi sul panorama nazionale, quello della Motor Valley. Le tre realtà si sono prefissate l'impegno sociale di portare sul territorio le competenze necessarie per vivere e lavorare coerentemente con le esigenze del mondo del futuro.

2.1 Istituto Fermo Corni

L'IIS "Fermo Corni" di Modena è un Istituto di Istruzione Superiore, una Scuola Secondaria di Secondo grado di eccellenza al cui interno sono presenti un Istituto Tecnico con diversi indirizzi tecnici (Meccanica e Meccatronica, Energia, Informatica, Automazione, Elettronica, Elettrotecnica, Chimica-Biotecnologie ambientali) e un Liceo Scientifico opzione Scienze Applicate (LSSA); ad oggi la scuola accoglie poco meno di 2000 studenti, con utenza proveniente sia dalla città sia da molti comuni della provincia. L'istituto pone tra i suoi principi regolatori "costruire una permanente interazione col territorio nella progettazione e gestione dei processi formativi per la crescita della cultura, delle competenze professionali, della imprenditorialità in ambito locale e per un più efficace raccordo formativo tra la scuola e il lavoro".

2.2 Ferrari

Ferrari non ha bisogno di presentazioni: è la punta di eccellenza mondiale nel settore motoristico. L'azienda si distingue per l'innovazione, la qualità dei suoi prodotti e la continua ricerca di soluzioni all'avanguardia. Inoltre, l'azienda investe costantemente in programmi di formazione per i propri dipendenti, collaborando con istituzioni accademiche e professionali per offrire opportunità di aggiornamento e sviluppo delle competenze. In questa iniziativa Ferrari ha attivato il suo impegno sociale e si configura come promotore del progetto, l'azienda ha infatti donato il laboratorio *e.DO Learning Center* con l'intento di diffondere sul territorio le competenze in ambito coding e per le materie STEM in un'ottica generativa.

2.3 Comau

Comau è l'azienda italiana leader nella robotica e dell'automazione industriale. La sua produzione di robot industriali comprende una vastissima gamma di soluzioni che presentano caratteristiche personalizzate in base alle esigenze della committenza. *e.DO Learning Center* è un'iniziativa di eccellenza e simbolica dell'impegno di COMAU in ambito educativo e didattico, in particolare con lo scopo di promuovere lo studio delle discipline STEM attraverso l'uso della robotica come innovativo strumento di apprendimento. In questo scenario COMAU rappresenta il partner tecnico e didattico, fornendo le tecnologie, professionisti della formazione, metodologie e materiali didattici che costituiscono *e.DO Learning Center*.

3 Il progetto

e.DO Learning Center nasce come risposta alle esigenze territoriali per rispondere alle sfide innescate dall'Industria 5.0, per sviluppare le competenze necessarie a vivere e lavorare nel futuro. Il laboratorio, situato nell'IIS Fermo Corni di Modena, è aperto a tutti gli istituti del territorio emiliano e accoglie in forma gratuita studenti e studentesse tra gli 8 e i 18 anni per attività della durata di 4 ore

3.1 Obiettivi

Utilizzando *e.DO Robot* come strumento di apprendimento, l'obiettivo del laboratorio è quello di sviluppare le competenze curriculari e non relative al coding, orientando questa trasmissione di contenuti verso studenti di ogni ordine di scuola.

Punta inoltre a sviluppare le competenze STEM e le competenze trasversali.

3.2 Metodologia didattica

L'approccio didattico è quello della robotica educativa, che si concretizza in attività laboratoriali di stampo esperienziale attraverso l'utilizzo di e.DO Robot. Si tratta di un approccio di learning by doing che utilizza e.DO come strumento di apprendimento per veicolare contenuti curricolari di coding, robotica, matematica. Contestualmente, attraverso metafore lavorative che attivano dinamiche collaborative o competitive tra i partecipanti, è possibile stimolare competenze trasversali che possono riflettersi ad esempio nella comunicazione, nella collaborazione o nel *problem solving*. La metodologia didattica pone le sue fondamenta sul modello di Kolb, nel quale l'attività prevede il costante interscambio di 3 momenti:

- **Contenuto** (25%): momenti dedicati all'apprendimento di nozioni scolastiche legate al coding e alle discipline STEM
- **Azione** (50%): momento esperienziale in cui, attraverso attività ingaggianti gli studenti mettono in pratica i contenuti scolastici trattati e li sperimentano in prima persona
- **Relazione** (25%): momento di scambio, confronto e feedback dedicato alla costruzione sociale delle conoscenze.

3.3 Tecnologie e ambiente didattico

Il laboratorio prevede cinque *e.DO Robot* a sei assi prodotto da COMAU, situati su cinque isole didattiche che permettono un'interazione in piccolo gruppo ai partecipanti. Il controllo di e.DO avviene tramite tablet, PC e un'app appositamente studiata per queste esperienze didattiche. È possibile implementare dei codici di programmazione che movimentano e.DO grazie a delle interfacce software grafiche e open-source. In questo ambiente sono anche presenti una LIM ed una whiteboard di ultima generazione per garantire un'esperienza multimediale ottimale.

4 Contenuti didattici

Il laboratorio permette a Istituti Scolastici provenienti da tutta l'Emilia Romagna di partecipare gratuitamente a diverse attività didattiche a seconda del grado scolastico e degli argomenti trattati. Rispetto al coding, l'ingegneria didattica di COMAU ha sviluppato 4 pacchetti didattici della durata di 4 ore ciascuno:

- “RoboBLOCK” (8-10 anni) - introduzione alla programmazione visuale a blocchi
- “RoboBIT” (11-13 anni) - numerazione binaria e pensiero computazionale.
- “RoboPYTHON” (14+ anni) - programmazione di e.DO tramite Python.

Si riportano qui le tipologie di contenuti didattici trasmessi attraverso questi pacchetti: RoboBIT: RoboPhyton: Le attività durano 4 ore e sono condotte da un formatore di Comau mentre ogni classe è accompagnata da un docente curricolare. Gli studenti vengono divisi in 5 gruppi e, dopo una presentazione iniziale del laboratorio, vengono veicolati i primi contenuti relativi alla componentistica e all'uso di e.DO Robot. In seguito sono proposti dei task di stampo cooperativo o competitivo basati sul coding tramite interfacce specifiche. I pacchetti didattici sono inseriti in un contesto di metafora lavorativa in cui gli studenti imparano i principi del Coding e li mettono in pratica programmando un robot reale e sperimentando modalità



Figura 1: Progetto del laboratorio e.DO Learning center

RoboBLOCK:

Modulo	Finalità	Unità	Risultati di apprendimento
ROBOTICA	Utilizzare i comandi per muovere un robot	I MOVIMENTI DEL ROBOT	Attuare i movimenti in giunti di un robot
CODING	Comprendere le differenti modalità di programmazione di un robot e sviluppare un programma per muoverlo	PRINCIPI DI CODING: MOVIMENTAZIONE	Attuare i movimenti in giunti di un robot attraverso la programmazione a blocchi
		PRINCIPI DI CODING: CICLI	Eseguire operazioni cicliche con un robot attraverso la programmazione a blocchi
		PRINCIPI DI CODING: DISEGNO	Orientare il movimento di un robot per disegnare su un piano attraverso la programmazione a blocchi

Figura 2: Enter Caption

di apprendimento che seguono il modello del *Learning by doing* e *Project based learning*. Vi rimandiamo al sito (<https://sites.google.com/istitutocorni.it/edo/moduli-didattici>) per ulteriori dettagli.

5 Risultati

Il progetto e.DO Learning Center di Modena è partito nel 2022 con l'obiettivo di intercettare, ogni anno scolastico, 3000 studenti e studentesse da tutto il territorio dell'Emilia Romagna e

Modulo	Finalità	Unità	Risultati di apprendimento
ROBOTICA	Distinguere e utilizzare le diverse tipologie di movimento di un robot	I MOVIMENTI DEL ROBOT	Utilizzare i movimenti cartesiani di un robot tramite interfaccia grafica e programmazione a blocchi
CODING	Organizzare un processo in sequenze interconnesse Sviluppare un programma per movimentare un robot	IL PENSIERO COMPUTAZIONALE E LE STRUTTURE DI BASE DI UN LINGUAGGIO DI PROGRAMMAZIONE	Analizzare un problema utilizzando la programmazione, sviluppando un pensiero computazionale Apprendere la sintassi e i costrutti logici di base della programmazione
		IL SISTEMA BINARIO NELLA GESTIONE DELL'INFORMAZIONE	Utilizzare la codifica del sistema binario per gestire informazioni in entrata e uscita

Figura 3: Enter Caption

Modulo	Finalità	Unità	Risultati di apprendimento
ROBOTICA	Descrivere le parti di un braccio robotico	e.DO ROBOT	Distinguere le varie parti che compongono un braccio robotico e controllarlo in un sistema di riferimento cartesiano
CODING	Definire un linguaggio di programmazione Utilizzare Python per programmare un robot	LE STRUTTURE BASE DI UN LINGUAGGIO DI PROGRAMMAZIONE	Interpretare la sintassi e i costrutti logici di base della programmazione
		LA MOVIMENTAZIONE DEL ROBOT CON PYTHON	Movimentare un robot utilizzando il linguaggio di programmazione Python
		LA CONDIZIONE "if" CON PYTHON	Applicare le strutture condizionali nella scrittura di un programma
		I CICLI E LA CONDIZIONE "while" CON PYTHON	Usare condizioni iterative nella scrittura di un programma

Figura 4: Enter Caption



Figura 5: Braccio meccanico e.DO



Figura 6: Il laboratorio

avvicinarli al mondo del coding, della robotica e delle STEM. Tale obiettivo è stato superato sia nell' A.S. 2022/23 sia nel successivo, con 3018 e 3207 studenti, rispettivamente. Complessivamente hanno partecipato 283 classi per un totale di 1084 ore di attività. I risultati attesi nei prossimi anni si attestano su queste cifre, con la volontà di coinvolgere nuove scuole presenti sul territorio. Nei risultati di apprendimento attesi possono essere individuate delle competenze comuni che il laboratorio mira a sviluppare:

- Distinguere le varie parti che compongono un braccio robotico e controllarlo entro un sistema di riferimento cartesiano.
- Interpretare la sintassi e i costrutti logici di base della programmazione.
- Movimentare un robot utilizzando un linguaggio di programmazione a blocchi o testuale.
- Applicare le strutture condizionali nella scrittura di un programma.
- Usare condizioni iterative nella scrittura di un programma.
- Rafforzamento di competenze trasversali (collaborazione, comunicazione, problem solving, apertura mentale, etc.)

6 Replicabilità

Il progetto e.DO Learning Center è replicabile, come testimoniato da varie esperienze già presenti sul territorio nazionale. Il progetto di Modena qui presentato rappresenta la punta d'ecceellenza di questa tipologia di progettualità, nata dalla sinergia tra Comau, Ferrari ed Istituto Corni. Al di là di queste specificità, l'e.DO Learning Center è un prodotto sviluppato dal team Academy di Comau. Esso prevede gli stessi pacchetti didattici e le stesse tecnologie utilizzate per veicolare i contenuti qui descritti. Presso alcuni contesti il progetto è stato concretizzato attraverso l'uso dell'"e.DO Cube", un simulatore virtuale che permette di sperimentare l'uso di e.DO robot attraverso un'interfaccia digitale. Gli e.DO Learning Center presenti ad oggi in Italia rappresentano, come nel caso di Modena, il frutto della collaborazione degli enti territoriali in ottica di ecosistema, con il fine di offrire progettualità didattiche di altissimo valore.

7 Conclusioni

La vera innovazione di questo laboratorio consiste nell'incontro tra esperienza didattica scolastica e realtà aziendale, che vuole essere un modello di formazione rivolto ai giovani studenti per favorire l'orientamento e la crescita di competenze collegate al territorio.

Il laboratorio e.Do ha dimostrato di essere un efficace strumento didattico, capace di coniugare teoria e pratica in modo coinvolgente e stimolante. E' un esempio di modello didattico innovativo che potrebbe (o dovrebbe) essere replicato in altre scuole. I benefici osservati suggeriscono che un approccio pratico e interattivo alla formazione tecnica non solo facilita l'apprendimento, ma rende anche gli studenti più preparati e motivati ad affrontare le sfide future.

Riferimenti Bibliografici

- Ivaldi, S., Scaratti, G., & Fregnan, E. (2022). Dwelling within the fourth industrial revolution: organizational learning for new competences, processes and work cultures. *Journal of Workplace Learning*, 34(1), 1-26.
- Fregnan, E., Ivaldi, S., & Scaratti, G. (2020). HRM 4.0 and New Managerial Competences Profile: The COMAU Case. *Frontiers in psychology*, 11.
- Fregnan, E., Scaratti, G., Ciocca, L., & Ivaldi, S. (2022). New working capabilities for coping with COVID time challenges. *Frontiers in Psychology*, 13, 814348.
- Fregnan, E., Pinto, D., & Scaratti, G. (Eds.). (2022). *Instilling Digital Competencies Through Educational Robotics*. IGI Global.
- Lee M.X., Lee Y.C., Chou C.J., 2017, Essential implications of the digital transformation in industry 4.0, *Journal of Scientific & Industrial Research*, vol. 76, pp. 465-467.
- Pinto, D., Scaratti, G., & Fregnan, E. (2018). *Giovani, Università e Azienda: il nuovo perimetro formativo*. Pearson.
- Pinto, D., Scaratti, G., & Fregnan, E. (2021). *Learnin agility 4.0. Ecosistemi e trasformazione culturale*. Pearson Italia.
- Prinsley, R. T., & Baranyai, K. (2013). *STEM skills in the workforce: what do employers want?*. Brisbane: Office of the Chief Scientist.
- Scaratti, G., Fregnan, E., & Ivaldi, S. (2018). Le situazioni di apprendimento. In *Giovani, università e azienda: il nuovo perimetro formativo: l'esperienza della Project & people management school* (pp. 76-151). Pearson.
- Stice, J. E. (1987). Using Kolb's Learning Cycle to Improve Student Learning. *Engineering education*, 77(5), 291-96.
- Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *The journal of strategic information systems*, 28(2), 118-144.
- Warner, K. S., & Wäger, M. (2019). Building dynamic capabilities for digital transformation: An ongoing process of strategic renewal. *Long Range Planning*, 52(3), 326-349.
- Yeow, A., Soh, C., & Hansen, R. (2018). Aligning with new digital strategy: A dynamic capabilities approach. *The Journal of Strategic Information Systems*, 27(1), 43-58.

LABORATORI PER LA SECONDARIA DI SECONDO GRADO

Il Coding nell'Educazione Inclusiva

Lorenzo Notaro, Roberta Cavalli, Enrica Gentile, Veronica Rossano

Dipartimento di Informatica Università degli Studi di Bari Aldo Moro
enrica.gentile@uniba.it, veronica.rossano@uniba.it

Abstract

Il Coding e la Robotica Educativa hanno acquisito crescente rilevanza come metodologie didattiche, soprattutto nell'ambito dell'educazione inclusiva. Queste metodologie si sono rivelate efficaci nel promuovere la collaborazione tra studenti ed educatori, utilizzando una varietà di stili di apprendimento dal laboratoriale al tangibile. In questa esperienza abbiamo portato il coding all'interno di una realtà educativa permanente di accoglienza di ospiti con disabilità motoria e cognitiva.

1 Il Coding e la Robotica Educativa

Il Coding è una forma elementare di programmazione che permette di applicare la logica algoritmica di risoluzione dei problemi tradotti in un linguaggio comprensibile ed eseguibile da un computer o da un robot. Questo processo sviluppa competenze tecniche e abilità cognitive fondamentali quali, ad esempio, il problem solving, il pensiero logico e la capacità di pianificazione.

Seymour Papert, uno dei pionieri nell'uso del computer per l'educazione e padre del costruzionismo, sostiene che l'apprendimento è più efficiente e proficuo se avviene mediante la produzione da parte di chi apprende di oggetti concreti e reali. Papert ha concluso che la programmazione aiuta a "imparare a imparare", promuovendo un apprendimento attivo e creativo. [1]

A questo si aggiunge il pensiero di Resnik e Siegel che presentano l'importanza che il Coding riveste facendo riferimento all'ambiente di programmazione "*Scratch*", il nuovo linguaggio che implementa la programmazione a blocchi mediante l'utilizzo di oggetti grafici che possono essere agganciati l'uno con l'altro per eseguire una sequenza di istruzioni. Resnik dichiara che "Il Coding, per noi, non rappresenta un insieme di competenze tecniche, ma piuttosto un nuovo tipo di alfabetizzazione e una forma di espressione personale, preziosa per tutti, proprio come imparare a scrivere. Noi vediamo il Coding come un nuovo modo, per le persone, di organizzarsi, esprimersi e condividere le proprie idee." [2]

L'importanza di questa modalità di programmazione nell'ambito delle attività educative e didattiche ci porta a comprendere come questo modalità di sviluppo delle abilità cognitive porta alla messa in atto del pensiero computazionale.

Il pensiero computazionale si basa sull'analisi e la scomposizione di problemi complessi in parti più gestibili, sviluppando soluzioni algoritmiche. Il termine di "Computational Thinking", coniato da Jeanette Wing, definisce un approccio impiegato dagli informatici per la risoluzione di problemi che implica la formulazione delle soluzioni algoritmiche come rappresentazione di sequenze di passi computazionali. Questa metodologia di problem solving non è limitata all'informatica ma è applicabile a diverse discipline, facilitando così l'apprendimento interdisciplinare. [3]

L'utilizzo del coding nell'educazione può quindi essere allargato a contesti multidisciplinari, consentendo di includere studenti con diversi stili di apprendimento e background. Attraverso attività pratiche come il coding e la robotica educativa, i partecipanti possono interagire con i

concetti e creare artefatti, il che può essere particolarmente benefico per coloro che trovano difficoltà con metodi più tradizionali di istruzione.

La robotica educativa, integrata con il coding, rappresenta un potente strumento didattico che utilizza robot programmabili per insegnare concetti attraverso esperienze pratiche e interattive. Questo approccio multidisciplinare coinvolge gli studenti in campi come matematica, italiano, storia, geografia e arte, rendendo l'apprendimento più accessibile e coinvolgente. Attraverso l'uso di robot, gli studenti possono apprendere concetti di coordinate e geometria, narrare storie e creare scenari di gioco, migliorando sia le loro competenze tecniche che narrative. Inoltre, la robotica educativa favorisce lo sviluppo delle competenze sociali ed emotive attraverso attività collaborative, riducendo lo stress e aumentando la motivazione grazie ai risultati immediati e tangibili delle loro attività didattiche.

In questo contesto, la ricerca ha portato alla definizione di un percorso di coding inclusivo che ha visto la combinazione di attività unplugged e attività in Scratch per introdurre il coding ad un gruppo di ospiti del centro diurno "Fratelli Tutti" di Melpignano (LE), della Caritas diocesana di Otranto. Le attività hanno visto coinvolti soggetti con disabilità cognitive di diversa tipologia e livello di gravità. L'esperienza è stata accolta e svolta positivamente dai partecipanti, dagli operatori del centro, che hanno collaborato ognuno nell'ambito della propria professionalità, e dai familiari dei partecipanti.

2 Educazione Inclusiva e Disabilità

Nel contesto dell'educazione inclusiva, la disabilità rappresenta una delle sfide più significative. La disabilità può manifestarsi in molte forme diverse, ciascuna con le proprie necessità e requisiti specifici. L'obiettivo principale dell'educazione inclusiva è garantire che tutti i partecipanti, indipendentemente dalle loro abilità fisiche o cognitive, abbiano accesso a un'istruzione di qualità.

I vantaggi del coding nell'educazione inclusiva sono molteplici:

- **Sviluppo di Abilità Cognitive:** Il coding richiede una sequenza logica di pensiero e la capacità di risolvere problemi complessi. Queste abilità sono cruciali per tutti i partecipanti ma possono essere particolarmente benefiche per quelli con disabilità cognitive, aiutandoli a sviluppare capacità di pensiero critico e di risoluzione dei problemi.
- **Strumenti Inclusivi e Accessibili:** Molti strumenti di coding e piattaforme sono progettati per essere inclusivi e accessibili. Esistono linguaggi di programmazione visivi come Scratch particolarmente utili per partecipanti con difficoltà di apprendimento o disabilità cognitive grazie a un'interfaccia grafica intuitiva che facilita la comprensione dei concetti di base della programmazione.
- **Aumento dell'Autonomia e Fiducia in Sé Stessi:** Imparare a programmare può aumentare l'autonomia e la fiducia in sé stessi dei partecipanti con disabilità. La capacità di creare qualcosa di nuovo e vedere immediatamente i risultati del proprio lavoro può essere estremamente gratificante e motivante.
- **Inclusione Sociale e Collaborazione:** Lavorare in team su progetti di programmazione può favorire l'inclusione sociale e aiutare i partecipanti a sentirsi parte di una comunità, sviluppando in questo modo abilità sociali e di collaborazione.

L'uso dei computer e delle tecnologie di programmazione offre strumenti personalizzabili e accessibili, promuovendo l'autonomia e la fiducia in sé stessi. Dal punto di vista psicologico, questa collaborazione riduce il senso di isolamento e favorisce lo sviluppo di un forte senso di appartenenza e sostegno reciproco, elementi chiave per il benessere emotivo e la crescita personale.

L'integrazione del coding e della robotica educativa nell'educazione inclusiva contribuisce a creare un ambiente di apprendimento più equo e inclusivo, dove ogni studente può sviluppare le proprie potenzialità [4, 5] e sentirsi parte di una comunità solidale e collaborativa [6]. Questo

approccio non solo sviluppa competenze tecniche e cognitive fondamentali, ma favorisce anche l'inclusione sociale, migliorando l'esperienza educativa per tutti [6].

3 L'esperienza del percorso inclusivo in un centro diurno

L'esperienza didattica è stata applicata in un ambiente educativo permanente, il Centro Diurno "Fratelli Tutti" di Melpignano (LE), della Caritas diocesana di Otranto. Sono stati coinvolti 8 soggetti con disabilità lievi ed età compresa tra i 25 e i 63 anni.

Per lo svolgimento delle attività, si è deciso di dividere i partecipanti in due gruppi:

- Gruppo ad **alto funzionamento** composto da Domenico, Andrea, Pierpaolo
- Gruppo a **basso funzionamento** composto da Carlo, Gabriele, Concetta, Luisa, Ausilia

Questa divisione in due gruppi ha permesso di organizzare le attività in modalità differenti con vari livelli di difficoltà.

3.1 Gruppo ad Alto Funzionamento

Per il gruppo ad alto funzionamento, è stato pensato un percorso con attività mirate alla logica e al Pensiero Computazionale entrando, sin da subito, nella disciplina Informatica.

Questo percorso è stato organizzato in undici incontri (Tabella 1):

Tabella 1: Organizzazione degli incontri del gruppo ad *alto funzionamento*

Obiettivo	Incontro	Strumento/attività
LOGICA	Primo incontro	Giochi Bebras (con vincolo tempo)
	Secondo incontro	Giochi Bebras (senza vincolo tempo)
SEQUENZA, SELEZIONE, ITERAZIONE IN SCRATCH	Terzo incontro	Presentazione dei concetti di Sequenza, Selezione e Iterazione Primo progetto in Scratch
	Quarto incontro	Introduzione del concetto di variabile
	Quinto incontro	Presentazione della scheda elettronica Makey Makey
	Sesto incontro	Progettazione su carta del quiz finale e realizzazione delle parti grafiche
	Settimo incontro	Utilizzo di una nuova variabile per definire chi è il prossimo giocatore
	Ottavo incontro	Inserimento del punteggio di gioco
	Nono incontro	Realizzazione Dettato Direzionale
	Decimo incontro	Collegamento fisico a scheda Makey Makey
	Undicesimo incontro	Test finale

Logica - Giochi Bebras:

L'obiettivo è stata la valutazione delle abilità di ragionamento e risoluzione di problemi/quesiti in un tempo determinato. Successivamente, gli esercizi sono stati ripetuti senza vincoli di tempo per osservare eventuali miglioramenti.

Sequenza, Selezione, Iterazione - Scratch:

Il percorso su Scratch è stato suddiviso in 8 incontri della durata che variava da 30 minuti (all'inizio del percorso) a 70 minuti (alla fine). Per stimolare e motivare il gruppo dei partecipanti, è stato chiesto loro di realizzare un quiz da somministrare al gruppo di basso funzionamento.

Per introdurre i 3 principali modelli di composizione delle istruzioni (Sequenza, Selezione, Iterazione) sono stati utilizzati degli esempi di vita quotidiana.

Dopo aver verificato che i partecipanti avessero acquisito questi concetti preliminari ed essenziali, si è passati alla piattaforma Scratch e alla sua esplorazione con alcuni esempi molto elementari (Tabella 2).

Tabella 2: Primo esercizio per il gruppo ad *alto funzionamento*

Pseudocodice	Script Scratch
<p>Ripeti per tre volte queste istruzioni</p> <p>quando si preme il tasto ‘a’:</p> <p>(5 passi in avanti</p> <p>Lo Sprite dice “Ciao” per due secondi</p> <p>10 passi in avanti</p> <p>Lo Sprite dice “Piacere” per tre secondi)</p>	

Questo primo esercizio è stato seguito da altri con difficoltà crescente per introdurre il concetto di variabile. Anche in questo caso sono stati sempre usati esempi pratici e concreti, presi dalla vita quotidiana dei partecipanti.

Dopo aver implementato le varie funzionalità previste per il quiz è stata presentata la scheda elettronica Makey Makey e i suoi vantaggi implementativi. La scheda elettronica Makey Makey codifica il segnale derivante dalla pressione delle frecce di direzione della tastiera, del tasto spazio e del click del mouse. La scheda collegata tramite dei cavetti in metallo permette di collegare altri elementi programmati tramite Scratch (Figura 1).

Per motivare i partecipanti è stato proposto di realizzare i disegni che avrebbero composto la grafica del quiz. Domenico e Pierpaolo hanno riportato di essere orgogliosi di aver creato, grazie all’informatica, un prodotto utile per i loro amici andando incontro ad alcune delle loro difficoltà.



Figura 1: Makey Makey in uso: due soggetti rispondono al quiz realizzato in Scratch usando dei bottoni realizzati con la plastilina

3.2 Gruppo a Basso Funzionamento

Per il gruppo a basso funzionamento è stato pensato un percorso composto da attività più elementari ma al tempo stesso divertenti e stimolanti. L’intero percorso è stato suddiviso in 9 incontri (Tabella 3):

Tabella 3: Organizzazione degli incontri del gruppo a *basso funzionamento*

Obiettivo	Numero incontro	Strumenti/attività	
Logica	Primo incontro	Cody&Roby	
Selezione	Secondo incontro	Percorsi Cerchi-Pallina	
Migliorare le capacità di orientamento	Terzo incontro	Schede Didattiche	
	Quarto incontro	Destra-Sinistra,	
	Quinto incontro		
	Sesto incontro	Riconosciamo la destra e la sinistra,	
	Settimo incontro		
	Ottavo incontro	Giochi interattivi su Wordwall	Creazione ambiente positivo: arteterapia e psicomotricità
	Nono incontro		Creazione ambiente positivo: musicoterapia

Logica: Cody&Roby

La prima attività proposta è stata il Coding Unplugged con il gioco Cody&Roby. I partecipanti hanno utilizzato carte che rappresentano diverse azioni elementari:

- vai avanti
- girati a sinistra
- girati a destra

per muovere una pedina sulla scacchiera (Figura 2).

Durante il gioco, i ruoli tra chi dà le istruzioni (Cody) e chi le esegue (Roby) si invertono.

Per semplificare, l'attività è stata trasformata in un gioco umano utilizzando il pavimento (Figura 2) e una lavagna per scrivere le regole e attaccare le carte (Figura 3).



Figura 2. La scacchiera costruita sul pavimento

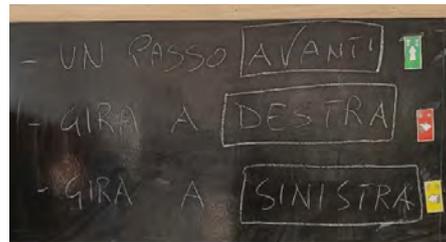


Figura 3. La lavagna per ricordare le regole e le carte da usare.

Dalle osservazioni è emerso che i partecipanti avevano difficoltà a riconoscere la destra e la sinistra. Pertanto, è stato sviluppato un nuovo percorso con attività specifiche per migliorare queste abilità:

- Percorsi Cerchi-Pallina
- Schede Didattiche Destra e Sinistra
- Giochi interattivi su Wordwall
- Riconosciamo la destra e la sinistra

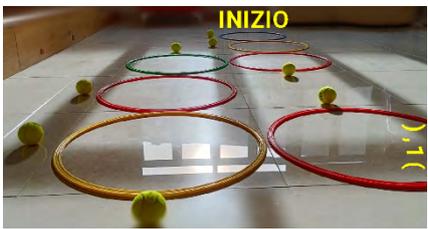
Selezione: Percorsi Cerchi-Pallina

Questa attività chiedeva di attuare due movimenti simili a quelli già visti nell'attività precedente: saltare o camminare nel cerchio successivo e girare a destra o sinistra. Le regole erano semplici e prevedevano di pronunciare a voce alta ogni movimento eseguito. Accanto ad ogni cerchio, era posizionata una pallina da raccogliere con la mano destra o sinistra, a seconda della posizione (Figura 4). L'obiettivo era rinforzare il riconoscimento della destra e della sinistra, incorporando i concetti base della programmazione come sequenza, selezione e iterazione.

Migliorare le capacità di orientamento: attività didattiche varie

Questa fase prevedeva varie sessioni di lavoro per potenziare le capacità di orientamento nello spazio, con l'obiettivo principale di riconoscere la destra e la sinistra. Le attività includevano:

- Attività mano destra e sinistra (Figura 5): i ragazzi dovevano rispondere ai comandi sia in gruppo che singolarmente.
- Attività in Wordwall su destra e sinistra (Figura 6): attività per identificare la posizione di oggetti rispetto a un albero al centro della figura.
- Attività Schede Didattiche (Figura 7): schede per determinare se un oggetto si trovava a destra o sinistra.

	
<p>Figura 4. Attività Cerchi Pallina</p>	<p>Figura 5. Attività Mano destra e mano sinistra</p>
	
<p>Figura 6. Attività wordwall</p>	<p>Figura 7. Attività Schede didattiche</p>

Queste 4 attività sono state ripetute in sequenza per ogni sessione di lavoro.

Durante le sessioni di lavoro è stato essenziale creare un ambiente positivo organizzando delle attività di musicoterapia e arteterapia quando la motivazione e l'attenzione dei ragazzi veniva meno. A circa metà del percorso, ad esempio, Concetta dimostra di subire questo calo risolvendo solo 2 quesiti su 5 nell'attività delle schede didattiche. Lei stessa afferma di essere stanca delle attività proposte e di non volerne più fare parte. Così si decide di sospendere le attività per creare un ambiente positivo. Durante l'attività di arteterapia è stato realizzato dai ragazzi un cartellone con due enormi mani gialle e blu che ha aiutato molto i ragazzi nel riconoscimento della destra e della sinistra. Infatti, Concetta, ad esempio, ha mostrato notevoli miglioramenti in quanto ha concluso il suo percorso con ottimi risultati e con il massimo del punteggio nell'ultimo incontro. Allo stesso modo, durante l'attività di musicoterapia i ragazzi hanno avuto la possibilità di creare un ritmo musicale impugnando due strumenti musicali (uno nella mano destra e uno nella mano sinistra) dal suono diverso tra loro. Questo ha permesso loro di migliorare le loro capacità di riconoscimento di destra e sinistra grazie all'uso del suono e della musica.

I ragazzi si sono dimostrati interessati alle nuove modalità di esecuzione delle diverse attività e questo ha aumentato la loro motivazione.

4 Conclusioni

L'esperienza dei laboratori di coding con adulti con disabilità ha dimostrato come queste metodologie possano essere potenti strumenti di inclusione ed empowerment. Attraverso il coding, i partecipanti non solo acquisiscono competenze tecniche ma sviluppano anche capacità cognitive e sociali, contribuendo a migliorare la loro qualità di vita e favorendo una maggiore autonomia.

Le attività proposte hanno portato a miglioramenti evidenti nei partecipanti. Il gruppo a basso funzionamento ha notevolmente migliorato le abilità di orientamento nello spazio, fondamentali per il coding, e ha acquisito i concetti base della programmazione. Per il gruppo ad alto funzionamento, queste attività hanno agevolato l'acquisizione di competenze informatiche e lo sviluppo di un pensiero critico nella risoluzione dei problemi. Inoltre, le attività hanno portato ad una maggiore coesione nel gruppo, tutti i partecipanti si sono aiutati vicendevolmente. Il Peer Tutoring è stato fondamentale, poiché supportare i compagni con maggiori difficoltà ha ulteriormente stimolato la crescita delle loro conoscenze e competenze. Gli effetti delle attività potranno essere valutati nel lungo periodo, ma i percorsi sviluppati, anche in base alle testimonianze dei familiari dei partecipanti, hanno contribuito sia a far frequentare il centro diurno con maggiore entusiasmo, sia a migliorare l'autonomia nelle attività quotidiane in cui si richiede di seguire piccole procedure.

Ringraziamenti

Si ringraziano tutti i partecipanti Andrea, Ausilia, Carlo, Concetta, Domenico, Gabriele, Luisa, Pierpaolo e i loro familiari. Inoltre, si ringraziano tutti gli operatori del Centro Diurno "Fratelli Tutti" di Melpignano (LE), della Caritas diocesana di Otranto, che hanno contribuito ad arricchire l'esperienza di apprendimento con le loro diverse professionalità.

Bibliografia

- [1] S. Papert, in *Mindstorms: Children, Computers, and Powerful Ideas.*, Basic Book, 1980.
- [2] M. Resnik e D. Siegel, «A different approach to Coding,» Medium:, 10 November 2015. [Online]. Available: <https://brightthemag.com/a-differentapproach-to-coding-d679b06d83a>.
- [3] J. Wing, «Computational Thinking,» Communications of the ACM, 2006.
- [4] Demir, Ü., The effect of unplugged coding education for special education students on problem-solving skills. *International Journal of Computer Science Education in Schools*, 4(3), 3-30, 2021.
- [5] Knight, V. F., Wright, J., Wilson, K., & Hooper, A. (2019). Teaching digital, block-based coding of robots to high school students with autism spectrum disorder and challenging behavior. *Journal of Autism and Developmental Disorders*, 49, 3113-3126.
- [6] Schmidt, T., Schneider, J., & Schuberth, K. Coding promotes the social participation of young people with disabilities. *SOCIAL INCLUSION AND DIGITALISATION*, 177.

Una Metodologia per l'Integrazione del Coding nell'Educazione Scolastica

Alessandro Marcelletti¹, Sara Pettinari², Barbara Re¹, and Beatrice Strappa¹

¹ Università degli Studi di Camerino, Sezione di Informatica, Camerino, Italia
{nome.cognome}@unicam.it

² Gran Sasso Science Institute, L'Aquila, Italia sara.pettinari@gssi.it

1 Introduzione

In un'era di rivoluzione digitale, l'informatica è presente in ogni aspetto della vita quotidiana, dai dispositivi intelligenti alle attività lavorative. In questo contesto, sviluppare pensiero critico e acquisire competenze pratiche sin dai primi anni di scuola prepara gli studenti al futuro mondo del lavoro e crea una società tecnologicamente consapevole [8]. Questo è reso possibile attraverso il *coding*, ossia tutte quelle attività che, in modo divertente ed efficace, permettono di apprendere come usare in modo critico le nuove tecnologie [11]. Fin da bambini, l'educazione nelle scuole primarie può infatti aiutare ad acquisire queste competenze grazie alla creazione di moduli formativi dedicati. Tra i punti di forza delle attività di coding troviamo l'*interdisciplinarietà*, che consente di combinare attività formative per lo studio e l'approfondimento di diversi argomenti scolastici [1]. Inoltre, il coding stimola la *creatività* grazie alla definizione di idee e alla creazione di progetti pratici [7]. Parallelamente, l'insegnamento del coding favorisce lo sviluppo della *logica* e del *pensiero computazionale* attraverso la comprensione strutturata e sequenziale richiesta dalla programmazione [9]. Infine, le attività di coding aprono nuove dinamiche di apprendimento *collaborativo*, richiedendo la condivisione di idee, la risoluzione collettiva di problemi e la realizzazione di progetti di gruppo [12].

Nonostante le possibilità offerte dalle attività di coding, l'insegnamento di questa disciplina risulta relativamente recente, specialmente a livello di istruzione primaria e secondaria inferiore [2]. Negli ultimi due decenni, l'Europa ha compiuto un significativo passo avanti nel panorama dell'istruzione digitale, concentrando particolare attenzione all'introduzione del coding nelle scuole [3]. Anche l'Italia ha progressivamente riconosciuto la crescente importanza delle competenze digitali e del coding nell'economia moderna e nell'educazione, promuovendo iniziative e politiche nazionali mirate [6]. Tuttavia, nonostante questa attenzione emergente verso il coding, rimangono ancora diversi ostacoli da superare come la carenza di approcci metodologici strutturati e pensati a lungo termine.

Per questo motivo, la Sezione di Informatica dell'Università degli Studi di Camerino ha lavorato ad una nuova metodologia per la definizione ed implementazione di moduli formativi per l'insegnamento del coding. In particolare, il primo passo è stato quello di identificare tutti gli elementi che caratterizzano i moduli formativi e le loro relazioni. Questi sono stati poi presentati come un database a grafo che permette di visualizzare e analizzare graficamente tali caratterizzazioni. Successivamente, è stata proposta una metodologia a supporto degli insegnanti, per definire ed implementare moduli formativi per la didattica del coding. Infine, la metodologia proposta è stata validata tramite la creazione di alcuni moduli didattici, erogati in due scuole primarie del territorio.

2 Caratterizzazioni Moduli Formativi

Il primo aspetto fondamentale del lavoro è stato l'identificazione ed analisi degli elementi essenziali che caratterizzano la definizione di un modulo formativo orientato al coding. Tali elementi caratterizzanti sono riportati di seguito:

- **Strumenti educativi:** rappresentano il mezzo con cui si mette in pratica un'attività di coding. Gli strumenti comunemente utilizzati si dividono in *unplugged*, *software* (a blocchi di icone o blocchi testuali) e *hardware* (assemblati o da assemblare).
- **Metodologie didattiche:** rappresentano le modalità dei processi di insegnamento e apprendimento [5]. Alcuni esempi sono *Learning-By-Doing*, *Game-Based Learning*, *Storytelling*, *Project-Based Learning* e *Problem-Solving*.
- **Organizzazione dell'aula:** rappresenta la disposizione dei banchi a seconda dell'obiettivo formativo e influenzano l'efficacia dell'insegnamento [4]. Questi sono *Ferro di cavallo*, *Serpentone*, *Gruppi*, *Isole*, *Corridoio*, *Conferenza*, *Stadio*, *Partnership*.
- **Tematiche:** rappresentano le tematiche del modulo come le discipline scolastiche o temi di attualità.
- **4C:** rappresentano le competenze trasversali ritenute fondamentali per la crescita e l'apprendimento [10]. Queste sono *Pensiero Critico*, *Comunicazione*, *Collaborazione* e *Creatività*.

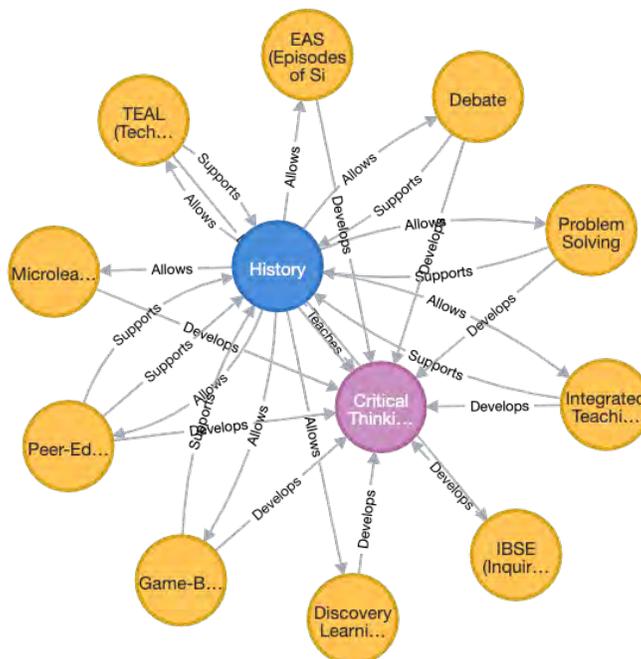


Figura 1: Visualizzazione di metodologie didattiche che collegano la materia *storia* alla competenza *pensiero critico*.

Al fine di racchiudere queste caratterizzazioni ed i loro collegamenti, si propone un database a grafo in grado di rappresentare dati fortemente collegati tra loro, sotto forma di nodi ed archi. Il grafo aiuta a pianificare lezioni e attività in modo efficace supportando l'insegnante nella selezione dei migliori elementi a seconda degli obiettivi e del contesto educativo. Il grafo risultante è costituito da 77 nodi e 1746 relazioni, che possono essere facilmente interrogabili per visualizzare intuitivamente le connessioni tra i concetti chiave che caratterizzano le attività di coding. Nel dettaglio, i nodi sono etichettati in base ai concetti descritti e sono relazionati tra loro in modo da definire come questi concetti si influenzano reciprocamente. Il punto di forza di questo sistema risiede nella sua capacità di fornire soluzioni diverse e flessibili per ogni esigenza educativa, permettendo agli insegnanti di personalizzare le loro pratiche didattiche in modo efficace e mirato, considerando variabili come il numero di alunni, la flessibilità dell'ambiente operativo, la disponibilità di strumenti e altri fattori cruciali.

Ad esempio, la Figura 1 visualizza le metodologie didattiche che collegano la materia *Storia* alla competenza *Pensiero Critico*. Grazie a questa visualizzazione, è possibile ottenere in modo intuitivo le possibili metodologie didattiche che permettono di insegnare la materia d'interesse e al contempo sviluppare il pensiero critico. Il database è accessibile online¹ e può essere importato ed interrogato nel software Neo4j².

3 Metodologia Proposta

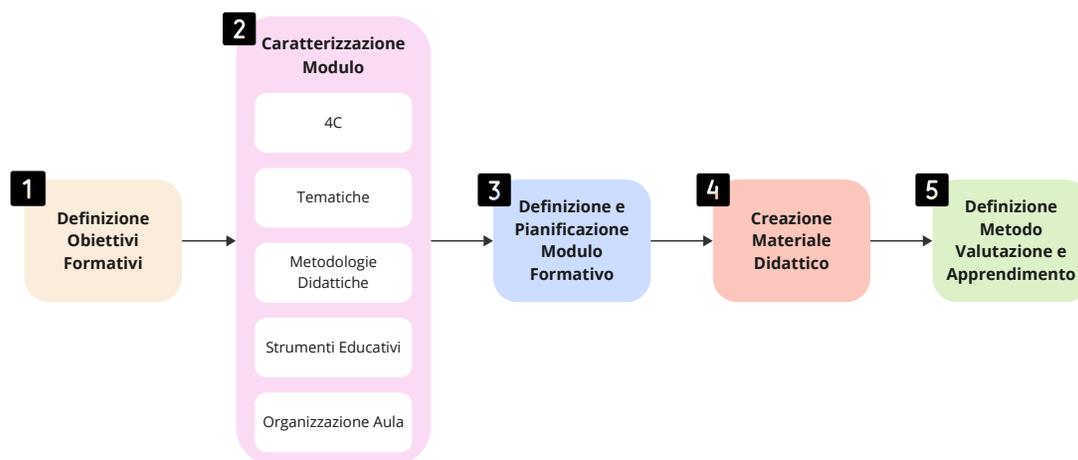


Figura 2: Metodologia proposta

In questa sezione si descrive la metodologia proposta per la definizione e implementazione di moduli formativi per la didattica del coding. Attraverso questa metodologia, l'insegnante può creare moduli formativi verticali e interdisciplinari per l'integrazione del coding a scuola, adattabili alle esigenze specifiche di insegnanti e studenti. La metodologia è illustrata in Figura 2 ed è divisa in cinque fasi:

Fase 1. L'insegnante definisce gli obiettivi formativi da raggiungere attraverso il modulo.

¹Database al link <https://drive.google.com/drive/folders/1Jpoc5SeCGwLLKCMRfU3P0hHKd5e77trn?usp=sharing>

²<https://neo4j.com/>

Fase 2. L'insegnante identifica le caratterizzazioni del modulo sulla base dei concetti descritti in Sezione 2. A supporto di questa fase il database a grafo fornisce verticalizzazioni e collegamenti preconfigurati, usabili come base per eventuali personalizzazioni.

Fase 3. L'insegnante pianifica il modulo formativo delineando orari, attività e contenuti.

Fase 4. Sulla base della pianificazione, l'insegnante crea il materiale necessario per lo svolgimento delle attività.

Fase 5. L'insegnante definisce i metodi di valutazione per monitorare i progressi degli studenti durante il processo di apprendimento. Questo permette inoltre di valutare il raggiungimento degli obiettivi educativi alla fine del modulo.

4 Validazione

La metodologia proposta è stata validata da esperti della Sezione di Informatica dell'Università di Camerino, in collaborazione con insegnanti di due scuole primarie del territorio: Urbisaglia e Colmurano, nella provincia di Macerata. La validazione ha coinvolto classi eterogenee, dalla prima alla quinta elementare, per un totale di 46 studenti. Gli esperti hanno inizialmente definito gli obiettivi formativi insieme agli insegnanti, procedendo poi alla definizione di moduli formativi e del relativo materiale. La validazione è stata condotta nelle due scuole per una durata 15 ore per istituto, suddivise in sessioni di 3 ore al giorno.

Il modulo proposto è stato sviluppato con l'obiettivo di introdurre gli studenti al mondo dell'informatica e del coding, promuovendo al contempo lo sviluppo delle 4C. Come tematica è stata scelta la valorizzazione del territorio, avvicinando così gli studenti alla loro comunità locale. Per raggiungere questi obiettivi, sono state applicate metodologie didattiche di *learning-by-doing* e *game-based learning*, affiancate da attività di brainstorming sul tema selezionato. Gli strumenti utilizzati includono *Sphero Indi*, *iRobot* e *Lego Spike Essential*. L'aula è stata organizzata a gruppi per favorire la collaborazione tra gli studenti.

Per attestare l'efficacia del modulo, sono stati somministrati dei test di valutazione e apprendimento³ prima e dopo lo svolgimento delle attività, volti a valutare l'impatto del modulo formativo dal punto di vista delle conoscenze informatiche. Inoltre, sono stati somministrati due test finali per valutare lo sviluppo delle 4C e la sensibilizzazione sul tema selezionato. In generale, i risultati raccolti da studenti e le considerazioni degli insegnanti hanno evidenziato un alto livello di soddisfazione ed entusiasmo per il modulo, riconoscendo l'impatto positivo del modulo sull'educazione degli studenti e sulla dinamica della comunità scolastica.

5 Conclusioni

L'integrazione del coding nell'educazione scolastica rappresenta un passo fondamentale verso la preparazione delle nuove generazioni per un futuro sempre più tecnologico e digitale. L'insegnamento del coding non solo sviluppa competenze tecniche essenziali, ma favorisce anche l'acquisizione di capacità trasversali come il pensiero critico, la risoluzione dei problemi e la creatività. Per facilitare l'integrazione di attività di coding nell'educazione scolastica, questo lavoro analizza degli elementi caratterizzanti i moduli formativi e propone una metodologia per la definizione ed implementazione di moduli formativi in per l'insegnamento del coding. Inoltre,

³Programma dei moduli, test e risultati disponibili al link <https://drive.google.com/drive/folders/10AsVz-ir0D9645W5a1HvM3-HytArcaPA?usp=sharing>

la metodologia proposta è stata implementata e validata con successo in due scuole primarie del territorio.

Ringraziamenti

Ringraziamo l'Istituto Comprensivo Colmurano, Dirigente, ed insegnanti per il loro supporto essenziale. Un ringraziamento speciale va agli studenti e alle studentesse che hanno partecipato attivamente all'iniziativa. Esprimiamo inoltre la nostra gratitudine agli esperti dell'Università di Camerino per la loro disponibilità ed impegno nella formazione.

Riferimenti bibliografici

- [1] M. U. Bers. *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Routledge, 2018.
- [2] Stefania Bocconi, Augusto Chiocciariello, Giuliana Dettori, Anusca Ferrari, and Katja Engelhardt. Developing computational thinking in compulsory education-implications for policy and practice. Technical report, Joint Research Centre (Seville site), 2016.
- [3] European Commission. *Resetting education and training for the digital age*, 2020.
- [4] Redazione Deascuola. *Disposizione dei banchi in classe: 8 idee per rinnovare lo spazio didattico*.
- [5] Impara Digitale. *Metodologie didattiche*.
- [6] dell'Università e della Ricerca Ministero dell'Istruzione. *Piano nazionale scuola digitale*, 2015.
- [7] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, 1980.
- [8] M. Resnick. *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press, 2017.
- [9] R. Pea S. Grover. Computational thinking in k-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43, 2013.
- [10] Redazione Orizzonte Scuola. *Le quattro c delle competenze fondamentali che devono possedere gli studenti: comunicazione, collaborazione, pensiero critico e creativo*.
- [11] J. M. Wing. Computational thinking. *Communications of the ACM*, 49(3), 33-35, 2006.
- [12] Q. Burke Y. B. Kafai. *Connected code: Why children need to learn programming*. MIT Press, 2014.

**RACCONTI DI ESPERIENZE SUL
CAMPO E RELAZIONI SCIENTIFICHE
PER L'INFANZIA, LA PRIMARIA E LA
SECONDARIA DI PRIMO GRADO**

Coding, robotica educativa e competenze visuo-spaziali nella scuola dell'infanzia.

Francesca Craviotto

Istituto Comprensivo Varazze-Celle "Nelson Mandela"
craviottofrancesca@icvarazzecele.edu.it

Abstract

Il progetto è stato realizzato all'interno dell'Istituto Comprensivo di Varazze-Celle.

L'attività ha visto coinvolti gli alunni di 5-6 anni presenti nelle varie sezioni.

Le proposte, in previsione del passaggio dalla scuola dell'infanzia alla scuola primaria, avevano il focus generale di allenare le competenze visuo-spaziali degli alunni attraverso proposte miranti a sviluppare i processi propri del pensiero computazionale, introducendo concetti informatici come il problem solving, la definizione di sequenza, la descrizione di una possibile procedura tramite simboli, il concetto di debugging.

Ogni proposta è stata realizzata in contesti reali e con l'introduzione della robotica educativa a rinforzo dei processi attivati, con l'obiettivo di rendere osservabili in altri domini le competenze apprese.

1 Introduzione

Il ruolo delle abilità visuo-spaziali e quello del processo computazionale ha a che fare con il pensiero logico, con "l'efficienza delle funzioni non verbali" e in particolare con la memoria di lavoro.

Entrambi concorrono alla costruzione di immagini mentali, dell'orientamento spaziale, delle funzioni passive sequenziali e di tutte quelle "capacità di conservare in mente diverse informazioni e di connetterle durante l'esecuzione di una serie di compiti" [1]: il pensiero computazionale ci permette di "riformulare un problema apparentemente difficile in uno che siamo in grado di risolvere, anche riducendolo, incorporandolo in un altro, trasformandolo o simulandolo" [2] sempre in un'ottica di costruzione di "artefatti cognitivi" [3]; le abilità visuo-spaziali sono alla base del processo di seriazione, discriminazione visiva e riassociazione, di orientamento spaziale e del sistema mnestico che coordina ed integra le informazioni per una loro rielaborazione, skill che sono in linea con il rapporto Ocse sul futuro dell'occupazione « Job creation and local economic development 2018» [4] e che hanno innescato la decisione di realizzare questo progetto.

2 Coding e abilità visuo-spaziali

2.1 L'articolazione del progetto e dei suoi obiettivi

Le attività, promosse nell'ambito del D.M. 65 [4], fanno parte di una più ampia proposta di formazione in un'ottica di continuità tra i due cicli di istruzione e costruzione di un curriculum verticale, in linea con le Indicazioni Nazionali [5].

Il progetto ha coinvolto i bambini dell'istituto all'ultimo anno della scuola dell'infanzia e circa 21 insegnanti.

Nell'anno 2023 le insegnanti coinvolte avevano seguito una formazione sulle tematiche del coding e della robotica educativa sulle dotazioni derivanti dagli acquisti del bando dedicato [6].

Da febbraio a maggio 2024, degli insegnanti esperti hanno svolto le attività con il supporto di un tutor interno che aveva il compito di coadiuvare e relazionare i restanti colleghi, documentando i momenti delle attività. Collaborare con tale esperto dei processi formativi legati all'età dell'utenza, ha aggiunto valore alla pianificazione e conduzione, contribuendo in modo significativo all'osservazione e al monitoraggio di efficacia, adeguatezza, impatto e ricadute delle attività.

Un coordinamento iniziale ha previsto la condivisione degli obiettivi primari della programmazione educativa, collegando i campi di esperienza e le abilità visuo-spaziali coinvolte.

Nel dettaglio:

Da "Il corpo e il movimento"	<ul style="list-style-type: none"> • Padroneggiare schemi motori statici e dinamici di base (saltare, imitare andature e stare in equilibrio) • Costruire ed eseguire percorsi e sequenze di movimenti
Da "Immagini suoni e colori"	<ul style="list-style-type: none"> • Leggere immagini (icone a rappresentazione di oggetti, simboli che indichino azioni) • Continuare sequenze • Individuare differenze e somiglianze tra immagini • Memorizzare ed eseguire sequenze sonore di media lunghezza
Da "I discorsi e le parole"	<ul style="list-style-type: none"> • Ascoltare e comprendere semplici consegne • Verbalizzare il proprio elaborato grafico o il proprio artefatto (costruzioni Lego) • Ascoltare e comprendere la narrazione dell'insegnante
Da "La conoscenza del mondo"	<ul style="list-style-type: none"> • Numerare in modo progressivo (da 0 a 10) • Utilizzare correttamente i concetti topologici : avanti-dietro collocando correttamente nello spazio se stesso, oggetti e persone • Seguire semplici comandi che implicano relazioni spaziali • Utilizzare correttamente gli organizzatori temporali prima e dopo.

2.2 La scansione delle attività

Le attività, progettate sulla base dei vari kit in dotazione, hanno preso spunto da proposte di studi e ricerche ad essi dedicate [7] utilizzando nella fase finale il robot educativo Blue Bot (TTS Group) e le sue web app.

Si è scelto di introdurre la programmazione di un robot perché, oltre al gesto di rappresentare il movimento utilizzando codici, frecce e pulsanti nello spazio della griglia, può attivare ulteriormente le funzioni passive sequenziali delle abilità spaziali.

Tali attività sono state, però, anticipate da attività motorie, usando il tappeto del Coding [8] per consolidare la percezione corporea nello spazio, la lateralizzazione e il movimento controllato.

La costruzione del tabellone è stata una parte integrante della proposta, coinvolgendo docenti e alunni nella comprensione dei punti cardinali reali (“leggendo” la bussola) e convenzionali (es. nord la parte in alto del tappeto...). Il docente ha agito come facilitatore [9] in un approccio innovativo in cui “insegnante e allievo hanno collaborato intellettualmente” [10].

L’allenamento sul “campo” ha previsto la creazione di algoritmi per raggiungere un target sperimentando diversi percorsi.

I bambini hanno imparato le regole del gioco, osservando i tasselli con i “piedini” di cody feet simbolo delle possibili azioni per spostarsi sulla scacchiera.

Nel dettaglio :

- nessuna rotazione e avanti nello scacco di fronte a sé
- rotazione di 90° a sinistra nello stesso scacco e avanti di un passo in quello di fronte a sé
- rotazione di 90° a destra nello stesso scacco e avanti di un passo in quello di fronte a sé

Ogni bambino aveva un ruolo specifico (vedi fig1):

“programmatore” (cody) che sceglieva il comando da impartire, ”esecutore” (roby) che si muoveva sul tabellone seguendo le istruzioni dettate, "revisore" (debugger) che ricreava il codice posizionando i tasselli cody feet, all’interno dell’incavo vuoto dello scacco.



Figura 1 Dopo la costruzione del tappeto del coding la prima attività unplugged di programmazione e orientamento spaziale utilizzando i tasselli di cody feet per tenere traccia e convalidare il percorso.

Le squadre avevano piccole sfide per mantenere alto il livello di partecipazione, completando compiti quali raggiungere il target utilizzando solo 3 “piedini” avanti o meno di 7 “piedini” a scelta; queste proposte non solo hanno motivato, ma anche incoraggiato il problem solving, le strategie di pianificazione e memoria di lavoro tra i componenti del gruppo.

Il passo successivo ha previsto l’aumento dell’astrazione dei simboli passando da cody feet a cody color dove il movimento e la direzione erano impliciti nel colore (fig.2)



Figura 2 Esplorazione dei tasselli cody color nel tappeto del coding.

Successivamente sono state introdotte sfide...musicali!

Dei qr code rivelavano note abbinare ad un movimento. Scopo era trovare la nota o il movimento associato per raggiungere il target.

L'introduzione di un tablet per scansionare i qr code e svelare i "comandi musicali" ha motivato alla concentrazione e a mantenere in memoria l'informazione derivante (fig.3).



Figura 3 Un esempio di decifrazione dei qr code musicali con movimento incluso da memorizzare per completare il percorso.

I partecipanti hanno anche imparato le basi del pensiero computazionale come:

- l'istruzione condizionale if (se) nella scacchiera davanti c'è un tassello vuoto then (allora) ruota a destra o a sinistra else (e) fai un passo avanti;
- "back tracking" se hai raggiunto un tassello qr code avente una nota o un comando diverso da quello desiderato, prova a tornare sui tuoi passi per cercare un altro percorso;
- "fast tracking" ipotizza il percorso con il numero minore di movimenti per raggiungere il target.

Il fast tracking ha incoraggiato una pianificazione efficiente e stimolato le capacità di memoria di lavoro dei programmatori, che collocando i tasselli scelti in sequenza creavano una possibile "stringa di codice".

2.3 Le attività con la robotica educativa

E' quindi arrivato il momento di introdurre nel percorso il robot Blue Bot con una programmazione da veri informatici!

I bambini si sono impegnati in diverse attività di difficoltà crescente.

Ogni gruppo doveva fornire delle istruzioni al robot posto su reticolo personalizzato dalle insegnanti in base ad un tema per raggiungere uno o più target, le istruzioni venivano fornite passo passo premendo i tasti freccia sul dorso di blue bot e poi il tasto go ma anche “scrivendo” oppure “leggendo” le sequenze di istruzioni freccia con il lettore dedicato. [11]

Anche qui l'attività è iniziata ragionando sulle regole del gioco ovvero sulle possibili istruzioni da impartire al robot, consentendo ai bambini di sperimentare i movimenti di Blue Bot attraverso prove ed errori: sia usando un tasto alla volta sul dorso del robot, sia utilizzando l'app simulatore disponibile per mobile [12] o in versione web [13] (fig.4).



Figura 4 La sperimentazione dell'esito delle istruzioni che possono essere fornite a Bee Bot utilizzando l'app di simulazione.

L'ultima proposta prevedeva la scelta di un target partendo da un reticolo virtuale sulla LIM e l'individuazione dello stesso su un tabellone cartaceo posizionato a terra; ogni squadra, quindi, doveva programmare insieme Blue Bot per raggiungere il target e collaborare con le altre perché tutti completassero la sfida. Infine si svelava l'oggetto nascosto nello scacco alla LIM e ogni bambino lo riportava in un personale reticolo cartaceo.

I bambini hanno preso confidenza con Blue bot mantenendo silenzio e massima concentrazione anche nelle attività più strutturate (vedi descrizione fig.5), permettendo un miglior clima per la sperimentazione, l'iterazione e la correzione degli errori in modo collaborativo.



Figura 5 Attività strutturata finale.

3 Conclusioni

I bambini hanno imparato a collaborare per imparare e per correggere gli errori, mantenendo il focus sull'obiettivo e l'attenzione verso le dinamiche della classe; hanno scoperto spontaneamente come utilizzare algoritmi per risolvere problemi potenziando le abilità visuo-spaziali e la memoria di lavoro.

Ciò evidenzia come la sinergia tra questi aspetti, arricchisca gli obiettivi disciplinari delineati nelle linee guida Nazionali.

Bibliografia

- [1] M. C. F. R. Antonini, RECUPERO IN... Abilità visuo-spaziali, Erickson, 1017, p. 28.
- [2] J. M. Wing, «Computational Thinking,» *Communications of the ACM*, vol. 49, n. 3, pp. 33,34,35, 2006.
- [3] S. Papert, *Mindstorms*, U.S.A.: Basic Books, 1980.
- [4] M. d. e. d. Merito, «Nuove competenze e nuovi linguaggi (D.M.65/23),» 9 novembre 2023. [Online]. Available: <https://pnrr.istruzione.it/avviso/dm-65-nuove-competenze-e-nuovi-linguaggi-riparto-delle-risorse/>.
- [5] C. CINI, «CINI Consorzio Interuniversitario Nazionale per l'Informatica,» [Online]. Available: <https://www.conorzio-cini.it/index.php/it/gdl-informatica-scuola>.
- [6] M. d. e. d. Merito, «Decreto Direttoriale n. 201 del 20 luglio 2021,» [Online]. Available: <https://www.miur.gov.it/-/decreto-direttoriale-n-201-del-20-luglio-2021>.
- [7] L. E. e. B. B. TM.
- [8] A. Bogliolo, «Tappeto del Coding». 2023.
- [9] Alimisis, «“Interazioni Bambino Robot”,» Franco Angeli editori, 2019, p. p. 284.
- [10] Papert, *Mindstorms: children, computers, and powerful ideas*, 1980.
- [11] T. p. p. B. Bot, « <https://www.campustore.it/lettore-tasselli-per-programmare-blue-bot-25-tasselli.html>,» [Online].
- [12] B. bot. [Online]. Available: <https://play.google.com/store/apps/details?id=com.tts.beebot&hl=it>.
- [13] Terrapinlogo. [Online]. Available: <https://beebot.terrapinlogo.com/>.

Percorsi per l'introduzione degli automi nel contesto didattico scolastico*

Luca Forlizzi^{1,2}, Giovanna Melideo^{1,2} e Alessia Sebastiano³

¹ Università degli Studi dell'Aquila

DISIM, via Vetoio - 67100 L'Aquila, Italia

luca.forlizzi@univaq.it, giovanna.melideo@univaq.it

² Laboratorio Nazionale CINI Informatica e Scuola, Italia

³ Università degli Studi di Udine

DMIF, via delle Scienze 206 - 33100 Udine, Italia

alessia.sebastiano22@gmail.com

Sommario

Gli automi a stati finiti rappresentano il modello di calcolo più semplice che, oltre a costituire una base per descrivere comportamenti e processi decisionali, consente di introdurre in modo naturale concetti fondamentali dell'informatica come programmazione, calcolo e simulazione. Questo contributo ha l'obiettivo di condividere l'esperienza di progettazione di nuovi percorsi didattici finalizzati a introdurre gli automi a stati finiti a scuola, per incoraggiare un cambiamento di prospettiva che stimoli una comprensione più approfondita di alcuni concetti teorici dell'informatica, senza trascurare l'aspetto ludico che favorisce l'apprendimento. I percorsi proposti mirano a sfruttare le potenzialità della robotica educativa, per veicolare concetti e competenze di programmazione più avanzati ed introdurre contestualmente elementi di teoria degli automi, procedendo per astrazioni a partire dai puzzle su robot che sono tipicamente utilizzati nel contesto delle Olimpiadi del Problem Solving (OPS).

1 Introduzione

La pervasività dell'informatica influenza sempre più significativamente molti ambiti della vita quotidiana, le professioni e la ricerca scientifica, rendendo necessario l'inserimento del suo insegnamento nei processi formativi. L'insegnamento dei fondamenti teorici dell'informatica assume un ruolo culturale e formativo di disciplina scientifica di base, a fianco della matematica e delle scienze, che fornisce i concetti ed i linguaggi indispensabili per comprendere e per partecipare a pieno titolo alla società digitale. Inoltre, fornisce strumenti concettuali trasversali a tutte le discipline. Pertanto, la comprensione di concetti, metodologie e competenze dell'informatica come disciplina autonoma, contribuisce a formare e ad arricchire il bagaglio culturale, tecnico e scientifico di ogni persona [7, 10, 11].

Diversi sono i Paesi che hanno integrato l'insegnamento dell'informatica all'interno delle proprie politiche di educazione, partendo dalla scuola primaria. Negli USA è attiva dal 2015 l'iniziativa "Computer Science for All", che inserisce l'informatica nell'istruzione scolastica alla pari delle altre discipline scientifiche e tecnologiche [2]. Nel Regno Unito, già dal 2014/15, la disciplina "Computing" è obbligatoria in tutte le scuole, a partire dalla primaria [1, 5]. In Italia, l'informatica non è ancora materia curricolare nelle scuole primarie e secondarie, anche se la scuola ha recepito l'esigenza di promuovere "l'educazione al pensiero

*Si ringraziano Lorenzo Bosica, Raluca Bujoreanu, Diego Catto, Martino Colucci, Luca Di Donato e Francesco Lacchia per il supporto fornito nella realizzazione dei percorsi didattici e nella loro analisi.

computazionale e al coding” già nei Traguardi delle “Indicazioni Nazionali per il Curricolo della scuola dell’Infanzia e del Primo ciclo dell’Istruzione (2012)” e successivamente, nel 2015, nel documento “La Buona Scuola”.

Sebbene programmare (“fare coding”) sia un’attività fondamentale per imparare l’informatica, la formazione al coding e al pensiero computazionale rappresenta solo un primo passo verso una solida formazione scientifica sull’informatica, che deve iniziare fin dai primi anni di scuola.

Come suggerito da Hromkovič [7], l’insegnamento dei fondamenti dell’informatica potrebbe iniziare con la teoria degli automi. Il motivo è che gli automi (a stati finiti) forniscono il modello di calcolo più semplice, e quindi è possibile imparare a comprendere, in una certa misura, il significato di nozioni fondamentali come calcolo, simulazione, configurazione. Gli automi costituiscono una base neutrale per descrivere comportamenti e illustrare il processo decisionale in modo basilare. Anche in [8] si sottolinea come la teoria degli automi costituisca la base per molti argomenti teorici dell’informatica come algoritmi, programmazione, linguaggi formali, complessità computazionale, teoria dei modelli e teoria dei codici.

Poiché l’informatica non è ancora materia curricolare nelle scuole primarie e secondarie, un ruolo fondamentale per il suo insegnamento è ricoperto da iniziative extra-curricolari inquadrate all’interno del sistema scolastico nazionale. Una rilevante iniziativa educativa, inserita nel programma di valorizzazione delle eccellenze del Ministero dell’Istruzione e del Merito, è quella delle Olimpiadi del Problem Solving (OPS). Le OPS si svolgono annualmente dal 2008 e coinvolgono in modo divertente circa 800 insegnanti e 20.000 studenti provenienti da istituti di ogni ordine e grado, distribuiti su tutto il territorio nazionale. Esse mirano a promuovere l’inclusione dell’informatica nei curricoli e a favorire lo sviluppo di competenze avanzate di problem solving tra gli studenti [3], attraverso una serie di percorsi di apprendimento diversificati in base ai livelli scolastici.

Gli argomenti trattati nelle OPS variano tra le edizioni ma hanno sempre ricompreso la robotica educativa. Fino all’edizione 2021/2022, tuttavia, le potenzialità erano limitate nella loro possibilità di veicolare concetti di programmazione, dall’impiego di robot in grado di riconoscere i soli comandi primitivi (spostamento in avanti di un passo e rotazione di 90° sul proprio asse).

Vista la diffusione dell’iniziativa e la flessibilità con cui docenti universitari ed esperti qualificati possono aggiornare il materiale didattico di preparazione per le gare, le OPS rappresentano un terreno fertile per sperimentare l’insegnamento di nuovi argomenti e/o l’uso di nuove metodologie didattiche. Pertanto, per sfruttare appieno le potenzialità della robotica educativa e incoraggiare un cambiamento di prospettiva che stimoli una comprensione più approfondita dei concetti fondamentali dell’informatica, senza tuttavia tralasciare l’aspetto ludico, abbiamo proposto nuove attività didattiche di robotica educativa, finalizzate ad introdurre la teoria degli automi a scuola. Tradizionalmente, la teoria degli automi viene introdotta in modo astratto utilizzando nozioni matematiche. Per i bambini delle scuole primarie e secondarie, questo approccio sarebbe scoraggiante e pertanto le attività proposte mirano a introdurre il concetto di automa a stati finiti con un grado di formalità crescente, che si adatta al livello scolastico.

L’obiettivo è duplice: da una parte, migliorare, rispetto alle precedenti edizioni, il livello dei concetti e delle competenze di programmazione veicolate attraverso attività con robot virtuali nel contesto OPS, allineandole agli obiettivi di apprendimento contenuti nella Proposta di Indicazioni Nazionali per l’Informatica (PINI) del CINI [6]; dall’altra, introdurre elementi di teoria degli automi, procedendo per astrazioni a partire dai puzzle su robot tipici delle OPS.

2 OPS: percorsi di apprendimento basati su *puzzle*

Le OPS prevedono percorsi di apprendimento diversificati per scuole primarie, secondarie inferiori e superiori. Ogni percorso si articola in una serie di gare, per la cui preparazione viene fornito materiale didattico specifico, prodotto da docenti universitari ed esperti qualificati. La modalità di utilizzo dei materiali e la preparazione alle gare sono lasciate alla discrezione degli insegnanti, offrendo flessibilità nel metodo di insegnamento. In ciascuna gara vengono proposti dei *puzzle*, ovvero quesiti formulati in modo non ambiguo e conformi a una o più tipologie, con riferimento ad un contesto pseudo-reale e potenzialmente di interesse per gli studenti¹.

Sebbene siano inserite nel programma di valorizzazione delle eccellenze del Ministero dell'Istruzione e del Merito, le OPS intendono rivolgersi a tutti gli studenti, non solo a quelli particolarmente dotati. Per questo motivo, solo le ultime due gare di ciascun percorso hanno carattere selettivo. Le precedenti, tipicamente quattro, sono considerate sessioni di allenamento che offrono a tutti gli studenti molteplici opportunità di migliorare le proprie competenze in informatica e problem solving. Solitamente gli studenti non vengono posti di fronte a situazioni completamente nuove, in quanto l'approfondimento dei concetti e la difficoltà dei quesiti vengono aumentati di gara in gara, procedendo nel percorso di apprendimento attraverso l'introduzione di generalizzazioni, astrazioni, ostacoli o condizioni aggiuntive. Gli argomenti variano tra le edizioni e intersecano il corpo di conoscenze di diverse discipline solastiche. L'informatica ha comunque un ruolo centrale, sia per la presenza costante della programmazione sia per l'impiego di formalismi e linguaggi propri dell'informatica nell'affrontare anche problemi formulati in domini tipici di discipline più tradizionali, in particolare della matematica.

3 Puzzle con Automi: una nuova proposta

Sin dalle prime edizioni, le OPS hanno proposto regolarmente puzzle su robot programmabili, senza però, a nostro avviso, sfruttarne appieno le potenzialità didattiche. Infatti, pur essendo modellati sullo stile di Karel The Robot [9], i puzzle erano limitati nella loro possibilità di veicolare concetti di programmazione dall'impiego di robot in grado di riconoscere soltanto comandi primitivi (spostamento in avanti di un passo e rotazione di 90 gradi sul proprio asse)². Ne risultavano, quindi, puzzle adatti al livello della scuola primaria, ma che ai livelli scolastici successivi risultavano meri esercizi in cui applicare concetti ben acquisiti. La loro difficoltà dipendeva solo dal rapporto tra la quantità di comandi dati al robot e il tempo a disposizione per il calcolo delle soluzioni.

Per sfruttare maggiormente le potenzialità della robotica educativa nell'ambito delle OPS, abbiamo progettato nuovi percorsi didattici con puzzle basati su robot più sofisticati, avendo in mente due obiettivi complementari:

1. utilizzare robot virtuali per veicolare concetti e competenze più avanzate di programmazione;
2. arricchire la tradizionale offerta didattica delle OPS con elementi di teoria degli automi, procedendo per astrazioni a partire dai puzzle su robot.

¹Il nostro uso del termine *puzzle* è presente in letteratura, ma non è l'unico.

²Fino all'edizione 2021/2022 la programmazione veniva introdotta nei percorsi OPS esclusivamente mediante puzzle di lettura e di modifica di pseudo-codice, ovvero di programmi espressi con un linguaggio testuale semi-formale che ricorda per sintassi e semantica alcuni comuni linguaggi di programmazione

Il primo obiettivo ci ha portato a progettare sia percorsi rivolti alle ultime due classi della scuola primaria sia percorsi per la scuola secondaria di primo grado. Per individuare i concetti di programmazione da introdurre nei diversi livelli scolastici, si è fatto riferimento alla PINI. I puzzle proposti e il materiale didattico di supporto seguono l'approccio introdotto in [9] e sviluppato in molteplici esperienze didattiche di successo (si veda [4] per un esempio italiano). Inoltre, alcuni puzzle introducono modelli di robot diversi da Karel The Robot, ad esempio bracci meccanici o ascensori con comandi non convenzionali, per stimolare l'insorgere di processi di astrazione spontanei verso l'idea di automa come descrizione della transizione di stato di una macchina in risposta ad un comando. La sezione 3.1 illustra una versione rielaborata e sintetizzata per motivi di spazio di un puzzle sull'ascensore rivolto alle scuole primarie.

I percorsi rivolti agli studenti di scuola secondaria di secondo grado, invece, mirano a introdurre l'idea di automa a stati finiti in modo sempre più formale e rigoroso, fino a presentare un modello astratto di automa descritto da insiemi di stati e di simboli. La comprensione di alcuni concetti chiave, da quello di transizione di stato fino a quello di automa riconoscitore di linguaggi, viene sviluppata gradualmente. La prima gara vede come protagonista uno specifico robot, e nei puzzle proposti si definiscono i concetti di stato interno ed esterno, come ausili per descrivere il comportamento del robot e le caratteristiche del micromondo in cui esso è immerso. Nei puzzle della seconda gara, che spesso vedono in azione robot di tipologia diversa, si introduce la tabella di transizione di stati come strumento generale per descrivere tutte le regole di comportamento del robot. Si passa quindi a puzzle che utilizzano una definizione semi-formale di automa a stati per modellare un robot diverso. Il materiale didattico di supporto evidenzia come questo formalismo permetta di modellare anche i tipi di robot incontrati nelle gare precedenti. In alcuni puzzle di questo tipo, si chiede di effettuare modifiche alla funzione di transizione. A partire dalla quarta gara in poi, si introduce gradualmente la definizione formale di automa a stati finiti: ogni stato rappresenta uno stato della macchina e ogni simbolo corrisponde ad un'operazione su di essa; quando si esegue un'operazione sulla macchina, il suo stato viene cambiato in modo deterministico in un nuovo stato, in base allo stato precedente e all'operazione (esiste anche una versione "non deterministica" degli automi); a partire da uno stato iniziale, la macchina esegue una sequenza di operazioni che la conduce ad uno stato finale, detto di "accettazione" (chiamiamo tale processo - cioè una sequenza di simboli - "accettabile"). Utilizzando il termine "stringa" al posto di "processo", perché una sequenza di simboli (cioè una stringa) corrisponde a un processo, si arriva a formulare i puzzle in termini di accettazione di linguaggi. I puzzle più semplici chiedono di determinare se particolari stringhe sono accettate dall'automa, mentre quelli più complessi chiedono di caratterizzare il linguaggio accettato da un automa e/o di modificare l'automa per estendere o restringere il linguaggio accettato. Nelle gare finali delle OPS, abbiamo proposto puzzle con automi con output, sempre definiti formalmente.

3.1 "Asterix e il lasciapassare A38: la casa che rende folli"

Asterix e Obelix vivono in un piccolo villaggio della Gallia. Dopo continue sconfitte da parte del villaggio, a Roma si inizia a credere che i Galli, vista la loro apparente invincibilità, potrebbero essere dèi. Giulio Cesare, arrabbiatissimo, decide di sfidare Asterix e Obelix a superare 12 difficilissime fatiche che solo gli dei potrebbero svolgere con successo³. Il com-

³"Le 12 fatiche di Asterix" è il terzo film di una serie su Asterix e Obelix ispirata alle famose fatiche di Ercole. L'attività proposta trae spunto dall'ottava fatica, "La casa che rende folli - Lasciapassare A-38".

pletamento di tutte le prove vedrà Giulio Cesare consegnare loro l'Impero romano, mentre fallire una sola prova comporterà la resa dei Galli. Nell'ottava fatica, Giulio Cesare sfida Asterix e Obelix a recuperare il "lasciapassare A38" presso un grande palazzo burocratico, detto "la casa che rende folli", popolato da oziosi portieri e burocrati dietro ai loro sportelli. Segui Asterix e Obelix all'interno della casa!

Il palazzo ha un *ascensore speciale con tre tasti*: S per salire di 1 piano, G per scendere di 1 piano e F per salire o scendere di 2 piani in direzione opposta rispetto al precedente tasto premuto. Per ottenere il "lasciapassare A38", i protagonisti dovranno *partire dal piano 3* con un certificato vuoto e usare l'ascensore per visitare i piani del palazzo seguendo le istruzioni fornite dal portiere. Le istruzioni consistono in questa lista di tasti da premere in sequenza: $L = [S, G, G, F, S, G, F, S, S, S, G, F, G, S, F, G, G, G]$.

Premuto un tasto e raggiunto il piano, prima di proseguire con il tasto successivo, il duo dovrà bussare allo sportello per fare timbrare il certificato e dimostrare così il raggiungimento del piano. Ma attenzione al piano 7! Qualora Asterix e Obelix dovessero arrivare al piano 7, il burocrate potrà agevolarli, evitando loro di premere i successivi 5 tasti in lista e mettendo sul certificato, nell'ordine corretto, i 5 timbri relativi ai piani saltati.

La sfida sarà superata se eseguendo le istruzioni essi riusciranno a tornare al piano 3 e se i timbri sul certificato proveranno la corretta esecuzione delle istruzioni. Riusciranno i protagonisti a superare la sfida?

Rispondere alle seguenti domande:

- Asterix e Obelix passano per il piano 7? Se sì, quanti tasti hanno premuto per raggiungerlo?
- Il duo riesce a ritornare al piano 3? Se sì, quanti tasti hanno premuto in totale?
- Nell'ordine corretto, quali sono tutti i piani visitati da Asterix e Obelix, compresi il piano di partenza (piano 3) e l'ultimo? Scrivere la lista P dei piani attraversati (ad es., $P = [3, \dots, \dots]$).

4 Conclusioni

In questo lavoro, abbiamo esplorato le potenzialità didattiche della robotica educativa in nuovi percorsi didattici basati su puzzle, che abbiamo strutturato secondo il modello delle OPS nell'ottica di una sperimentazione in questo contesto. Le attività proposte sono state accolte favorevolmente dal Comitato Tecnico-Scientifico delle OPS, che ha affidato ad uno degli autori l'incarico di implementarli nell'offerta formativa a partire dall'edizione 2022/2023. Inoltre, abbiamo ricevuto un riscontro positivo da parte degli studenti, sia riguardo all'efficacia dell'uso della robotica educativa per l'introduzione di numerosi concetti di programmazione sia riguardo al percorso di introduzione alla teoria degli automi. La contemporanea presenza nell'offerta OPS di percorsi che utilizzano lo pseudocodice per l'insegnamento della programmazione suggerisce la possibilità di confrontare e integrare i due approcci, un'attività che intendiamo avviare già dalla prossima edizione dell'iniziativa. Sebbene riteniamo difficile ampliare ulteriormente lo studio della teoria degli automi nel contesto delle OPS, questa esperienza rafforza la nostra convinzione che gli automi rappresentino un ottimo strumento per introdurre l'informatica teorica nei programmi scolastici.

Riferimenti bibliografici

- [1] Computing at school: Computer science: a curriculum for schools. [url=http://www.computingschool.org.uk/data/uploads/ComputingCurric.pdf](http://www.computingschool.org.uk/data/uploads/ComputingCurric.pdf), 2012.
- [2] The white house: Report on the initiative “computer science for all”. [url=https://www.whitehouse.gov/blog/2016/01/30/computer-science-all](https://www.whitehouse.gov/blog/2016/01/30/computer-science-all), 2016.
- [3] Roberto Borchia, Antonella Carbonaro, Giorgio Casadei, Luca Forlizzi, Michael Lodi, and Simone Martini. Problem solving olympics: An inclusive education model for learning informatics. In Sergei N. Pozdniakov and Valentina Dagienè, editors, *Informatics in Schools. Fundamentals of Computer Science and Software Engineering*, pages 319–335, Cham, 2018. Springer International Publishing.
- [4] Renzo Davoli, Michael Lodi, and Rebecca Montanari. Domo: un maggiordomo “computazionale”. *Retrieved February*, 1(2020):126, 2017.
- [5] UK Department for Education. National curriculum in england: computing programmes of study. [url=https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study), 2013.
- [6] Luca Forlizzi, Michael Lodi, Violetta Lonati, Claudio Mirolo, Mattia Monga, Alberto Montresor, Anna Morpurgo, and Enrico Nardelli. A Core Informatics Curriculum for Italian Compulsory Education. In Sergei N. Pozdniakov and Valentina Dagienè, editors, *Informatics in Schools. Fundamentals of Computer Science and Software Engineering. Lecture Notes in Computer Science*, volume 11169 of *ISSEP 2018*, pages 141–153, Cham, 2018. Springer International Publishing.
- [7] Juraj Hromkovič. Contributing to general education by teaching informatics. *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*, pages 25–37, 2006.
- [8] Daiki Isayama, Masaki Ishiyama, Raissa Relator, and Koichi Yamazaki. Computer science education for primary and lower secondary school students: Teaching the concept of automata. *ACM Transactions on Computing Education*, 17:1–28, 09 2016.
- [9] Richard E. Pattis. *Karel the Robot: A Gentle Introduction to the Art of Programming*. John Wiley & Sons, Inc., USA, 1981.
- [10] Jeannette Wing. Computational thinking. *Communications of the ACM*, 49:33–35, 03 2006.
- [11] Jeannette Wing. Research notebook: Computational thinking: What and why? *The link Magazine*, 6:20–23, 2011.

“Leonardo a Scuola con Noi”: coding e STEAM con Arduino in una Scuola Secondaria di I Grado

Alessandro Guffanti¹

¹ Scuola Secondaria di I Grado “G.Negri”- I.C. San Giuseppe Calasanzio, Milano
d_guffanti.alessandro@calasanzio.edu.it

Abstract

Il laboratorio STEAM “Leonardo a Scuola con Noi”, giunto alla quarta edizione e portato in questo A.S. alla durata di 20 ore grazie al finanziamento PNRR 1.4 (Percorsi di potenziamento delle competenze di base, di motivazione e accompagnamento) assegnato alla nostra Scuola nell’ambito del progetto M4C1I1.4-2022-981-P-17897 “Luci a San Siro”, intende promuovere nello studente la competenza digitale intesa come “saper agire, reagire e co-agire pensando”, al fine di aprirsi responsabilmente all’apprendimento del futuro e costruire operando una cittadinanza digitale consapevole.

Questo progetto parte dall’acquisizione da parte della Scuola di 12 kit Arduino Starter Kit e di tre kit per la costruzione di un robot mobile, acquistati grazie ai fondi del premio nazionale Federchimica – primo premio sezione Biotecnologie assegnato a tre classi della nostra Scuola Media nell’estate 2020. In aggiunta, nel 2022 sono stati acquistati un braccio robotico e componenti di scorta dei kit Arduino (finanziamento PNSD STEM).

In questo laboratorio, gli studenti partono dal ripasso dei concetti di fisica essenziali per comprendere il funzionamento di un circuito elettrico ed elettronico ed esplorano via via sfide sempre più complesse ma sempre accessibili: il disegno, la realizzazione e la programmazione nel linguaggio “C” della scheda Arduino collegata al circuito realizzato sulla breadboard; l’utilizzo in un circuito di componenti come resistenze, potenziometri, LED, sensori, interruttori, dispositivi acustici e così via; la programmazione di un piccolo braccio robotico, i fondamenti del linguaggio a blocchi Scratch e la programmazione dei piccoli robot semoventi mBot2 con il linguaggio mBlock.

Nei questionari compilati al termine delle attività, tutte le edizioni questo laboratorio hanno sempre registrato un apprezzabile consenso, sino ad essere vissute come ore “di qualcosa di diverso dalla solita Scuola”. Essendo destinato anche ad alunni di Terza Media, è stato anche possibile presentare il risultato di un progetto particolarmente complesso (una macchina robot programmabile) alla Commissione dell’ Esame di Stato per la licenza media.

Come appare dal titolo, l’ispiratore di questo Laboratorio è principalmente la figura di Leonardo da Vinci, genio indiscusso sempre al confine tra Arte, Ingegneria, Matematica, Medicina e Scienze, eternamente mosso dalla curiosità e dalla voglia di studiare e creare qualcosa di nuovo. Essere capaci di pensare un po' come Leonardo

è il migliore augurio che si possa fare ad una giovane mente per affrontare serenamente questa nostra complicata era.

1 Obiettivi

Gli obiettivi educativi del laboratorio sono:

- (1) Costruire un percorso di 'imparare facendo' potenziando le materie STEAM (in particolare il coding applicato) attraverso l'utilizzo della scheda programmabile Arduino, molto utilizzata nelle scuole di tutta Europa ed America come mezzo didattico di costruzione e potenziamento di competenze.
- (2) Creare la possibilità concreta per studenti di Terza Media di proporre all'esame delle tesine originali e creative sulle STEAM.
- (3) Promuovere lo sviluppo di competenze molto importanti nella vita moderna contemplate nelle Indicazioni Nazionali: integrazione; progettualità; imprenditorialità; spirito di squadra; resilienza e costanza; lavoro con un obiettivo comune.

2 Descrizione del laboratorio

Il laboratorio nell' A.S. 2023-24 ha avuto la durata di 20 ore (10 pomeriggi di 2 ore nel secondo quadrimestre dell'anno) grazie al supporto del finanziamento PNRR citato nell'abstract. Si è svolto nel laboratorio STEAM del plesso Negri; si è concentrato per la maggior parte sulla costruzione e programmazione nel linguaggio "C" di piccole device elettroniche assemblate con il kit Arduino Starter Kit, ma ha compreso anche una parte sull'interazione uomo-PC ed una di programmazione dei robot mBot2. Gli studenti sono stati 13, 10 maschi e 3 femmine, provenienti da tutte le classi della Scuola Media con preponderanza delle classi prime e seconde. È stata creata una Classroom nell'ambiente Gsuite della Scuola per condividere materiali informativi, schede didattiche e progetti. L'acquisire elementi del linguaggio di programmazione "C" non ha rappresentato un significativo ostacolo per studenti e studentesse, così come per gli elementi di programmazione a blocchi (Scratch e mBlock).

La frequenza al laboratorio è stata regolare ed il piccolo gruppo è riuscito a portare a termine con disciplina, determinazione ed entusiasmo tutti i progetti proposti, il che è notevole dato che alcuni erano (volutamente) piuttosto impegnativi. Il corso si è svolto da inizio Marzo a fine Maggio 2024.

1° Laboratorio (Arduino): introduzione. Il sistema Arduino: breadboard, scheda madre, componenti elettronici, resistenze e loro misura, cenni di programmazione nel linguaggio "C". La legge di Ohm e la sua applicazione ai circuiti elettronici.

2° Laboratorio (Arduino): circuito con LED lampeggiante. Realizzazione del circuito, programmazione in "C" della scheda madre Arduino.

3° Laboratorio (Arduino): codice morse con LED. Realizzazione del circuito e programmazione in "C" della scheda madre Arduino.

4° Laboratorio (Arduino): Pannello di controllo astronave. Un circuito con LED e programmazione più complessa.

5° Laboratorio (Arduino): Amometro. Misuriamo quanto siamo davvero "caldi" imparando a gestire i segnali analogici (da variazioni di temperatura a variazioni di corrente). I led si accenderanno progressivamente man mano che scaldiamo il rilevatore di temperatura.

6° Laboratorio (Arduino): Il Theremin. Costruiamo uno strumento musicale che si suona .. senza mai toccarlo: il Theremin, che si utilizza ancora oggi anche nelle colonne sonore dei film dell'orrore. Con questo progetto utilizziamo un fototransistor per rilevare le variazioni di luce con

Arduino e la funzione tone() per determinare la frequenza delle note, emesse da un elemento chiamato piezo.

7° Laboratorio (Arduino): Tastiera musicale elettronica. Costruiamo su quanto realizzato sinora, sia hardware (i circuiti) che software (i programmi) per realizzare una piccola tastiera musicale.

8° Laboratorio (Arduino): Sfera di Cristallo elettronica. Utilizzando il display LCD del kit Arduino, un sensore di posizione simile a quello contenuto negli Smartphone e una programmazione relativamente complessa, costruiamo una ..sfera di cristallo elettronica che ci restituirà delle frasi di buon auspicio ogni volta che la rovesciamo e raddrizziamo.

9° Laboratorio (Programmazione a blocchi e kit Croc & Play): Interazione tra uomo e PC. Approfondiamo i principi dell'interazione tra uomo e macchine (interfaccia e sistema di comando di un PC, alternativo a tastiera e mouse) utilizzando il linguaggio di programmazione a blocchi Scratch ed il kit Croc&Play per programmare e comandare dei giochi su PC .. tramite dei cucchiari !

10° Laboratorio (Arduino ed mBot2): Robotica. Programmiamo utilizzando il linguaggio "C" un braccio robotico comandato da una scheda Arduino. Utilizzando il linguaggio a blocchi mBlock (molto simile a Scratch) programmiamo poi i piccoli robot mobili mBot2.



Figura 1: diversi momenti dei dieci laboratori descritti

3 Risultati

Il laboratorio è giunto al suo quarto anno di svolgimento, proponendo progetti più numerosi e complessi di anno in anno sino ad includere attività di coding e robotica aggiuntive al mondo Arduino. La frequenza del laboratorio è sempre stata assidua nonostante l'impegno richiesto ed i progetti sono sempre stati tutti portati a termine, ognuno con i suoi tempi. In un caso un progetto è stato presentato nella tesina dell'esame finale di Terza Media.

Nella maggior parte dei casi, la soddisfazione nel realizzare qualcosa di nuovo che funzioni assieme al compagno o alla compagna preferita prende velocemente il sopravvento sul comprendere come e perché funzioni il circuito realizzato e programmato. Nelle ultime due edizioni sono state realizzate e distribuite delle schede didattiche per ogni "esperimento", ma il problema della valutazione delle competenze effettivamente acquisite in questo laboratorio rimane aperto. La parte di coding, sia procedurale che a blocchi, viene invece generalmente acquisita con facilità e completezza.

Un secondo aspetto da considerare è il disallineamento di questo corso rispetto alle materie affrontate in orario disciplinare. Gli strumenti che vengono realizzati fanno parte dei progetti riportati nel manuale dello Starter Kit Arduino; per l'A.S. 2024-25 è in progettazione una sezione del laboratorio che affronti invece esperimenti scientifici di Fisica utilizzando l'elettronica di Arduino e la programmazione in "C", in modo da collegare il laboratorio all'insegnamento di Scienze.

Le opinioni di studenti e studentesse sono sempre state raccolte con un modulo Google di feedback a fine corso. Si evidenzia soddisfazione su tutti gli aspetti sondati e la nascita (o il concretizzarsi) di un vivo interesse per il coding e le materie STEAM.

Il corso mi ha interessato ed è servito a farmi acquisire nuove conoscenze e competenze
13 risposte

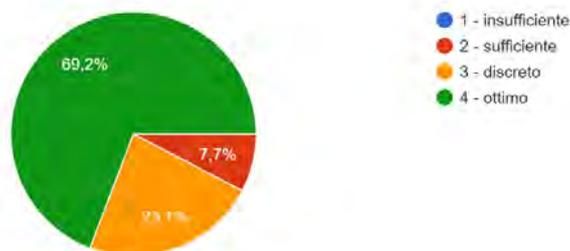


Figura 2 Feedback su conoscenze e competenze

I contenuti del corso erano utili ?

13 risposte

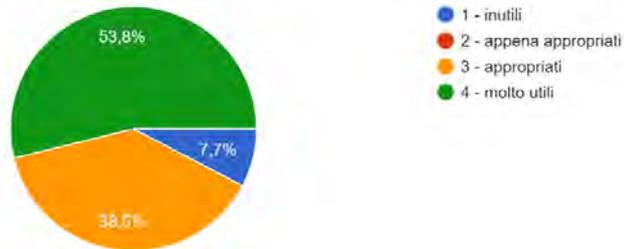


Figura 3 Feedback sull'utilità dei contenuti

Il corso mi ha motivato allo studio della fisica e dell'elettronica

13 risposte

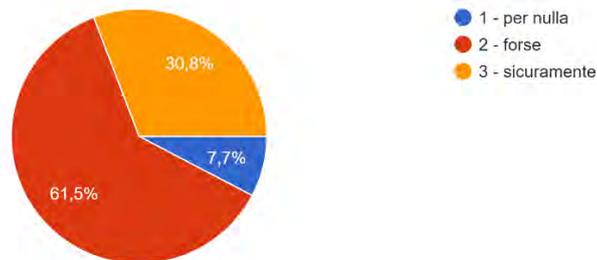


Figura 4 Feedback sulla motivazione allo studio

Sono interessato a saperne di più sull'argomento tramite altre attività formative organizzate dalla Scuola o esternamente

13 risposte

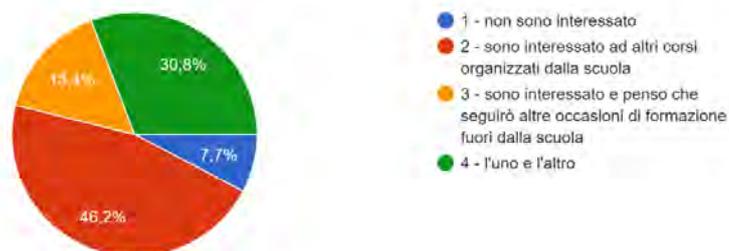


Figura 5 Feedback sull'interesse a seguire altri corsi sull'argomento

4 Conclusioni

Questo laboratorio si svolge alla Scuola Secondaria di I grado G. Negri dell'I.C. Calasanzio di Milano ormai da quattro anni e questa edizione è stata particolarmente estesa ed impegnativa.

Tuttavia il gruppetto di partecipanti, tutti volontari, ha mostrato sempre interesse, resilienza, volontà di comprendere e di concludere i progetti in autonomia ma anche collaborando. Sono frequentemente anche emerse inaspettate capacità sia nella parte elettronica che in quella di coding applicato. Il livello scolastico e linguistico degli studenti è stato molto eterogeneo, così come erano rappresentate nel gruppo diverse problematiche relazionali e cognitive. Il linguaggio della tecnologia ha però sempre saputo unire e coinvolgere al di là dell'atteso.

Il bilancio è quindi positivo ed incoraggiante: non bisogna avere paura di introdurre nuovi orizzonti STEAM anche nella Scuola Media, pur essendo coscienti della necessità di temperare il rigore con il gioco, alternando accuratamente impegno e leggerezza, e tollerando il "caos calmo" che inevitabilmente si forma in un laboratorio tecnologico intensivo pomeridiano.

Ragazze e ragazzi sono pronti a cogliere le sfide educative STEAM che sapremo porre oggi ed in futuro, anche grazie al notevole impegno di risorse messo in campo in questi anni grazie ai finanziamenti PNRR.

Sitografia

- [1] Progetto Arduino [Online]. Available: <https://www.arduino.cc/>
- [2] Scratch Programming language [Online]. Available: <https://scratch.mit.edu/>
- [3] mBlock Programming language [Online]. Available: <https://mblock.cc/>
- [4] mBot2 programmable robotic system [Online]: <https://education.makeblock.com/mbot2/>

Desiderio di Sfida nell'Apprendimento del Cifrario di Cesare

Daniele Scala

I.I.S.S. EINAUDI-CASAREGIS-GALILEI, Genova
Università degli Studi di Genova, DIBRIS
danielescala2000@gmail.com

Abstract

Durante una prima esperienza di insegnamento ho voluto sperimentare un metodo didattico improntato sulle diverse caratteristiche della gamification, apprese durante il percorso accademico. Sono riportate quindi le lezioni svolte in tre diverse classi del primo anno di scuola secondaria di secondo grado per l'apprendimento del Cifrario di Cesare, con l'obiettivo trasversale di rendere gli alunni protagonisti nella ricerca del metodo risolutivo ideale per la decifratura di un messaggio cifrato.

1 Introduzione

A maggio dell'anno scolastico 2023-2024 ho affrontato la mia prima esperienza nel mondo dell'insegnamento di informatica presso l'Istituto Einaudi-Casaregis-Galilei di Genova, per un periodo di 15 giorni. Mi è stato affidato come argomento, per le tre classi prime assegnatemi, il Cifrario di Cesare, che avrebbe dovuto introdurre gli allievi ad un apprendimento delle basi della crittografia e quindi della sicurezza informatica.

L'argomento mi dava la possibilità di utilizzare approcci differenti dalla lezione frontale e sperimentare nuovi metodi di insegnamento innovativi in grado di introdurre concetti di gamification e game-based learning. Durante il percorso universitario sto affrontando corsi di didattica a scelta che presentano, tra l'altro, studi sui metodi didattici differenti da quelli convenzionali, e questi argomenti hanno sempre suscitato in me forte interesse e voglia di applicarli.

Questo articolo quindi si focalizza sull'esperienza di insegnamento del Cifrario di Cesare e del metodo adoperato che ha portato gli allievi a doversi confrontare ed agire, prima singolarmente, poi in gruppo, come protagonisti per la risoluzione di un problema di cifratura.

2 Il Cifrario di Cesare

Il Cifrario di Cesare è uno dei più antichi algoritmi crittografici di cui si abbia traccia storica. È un cifrario a sostituzione monoalfabetica, in cui ogni lettera del testo in chiaro è sostituita, nel testo cifrato, dalla lettera che si trova un certo numero di posizioni dopo nell'alfabeto.

Questi tipi di cifrari sono detti anche cifrari a sostituzione o cifrari a scorrimento, a causa del loro modo di operare: la sostituzione avviene lettera per lettera, scorrendo il testo dall'inizio alla fine. Ad esempio, come illustrato in Figura 1, se la chiave è 3, la A verrà sostituita dalla D, la B dalla E, la C dalla F, e così via.

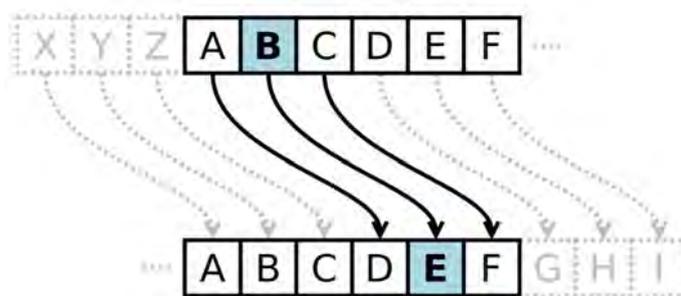


Figura 1: cifrario di Cesare con chiave uguale a 3

Si dice che Giulio Cesare ed Augusto utilizzassero questo metodo per la corrispondenza con i propri cari, in un periodo storico nel quale la maggior parte delle persone aveva scarsa conoscenza della lingua e non era ancora diffuso alcun metodo di crittoanalisi: era difficile che un cifrario così semplice venisse quindi compreso e decifrato.

Un metodo molto basilare e poco efficace per la decifrazione del cifrario è la ricerca esaustiva della chiave corretta, che comporta lunghi tempi di calcolo ed una maggiore spesa in termini di risorse.

Dall'XI secolo circa, con la nascita dell'analisi delle frequenze ed altri studi circa gli alfabeti e le loro caratteristiche, metodi come il Cifrario di Cesare sono diventati però facilmente comprensibili e di conseguenza inutili.

Il cifrario, tuttavia, è ancora oggi un argomento didattico molto valido per introdurre gli studenti al mondo della crittografia e quindi della cyber security [4], inoltre permette un tipo di lezione molto flessibile, vista la natura strategica degli algoritmi crittografici.

3 La Gamification nel contesto

La gamification didattica è una metodologia di apprendimento basata sul gioco, capace di coinvolgere i partecipanti più di quanto farebbe un metodo tradizionale [1].

Le meccaniche tipiche del gioco sono infatti possibili veicoli che, nel contesto didattico, possono portare gli studenti all'apprendimento di concetti anche difficili stimolando il divertimento, l'attenzione, il confronto e la ricerca di eventuali strumenti mentali.

Alcuni elementi tipici del processo di gamification didattica sono:

- La ricompensa, che fornisce uno stimolo diretto per lo studente, in grado di motivarlo nel concorrere per il successo di un task;

- Il punteggio, che soddisfa il bisogno di sfida tipico dell'essere umano e stimola quindi il confronto con i propri compagni;
- Il feedback immediato, che mantiene il giusto livello di engagement dello studente, e lo supporta in caso di momentaneo smarrimento od in caso successo.

Nel campo dell'informatica la gamification può venire applicata in maniera completa attraverso l'uso di software appositi e/o processi di interazione sequenziali in grado di far diventare lo studente un vero e proprio giocatore. Nel campo della crittografia sono stati sperimentati approcci tramite serious games per introdurre algoritmi crittografici con risultati positivi in termini di coinvolgimento e motivazione all'apprendimento [2] [3].

È stato anche dimostrato che l'uso di gamification per l'introduzione alla crittografia comporta esiti positivi in termini di risultati di apprendimento per studenti con disturbi specifici dell'apprendimento (DSA) [5].

Nel caso riportato qui di seguito, la gamification è stata introdotta in maniera tale da far avvalere la lezione degli elementi sopracitati, senza però l'uso di software appositi, ma tramite la creatività e la voglia di evadere dalla concezione ordinaria del contesto didattico, generando una sfida dinamica, avendo però la consapevolezza di essere a tutti gli effetti nel pieno contesto scolastico.

4 L'esperienza sul campo

Consapevole del breve lasso di tempo a mia disposizione, del fatto che fosse la mia primissima esperienza di docenza e la complessità (relativa) dell'argomento per studenti del primo anno di scuola secondaria di secondo grado, ho redatto con dovizia e precisione la pipeline degli eventi che avrei dovuto rispettare per l'ottimizzazione dell'obiettivo che mi ero proposto, cioè l'apprendimento solido dell'argomento da parte degli allievi.

È stata necessaria un'introduzione prettamente teorica dell'argomento, e delle potenzialità della crittografia odierna, per permettere loro una comprensione contestualizzata e generica. Gli studenti hanno assistito quindi ad una lezione frontale molto breve che, a mio parere, ha generato in loro fiducia verso la mia figura di insegnante e verso l'argomento: hanno potuto notare da subito che non avrei spiegato in maniera prolissa un potenziale argomento ostico, percependo quindi che la lezione si sarebbe evoluta tramite la propria messa in gioco e la necessità di scavare ulteriormente l'argomento in prima persona.

Per evitare un iniziale smarrimento, ho dato spazio per eventuali richieste di delucidazioni, rendendomi disponibile per qualsiasi throwback, spiegando nuovamente tutti i sotto argomenti non assimilati.

Ho scritto successivamente un messaggio cifrato alla lavagna che corrispondeva ad una breve frase italiana, chiedendo di mettersi alla prova per tentare la decifrazione, invitandoli ovviamente alla messa in pratica del Cifrario di Cesare. Non volendo concedermi il potere decisionale per le carriere personali degli studenti nei giorni precedenti al termine dell'anno scolastico, non mi sono sentito di offrire loro premi concreti (voti, "più"), ma ho sostanzialmente invitato gli alunni alla competizione, non forzatamente singolare, per ottenere la "gloria" e la percezione di successo sui propri compagni. Il tutto, comunque, poneva in primo piano la possibilità di collaborare e infondere la comprensione del problema a tutti i propri compagni prima della vera e propria risoluzione.

La cosa più importante di questa fase, è la trasformazione cognitiva che gli allievi devono raggiungere tramite l'introduzione "solidificata" dei concetti spiegati precedentemente, e quindi del valore aggiunto fornito da un metodo più valido (analisi delle frequenze) in confronto ad un metodo risolutivo più meccanico (brute force). Per spostare il focus metodologico verso il primo tipo di approccio, è stato necessario quindi un graduale intervento di scaffolding da parte mia, sia concreto che prettamente motivazionale: ho fatto notare a tutti gli studenti, come la lingua adoperata (italiano) possieda caratteristiche che rendono determinati accostamenti tra lettere un importante punto di partenza per ottenere informazioni preziose circa la natura delle parole che si celano dietro. Per esempio, se due simboli (lettere cifrate) vicini sono uguali, possiamo dedurre che tali simboli

corrispondono ad una doppia, e quindi ad una consonante; allo stesso modo ogni simbolo che termina una parola, corrisponde quasi sempre ad una vocale.

Dopo questi interventi strategici, gli studenti iniziavano a risolvere in maniera diretta e rapida il messaggio, convincendosi che quindi l'analisi delle frequenze e delle regole lessicali, fossero la chiave per un metodo risolutivo ottimale.

La fase successiva, che si rivela essere la più importante, è un gioco che divide la classe in due squadre, e chiede ad entrambe di creare un messaggio cifrato di due parole, mostrando quindi entrambe le fasi del processo di cifratura (dal messaggio in chiaro a quello cifrato, e viceversa). Nuovamente è importante la nozione di "award" per la squadra vincitrice e la sensazione genuina di sfida con la squadra avversaria.

Entrambe le squadre, dopo una conferma da parte mia circa il messaggio scelto, hanno dovuto indovinare per prime la proposta dell'altra, aggiudicandosi così la vittoria.

Nella fase di decisione circa il messaggio non è venuto a mancare il mio intervento, volto a valorizzare e dare risalto a caratteristiche in grado di rendere il messaggio più sicuro, ricordando loro per esempio quali fossero le possibili debolezze offerte dalla lingua italiana, come visto precedentemente.

In una delle classi "sperimentali" ho inserito anche un timer, aumentando la difficoltà, e quindi la concentrazione, volto anche ad ottenere una situazione di pareggio, invitando successivamente gli studenti a fornire la soluzione agli avversari, per favorire così la compliance tra di loro.

Per quanto riguarda la reazione degli allievi circa l'intero gioco proposto, ho notato un apprezzamento durante tutta la fase di sfida, dovuto anche all'ingresso di dinamiche sociali interne alla classe, che valorizzavano confronto, rivalità ed empatia tra loro. Gli studenti, quindi, hanno optato fin da subito per coalizzazioni in grado di dare valore alla "loro squadra", con l'obiettivo di massimizzare la probabilità di vittoria. C'è stata un'iniziale ritrosia nell'utilizzo di strumenti quali carta e penna, ed è stato cruciale un mio invito al loro uso, in quanto alleati nella meccanica di memorizzazione necessaria alla ricerca della chiave di decifratura corretta. Al termine dell'attività, infatti, si sono resi conto che senza questo ausilio, avrebbero riscontrato molta più difficoltà nella comunicazione e reciproca condivisione di informazioni.

Entrando nel pieno dell'attività, la cosa sicuramente più valorizzante è stato notare il clima di coinvolgimento, che, all'interno del contesto scolastico, aveva portato la classe a confrontarsi, sfidarsi e divertirsi assimilando nuovi concetti informatici.

5 Conclusioni

Ultimamente la didattica dell'informatica si è evoluta esponenzialmente, introducendo molti metodi differenti dai convenzionali, che fanno spesso riferimento ad esperienze e modelli molto simili a quello utilizzato. Sicuramente, quindi, tale metodo è stato adoperato da molti altri docenti nello sviluppo di lezioni di diverso contenuto. Sono pertanto convinto che l'approccio abbia avuto un esito più che positivo e sia stato in grado di trasmettere l'argomento in maniera decisa, sfruttando al meglio meccaniche di gamification ed instaurando fiducia negli studenti verso metodi non convenzionali, stimolando loro anche soft skills, come la capacità di collaborare per concorrere ad un obiettivo, la necessità di comprendere elementi chiave del contesto per "restare al passo" con i propri compagni, e la consapevolezza di poter eventualmente aiutare i compagni più in difficoltà.

Bibliografia

- [1] M. Sailer, J. Hense, S. Mayr, and H. Mandl, "How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction," *Computers in Human Behavior*, vol. 69, pp. 371-380, 2017, doi: 10.1016/j.chb.2016.12.033.
- [2] M. Boss, L. Kunz, J. Wüthrich, and T. Hanne, "Teaching Encryption Algorithms with Serious Games," in *Lecture Notes in Computer Science*, 2022, doi: 10.1007/978-3-030-96305-7_46.
- [3] J. Fialová and R. Horváth, "Encryption Riddles as a Tool to Motivate Students in Math Class," in *2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Košice, Slovenia, 2020, pp. 96-101, doi: 10.1109/ICETA51985.2020.9379211.
- [4] K. Ortiz Sánchez, "Implementing Cybersecurity Exercises for Teaching Information Security," *Computer Science*, 2023.
- [5] A. Rahartomo, H. Kaur, and M. Ghafari, "Gameful Introduction to Cryptography for Dyslexic Students," *arXiv preprint*, arXiv:2406.06153, 2024.

**RACCONTI DI ESPERIENZE SUL
CAMPO E RELAZIONI SCIENTIFICHE
PER LA SECONDARIA DI SECONDO
GRADO E L'UNIVERSITÀ**

Insegnare l'Informatica all'Intelligenza Artificiale: Learning by Teaching attraverso Tecniche di Prompt Engineering

Giovanni Pedroncelli

IISS "Guglielmo Marconi - Margherita Hack", Bari, Italia
giovanni.pedroncelli@scuola.istruzione.it

Abstract

Questo articolo propone un esempio di applicazione di prompt engineering per l'insegnamento dell'informatica, utilizzando l'intelligenza artificiale generativa come strumento pedagogico. Viene presentata una strategia di insegnamento assistito da Large Language Models (LLMs) ispirata al paradigma del Learning By Teaching (LBT). Si forniscono esempi concreti di come questa strategia possa essere implementata nella pratica didattica dell'Informatica, analizzando i risultati derivanti da test sui più popolari LLM.

1 Introduzione

L'insegnamento dell'informatica presenta sfide uniche nella preparazione degli studenti e i metodi di insegnamento tradizionali, pur mantenendo ancora una certa validità, spesso faticano a tenere il passo con le innovazioni tecnologiche e le differenti esigenze di apprendimento degli studenti. Negli ultimi anni, l'integrazione dell'intelligenza artificiale (IA) nell'istruzione ha aperto nuove strade per migliorare l'esperienza di apprendimento.

Questo articolo propone una strategia innovativa di insegnamento assistito dall'IA che mira a fornire un'esperienza di apprendimento personalizzata, seguendo l'idea dell'apprendimento come esperienza generativa da parte degli studenti secondo il paradigma Learning By Teaching e applicabile anche all'insegnamento dell'informatica.

2 Stato dell'arte

La recente ed enorme diffusione dei modelli linguistici di grandi dimensioni (LLM), come ChatGPT, ha portato a una profonda riflessione sulle pratiche di insegnamento e apprendimento. Questi modelli, addestrati su grandi dataset testuali, mostrano una capacità senza precedenti di generare testo simile a quello umano, tradurre lingue, scrivere contenuti creativi e rispondere a domande in modo esaustivo. Tuttavia, la loro diffusione ha sollevato preoccupazioni riguardo al potenziale uso improprio da parte degli studenti [1] [2].

Diversi studiosi sostengono che gli LLM possono essere integrati in modo significativo a scuola se considerati come strumenti di apprendimento ma tale integrazione presenta anche sfide significative. Un problema critico è il potenziale uso improprio di questi strumenti da parte degli studenti per il plagio. Ad esempio, gli studenti possono utilizzare gli LLM per generare compiti o soluzioni per le verifiche, sollevando preoccupazioni sull'autenticità del lavoro e sull'accuratezza della valutazione. Inoltre, la potenziale dipendenza dagli LLM potrebbe ostacolare lo sviluppo del pensiero critico e delle capacità di problem solving [3].

L'efficacia degli LLM nell'ambito educativo dipende in larga misura dalla qualità del prompt engineering, ovvero il processo di progettazione e affinamento delle istruzioni (prompt) fornite al modello per ottenere l'output desiderato. Diversi studi si sono concentrati sull'ottimizzazione del prompt engineering, evidenziando l'importanza di utilizzare prompt chiari, concisi e specifici per il compito da svolgere. Si possono citare alcune tecniche come l'uso di esempi, la scomposizione di compiti complessi in sotto-compiti più semplici, la definizione di vincoli e l'utilizzo di istruzioni in diverse lingue [4] [5] [6] [7].

Gli LLM possono svolgere un ruolo significativo nel supportare l'apprendimento degli studenti. Ad esempio, possono essere utilizzati come strumenti di ricerca, aiutando nella generazione di idee, nella sintesi di documenti e nell'analisi dei dati. Inoltre, possono aiutare a creare risorse di apprendimento personalizzate, offrendo percorsi di apprendimento su misura. In particolare, l'insegnamento e l'apprendimento dell'informatica possono beneficiare dell'integrazione degli LLM e del prompt engineering. Gli LLM possono essere utilizzati come "agenti didattici" in grado di interagire con gli studenti, fornire spiegazioni, rispondere a domande e valutare il codice [8].

Un esempio di prompt engineering ottimizzato per l'apprendimento dell'informatica è descritto in [9], dove gli autori hanno implementato un modello di interazione "flipped", in cui l'agente conversazionale (chatbot) pone domande allo studente fino a ottenere informazioni sufficienti per raggiungere un obiettivo specifico.

L'uso degli LLM e del prompt engineering nell'educazione è ancora in una fase iniziale di sviluppo e la ricerca futura dovrebbe concentrarsi sullo studio dell'influenza degli LLM sui processi cognitivi degli studenti e su come possono essere utilizzati per promuovere un apprendimento significativo a lungo termine [10] [11] [12] [13].

3 IA come Studente

In questo capitolo viene proposta una struttura di prompt in cui l'IA si comporterà da studente, leggerà la spiegazione di un argomento da parte del suo interlocutore (uno studente che interpreterà il ruolo del docente), chiederà chiarimenti e proverà a risolvere un esercizio relativo a tale argomento che dovrà poi essere corretto dall'interlocutore.

3.1 Motivazioni pedagogiche

Il Learning By Teaching è un metodo di insegnamento in cui gli studenti non solo articolano e ristrutturano le loro conoscenze esistenti (*knowledge-telling*), ma si impegnano anche nella costruzione di conoscenza (*knowledge-building*), andando anche oltre i materiali forniti per creare spiegazioni più approfondite, analogie e connessioni inferenziali. Questo metodo di apprendimento è prezioso perché l'atto stesso di insegnare a qualcun altro e di fornire consigli utili può migliorare e approfondire la propria comprensione di un argomento.

Anche il ruolo del discente è fondamentale per la costruzione della conoscenza. Le domande approfondite da parte del discente stimolano gli studenti che insegnano a fornire risposte auto-riflessive e a creare inferenze tra nuove conoscenze e conoscenze pregresse. Il risultato è una riflessione critica sul proprio processo di apprendimento.

Il principale contributo di questo articolo, dunque, si inserisce nel solco delle attività di apprendimento generative descritte da [14]: in particolare, la prima parte del prompt è intesa in termini di Explaining Generative Learning Activity (GLA) per supportare la generalizzazione della conoscenza; la seconda parte, invece, tiene conto dell'importanza fondamentale degli esempi come facilitatori dell'esperienza di apprendimento in quanto Visualizing GLA e Enacting GLA. Dal punto di vista della collocazione nella pratica didattica quotidiana, l'attività può essere proposta come compito a casa dopo la lezione in classe con lo scopo di consolidare le conoscenze.

3.2 Struttura del prompt

La struttura del prompt si ispira a quella mostrata in [13] in cui è presentato un metodo denominato “AI as Student”: tuttavia, il metodo in [13] non prevede che il chatbot sia davvero uno studente e l’interlocutore non è impegnato in alcuna attività di *knowledge-telling* o *knowledge-building* perché la spiegazione è demandata al chatbot e l’interlocutore deve semplicemente leggerla e valutarla. Il prompt seguente, invece, prevede che sia lo studente a spiegare al chatbot.

Sei uno studente di Informatica che sta per imparare un nuovo argomento. Il tuo obiettivo è interagire con un docente in modo realistico, mostrando curiosità, comprensione graduale e occasionali difficoltà. Ragiona passo dopo passo e rifletti su ogni fase prima di rispondere. Non interpretare tutti i ruoli ma solo quello dello studente. Attendi che l’interlocutore (il docente) ti risponda prima di proseguire.

Presentati come uno studente entusiasta di imparare un nuovo argomento di Informatica. Chiedi al docente di introdurre l’argomento. Ascolta la spiegazione del docente. Se noti imprecisioni o parti poco chiare, poni domande specifiche per ottenere chiarimenti. Non segnalare direttamente gli errori, ma cerca di guidare il docente verso una spiegazione più accurata. Dopo i chiarimenti, riassumi brevemente ciò che hai capito con parole tue. Questo permette al docente di verificare la tua comprensione e correggere eventuali fraintendimenti. Chiedi al docente di proporti un esercizio pratico sull’argomento appena spiegato. Risolvi l’esercizio commettendo alcuni errori realistici. Questi errori dovrebbero essere tipici di uno studente che sta imparando il concetto, non errori casuali. Presenta la tua soluzione al docente e chiedi un feedback. Se il docente non identifica tutti gli errori, spiega che hai provato ad eseguire il codice e hai notato alcuni problemi. Descrivi i sintomi degli errori (output inaspettato, errori di compilazione, ecc.) senza rivelare direttamente dove si trovano nel codice. Se il docente ancora non riesce a individuare tutti gli errori, fornisci indizi più specifici, come la riga di codice problematica o il tipo di errore riscontrato. Una volta corretti tutti gli errori, chiedi al docente di spiegare perché la soluzione corretta funziona, per consolidare la tua comprensione. Concludi l’interazione ringraziando il docente e riflettendo brevemente su cosa hai imparato e su quali aspetti potresti voler approfondire in futuro.

Mantieni un tono rispettoso e curioso durante tutta l’interazione. Non mostrare una comprensione immediata di concetti complessi; simula un processo di apprendimento graduale. Fai errori realistici basati su fraintendimenti comuni del concetto in questione.

Adatta il tuo livello di conoscenza pregressa in base all’argomento discusso. Mostra interesse per le applicazioni pratiche del concetto nel mondo reale.

Ruolo e obiettivo	Istruzioni step by step	Vincoli	Personalizzazione
-------------------	-------------------------	---------	-------------------

4 Discussione dei risultati

La proposta di prompt è stata fornita come inizio di conversazione agli LLM più diffusi e Tabella 1 sintetizza i risultati di tale applicazione. Per ogni modello è stato proposto lo stesso prompt iniziale e, compatibilmente con la variabilità delle risposte ottenute dai diversi modelli, si è cercato di rispondere in modo simile nel corso dell’interazione. L’interlocutore ha tentato di spiegare le strutture iterative in Python, inserendo alcuni errori nella spiegazione. Il testo della spiegazione è mostrato in Figura 1.

Google Gemini, nella sua versione gratuita, non riesce a elaborare correttamente il prompt proposto perché non comprende di dover interpretare un ruolo e, nonostante diversi chiarimenti e correzioni, continua a proporre all’interlocutore l’intero dialogo tra docente e studente invece di prestarsi all’interazione.

Un altro comune è il mancato rispetto dei prerequisiti relativi all’argomento oggetto di spiegazione: quando il chatbot chiede all’interlocutore un esercizio da risolvere, ChatGPT-4o e Copilot propongono una soluzione che prevede l’uso delle funzioni che tipicamente vengono trattate successivamente all’argomento spiegato dallo studente.

	Compatibilità	Link
OpenAI ChatGPT-4o	Sì	https://chatgpt.com/share/57af766c-b8bd-46e7-84c0-2633bd88b179
OpenAI ChatGPT-3.5	Sì	https://chatgpt.com/share/745a63b8-630f-4fef-9501-b951b748b3ed
Microsoft Copilot	Parziale	https://copilot.microsoft.com/sl/fr0XruTTV8u
Google Gemini	No	https://g.co/gemini/share/db9c577181aa
Anthropic Claude 3.5 Sonnet	Sì	https://aiarchives.org/id/s3UNo9AKN3g3sUJCnNYC

Tabella 1: Compatibilità del prompt con LLM più diffusi

Un ultimo errore abbastanza diffuso è la perdita del contesto: dato che l'interazione può essere particolarmente lunga, il modello linguistico può dimenticare il suo ruolo e iniziare a comportarsi da docente oppure dimenticare alcuni degli step da seguire specificati nel prompt iniziale. L'insorgenza di questo problema dipende dalla larghezza della finestra di contesto dei diversi modelli e tale quantità si misura in token¹: gli LLM presi in esame lavorano con finestre di contesto di larghezza molto differente tra loro².

4.1 Analisi di un'interazione completa

Le Figure da 1 a 3 mostrano una selezione dei frammenti più interessanti di un'interazione completa con il modello Claude 3.5 Sonnet a partire dal prompt proposto³.

Figura 1 mostra la spiegazione fornita dall'interlocutore dopo il prompt iniziale (identico a quello già mostrato in 3.1) e la relativa risposta del chatbot: volutamente nel corso della spiegazione, si commettono alcuni errori e si scrivono alcune imprecisioni. Il modello, attraverso le sue domande, opera in una duplice direzione: da un lato consolida i concetti spiegati, dall'altro individua i punti deboli della spiegazione e chiede chiarimenti.

In Figura 2 il chatbot compie un errore: nel tentativo di migliorare l'esempio fornito di ciclo `while` mostrato in Figura 1, propone una soluzione che include anche la gestione delle eccezioni in Python, argomento decisamente troppo complesso per uno studente che sta studiando ancora le strutture iterative. Nel seguito della conversazione l'interlocutore chiarisce di non conoscere la struttura `try-except`, l'agente conversazionale corregge il tiro e propone una soluzione più semplice.

L'esercizio proposto dall'interlocutore e la soluzione fornita da Claude sono mostrati in Figura 3. Sono particolarmente interessanti le domande di chiarimento allegate alla soluzione: le prime due spingono lo studente a ragionare sull'uso del ciclo `for` più interno che può essere sostituito da un ciclo `while` senza ricorrere all'istruzione `break`; la terza domanda è un follow-up della conversazione precedente relativa al miglioramento dell'esempio fornito con il ciclo `while` mentre l'ultima domanda stimola lo studente nell'ottimizzazione dell'approccio per la ricerca dei divisori di un numero intero.

¹ Un token è l'unità di testo che un LLM può elaborare e può essere una parola, parte di una parola o un carattere, a seconda della lingua utilizzata e del modello considerato.

² Per un confronto <https://codingscape.com/blog/most-powerful-llms-large-language-models>

³ L'interazione completa è disponibile al link indicato in Tabella 1.

G L'argomento di oggi sono le strutture iterative in Python. Le strutture iterative ci permettono di scrivere programmi che contengono operazioni da ripetere più volte. Ci sono le strutture iterative definite e indefinite, quelle definite si ripetono un numero definito di volte, quelle indefinite un numero infinito di volte. Possono essere anche strutture iterative precondizionali o postcondizionate in base a dove si trova la condizione.

In Python c'è il for che si può usare in due modi diversi:

```
for i in range(1,10):  
    print(i)
```

stampa i numeri da 1 a 10

```
for i in lista:  
    print(i)
```

stampa gli elementi di una lista.

Poi c'è il while che si usa per le strutture iterative indefinite:

```
while num > 0:  
    print(num)  
    num = num - 1
```

Questo codice stampa una specie di countdown con num che deve essere preso in input dall'utente.

Grazie per l'introduzione, professore! Le strutture iterative sembrano un argomento molto interessante e utile. Ho alcune domande per assicurarmi di aver capito bene:

1. Ha menzionato strutture iterative definite e indefinite. Il ciclo for è un esempio di struttura definita, giusto? E il while sarebbe quindi un esempio di struttura indefinita?
2. Per quanto riguarda il for con range(1,10), ho notato che ha detto che stampa i numeri da 1 a 10. Ma non dovrebbe stampare da 1 a 9? Ho sempre pensato che in Python il secondo numero nel range fosse escluso.
3. Potrebbe spiegarmi meglio la differenza tra strutture precondizionali e postcondizionate? Non sono sicuro di aver afferrato questo concetto.
4. Nell'esempio del while, vedo che stiamo decrementando num ad ogni iterazione. Questo significa che il ciclo terminerà quando num diventa 0 o negativo, giusto?

Mi scusi se faccio troppe domande, ma voglio essere sicuro di aver compreso bene le basi prima di procedere!

Figura 1: Spiegazione argomento e risposta del chatbot

Grazie per il feedback, professore. Apprezzo molto la sua pazienza nel rispondere alle mie domande.

Riguardo al punto 3, ho fatto una piccola ricerca e ho scoperto che in Python possiamo usare l'operatore `--` per decrementare una variabile. Quindi, potremmo scrivere `num -= 1` invece di `num = num - 1`. È corretto?

Inoltre, ho provato ad eseguire il codice che ho scritto e ho notato che funziona come previsto per input positivi, ma se inserisco una lettera o un simbolo, il programma si interrompe con un errore. Ecco l'errore che ho ottenuto:

```
ValueError: invalid literal for int() with base 10: 'a'
```

Questo mi fa pensare che forse dovremmo aggiungere un controllo per assicurarci che l'input sia effettivamente un numero. Potrei provare a modificare il codice per gestire questo caso? Ecco un tentativo:

```
python

while True:
    try:
        num = int(input("Inserisci un numero positivo: "))
        if num > 0:
            break
        else:
            print("Il numero deve essere positivo. Riprova.")
    except ValueError:
        print("Input non valido. Inserisci un numero intero.")

while num > 0:
    print(num)
    num -= 1
```

Questo codice dovrebbe continuare a chiedere all'utente di inserire un input finché non viene fornito un numero intero positivo. È un approccio corretto per gestire possibili errori di input?

Infine, anche se non possiamo simulare esattamente un `do-while` in Python, mi chiedo se ci siano situazioni in cui potremmo voler eseguire un blocco di codice almeno una volta prima di controllare una condizione. Come potremmo gestire tali situazioni in Python?

⚠ Claude does not have the ability to run the code it generates yet.

Figura 2: Mancato rispetto dei prerequisiti

G Chiedi un numero in input all'utente e stampa tutti i numeri primi inferiori a quel numero

Grazie per l'esercizio, professore. Cercherò di risolverlo utilizzando le strutture iterative che abbiamo appena studiato. Ecco il mio tentativo:

```
python

numero = int(input("Inserisci un numero positivo: "))

for i in range(2, numero):
    is_primo = True
    for j in range(2, i):
        if i % j == 0:
            is_primo = False
            break
    if is_primo:
        print(i)
```

Ho cercato di usare un ciclo for esterno per iterare su tutti i numeri da 2 fino al numero inserito dall'utente (escluso). Per ogni numero, ho usato un altro ciclo for per verificare se è primo.

Tuttavia, ho alcune domande e dubbi:

1. Ho usato una variabile booleana `is_primo`. È il modo corretto di farlo in Python?
2. Non sono sicuro se l'uso del `break` nel ciclo interno sia la soluzione migliore. Potrebbe esserci un modo più efficiente?
3. Non ho gestito il caso in cui l'utente inserisca un numero negativo o 1. Dovrei aggiungere un controllo all'inizio?
4. Mi chiedo se ci sia un modo per ottimizzare questo algoritmo. Per esempio, devo davvero controllare tutti i numeri fino a $i-1$ per determinare se i è primo?

Potrebbe darmi un feedback su questo codice e suggerirmi eventuali miglioramenti?

⚠ Claude does not have the ability to run the code it generates yet.

Figura 3: Risoluzione di un esercizio

5 Conclusioni e sviluppi futuri

Questo articolo ha proposto un innovativo approccio all'insegnamento dell'informatica (e di altre discipline) basato sull'utilizzo di LLM come "studenti virtuali", implementando una strategia di prompt engineering ispirata al paradigma del Learning By Teaching. L'approccio proposto si è dimostrato promettente nel favorire processi di apprendimento generativi e nello stimolare la riflessione critica degli studenti sul proprio processo di apprendimento.

I risultati dell'applicazione di questo metodo su diversi LLM hanno evidenziato sia il potenziale che le sfide di questo approccio in base all'LLM utilizzato.

Per gli sviluppi futuri ci si potrebbe orientare verso l'ottimizzazione del prompt, la conduzione di studi sperimentali per misurare l'efficacia di questo approccio e l'integrazione con altri strumenti didattici.

Bibliografia

- [1] J. Rudolph, S. Tan e S. Tan, «ChatGPT: Bullshit Spewer or the End of Traditional Assessments in Higher Education?», *Journal of Applied Learning and Teaching*, n. 6, pp. 342-363, 2023.
- [2] H.-W. Choi e S. Abdirayimov, «Revolutionizing Education: The Era of Large Language Models», *Journal of Multimedia Information System*, vol. 11, n. 1, pp. 97-100, 2024.
- [3] R. E. Wang e D. Demszky, «Is ChatGPT a Good Teacher Coach? Measuring Zero-Shot Performance For Scoring and Providing Actionable Insights on Classroom Instruction», 2023. [Online]. Available: <https://arxiv.org/abs/2306.03090>.
- [4] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith e D. C. Schmidt, «A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT», 2023. [Online]. Available: <https://arxiv.org/abs/2302.11382>.
- [5] Z. Chen, K. Zhou, B. Zhang, Z. Gong, W. X. Zhao e J.-R. Wen, «ChatCoT: Tool-Augmented Chain-of-Thought Reasoning on Chat-based Large Language Models», 2023. [Online]. Available: <https://arxiv.org/abs/2305.14323>.
- [6] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao e D. Jiang, «WizardLM: Empowering Large Language Models to Follow Complex Instructions», 2023. [Online]. Available: <https://arxiv.org/abs/2304.12244>.
- [7] G. Kim, P. Baldi e S. McAleer, «Language Models can Solve Computer Tasks», 2023. [Online]. Available: <https://arxiv.org/abs/2303.17491>.
- [8] J. M. Markel, S. G. Opferman, J. A. Landay e C. Piech, «GPTeach: Interactive TA Training with GPT-based Students», in *Proceedings of the Tenth ACM Conference on Learning @ Scale*, Copenhagen, 2023.
- [9] H. Jin, S. Lee, H. Shin e J. Kim, «Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education», 2024. [Online]. Available: <https://arxiv.org/abs/2309.14534>.
- [10] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi e H. Hajishirzi, «Self-Instruct: Aligning Language Models with Self-Generated Instructions», 2023. [Online]. Available: <https://arxiv.org/abs/2212.10560>.
- [11] E. R. Mollick e L. Mollick, «New Modes of Learning Enabled by AI Chatbots: Three Methods and Assignments», 13 December 2023. [Online]. Available: Mollick, Ethan R. and Mollick, Lilach, New Modes of Learning Enabled by AI Chatbots: Three Methods and Assignments (December 13, 2022). <https://ssrn.com/abstract=4300783>.
- [12] E. R. Mollick e L. Mollick, «Using AI to Implement Effective Teaching Strategies in Classrooms: Five Strategies, Including Prompts», 17 March 2023. [Online]. Available: <https://ssrn.com/abstract=4391243>.
- [13] E. R. Mollick e L. Mollick, «Assigning AI: Seven Approaches for Students, with Prompts», 23 September 2023. [Online]. Available: <https://ssrn.com/abstract=4475995>.
- [14] L. Fiorella, «Making Sense of Generative Learning», *Educational Psychology Review*, vol. 35, n. 50, 2023.

Lezioni “*AI-Aumentate*”: come trasformare una lezione in una performance multi-attore

Michele Iacobellis¹

¹ Colamonico - Chiarulli

michele.iacobellis@gmail.com

Abstract

La diffusa tentazione di sviluppare un *atteggiamento ostile* nei confronti dell'intelligenza artificiale va arginata mediante il disegno intelligente di un atteggiamento propositivo, curioso e assertivo che consenta ai docenti di piegare la tecnologia ai propri scopi pedagogici e culturali, a vantaggio della comunità, spesso travolta dal progresso tecnico, che viene subito, ma non sempre compreso. Per cominciare questo percorso (si spera virtuoso), si propone, in forma ingenua ed embrionale, un metodo didattico che inserisca l'AI nel flusso di interazione con docente e studenti, come fosse autonoma e di pari dignità rispetto al conduttore dell'esperienza didattica, cioè il collettivo docenti-studenti. *Questo articolo non è stato scritto mediante AI generativa!*

1 Introduzione

L'intelligenza artificiale generativa si è presentata al pubblico dapprima in modo dirompente (direi di impiegare il termine “disruptive”) creando un'onda di crescente interesse e meraviglia, in primo luogo da parte dei ragazzi e poi, a seguire, da parte degli adulti, per poi tornare ad un atteggiamento più cauto e misurato, come si evince dalle recenti interviste di Sam Altman, disponibili su YouTube [1]. I ragazzi hanno colto, in modo conscio o inconscio, immediatamente la possibilità di “*barare*”, di impiegare l'AI generativa per poter svolgere compiti e attività in modo più rapido, meno oneroso in termini di energie psichiche e intellettive: il minor sforzo con il massimo risultato. Nel tempo, la AI generativa è diventata di impiego consueto da parte di studenti, docenti e genitori, come riportano i media di settore (un esempio in [2]). Contestualmente, altre forme di AI hanno preso piede: 1) le AI che generano Avatar foto realistici, in grado di recitare con voci, espressioni del volto, prossemica, i più disparati testi, scritti da esseri umani oppure generati da LLM; 2) le AI che generano musiche complete di voce, di qualità professionale; 3) le AI che generano immagini e video. In questo “*universo di minacce*”, una “*cascata di opportunità*” didattiche per tutti noi docenti.

2 Le AI che generano Avatar

Per i nostri scopi, risulta rilevante quella tecnologia che consente di generare l'immagine in movimento di uno o più Avatar: per esempio, persone comuni, con qualità foto realistica, inserite all'interno di una presentazione multimediale, secondo una sequenza di diapositive.

Esistono alcune aziende (es. Synthesia, www.synthesia.io) che forniscono questo tipo di servizio, cioè la possibilità di realizzare complesse presentazioni multimediali dotate di uno o più Avatar a supporto, in sovraimpressione sulle slide, che recitano il testo inserito dall'utente e ne interpretano adeguatamente il contenuto e il tono, con la prossemica inferita dal testo o indicata

esplicitamente dall'utente mediante meta istruzioni. Nella Figura 1 è riportato un fotogramma di uno dei video didattici messi a disposizione degli studenti, rigenerato con l'ultima versione della piattaforma Synthesia Studio. A sinistra del video possiamo vedere l'Avatar, su uno sfondo moderatamente animato di colore nero-grigio, che legge e spiega ciascuno degli elementi riportati a destra, argomenti cardine trattati a lezione, attinenti al pensiero computazionale applicato.



Figura 1 - Un frame di un video didattico, con Avatar generato da AI.

In questo video, l'Avatar è di sesso femminile, esile, con lineamenti morbidi, che non distrae il discente o la discente quando questi lo mette (per certo) a confronto con le qualità estetiche usualmente proposte o imposte dai mezzi di comunicazione di massa, dalla cultura degli influencer, dagli standard estetici dei social network.

L'abito è anonimo, bianco.

La voce, selezionabile tra una ricca palette di possibilità, può essere scelta dal docente sulla base della fermezza, del tono, della ricchezza armonica, della musicalità, della **modalità cognitiva** che si vuole favorire o con la quale si presume di avere a che fare.

L'Avatar si rivolge indifferentemente al docente e a ciascuno studente, ponendo domande, per le quali tuttavia, ad oggi, non potrà ascoltare la risposta, né tantomeno darne un giudizio critico in tempo reale.

È stato molto interessante osservare in che modo i ragazzi hanno reagito a questo tipo di video, di cui il presente è solo un campione (nei prossimi paragrafi esamineremo le altre possibilità).

Per ogni classe, in occasione del primissimo impiego di contenuti didattici di questo tipo, si può sfruttare un favorevole *effetto sorpresa* con il fine di catturare l'attenzione dei discenti all'inizio della lezione. In generale si è rilevato un evidente interesse da parte dell'intera classe. La struttura del video ha poi il ruolo di tenere alto l'interesse e ridurre al minimo gli eventi distrattori, riacquisendo di volta in volta l'attenzione di ogni studente.

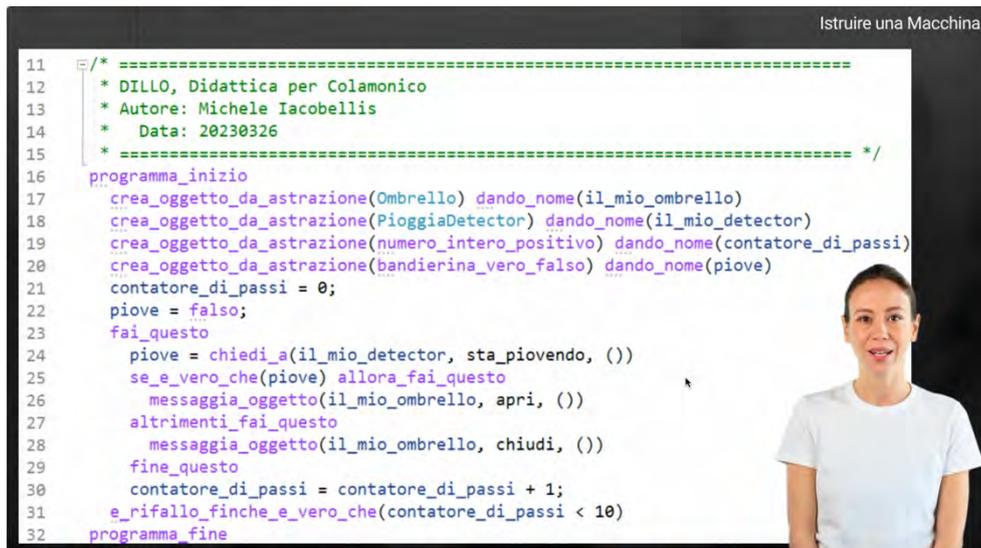


Figura 2 - L'Avatar chiede al docente di spiegare una parte del codice DILLO.

3 Una performance multi-attore

La tecnologia descritta nella sezione precedente, se interpretata come generatore di una semplice presentazione multimediale, offre dei vantaggi indubbi, ma rimane pur sempre il generatore di un video.

In realtà, il presente articolo offre una prospettiva differente. Partiamo da un importante postulato, che dobbiamo condividere, per poter dare considerazione al resto dell'articolo:

“Una lezione è una performance teatrale in cui il docente e i discenti sono tutti attori”

Fissato questo postulato fondamentale, gli Avatar diventano parte integrante della lezione, anch'essi protagonisti attivi della performance, seppur pre-programmati. Immaginiamo una sessione di musica da camera oppure una jam-session:

“La presentazione arricchita con gli Avatar rappresenta la coppia metronomo e percussioni: essa dà il tempo, il ritmo e di fatto guida tutta la classe accompagnandola attraverso i vari passaggi della lezione.”

È opportuno strutturare la lezione secondo uno schema a tre momenti:

1. Un momento introduttivo, squisitamente motivazionale, il cui scopo è smaltire le eventuali cause di distrazione del cambio d'ora e riacquisire un principio di attenzione da parte degli studenti;
2. Un momento centrale, in cui si affrontano i temi scelti per la lezione;
3. Un momento finale, in cui si raccolgono le impressioni e si traggono le conclusioni, schematizzandole per quanto possibile.

Ogni docente conosce le proprie classi ed è in grado di stabilire che durata debbano avere i tre momenti, considerando le capacità medie dei ragazzi di tenere il focus su certe categorie di argomenti.

Questa stessa struttura deve essere in qualche modo riflessa all'interno del video guida, assegnando ad uno o più Avatar i ruoli più opportuni, potendo in effetti giocare su una serie di parametri che modulerebbero, se scelti con cura, l'efficacia del messaggio motivazionale, teorico, tecnico-pratico, comportamentale, ecc.

3.1 Stabilire lo Story Board

Il docente deve aver cura di strutturare anche il video guida (oltre alla lezione) secondo una successione di diversi momenti, sulla base del contenuto che sta proponendo per la lezione.

I vari momenti, oltre a presentare la macrostruttura già illustrata, devono essere ulteriormente suddivisi sulla base dei "palleggi" che ciascun Avatar farà verso il docente, con frasi del tipo "professore, può spiegare lei questo concetto, per favore?" oppure "prof, cosa avete detto a lezione sullo statement while?". Con qualche accortezza in più, è possibile dirigere le attenzioni dell'Avatar anche verso la classe o verso gruppi di studenti.



Figura 3 - L'Avatar chiede l'intervento del docente.

Per ogni domanda che l'Avatar pone al docente, è necessario prevedere un periodo di N secondi di pausa, in cui l'Avatar attende una qualche risposta o interviene con intercalari opportuni: in questo tempo il docente ha modo di illustrare i concetti presenti sul video, anche interagendo con i discenti. Con destrezza, il docente può agire sul sistema di presentazione (una smart board o altro dispositivo) per mettere in pausa il video, se necessita di più tempo per illustrare la slide corrente.

In Figura 3, l'Avatar si rivolge agli studenti, chiamando in causa il docente, chiedendo un intervento esplicativo sulle parole che l'Avatar stesso ha utilizzato nel discorso immediatamente precedente.

In Figura 4 invece, l'Avatar si rivolge direttamente al docente, chiedendo di illustrare quanto riportato nella slide corrente. Una volta fatta la richiesta, l'Avatar (Figura 5) si pone in attesa per un certo tempo (il docente infatti ha progettato lo storyboard della lezione) e di tanto in tanto interviene con intercalari o esclamazioni. Queste hanno l'effetto di intervenire in modo impulsivo nel discorso, in generale riattivando l'attenzione eventualmente affievolita degli studenti, sempre che il docente, nella progettazione, lo abbia ritenuto utile o necessario.



Figura 4 - Qui l'Avatar chiede al docente di spiegare quanto riportato a schermo.



Figura 5 - L'Avatar in attesa (il docente ha circa 30s per intervenire).

4 Varietà e Impatto Visivo

Questi strumenti consentono di abbinare testi complessi che il docente predispone sulla base della lezione con persone dall'aspetto improbabile, ma comunque efficace, quando si tratta di attrarre l'attenzione del discente e indurre stati d'animo in cui la distanza percepita rispetto al parlante si riduce (o si allarga), modulando l'effetto complessivo della lezione.

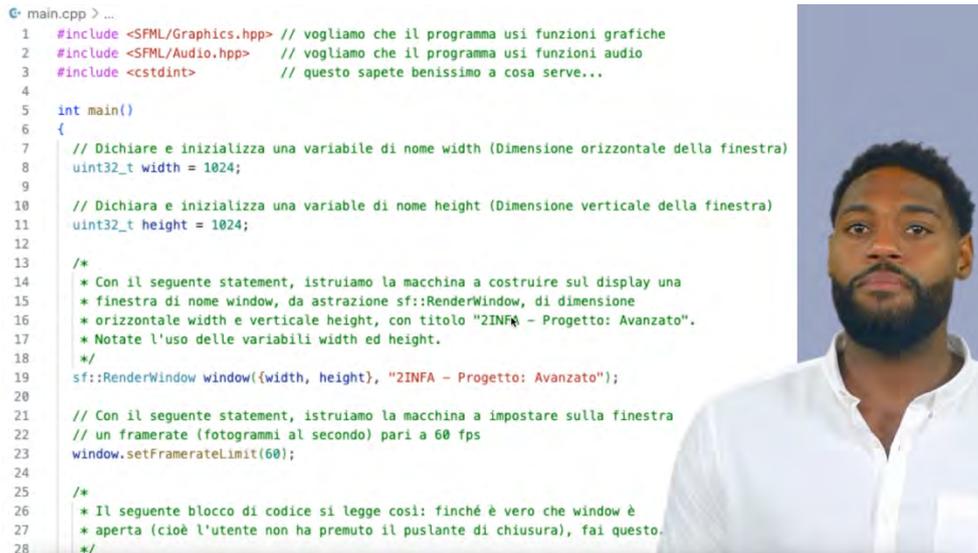


Figura 6 - Un ragazzo spiega con voce profonda un sorgente C++11.

Il docente, come possiamo vedere dalle figure, ha un elevato numero di gradi di libertà per disegnare una performance teatrale utile e coinvolgente.



Figura 7 - Un uomo dagli evidenti tatuaggi affronta un sorgente ASM per x86_64.

Bibliografia

- [1] Lex Friedman podcast, <https://youtu.be/jvqFAi7vkBc?feature=shared>.
- [2] Articolo CNBC, <https://www.cnbc.com/2024/06/11/ai-is-getting-very-popular-among-students-and-teachers-very-quickly.html#>.
- [3] C. G. Jung, "Energetica Psicica", Bollati Boringhieri.
- [4] J. Bruner, "La mente a più dimensioni", Editori Laterza.
- [5] A. Guasparri, "Il primate della moda", FrancoAngeli.
- [6] O. Lizzini, "Avicenna", Carocci Editore.

Le affermazioni, lo stile, il contenuto delle lezioni e delle immagini sono il frutto della lettura critica delle opere in bibliografia. Molte altre hanno contribuito, di cui si omette per ragioni di spazio il titolo.

Esercizi di programmazione non banali con ChatGPT nelle superiori: un esperimento

Davide Ponzini¹, Giovanna Guerrini¹, Flaviano Cipollone², and Giorgio Delzanno¹

¹ Università degli Studi di Genova
davide.ponzini@edu.unige.it
{nome.cognome}@unige.it

² I.I.S. “Cristoforo Colombo” Sanremo
flavio.cipollone@gmail.com

Sommario

Negli ultimi anni, i Large Language Models (LLM), tra cui ChatGPT, sono diventati strumenti popolari tra gli studenti delle scuole secondarie e delle università per risolvere esercizi di programmazione. Tuttavia, l’uso improprio di questi strumenti può danneggiare il processo di apprendimento. Questo studio analizza l’effetto di ChatGPT sulle prestazioni degli studenti delle scuole superiori in esercizi di programmazione, concentrandosi sulla comprensione di concetti non familiari. Abbiamo condotto un esperimento su una classe terza di un istituto tecnico informatico, suddivisa in due gruppi: uno con accesso a ChatGPT e l’altro senza. I risultati mostrano che gli studenti senza accesso a ChatGPT hanno ottenuto risultati leggermente migliori. Le difficoltà maggiori si sono riscontrate nella comprensione delle richieste e nell’uso delle funzioni fornite insieme all’esercizio. I risultati suggeriscono che l’accesso a ChatGPT senza una formazione adeguata può non essere vantaggioso per risolvere esercizi di programmazione complessi.

1 Introduzione

Negli ultimi anni, i Large Language Models (LLM), come ad esempio ChatGPT¹, sono diventati sempre più popolari tra gli studenti delle scuole secondarie e delle università. Gli strumenti basati su IA generativa sono in grado di generare risposte accettabili ad un’ampia gamma di domande. In particolare, possono permettere agli studenti di risolvere esercizi di programmazione in pochi prompt [2, 1, 4].

L’uso di questi strumenti può potenzialmente migliorare le prestazioni e l’apprendimento degli studenti [7]. Tuttavia, questi strumenti possono essere facilmente utilizzati dagli studenti in maniera impropria, ad esempio per risolvere in maniera automatica i compiti assegnati dai docenti, senza dover quindi studiare o ragionare su quanto visto a lezione, finendo per danneggiare il processo di apprendimento, così come lo sviluppo delle abilità di pensiero critico [5]. Allo stesso tempo, c’è ancora molta confusione tra gli studenti sulle capacità dei LLM e su come usarli in maniera efficace: alcuni credono che possano essere utili, mentre altri ritengono possano avere un impatto negativo sull’apprendimento [5, 6].

In questo studio, analizziamo gli effetti dei LLM sulle prestazioni degli studenti delle scuole superiori nel risolvere esercizi di programmazione su argomenti a loro non completamente familiari. Data la sua popolarità tra gli studenti, abbiamo deciso di concentrare la nostra attenzione su ChatGPT 3.5.

¹<https://chat.openai.com/>

2 Background

La capacità di ChatGPT di risolvere problemi di programmazione è stata esplorata principalmente a livello universitario. ChatGPT è in grado di fornire risultati ottimi per corsi di programmazione introduttivi, mentre fatica a comprendere appieno richieste più complesse, specialmente se le domande non sono strutturate in maniera adeguata [1, 4]. Per risultati ottimali, le istruzioni devono essere fornite in maniera chiara ed ordinata [4]. Per scenari semplici, ChatGPT è molto spesso in grado di fornire immediatamente la risposta corretta, mentre per domande più complesse è spesso necessario interagire con ulteriori prompt per indirizzare lo strumento verso la risposta corretta [1]. ChatGPT, inoltre, fatica a comprendere informazioni non testuali, come documentazione API e diagrammi UML [4].

Nonostante ChatGPT sia conosciuto tra gli studenti, solo un numero limitato lo utilizza con finalità di apprendimento [6]. Le ragioni principali sono che, da una parte, manca un insieme chiaro di norme riguardanti il suo uso a livello didattico, il che porta gli studenti a temere accuse di plagio o di violazione delle regole [6]. Dall'altra, l'educazione sull'uso degli LLM è molto limitata, e talvolta assente, e gli studenti spesso non sanno come utilizzare efficacemente questi strumenti per migliorare il loro apprendimento [6, 5]. Inoltre, alcuni studenti scelgono di non utilizzare lo strumento, pensando che possa ridurre la qualità del loro apprendimento [6].

Tra quelli che usano ChatGPT, l'uso più comune è per generare codice, seguito dal debugging, mentre la terza funzione più usata è farsi spiegare concetti difficili [3]. Gli studenti che utilizzano ChatGPT per i compiti di programmazione tendono a mostrare maggiore efficacia e competenze nel pensiero computazionale [7]. Un altro aspetto cruciale nel processo di apprendimento è la motivazione: l'uso di ChatGPT può di solito aumentare la motivazione degli studenti. Questo effetto diventa però insignificante se il compito è particolarmente difficile [7].

3 Impostazione dell'esperimento

Abbiamo condotto un esperimento su 21 studenti appartenenti ad una classe terza di un istituto tecnico informatico. Gli studenti stavano studiando C++ ed erano già familiari con diversi concetti base, come la sintassi, gli operatori, i tipi di dato, le variabili, le condizioni e i cicli. Inoltre, avevano da poco iniziato a vedere array e funzioni. Non erano invece ancora familiari con la gestione dei file o i principi della programmazione orientata ad oggetti. Gli esercizi con cui erano abituati erano brevi e semplici e gli studenti non erano abituati a svolgere progetti un po' più complessi, nè ad avere in input funzioni già implementate da utilizzare nello svolgimento dell'esercizio.

Abbiamo diviso la classe in due gruppi e gli abbiamo chiesto di risolvere due diversi esercizi preparati da noi. Gli esercizi erano volutamente di una difficoltà superiore a quella a cui gli studenti erano abituati ed includevano sia concetti con cui erano familiari, che concetti nuovi. In dettaglio, gli esercizi richiedevano di interagire con array bidimensionali (array contenenti array) e di leggere dati da file, operazioni a cui gli studenti non erano abituati.

Prima dell'inizio dell'attività, abbiamo fornito agli studenti un file contenente delle classi create da noi che permettevano di eseguire questo tipo di operazioni in maniera molto semplificata, in modo che gli studenti non si dovessero preoccupare dei dettagli implementativi relativi agli array bidimensionali e alle operazioni sui file. Abbiamo inoltre fornito la documentazione relativa alle classi e alle funzioni da noi create.

Entrambi gli esercizi avevano un processo di risoluzione analogo, che si può riassumere in:

1. Leggere i dati da file usando le apposite funzioni fornite

2. Salvare questi dati in un array bidimensionale usando le funzioni fornite
3. Iterare con dei cicli sull'array e stampare i dati in un formato specificato, con l'aiuto di alcune funzioni da noi fornite

Gli studenti sono stati divisi in due gruppi, in cui uno aveva accesso a ChatGPT ed era libero di usarlo come voleva, mentre l'altro poteva utilizzare internet ma non LLM.

4 Risultati

Un numero molto limitato di studenti è riuscito a risolvere completamente gli esercizi e solo un singolo studente li ha completati entrambi. Numerose difficoltà sono emerse durante la risoluzione, molte delle quali relative alla comprensione delle richieste e del codice che era stato fornito. Sorprendentemente, gli studenti senza accesso a ChatGPT hanno ottenuto risultati leggermente migliori rispetto a quelli che lo potevano utilizzare.

Diverse consegne contenevano errori di ragionamento, suggerendo che alcuni studenti avevano difficoltà a comprendere il compito o a determinare come risolvere il problema. Il primo esercizio è stato compreso meglio dagli studenti con accesso a ChatGPT, mentre quelli senza accesso hanno avuto notevoli difficoltà. Il secondo esercizio, invece, ha posto maggiori difficoltà agli studenti con accesso a ChatGPT, in quanto molti non hanno compreso appieno la richiesta, mentre tutti gli studenti senza accesso la hanno compresa.

La maggior parte delle consegne per entrambi i compiti non presentava errori di sintassi, con solo alcuni problemi minori come parentesi mancanti o tipi di dati errati, facilmente correggibili. Questi errori minori sono stati riscontrati in entrambi i gruppi, sia con che senza ChatGPT, indicando che ChatGPT non ha influenzato la correttezza del codice. Tuttavia, per il secondo compito, gli errori di sintassi comparivano solo negli studenti che utilizzavano ChatGPT, probabilmente a causa delle difficoltà nell'uso del codice generato.

Per quanto riguarda la lettura e la memorizzazione dei dati di input, gli studenti con ChatGPT hanno ottenuto risultati peggiori. ChatGPT genera spesso dati di esempio nel codice, e gli studenti non sono riusciti a sostituirli con le appropriate funzioni fornite da noi. Al contrario, gli studenti senza ChatGPT hanno utilizzato correttamente le funzioni fornite, ottenendo risultati particolarmente buoni nel secondo compito.

Per quanto riguarda invece la stampa dei dati, ChatGPT ha migliorato le prestazioni nel primo compito, anche se non sempre rispettando esattamente il formato richiesto. Tuttavia, nel secondo compito, gli studenti con ChatGPT hanno ottenuto risultati nettamente peggiori rispetto a quelli senza, probabilmente a causa della complessità maggiore delle istruzioni per il formato di output richiesto.

Un ulteriore problema emerso in molti studenti con accesso a ChatGPT è stata la reimplementazione del codice. Nel formulare i loro prompt, gli studenti includevano il codice sorgente delle funzioni fornite. ChatGPT, invece di invocare semplicemente le funzioni, procedeva a reimplementarne le funzionalità. Ad esempio, nel secondo compito, agli studenti è stata fornita la funzione `printValue`, che poteva stampare un valore utilizzando esattamente 16 caratteri. La funzione che fornita era la seguente:

```
void printValue(float value) {
    cout << fixed << setprecision(1) << setfill(' ') << setw(16) << value;
}
```

Molti studenti con accesso a ChatGPT hanno reimplementato questo codice nella loro funzione `main`, producendo risultati simili a questo:

```

int main() {
    std::string studentData[5][10] = {
        {"Gaia Rizzo", "5.2", "3.9", "3.0", "6.0", "5.4", "5.1", "7.2"},
        {"Mattia Marino", "10.0", "9.5", "9.4", "10.0", "9.1", "9.0", "10.0"},
        {"Pietro Martino", "6.5", "4.5", "5.1", "5.8", "3.4", "5.6", "5.0"},
        {"Giorgia Barone", "8.5", "6.5", "5.8", "7.6", "5.2", "7.6", "4.3"},
        {"Michele Giordano", "5.3", "8.0", "6.1", "7.82", "8.1", "6.5", "6.4"}
    };

    std::cout << std::left << std::setw(20) << "Nome"
              << std::setw(8) << "Italiano"
              << std::setw(8) << "Storia"
              << std::setw(8) << "Geografia"
              << std::setw(8) << "Inglese"
              << std::setw(8) << "Francese"
              << std::setw(12) << "Matematica"
              << std::setw(8) << "Scienze"
              << std::setw(10) << "Media"
              << std::setw(15) << "Ammissione" << std::endl;
}

```

Figura 1: Esempio di codice consegnato contenente dati di esempio generati da ChatGPT.

```
std::cout << std::setw(10) << std::fixed << std::setprecision(2) << averagegrade;
```

quando invece sarebbe stato sufficiente invocare la funzione fornita, scrivendo semplicemente

```
printValue(averagegrade);
```

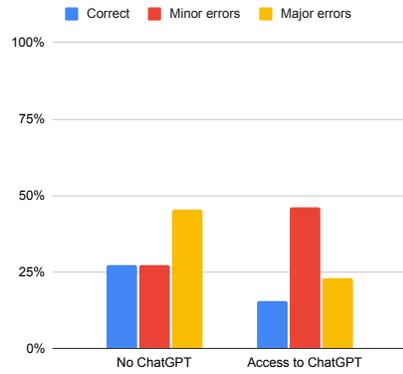
5 Discussione

I risultati dell'esperimento suggeriscono che gli studenti hanno dato priorità all'uso dello strumento rispetto alla comprensione del problema. Questo conferma le nostre osservazioni durante l'esperimento: abbiamo infatti notato che la maggior parte dei prompt si concentrava sull'implementazione del codice piuttosto che sulla spiegazione dei passaggi necessari per risolvere il problema. Molti studenti si sono concentrati nel trovare i prompt appropriati per far sì che ChatGPT risolvesse l'intero compito, anche se in molti casi di fatto la richiesta non era stata compresa appieno.

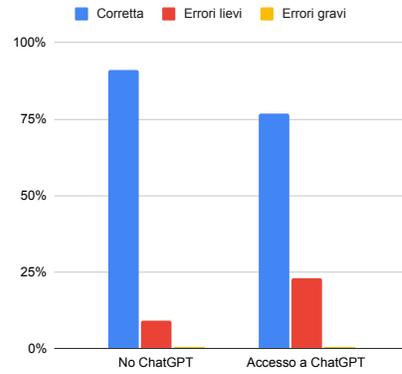
Gli studenti hanno spesso avuto difficoltà a formulare i prompt, soprattutto quando cercavano di stampare i dati in un formato specifico senza una descrizione testuale. Questo potrebbe spiegare perché gli studenti con accesso a ChatGPT abbiano ottenuto risultati peggiori rispetto a quelli senza nella stampa dei dati del secondo compito.

Nel secondo compito, gli studenti senza accesso a ChatGPT hanno dimostrato una maggiore attenzione e comprensione del codice fornito. Questo suggerisce che l'accesso a ChatGPT potrebbe aver ostacolato la capacità di comprendere e applicare il codice. Analogamente, gli studenti senza sono stati in grado di leggere correttamente i dati di input dai file, mentre gli studenti con accesso hanno spesso incluso dati di esempio direttamente nel loro codice (come mostrato nella Figura 1).

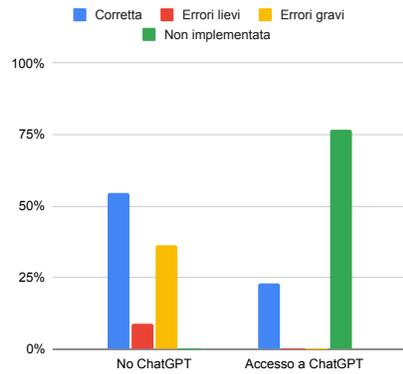
La reimplementazione automatica del codice eseguita da ChatGPT è un problema significativo. Molte delle funzionalità reimplementate hanno effettivamente ostacolato la comprensione del codice generato da parte degli studenti poiché utilizzavano concetti che gli studenti non



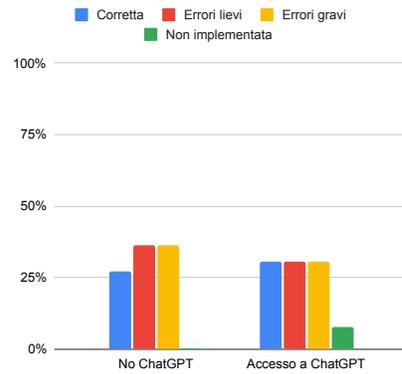
(a) Ragionamento.



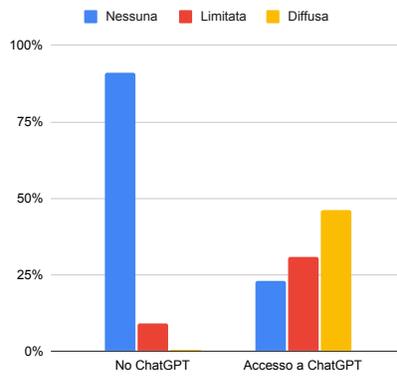
(b) Sintassi.



(c) Implementazione lettura dati.



(d) Implementazione stampa dati.



(e) Reimplementazione codice.

Figura 2: Confronto tra studenti con e senza accesso a ChatGPT.

conoscevano. Questo è confermato dal fatto che queste funzioni sono state utilizzate solo dai team con accesso a ChatGPT.

Abbiamo anche osservato che gli studenti che ricevevano codice dedicavano poco tempo alla sua comprensione, anche quando il codice non era loro chiaro, preferendo invece provare a copiarlo ed eseguirlo. Se l'output non era quello previsto, la maggior parte di loro formulava un prompt diverso anziché cercare di capire perché il codice generato non funzionasse. Inoltre, non abbiamo osservato nessuno studente chiedere a ChatGPT di semplificare o spiegare la soluzione fornita, nonostante i nostri incoraggiamenti. Infine, abbiamo osservato che alcuni studenti erano riluttanti a utilizzare ChatGPT nonostante fossero autorizzati a farlo perché volevano mettersi alla prova.

Abbiamo condotto un'analisi aggiuntiva sui risultati, confrontando le prestazioni degli studenti con e senza accesso a ChatGPT per entrambi i compiti. I risultati, mostrati in Figura 2, indicano che gli studenti con accesso a ChatGPT non hanno un vantaggio chiaro rispetto a quelli senza accesso in termini di comprensione del compito o di sintassi. Le prestazioni di implementazione variano notevolmente a seconda del compito, e la reimplementazione del codice colpisce quasi esclusivamente gli utenti di ChatGPT.

Questi risultati suggeriscono che, per gli studenti senza formazione preliminare sull'uso di LLM, ChatGPT da solo potrebbe non essere sufficiente per ottenere risultati soddisfacenti su compiti di programmazione non banali o non standard. Al contrario, un'eccessiva dipendenza dallo strumento potrebbe effettivamente ostacolare la comprensione del codice o del compito.

6 Riferimenti bibliografici

- [1] Chuqin Geng et al. *Can ChatGPT Pass An Introductory Level Functional Language Programming Course?* 2023. DOI: [10.48550/ARXIV.2305.02230](https://doi.org/10.48550/ARXIV.2305.02230).
- [2] Chung Kwan Lo. «What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature». In: *Education Sciences* 13.4 (apr. 2023), p. 410. ISSN: 2227-7102. DOI: [10.3390/educsci13040410](https://doi.org/10.3390/educsci13040410).
- [3] Abdessalam Ouazaki, Kristoffer Bergram e Adrian Holzer. «Leveraging ChatGPT to Enhance Computational Thinking Learning Experiences». In: *2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*. 2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE). Nov. 2023, pp. 1–7. DOI: [10.1109/TALE56641.2023.10398358](https://doi.org/10.1109/TALE56641.2023.10398358).
- [4] Eng Lieh Ouh et al. «ChatGPT, Can You Generate Solutions for my Coding Exercises? An Evaluation on its Effectiveness in an undergraduate Java Programming Course.» In: *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2023: Innovation and Technology in Computer Science Education. Turku Finland: ACM, 29 giu. 2023, pp. 54–60. ISBN: 9798400701382. DOI: [10.1145/3587102.3588794](https://doi.org/10.1145/3587102.3588794).
- [5] Basit Qureshi. «Exploring the Use of ChatGPT as a Tool for Learning and Assessment in Undergraduate Computer Science Curriculum: Opportunities and Challenges». In: *2023 9th International Conference on e-Society, e-Learning and e-Technologies*. 9 Giu. 2023, pp. 7–13. DOI: [10.1145/3613944.3613946](https://doi.org/10.1145/3613944.3613946). arXiv: [2304.11214\[cs\]](https://arxiv.org/abs/2304.11214).
- [6] Harpreet Singh, Mohammad-Hassan Tayarani-Najaran e Muhammad Yaqoob. «Exploring Computer Science Students' Perception of ChatGPT in Higher Education: A Descriptive and Correlation Study». In: *Education Sciences* 13.9 (set. 2023), p. 924. ISSN: 2227-7102. DOI: [10.3390/educsci13090924](https://doi.org/10.3390/educsci13090924).

- [7] Ramazan Yilmaz e Fatma Gizem Karaoglan Yilmaz. «The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation». In: *Computers and Education: Artificial Intelligence* 4 (1 gen. 2023), p. 100147. ISSN: 2666-920X. DOI: [10.1016/j.caeai.2023.100147](https://doi.org/10.1016/j.caeai.2023.100147).

Coding in Ambienti di Extended Reality

Lorenzo Gerini, Manuela Chessa, Giorgio Delzanno,
Giovanna Guerrini e Fabio Solari

DIBRIS, Università degli Studi di Genova, Italy
{name.surname}@unige.it

Sommario

In questo articolo presentiamo un sistema sperimentale, chiamato XRCoding, per combinare coding ed attività immersive in realtà virtuale. Il sistema è basato sul linguaggio a blocchi XRBlocks sviluppato per realtà estesa (XR) e prevede anche la possibilità di manipolare blocchi fisici mappati su blocchi virtuale.

1 Motivazioni e obiettivi

Nei lavori descritti in [1, 3, 4] sono stati sperimentati metodi e formati laboratoriali per aumentare il coinvolgimento dei partecipanti in attività di coding basate su sistemi a blocchi come Scratch¹. In recenti esperimenti abbiamo iniziato ad esplorare le potenzialità del coding immersivo [2], ovvero ambienti in realtà virtuale in cui i giocatori, dotati di visori, affrontano sfide di coding attraverso linguaggi ed editor a blocchi sviluppati in Unity.

In questo articolo, presentiamo un'estensione del sistema presentato in [2] basato su un nuovo linguaggio a blocchi per realtà estesa (XR) che integra la percezione aptica passiva, consentendo agli utenti di interagire con elementi virtuali corrispondenti a oggetti fisici ed introdurre quindi feedback tattile.

2 Il sistema a blocchi XRCoding

XRCoding è un nuovo sistema di codifica a blocchi basato su XR per giochi 3D che integra l'aptica passiva per introdurre un feedback tattile. Ciò si ottiene tracciando delle vere e proprie scatole di cartone che rappresentano i blocchi di codifica all'interno dell'ambiente virtuale. Il sistema è stato sviluppato utilizzando il motore di gioco Unity, con Meta Quest 2 che funge da Head-Mounted Display (HMD). Le capacità di hand-tracking di Quest 2 sono utilizzate per tracciare le mani dell'utente all'interno dell'ambiente virtuale e per visualizzarle nella scena virtuale. Questa funzionalità consente agli utenti di interagire direttamente con i blocchi di codice utilizzando le mani. L'ambiente virtuale è stato progettato come una semplice stanza con un tavolo allineato alla posizione della sua controparte reale, che funge da spazio di lavoro designato. La figura 1 mostra lo spazio di lavoro nell'ambiente virtuale. L'idea chiave è quella di completare un modello di gioco arcade 2D definito in Unity. Il gioco arcade viene mostrato su una tela proiettata nella scena 3D nell'area adiacente allo spazio di lavoro. L'obiettivo dell'attività di coding proposta è la programmazione delle funzionalità del personaggio principale del gioco arcade. Gli utenti possono utilizzare i controlli virtuali (il joystick virtuale e i pulsanti mostrati in Fig. 2) per sfruttare le funzionalità integrate nel modello (ad esempio, cambiare la direzione di uno sprite). Ulteriori funzionalità (ad esempio, il valore attuale della velocità, i flag di abilitazione all'azione, ecc.) possono essere associate al blocco Start nello spazio di lavoro

¹<https://scratch.mit.edu/>



Figura 1: La tavola virtuale è allineata a quella reale e utilizzata come spazio di lavoro.



Figura 2: Vista in VR del gioco arcade

tramite uno script a blocchi. In particolare, per iniziare a costruire il proprio programma, l'utente può trascinare e posizionare un blocco di codice a sua scelta nell'area bianca-trasparente sotto il blocco "Start" (vedi Fig. 1). L'utente può quindi premere il pulsante verde nello spazio di lavoro per avviare lo script. Il pulsante rosso rimuove il blocco posizionato, ripristinando così la scena. Lo script a blocchi collegato al blocco Start può essere interpretato come il gestore dell'evento associato alla pressione del pulsante verde. Gli esercizi di coding, discussi nella sezione 3, sono progettati secondo questo paradigma di interazione. Ad esempio, l'utente può iniziare a giocare al gioco arcade utilizzando le funzionalità e i widget di interazione integrati. Quando si incontra un ostacolo nel gioco, l'utente può ricorrere all'area di lavoro di codifica per programmare una funzionalità aggiuntiva per procedere con il livello successivo del gioco e così via.

Il nostro linguaggio di codifica a blocchi fornisce la seguente serie di comandi (i blocchi virtuali corrispondenti sono in Fig. 3): *Var*, *GetVar*, *Value*, *FloatOp*, *BoolOp*, *If*, *While*. I

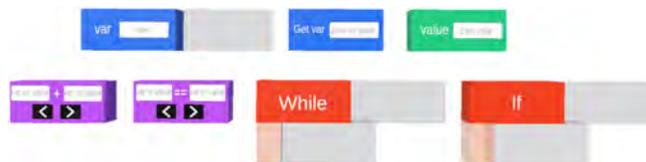


Figura 3: Blocchi del linguaggio XRCoding.

blocchi *Var* sono usati per creare o fare riferimento a una variabile. I blocchi *GetVar* sono utilizzati per ottenere il valore di una variabile. I blocchi *Value* sono usati insieme ai blocchi *Var* per assegnare un valore a una variabile. Il blocco *Value* converte il valore inserito nella casella di testo nel tipo corretto e lo restituisce al blocco *Var* collegato alla sua sinistra. I blocchi *FloatOp* sono utilizzati per calcolare operazioni tra variabili o valori. I blocchi *If* e *While* vengono utilizzati per introdurre istruzioni di controllo, come mostrato in Fig. 4. In

particolare, il blocco *If* implementa un salto condizionale, mentre il blocco *While* definisce un ciclo fino a quando la condizione diventa falsa, quindi procede ai blocchi successivi. In Fig. 4, viene presentato un esempio di programma che inizializza una variabile con un valore di 10 e ne decrementa il valore fino a raggiungere lo zero. Nel nostro lavoro, per migliorare la

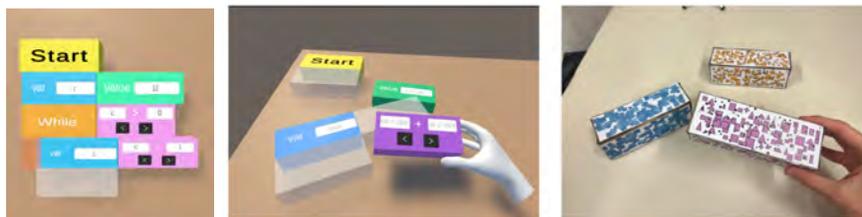


Figura 4: Esempio di programm (sinistra); Blocchi virtuali e fisici (destra).

manipolazione dei blocchi di coding, sfruttiamo l’aptica passiva: gli utenti interagiscono con i blocchi di cartone fisici mentre visualizzano contemporaneamente i corrispondenti blocchi di codifica virtuali nell’ambiente virtuale (vedi Fig. 4). Per interagire con oggetti fisici all’interno di ambienti virtuali, è essenziale tracciare la loro posa a 6 gradi di libertà (6DoF) in tempo reale. A tale scopo abbiamo utilizzato un approccio basato sull’SDK di Vuforia, seguendo la metodologia descritta in [5].

3 Esercizi di coding

Gli esercizi di coding sviluppati nei sistemi introducono in maniera graduale i concetti di base del coding (assegnazione, iterazione, alternativa, ecc). Gli esercizi devono essere risolvibili con poche righe di codice, a causa dello spazio limitato dell’area di lavoro, vincolata sia dai limiti fisici del tavolo sia dal campo visivo della telecamera utilizzata per seguire i blocchi fisici.

Vediamo come esempio il gioco arcade chiamato Albion. Per superare il primo livello occorre predisporre uno script per modificare il valore di una variabile che specifica la velocità dello sprite, come in Fig. 6 (a). Quando tale script viene eseguito, lo sprite può essere spostato in direzione orizzontale utilizzando il joystick virtuale mostrato in Fig. 2.

Per risolvere il secondo livello, occorre predisporre uno script per controllare il valore corrente della variabile velocità e quindi impostare un flag che abilita l’azione di salto per il personaggio cavaliere, come in Fig. 6 (b). Lo sprite può quindi essere spostato in direzione orizzontale usando un pulsante virtuale per l’azione di salto. Nel terzo livello, occorre predisporre uno



Figura 5: Livelli di Albion arcade

script per sconfiggere il nemico che appare nella scena, vedi Fig. 5, utilizzando gli attacchi attivati dalla pressione di un pulsante virtuale.

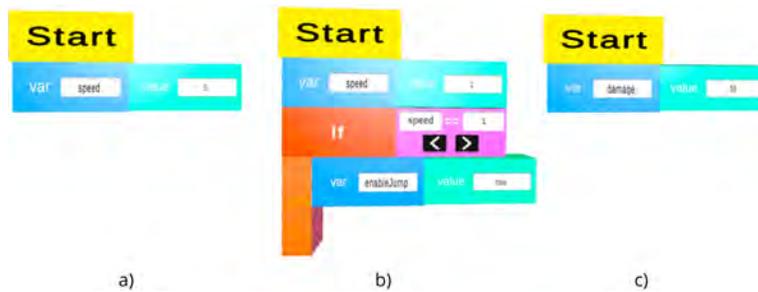


Figura 6: Soluzioni per il gioco Albion: a) primo livello, b) secondo livello, c) terzo livello.

4 Discussione

Per provare il sistema sviluppato, abbiamo condotto dei test iniziali coinvolgendo studenti della scuola secondaria confrontando un ambiente di coding desktop basato su Blockly e il sistema XRCoding. Nella valutazione dell'esperienza utente XRCoding ha ottenuto punteggi migliori rispetto a parametri come stimolo e novità. Blockly ha ottenuto tuttavia punteggi più alti in termini di efficienza e affidabilità per le difficoltà legate all'interazione con l'editor in VR.

Tra le possibili estensioni di nostro interesse, stiamo valutando la possibilità di implementare XRblocks in Cospaces in maniera da sfruttare la possibilità di generare istanze web, VR ed AR partendo dallo stesso progetto. Come in XRCoding, la costruzione ed esecuzione di un programma richiede dimestichezza con l'ambiente virtuale fornendo ulteriori elementi di gioco e sfida in questo tipo di attività.

Riferimenti bibliografici

- [1] Manuela Chessa, Giorgio Delzanno, Angelo Ferrando, Luca Gelati, Giovanna Guerrini, Viviana Mascardi, Nicoletta Noceti, Francesca Odone, and Francesca Vitali. Smart Rogaining for computer science orientation. *Frontiers in Education*, 7, 2022.
- [2] Manuela Chessa, Giorgio Delzanno, Davide Giovanetti, Giovanna Guerrini, Filippo Manini, Davide Miggiano, Marianna Pizzo, and Eros Viola. Work-in-progress - iCoding: Immersive coding in Unity. In *9th International Conference of the Immersive Learning Research Network (iLRN2023)*, 2023.
- [3] Giorgio Delzanno, Angelo Ferrando, Giovanna Guerrini, Matteo Pusceddu, and Giovanni Zanone. Coding maps: A distance laboratory on computational thinking inspired by modal logic. In Giovanni Fulantelli, Daniel Burgos, Gabriella Casalino, Marta Cimitile, Giosuè Lo Bosco, and Davide Taibi, editors, *Higher Education Learning Methodologies and Technologies Online - 4th International Conference, HELMeTO 2022, Palermo, Italy, September 21-23, 2022, Revised Selected Papers*, volume 1779 of *Communications in Computer and Information Science*, pages 497–508. Springer, 2022.
- [4] Giorgio Delzanno, Luca Gelati, Giovanna Guerrini, Angela Sugliano, and Daniele Traversaro. Experience-based training in computer science education via online multiplayer games on computational thinking. In Giovanni Fulantelli, Daniel Burgos, Gabriella Casalino, Marta Cimitile, Giosuè Lo Bosco, and Davide Taibi, editors, *Higher Education Learning Methodologies and Technologies Online - 4th International Conference, HELMeTO 2022, Palermo, Italy, September 21-23, 2022, Revised Selected Papers*, volume 1779 of *Communications in Computer and Information Science*, pages 459–470. Springer, 2022.

- [5] Lorenzo Gerini, Fabio Solari, and Manuela Chessa. A cup of coffee in mixed reality: analysis of movements' smoothness from real to virtual. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 566–569, 2022.

Esperienze di apprendimento non-convenzionali: insegnare informatica uscendo dagli schemi*

Ilenia Fronza¹, Gennaro Iaccarino², and Veronica Rossano³

¹ Libera Università di Bolzano
ilenia.fronza@unibz.it

² Direzione Istruzione e Formazione Italiana, Bolzano
gennaro.iaccarino@scuola.alto-adige.it

³ Università degli Studi di Bari Aldo Moro
veronica.rossano@uniba.it

Abstract

Le esperienze di apprendimento non convenzionali, come i coding camp, offrono alla didattica dell'informatica l'opportunità di uscire dagli schemi curricolari diventando attività laboratoriale immersiva, simulazione di ambienti di lavoro reali e virtuali. Tuttavia, la loro progettazione pone numerose sfide. OSCAR è un progetto europeo che mira a raccordare le esperienze di apprendimento non-convenzionali con la programmazione scolastica tradizionale, promuovendo una didattica ibrida ed efficace, e favorendo l'inclusione di studenti e studentesse con bisogni educativi speciali. Questo articolo illustra come OSCAR intenda rispondere alle sfide della didattica non convenzionale, riportando spunti di riflessione basati sulle prime fasi del progetto.

1 Esperienze di apprendimento non convenzionali

La *didattica non convenzionale* ha la caratteristica di uscire dagli schemi tradizionali diventando attività laboratoriale immersiva, simulazione di ambienti di lavoro reali o virtuali [19]. Un esempio sono i *coding camp*, eventi in cui studenti provenienti da contesti diversi si riuniscono per affrontare in modo collaborativo sfide complesse in un arco di tempo limitato, immergendosi in un ambiente di lavoro dinamico e completamente laboratoriale [16]. Iniziative di questo tipo godono già di una consolidata notorietà nella diffusione di *competenze informatiche* [19], promuovendo esercizi di prototipazione rapida, problem-solving e incoraggiando il pensiero creativo e l'implementazione delle proprie idee in un contesto reale [6].

Le esperienze di apprendimento non convenzionali contribuiscono a migliorare l'inclusione [22], ad avvicinare le ragazze alle discipline STEM [13], e a promuovere le competenze necessarie per diventare cittadini attivi [23]. Inoltre, contrastano la *dispersione implicita* [1], il nuovo malessere che affligge la scuola italiana [20, 23], stimolando l'interesse e la motivazione ad imparare. Non a caso infatti, il lavoro in team, solitamente protagonista di queste esperienze, promuove le competenze di comunicazione e l'assertività, nell'ottica dell'apprendimento continuo [16]. La didattica non convenzionale favorisce la partecipazione di tutti, indipendentemente dal proprio funzionamento, garantendo ad ognuno la possibilità di esprimersi, di trovare la soluzione e di partecipare con le proprie abilità. L'uso della tecnologia permette di personalizzare e valorizzare i differenti stili di apprendimento [23, 25].

*This study was co-funded by the European Union in the framework of the OSCAR project (ERASMUS-EDU-2023-PI-FORWARD, Project number: 101132432). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Da qualche anno il mondo della ricerca studia l'efficacia della didattica non convenzionale, le competenze trasmesse [19], il valore educativo professionalizzante (ad esempio, nell'ingegneria del software [8]), e la necessaria integrazione con il curriculum scolastico tradizionale [18, 19]. Le indicazioni per organizzare attività in formato online o ibrido [10] sono meno diffuse di rispetto al formato in presenza [8], anche se il formato online o ibrido è divenuto essenziale alla luce dei cambiamenti post-pandemia del mondo del lavoro: ricerca e pedagogia concordano nell'affermare che il formato ibrido sarà la "nuova normalità" di cui occorre trasmettere le competenze. Inoltre, negli ultimi anni questo formato è stato utilizzato per migliorare l'accessibilità alle nuove competenze digitali. Ad esempio, sono stati analizzati i benefici dei coding camp da remoto sulla comunicatività da parte di studenti autistici [2]; stesso approccio, ma con obiettivi diversi anche in [14], in cui viene proposto un coding camp per studenti delle scuole superiori con disturbi dello spettro autistico.

2 Le sfide da affrontare

La progettazione di coding camp, come esperienze di apprendimento non convenzionali, pone numerose sfide agli organizzatori (Figura 1). Una di queste è sicuramente quella di proporre una *didattica dell'informatica efficace e significativa*, puntando su tematiche accattivanti e coinvolgenti, ma allo stesso tempo rimanendo in linea con le richieste del mondo scolastico. A questo scopo, è essenziale raccordare le attività dei coding camp con la programmazione curricolare delle singole scuole [19], per garantire l'acquisizione e l'approfondimento non solo di competenze disciplinari contigue alla programmazione didattica dell'indirizzo di studio, ma anche di competenze trasversali di cittadinanza attiva spendibili all'interno del contesto scolastico. Alcuni esempi in letteratura spingono a lavorare in questa direzione. Ad esempio, in [3] e [7], dopo un coding camp, gli studenti sono stati incoraggiati a continuare lo sviluppo di app *socialmente rilevanti* anche durante la normale attività scolastica, così da mettere in pratica quanto appreso (programmazione a blocchi, algoritmi semplici, correttezza) e immergere il tutto in un contesto di educazione civica. Interessante esempio di *service learning* [17].

Altra sfida di primaria importanza è la realizzazione di una *didattica realmente inclusiva* personalizzando gli stili di apprendimento e sviluppando nuove competenze [11, 23] attraverso attività che tengano conto dei bisogni educativi di ognuno ma capaci di coinvolgere tutti. Questa sfida, già ambiziosa nella didattica convenzionale, lo è ancora di più nelle esperienze di apprendimento non convenzionali, che attraggono partecipanti con esigenze e background diversi e dove gli organizzatori in genere non conoscono a fondo le esigenze di ognuno. Una possibile strategia è puntare su tematiche *socialmente rilevanti* [11, 17], approccio motivazionale che si ispira al concetto di *literacy scientifica* ossia "l'abilità degli studenti di impegnarsi nelle questioni scientifiche e nelle idee della scienza, in quanto cittadini che riflettono" descritta in PISA [15]. Inoltre, si dovrebbero applicare il più possibile i sette principi chiave definiti dalla pedagogia per la realizzazione di ambienti di apprendimento inclusivi [25]: il lavoro in team, l'adattamento dei materiali didattici, l'utilizzo di mappe concettuali e supporti visivi, il problem solving come strumento didattico, il supporto alla metacognizione, utilizzo delle emozioni positive come variabili dell'apprendimento, valutazione e feedback.

Tuttavia, diversità non significa solo "bisogni educativi speciali". Ad esempio, l'equilibrio di genere nelle discipline STEM è una delle sfide più attuali del mondo della pedagogia. L'interesse delle ragazze per l'informatica cala precocemente durante l'istruzione primaria e secondaria, con un recupero minimo nelle fasi successive [9]. Per arginare questo problema e *attrarre più ragazze verso le discipline STEM*, la ricerca pedagogica ha analizzato diverse strategie [9, 13]. Le attività di apprendimento non convenzionali hanno stimolato la ricerca a livello mondiale. In



Figure 1: Le principali sfide delle esperienze di apprendimento non-convenzionali raggruppate per tipologia.

[12] si discutono vantaggi e svantaggi della gestione degli hackathon in modo neutrale rispetto al genere, in Italia si promuovono camp destinati esclusivamente alle ragazze¹ [4, 5, 21] e, in [13], si analizzano i fattori che possano rendere eventi come i coding camp più vicini al mondo femminile. Inclusione ed equilibrio (non solo) di genere sono sfide cruciali poiché, se non raggiunti, forniscono ai partecipanti una percezione di mancanza di opportunità e l'impossibilità di proiettarsi verso il futuro ed il mondo del lavoro [24]. Questa percezione distorta può portare a quella che viene definita "dispersione implicita" [20, 23]: studenti e studentesse, pur avendo conseguito un diploma, non hanno acquisito le competenze minime adeguate per intraprendere, con consapevolezza ed efficacia, un percorso accademico o professionale post-scolastico [1].

Infine, va considerato che le esperienze di apprendimento non convenzionali sono spesso proposte da università/istituzioni con uno scarso raccordo con il mondo della scuola, o perfino con divergenze comunicative, organizzative, e strutturali. Un'ulteriore sfida consiste quindi nel trovare un *punto di raccordo tra scuola e università/istituzioni*, capace di dialogare con entrambe le parti, puntando a migliorare la qualità delle esperienze didattiche non convenzionali unendo competenze, risorse e tecnologie, coordinando le tempistiche, e permettendo ai docenti di avere una più chiara visione delle possibilità da offrire ai propri studenti.

3 Il progetto OSCAR

Il progetto OSCAR² è un progetto *Erasmus + Forward-Looking Projects* che mira a sviluppare e testare un manuale e una piattaforma innovativa, di alta qualità, e inclusiva per promuovere competenze digitali trasversali attraverso esperienze di apprendimento non convenzionali, online e/o ibride, per gli studenti della scuola secondaria di secondo grado. Come casi d'uso,

¹www.ragazzedigitali.it, www.pinkamp.disim.univaq.it

²<https://oscar-codingcamps.eu>

OSCAR propone tre coding camp su sviluppo di app, serious game ed extended reality. Di seguito, vediamo come il progetto OSCAR miri a rispondere alle sfide identificate nella sezione precedente.

Come illustrato nella Figura 2, il progetto coinvolge sette partner in tre Paesi europei (Italia come capofila, Finlandia, e Spagna), comprendendo Università ed Istituzioni Scolastiche così da unire ricercatori e professionisti per proporre una *didattica dell'informatica efficace e significativa* e distribuire i risultati del progetto su scala europea. I partner accademici³ si completano a vicenda nelle competenze necessarie (tra cui ingegneria del software, formazione informatica, software startup, fattori umani nell'ingegneria del software) per garantire solide fondamenta e qualità alla formazione informatica che verrà proposta. OSCAR vuole inoltre essere un *punto di raccordo tra scuola e università/istituzioni*. A questo scopo, le Istituzioni⁴ contribuiscono a creare un ponte tra il mondo accademico e le scuole, supportando la comunicazione e le attività degli istituti scolastici, e promuovendo la partecipazione ai coding camp. Punti cardine del progetto sono inoltre *l'inclusione* e l'esplorazione di strategie per colmare il gender-gap e *attrarre più ragazze verso l'informatica* proprio attraverso le esperienze didattiche non convenzionali.

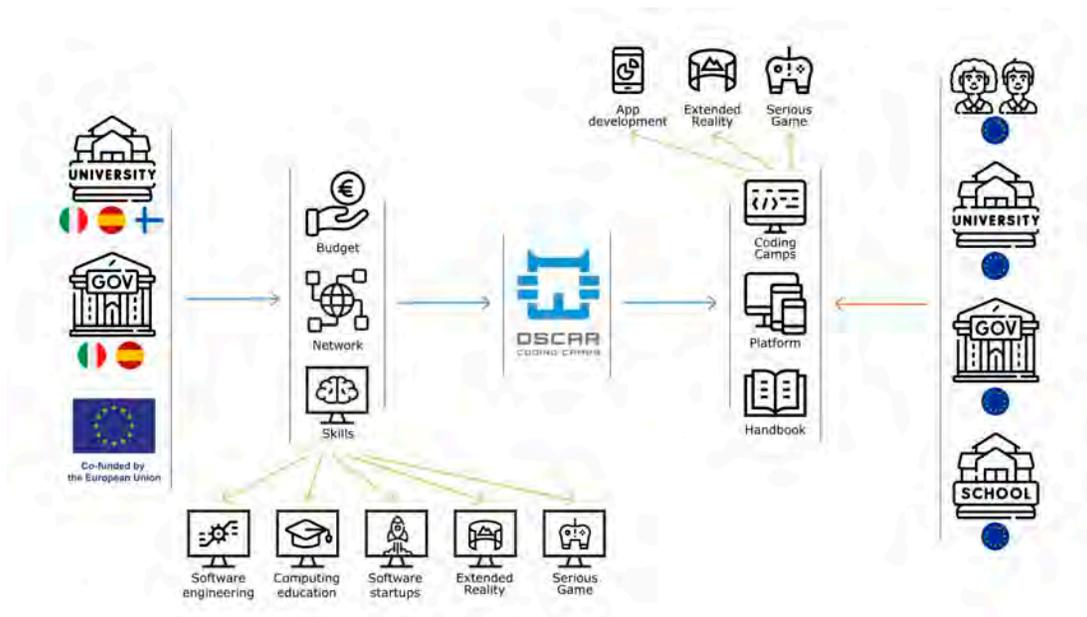


Figure 2: Il progetto OSCAR come punto di raccordo tra scuola e università/istituzioni che organizzano esperienze di apprendimento non convenzionali.

Pur essendo nelle sue prime fasi, il progetto OSCAR fornisce già numerosi spunti di riflessione e *lesson learned*. Ad esempio, la nascita di progetti di questo tipo richiede ovviamente esperienza nello specifico ambito della didattica non convenzionale, ma anche un grande lavoro di costruzione di una rete di contatti (tra Università, scuole, istituzioni), importante soprattutto in un campo di ricerca come quello della didattica dell'informatica, che vede un numero

³Libera università di Bolzano, Università di Bari Aldo Moro, Università Jaume I (Spagna), e Università di Jyväskylä (Finlandia).

⁴Direzione Istruzione e Formazione Italiana (Bolzano), Ministero dell'Istruzione e del Merito, Diputació de Castelló (Spagna).

piuttosto limitato di ricercatori/ricercatrici sparsi in diverse Università. Nel caso specifico, OSCAR è nato grazie agli sforzi dedicati per anni all'organizzazione di coding camp⁵ e alla rete di contatti creatasi anche nell'ambito del Laboratorio CINI *Informatica e Scuola*.

Durante i primi mesi di progetto si sono svolte alcune attività pilota. Ad aprile/maggio 2024 i due partner italiani di OSCAR (Bari e Bolzano) hanno collaborato ad un coding camp pilota incentrato sullo sviluppo di serious game per sensibilizzare gli utenti alla cybersecurity, applicando in modo laboratoriale i concetti introduttivi su sicurezza informatica, elementi di design dei serious game, e programmazione per la realizzazione di giochi per smartphone. Il coding camp si è svolto in modalità ibrida, coinvolgendo 82 studenti/esse della scuola secondaria di secondo grado (35 a Bolzano e 47 a Bari). Il camp ha consentito di testare il formato ibrido e valutare i primi risultati della ricerca. Inoltre, è stato possibile identificare criticità e problematiche da affrontare e risolvere nell'ambito del progetto, come la strutturazione della didattica in maniera completa senza togliere spazio all'attività laboratoriale, l'allineamento dell'infrastruttura tecnica tra le diverse sedi, le diverse esigenze di orario, la gestione della collaborazione con più scuole/docenti. A partire da quanto appreso in questo camp pilota, i prossimi mesi vedranno la progettazione e la realizzazione di nuovi camp improntati sullo sviluppo di applicazioni per dispositivi mobili e l'utilizzo della realtà virtuale estesa.

References

- [1] Autorità Garante per l'Infanzia e l'Adolescenza. La dispersione scolastica in Italia: un'analisi multifattoriale - Documento di studio e di proposta. <https://www.garanteinfanzia.org/sites/default/files/2022-06/dispersione-scolastica-2022.pdf>, 2022.
- [2] A. Begel, J. Dominic, C. Phillis, T. Beeson, and P. Rodeghero. How a remote video game coding camp improved autistic college students' self-efficacy in communication. In *SIGCSE '21*, page 142–148. ACM, 2021.
- [3] Anna Brancaccio, Luis Corral, Ilenia Fronza, and Gennaro Iaccarino. Building smart apps for smart cities: un esempio di sinergia tra pcto ed educazione civica, concluso ai tempi di covid-19. In *DIDAMATICA 2020 - Smarter School for Smart Cities. 12-13 Novembre 2020*, pages 380–389, 2020.
- [4] A. Di Marco and L. Tarantino. Pinkamp: A novel educational format for the recruitment of high school girls in stem subjects. In *EDULEARN20 Proceedings*, 12th International Conference on Education and New Learning Technologies, pages 6967–6976. IATED, 6-7 July, 2020 2020.
- [5] F. Faenza, C. Canali, and A. Carbonaro. Evaluating different approaches to closing the gender gap at ict summer camps in italy. In *4th International Conference on Gender Research, ICGR 2021*, pages 104–113, 2021.
- [6] Ilenia Fronza and Luis Corral. A facilitator's guide to create and consolidate a teenage coding camp. *ACM Inroads*, 15(2):31–38, may 2024.
- [7] Ilenia Fronza, Luis Corral, Claus Pahl, and Gennaro Iaccarino. Evaluating the effectiveness of a coding camp through the analysis of a follow-up project. In *Proceedings of the 21st Annual Conference on Information Technology Education*, page 248–253, New York, NY, USA, 2020. Association for Computing Machinery.
- [8] K. Gama. Developing course projects in a hack day: an experience report. In *Proc. of the ACM Conf. on Innovation and Technology in CS Education*, pages 388–394, 2019.
- [9] L. Happe, B. Buhnova, A. Koziolok, and I. Wagner. Effective measures to foster girls' interest in secondary computer science education. *Education and Information Technologies*, 26:2811–2829, 2020.

⁵<https://mobiledev.inf.unibz.it>, <https://gamedev.events.unibz.it>

- [10] A. Happonen, M. Tikka, and U.A. Usmani. A systematic review for organizing hackathons and code camps in covid-19 like times: Literature in demand to understand online hackathons and event result continuation. In *2021 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6, 2021.
- [11] Gennaro Iaccarino, Sara Tosi, Ilenia Fronza, and Luis Corral. Laboratorio di bioinformatica: un pcto innovativo in periodo pandemico e post pandemico. In *DIDAMATICA 2022 - La trasformazione digitale nella Scuola, negli ITS, nell'Università e nella formazione professionale*, pages 292–302, 2022.
- [12] Y. Kovaleva, A. Happonen, and V. Hasheela-Mufeti. Pros and cons of running educational hackathons in a gender-neutral fashion. In *3rd Int. Workshop on Gender Equality, Diversity and Inclusion in Software Engineering*, pages 27–34, 2022.
- [13] Y. Kovaleva, A. Happonen, and A. Mbogho. Towards gender balance in modern hackathons: literature-based approaches for female inclusiveness. In *Proc. of the 3rd W. on Gender Equality, Diversity, and Inclusion in Software Engineering*, page 19–26, New York, NY, USA, 2022. ACM.
- [14] M. Moster, E. Kokinda, M. Re, J. Dominic, J. Lehmann, A. Begel, and P. Rodeghero. "can you help me?": an experience report of teamwork in a game coding camp for autistic high school students. *ICSE-SEET '22*, page 50–61, NY, USA, 2022. ACM.
- [15] OCSE PISA. I risultati degli studenti italiani in letteratura, matematica e scienze. Rapporto nazionale. https://www.oecd.org/pisa/publications/PISA2018_CN_ITA_IT.pdf, 2018.
- [16] A. Oliveira, H. Assumpcao, J. Queiroz, L. Piardi, J. Parra, and P. Leitao. Hands-on learning modules for upskilling in industry 4.0 technologies. In *IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 1–6, 2022.
- [17] Lorenza Orlandini, Stefania Chipa, and Chiara Giunti. *Il Service Learning per l'innovazione scolastica*. Ricerche Indire - Ed. Carocci, 2020.
- [18] J. Porras, J. Khakurel, J. Ikonen, A. Happonen, A. Knutas, A. Herala, and O. Drögehorn. Hackathons in software engineering education: lessons learned from a decade of events. In *Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials*, pages 40–47, 2018.
- [19] J. Porras, A. Knutas, J. Ikonen, A. Happonen, J. Khakurel, and A. Herala. Code camps and hackathons in education - literature review and lessons learned. In *Hawaii Int. Conf. on System Sciences*, 01 2019.
- [20] Roberto Ricci. *L'editoriale - La dispersione scolastica implicita*. Invalsi open, 2019.
- [21] Laura Tarantino and Antiniscia Di Marco. Gender equality: the need for explicit address and actions in ict and socio-technical systems agendas. In *Proceedings of the 7th International Workshop on Socio-Technical Perspective in IS Development (STPIS 2021), Virtual conference in Trento, Italy, October 11-12, 2021*, volume 3016 of *CEUR Workshop Proceedings*, pages 258–267. CEUR-WS.org, 2021.
- [22] K. Thayer and A.J. Ko. Barriers faced by coding bootcamp students. *ICER '17*, page 245–253, New York, NY, USA, 2017. ACM.
- [23] Sara Tosi, Gennaro Iaccarino, and Ilenia Fronza. Didattica non convenzionale: un approccio inclusivo per la lotta alla dispersione implicita. In *Didattica e inclusione scolastica. 3-4 Marzo 2023, Bressanone (BZ)*, 2023.
- [24] UNESCO. Inclusion and gender equality: brief on inclusion in education. In *UNESCO briefs on inclusion in education*, pages 1–11, 2023.
- [25] F. Zambotti. *BES a scuola. I 7 punti chiave per una didattica inclusiva*. Erikson, 2015.

Formazione universitaria per l'insegnamento dell'informatica e casi studio con Arduino

Alessandra De Vitis¹, Guglielmo Abbruzzese²

¹ Università degli Studi di Roma "Sapienza"; I.I.S. "Guglielmo Marconi", Latina
devitis@di.uniroma1.it

² Università degli Studi di Roma "Tor Vergata"
abbruzzese@axp.mat.uniroma2.it

Abstract

A valle di una introduzione con un excursus storico sulle modalità di conseguimento dell'abilitazione all'insegnamento nella scuola secondaria, si illustrano i contenuti di un corso universitario dedicato alla didattica dell'informatica. Tale corso -che mira ad orientare i futuri laureati all'insegnamento secondario- propone metodi e strumenti per realizzare attività e progetti stimolanti ed inclusivi. I casi studio con Arduino presentati sono corredati di materiali ideati per coadiuvare gli insegnanti nella concreta attività didattica.

1 Introduzione

1.1 La formazione e il reclutamento dei docenti: breve excursus storico

Negli ultimi decenni, la formazione iniziale degli insegnanti in Italia è stata realizzata in modo discontinuo e frammentato, all'interno di un quadro legislativo continuamente mutevole. Limitandosi al solo ciclo secondario, è utile ripercorrere brevemente alcune tappe essenziali. Con la Legge n. 341 del 19 novembre 1990 è stata istituita la SSIS (Scuola di Specializzazione per l'Insegnamento Secondario), un percorso post-laurea biennale al termine del quale era possibile conseguire l'abilitazione per l'insegnamento nella scuola secondaria di primo e secondo grado.

Le università hanno avviato le SSIS diversi anni dopo (il primo ciclo è stato attivato negli anni 1999-2000); con le SSIS, infatti, la formazione iniziale dei docenti della scuola secondaria è stata affidata per la prima volta alle università e non al solo concorso statale.

Dopo il nono ciclo (negli anni 2008-2009) la SSIS è stata ufficialmente abolita (Di Pasqua, Grassilli, & Storti, 2009). Con la Legge n. 244 del 24 dicembre 2007 è subentrato il cosiddetto TFA (Tirocinio Formativo Attivo), la cui durata annuale consentiva di conseguire l'abilitazione in tempi minori. Anche il TFA, avviato nel 2010, ha avuto vita breve: la Legge n. 107 del 13 luglio 2015, seguita dal D. L. n. 59 del 13 aprile 2017, introduceva il percorso FIT (Formazione Iniziale e Tirocinio) articolato su tre anni: i primi due, diversamente distribuiti tra scuola e università, prevedevano un modesto compenso per le attività di tirocinio; il terzo, regolato dal CCNL docenti, era corrispondente al cosiddetto "anno di prova" per l'assunzione a tempo indeterminato (De Angelis, 2018).

Uno dei requisiti per l'ammissione al FIT era il conseguimento di 24 CFU nelle discipline antropo-psico-pedagogiche e nelle metodologie e tecnologie didattiche. Di fatto, il FIT non è mai partito, lasciando in eredità solo l'acquisizione dei 24 CFU per accedere al concorso statale. Successivamente, con la Legge n. 79 del 29 giugno 2022 e il successivo DPCM del 4 agosto 2023, sono stati istituiti i cosiddetti "60 CFU", corrispondenti a un nuovo tipo di percorso post-laurea

propedeutico al concorso, la cui denominazione ha suscitato non poche perplessità in una parte del mondo accademico: è infatti difficile intravedere un progetto pedagogico e una “idea di insegnante” in un corso identificato con il mero numero di crediti da acquisire (Baldacci, Bocciolesi, Michelini, Odini, & Oliverio, 2023).

Nel frattempo, un insieme disomogeneo di procedure concorsuali ha consentito di stipulare un contratto a tempo indeterminato a docenti di preparazione molto diversa. Esempi significativi sono forniti dai cosiddetti “concorsi STEM”,¹ attuati in tempi brevi per sopperire alla cronica carenza di insegnanti in alcune materie di area scientifico-tecnologica, fra cui l’informatica (Celestino, 2021). La prova scritta di questi concorsi, estremamente selettiva, ha escluso un numero notevole di candidati (fino al 90% circa); questi hanno però potuto rimettersi in gioco a distanza di pochi mesi tramite altri concorsi riservati a chi aveva già accumulato tre anni di supplenze, anche non interamente svolte nell’ambito della specifica classe di concorso.

Le prove di questi concorsi riservati (peraltro solo orali) sono state superate dalla quasi totalità dei concorrenti (dato facilmente verificabile dal confronto tra il numero di candidati e le graduatorie degli ammessi nelle varie province). Altri fattori hanno contribuito alla perdita di credibilità del sistema di reclutamento: basti pensare al comportamento non sempre professionale di alcune commissioni a cui sono state affidate le prove orali dei concorsi STEM (Celestino, 2021).

1.2 L’insegnamento dell’informatica nella scuola secondaria

La carenza di insegnanti STEM non riguarda solo l’Italia; si tratta di una emergenza a livello europeo, come illustrato dal Rapporto Longuet (Longuet, 2022). Tuttavia, elementi peculiari del sistema italiano aggravano notevolmente il quadro: non solo le modalità di reclutamento estremamente caotiche, ma anche la perdita di prestigio sociale evidenziata da stipendi assolutamente inadeguati. A ciò si aggiunge un numero di laureati particolarmente basso in alcune discipline di area tecnico-scientifica, tanto da indurre l’implementazione di un piano su scala nazionale (il cosiddetto PLS – Piano Lauree Scientifiche) (Vittorio & Predazzi, 2007) per aumentare l’attrattività dei corsi di laurea le cui prospettive occupazionali sono particolarmente promettenti. Il PLS coinvolge anche la classe di laurea L-31 in “Scienze e tecnologie informatiche”, con un progetto nazionale motivato dalle seguenti criticità: confusione rispetto allo status scientifico della disciplina (Abbruzzese, Celestino, & De Vitis, 2023); progressivo sbilanciamento di genere; scarsa presenza di professionalità informatiche nella scuola. Come si legge in corrispondenza dell’ultimo punto (Progetto nazionale di informatica, s.d.):

L’insegnamento di discipline informatiche nella scuola è affidato a laureati e laureate dalle provenienze molto diverse, anche nel caso della specifica classe concorsuale “A-41 Scienze e tecnologie informatiche”. In generale, inoltre, sono pochi coloro che si laureano in informatica che decidono di intraprendere una carriera nell’insegnamento. Il risultato è che a tutti i livelli scolastici le professionalità specializzate in informatica sono rare, il che aggrava il già menzionato problema di indicazioni e linee guida che mancano di chiarire senza ambiguità il ruolo delle discipline informatiche nella scuola secondaria non specialistica, confondendo aspetti scientifici e tecnologici con il piano più meramente strumentale e di servizio.

Figura 1 - Estratto dal sito web "Progetto nazionale di informatica"

La carenza di insegnanti abilitati nella classe di concorso A-41 è testimoniata dagli interpellati nazionali pubblicati in caso di irreperibilità di docenti locali sui siti specializzati nel settore scolastico. Mentre il governo britannico offre supplementi salariali esentasse per attirare laureati in “materie prioritarie” come matematica, informatica, fisica o chimica (Longuet, 2022), in Italia non sono previsti incentivi economici.

¹ Acronimo ormai di uso comune che sta per “Science, Technology, Engineering and Mathematics”

2 Un corso di didattica dell'informatica

Al fine di sopperire alla carenza di informatici che decidono di intraprendere la carriera dell'insegnamento secondario, dall'a.a. 2018/19 presso l'Università Sapienza di Roma è attivo il corso in lingua inglese "Methods in Computer Science Education: Design" nell'ambito della laurea magistrale "Computer Science - Informatica".

Il corso, di durata semestrale, presenta dapprima una panoramica generale sull'insegnamento dell'informatica nei diversi indirizzi di studio, concentrandosi successivamente sui vari aspetti della didattica. La visione d'insieme iniziale riveste particolare importanza, in quanto l'insegnamento dell'informatica differisce molto fra i vari indirizzi di studio; inoltre, la classe di concorso A-41 contempla materie di insegnamento differenti che spaziano dai linguaggi di programmazione alle reti di calcolatori, fino allo studio della gestione aziendale (inclusendo aspetti normativi di privacy e sicurezza sul lavoro). I 6 CFU ottenibili sono riconosciuti nell'ambito del percorso di abilitazione all'insegnamento attualmente previsto.

Gran parte del corso è dedicato ad attività di laboratorio consistenti nella progettazione di Unità di Apprendimento (UDA) per piccoli gruppi sotto la supervisione del docente; le attività sono basate su strategie di Game Based Learning (Connolly, Boyle, MacArthur, Hainey, & Boyle, 2012) e Gamification (Dicheva, Dichev, Agre, & Angelova, 2015).

Il microcontrollore Arduino è lo strumento principe nella realizzazione pratica dei progetti, in quanto rappresenta una risorsa preziosa per accrescere l'interesse degli studenti nelle materie STEM grazie alla sua flessibilità d'uso e alla possibilità di realizzare attività coinvolgenti basate sui principi dell'apprendimento per scoperta (De Vitis A. , 2019).

2.1 Struttura del corso

Il corso è strutturato in dieci lezioni della durata di 6 ore ciascuna, che prevedono introduzioni teoriche e attività laboratoriali.

Macro argomenti del corso "Methods in Computer Science: Design"
Analisi critica delle principali metodologie per l'insegnamento sviluppate nella ricerca in didattica dell'informatica, anche in riferimento allo specifico ruolo dell'insegnante, ai nodi concettuali, epistemologici, linguistici e didattici dell'insegnamento e apprendimento dell'informatica, ponendo in particolare l'accento sulla distinzione fra l'informatica come disciplina scientifica da una parte e le applicazioni dell'informatica dall'altra.
Progettazione e sviluppo di metodologie di insegnamento dell'informatica: illustrazione dei principi e dei metodi per la costruzione di attività e più in generale di un curriculum di informatica coerente con gli obiettivi fissati dalle indicazioni nazionali per i licei e dalle linee guida per gli istituti tecnici e professionali.
Esplicitazione del parallelismo fra metodologie informatiche e metodologie didattiche: tecniche di problem solving costruttivo; approccio epistemologico ai problemi; metodi cooperativi di sviluppo delle soluzioni.
Metodologie e tecnologie didattiche per lo studio del rapporto dell'informatica con la società attuale, e in particolare degli aspetti etici legati, per esempio, al trattamento dei dati personali, all'automazione di decisioni e raccomandazioni, alla ridefinizione del concetto di copyright.

Tabella 1 - Struttura del corso

2.2 Progetti Arduino

1. Panoramica generale

Durante le lezioni viene introdotto il funzionamento di Arduino e sono proposti vari esperimenti tipici da eseguire nelle classi di scuola superiore. L'esame finale prevede, relativamente alla componente pratica, la realizzazione di un progetto Arduino adatto a formare uno specifico *target*

di studenti (tipo di istituto, anno di corso, BES², etc.) (De Vitis & Abbruzzese, 2023). Lo studente universitario veste quindi i panni di progettista di attività didattiche, prefiggendosi contestualmente di produrre un artefatto laboratoriale della stessa tipologia attesa dai loro ipotetici allievi, da questi fruibile sia come esempio operativo che come stimolo per progetti da sviluppare in autonomia. I tempi previsti e riportati nelle schede dei progetti, dipendenti dalla complessità degli stessi, si riferiscono allo svolgimento in classe delle attività³.

2. Descrizione di dettaglio di un caso studio:

Tra i circa duecento progetti realizzati nell'ambito del corso in oltre un quinquennio, quello di seguito illustrato è ritenuto particolarmente rappresentativo perché semplice da realizzare e facilmente replicabile da uno studente del terzo anno delle superiori⁴.

Project name	DRACULINO 
Storyline	Draculino is a vampire, and -as everyone knows- vampires hate light. For this reason, he only leaves his mansion late at night, in search for preys, coming back well before dawn breaks.
Target	Third-year students of LSA and ITT (computer science/telecommunication address)
Level	Easy
Learning goals	<i>Introduction to basic programming Arduino, use of elementary electronic components</i>
<i>Hardware</i>	Arduino, correct LED polarization, use of photoresistor, servomotor and resistors.
<i>Software</i>	Basic information on Arduino IDE, definition of output/input pins, serial interface, increment instruction, if instruction.
Operating description	When the light sensor detects darkness (night), the servomotor rotates 180°, revealing Dracula's face and turning on the LED lights. When there is light (day), the servomotor performs another 180° rotation, hiding Dracula's face and turning off the LEDs.
Handiwork	The castle and Dracula are made out of cardboard.
Materials list	Arduino Uno R3, 1 10 Kohm resistor, 3 220ohm resistors 1 micro servo motor, 1white LED, 2 red LEDs, 1 photoresistor, wire for connection, 1 bread board
Lesson planning	Classroom lesson: 2 hours Cardboard construction: 1 hour Software production: 1 hour Assembly and final check: 2 hours
Project details	https://drive.google.com/drive/folders/1v1vdIDB8PIgN1Whg3bdfGu7t4ApvyJfP?usp=sharing

Figura 2 - Esempio di scheda di presentazione di progetto

² Bisogni Educativi Speciali

³ Attività di progettazione creativa e realizzazione di progetti in autonomia da parte degli alunni target non rientra negli obiettivi del presente articolo.

⁴ <https://drive.google.com/drive/folders/1v1vdIDB8PIgN1Whg3bdfGu7t4ApvyJfP?usp=sharing>

3. Altri casi studio selezionati

Nella *Tabella 2* sono elencati i progetti di un campione selezionato tra quelli realizzati dagli studenti, con livelli di difficoltà variabili. La maggior parte dei progetti descrive la realizzazione di attività a scopo ludico ma con precisi obiettivi di apprendimento.

Titolo	A. A.	Descrizione
Simon ⁵	22/23	equivalente elettronico del gioco di memoria Simon nato alla fine degli anni 70
Gioco luci ⁶	21/22	Gioco competitivo per n giocatori (n<5) con combinazioni di luci. Viene scelta una combinazione iniziale e lo scopo è ricordare e ripetere tale combinazione. Prende spunto da un minigioco di Mario Party 3 per Nintendo64
Doctor Smart ⁷	21/22	Attraverso un sensore ad ultra suoni misura il battito cardiaco
Hero's Quest ⁸	21/22	Hero's Quest è un videogioco realizzato con Arduino dove il giocatore impersona un eroe della luce il quale compito è scacciare il malvagio Dark Lord che ha avvolto nelle tenebre e portato in rovina il mondo. Per raggiungere il suo obiettivo l'eroe dovrà farsi strada attraverso 3 capitoli, in ognuno dei quali affronterà un nemico.
Automatic Plant Watering System ⁹	20/21	Sistema automatico di irrigazione piante

Tabella 2 - Casi studio selezionati

3 Conclusioni

L'utilizzo dei microcontrollori Arduino in un corso universitario di didattica dell'informatica si è dimostrato efficace nell'accrescere l'interesse degli studenti verso una potenziale carriera nell'ambito dell'insegnamento. L'approccio utilizzato nel corso ha favorito l'acquisizione di competenze interdisciplinari e metodologie innovative, sottolineando l'importanza di attività pratiche e coinvolgenti.

Bibliografia

Abbruzzese, G., Celestino, T., & De Vitis, A. (2023). Epistemologia, etica ed estetica dell'informatica: insegnamento tra tecnologie e mondo digitale. *Nuova Secondaria Ricerca*, 2, 431-450.

⁵ https://drive.google.com/drive/folders/1ymy_fmQNOJfF-3Gf3QgoddV1iZp4Ku6C?usp=sharing

⁶ https://drive.google.com/drive/folders/1vy9r4v1wjs8TCM1Q-F9aR42iSpawFKb9?usp=drive_link

⁷ https://drive.google.com/drive/folders/11xS82prZemvM9oUR4LBSM-1hXYRJPppK?usp=drive_link

⁸ https://drive.google.com/drive/folders/1sCmlgTfzSYRz_ojYbmGchOEIT62KdHCN?usp=drive_link

⁹ https://drive.google.com/drive/folders/1uI-sXNSK77IRj1H1M2IqwjCeMxxsHKmR?usp=drive_link

- Baldacci, M., Bocciolesi, E., Michelini, M.-C., Odini, L., & Oliverio, S. (2023). Per un'idea di insegnante. *Pedagogia più Didattica*, 9(2), 48-61. doi:10.14605/PD922304
- Celestino, T. (2021, 11 08). *Il tic delle TIC sull'ultimo concorso STEM*. Tratto da La letteratura e noi: <https://laletteraturaenoi.it/2021/11/08/il-tic-delle-tic-sullultimo-concorso-stem/>
- Connolly, T., Boyle, E., MacArthur, E., Hainey, T., & Boyle, J. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, 59(2), 661-686.
- De Angelis, F. (2018, 02 07). *Percorso FIT, tutte le info utili per diventare insegnante nella scuola secondaria*. Tratto da La tecnica della scuola: <https://www.tecnicadellascuola.it/percorso-fit-tutte-le-info-utili-diventare-insegnante-nella-scuola-secondaria>
- De Vitis, A. (2019). Methods In Computer Science Education In High Schools. *EDULEARN19 Proceedings*. Barcelona.
- De Vitis, A., & Abbruzzese, G. (2023). Metodi educativi dell'informatica per studenti con disturbi di apprendimento nella scuola secondaria di secondo grado. In M. M. E. Gentile (A cura di), *Convegno Italiano sulla Didattica dell'Informatica* (p. 237-243). Bari: Università di Bari.
- Di Pasqua, S., Grassilli, B., & Storti, A. (2009). *L'eredità della SSIS. "Luci e ombre" della Scuola per la formazione degli insegnanti*. Trieste: Edizioni Università di Trieste.
- Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in Education: A Systematic Mapping Study. *Educational Technology & Society*, 18(3), 75-88.
- Longuet, M. G. (2022). *Crise d'attractivité du métier d'enseignant: quelles réponses des pays européens? Sénat de la République française. Rapport d'information n° 649 (2021-2022)*. Paris: Sénat de la République française. Tratto da <https://www.senat.fr/rap/r21-649/r21-649.html>
- Progetto nazionale di informatica*. (s.d.). Tratto da Orientazione: <https://www.orientazione.it/il-progetto/piano-lauree-scientifiche/informatica/#premesse>
- Vittorio, N., & Predazzi, E. (2007). Scuola, università e mondo del lavoro: la filiera del progetto «Lauree Scientifiche». *Annali della Pubblica Istruzione*, 2-3, 1-27.

LabOrinto: Progettazione e Realizzazione di Stage di Orientamento verso l'Informatica

Giorgio Delzanno, Giovanna Guerrini e Davide Ponzini

DIBRIS, Università degli Studi di Genova, Italy
{name.surname}@unige.it

Sommario

In questo articolo presentiamo i principi alla base di LabOrinto, un formato adottato negli stage di orientamento per il corso di studi di informatica dell'Università degli Studi di Genova a partire dal 2022/23. La struttura alla base del formato proposto è centrata su attività di laboratorio progettati con principi di gamification ed interconnesse con attività in aula e di approfondimento.

1 Contesto e Motivazioni

Da diversi anni la commissione orientamento del corso di studi di Informatica dell'Università di Genova utilizza gli stage per gli studenti delle scuole superiori, tipicamente classi quarte e quinte, per sperimentare nuove metodologie didattiche e di orientamento verso temi di informatica. Lo stage rappresenta un contesto sfidante per questo tipo di sperimentazione sia per i tempi brevi in cui deve essere svolto sia per l'eterogeneità delle competenze dei partecipanti provenienti da scuole di indirizzi diversi e quindi con diversi livelli di conoscenza dell'informatica, problema comune anche tra gli studenti del primo anno della laurea triennale.

Uno degli obiettivi fondamentali nella progettazione di queste attività è quello di favorire e stimolare la partecipazione attiva non solo degli studenti delle scuole superiori ma anche degli studenti universitari (triennale, magistrale e dottorato) coinvolti come tutor e mentor universitari. Il programma proposto cerca di bilanciare interventi frontali e attività di laboratorio con spazio per approfondimenti da parte dei partecipanti e per discussioni sia con docenti che con tutor e mentor universitari. Il formato è il risultato di successivi raffinamenti di un modello proposto alcuni anni fa e descritto in [2].

Nel resto dell'articolo vedremo in dettaglio la pianificazione e le diverse fasi adottate nelle edizioni di stage e PCTO degli ultimi due anni accademici partendo dal nodo centrale che è appunto la proposta di attività laboratoriali chiamata LabOrinto e le attività in aula ispirate agli approcci presentati in [1, 4, 5].

2 Struttura di LabOrinto

Il razionale che ha guidato la progettazione del formato adottato per LabOrinto è illustrato in Fig. 1. Il formato presenta una serie di attività tutte fortemente collegate tra loro non solo come spunti tematici ma soprattutto come approccio a studio e lavoro individuale e di gruppo. Le attività individuali e di gruppo sono usualmente affiancate dalla presentazione dei percorsi di laurea triennale e magistrale, da simulazioni di lezioni ed esami, dalla presentazione di prospettive professionali, interventi su sul funzionamento dell'università e sui metodi di studio da parte di referenti di orientamento del nostro ateneo.

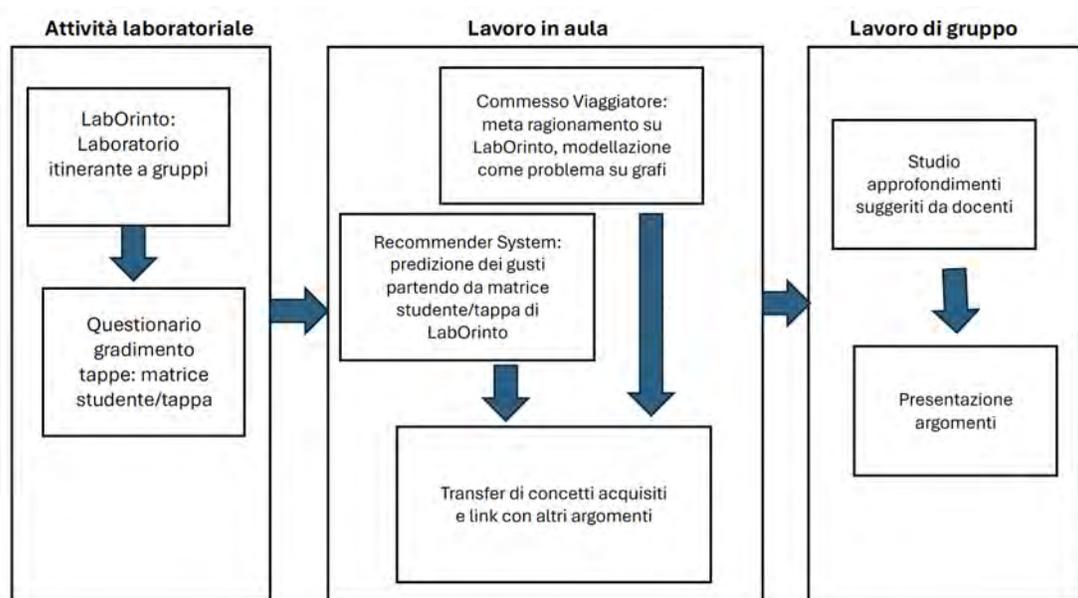


Figura 1: Mappa concettuale di LabOrinto

2.1 Laboratorio di orientamento itinerante

L'attività laboratoriale di LabOrinto è un punto centrale dello stage mirata a stimolare la partecipazione attiva degli studenti e il contatto diretto con tutor e mentor. L'attività ha una durata di 3 ore durante la quale i partecipanti vengono divisi in gruppi (di circa dieci studenti ognuno) per affrontare una serie di sfide collocate in diverse tappe all'interno degli spazi del nostro dipartimento. Ogni sfida è organizzata come un laboratorio di massimo 30 minuti che mette a disposizione un numero di punti proporzionale alla difficoltà dell'attività e alla distanza dal punto iniziale della gara. Il tempo totale del LabOrinto non consente di completare tutte le sfide. In questo modo si vuole favorire la collaborazione all'interno dei gruppi per ottimizzare percorsi da rischedulare quando necessario. I partecipanti hanno inoltre a disposizione una app realizzata nella piattaforma IoT Nodered per visualizzare in tempo reale la lista di tappe libere tra le quali scegliere l'attività successiva. I tutor si occupano di gestire le notifiche relative allo stato delle tappe. Le tappe proposte nelle ultime edizioni sono riassunte nella Tabella 1. Alla fine di LabOrinto viene compilata una classifica usata come priorità nella scelta di una serie di argomenti di approfondimento preparati dai docenti (articoli divulgativi o scientifici). In premio viene quindi messa la possibilità di scegliere un argomento più gradito di altri. I gruppi hanno quindi a disposizione un pomeriggio per analizzare il materiale, preparare dei lucidi e quindi presentare il lavoro svolto. In quest'ottica questa parte dello stage è vista come un'anticipazione delle diverse attività mirate allo sviluppo di soft skill che verranno proposte durante gli anni di università.

2.2 Approfondimenti Tematici

Le attività proposte durante LabOrinto hanno un'ossatura comune per presentare alcuni argomenti in varie modalità e da vari punti di vista.

Attività	Argomento	Strumenti	Descrizione
<i>Al-Khwarizmi</i>	Algoritmi	Unplugged	Ricerca di una soluzione ad un problema NP completo.
<i>Torre di Hanoi</i>	Strutture dati	Unplugged	Ricerca di una soluzione al problema delle torre di Hanoi.
<i>Capture the Flag</i>	Cybersecurity	Piattaforma web	Sfida di cybersecurity del tipo Capture-the-Flag.
<i>Gioco di squadra</i>	Algoritmi paralleli	Unplugged	Sfida tra gruppi per ordinare in parallelo un mazzo di carte.
<i>Codice Segreto</i>	Privacy	Unplugged	Sfida tra gruppi per decifrare un messaggio codificato.
<i>Kevin Bacon</i>	Database	Neo4j	Laboratorio per inferire relazioni tra membri di una rete sociale.
<i>Gossip</i>	Sistemi embedded	Micro:bit	Implementazione di un algoritmo di gossiping attraverso comunicazione radio.
<i>Pocketcode</i>	Eventi e concorrenza	PocketCode	Creazione di una game app su tablet.
<i>RISChiatutto</i>	Architetture	Simulatore	Programmazione con sole 6 istruzioni nello stile delle architetture RISC.
<i>Coding immersivo</i>	Pensiero computazionale	Unity	Coding in un sistema di realtà virtuale immersivo.
<i>Coding imperfetto</i>	Gaming	Wooclap	Sfida con domande ispirate a bug e glitch di videogame storici.
<i>Dama Cooperativa</i>	A.I.	Unplugged	Sfida a dama dove le pedine sono autonome di decidere le mosse di un giocatore.
<i>Teachable Machine</i>	Computer vision	Teachable Machine	Classificazione di immagini/suoni con deep learning.

Tabella 1: Attività proposte

Nelle edizioni 2022/23 e 2023/24 abbiamo utilizzato come primo tema i sistemi di raccomandazione come suggerito in [1]. Oltre ad essere un tema legato ad applicazioni usate quotidianamente dagli studenti, permette di isolare sottoproblemi adatti per una lezione interattiva: rappresentazione dei dati come matrice utente-prodotto, misure di similarità tra utenti, algoritmi per predire i dati mancanti. Alla fine dell'attività in classe di due ore si giunge alla definizione di un algoritmo condiviso con la classe che viene testato a partire dai dati raccolti con questionari di gradimento sui laboratori svolti durante LabOrinto per prevedere i gusti dei partecipanti.

Un secondo tema riguarda il problema del commesso viaggiatore. La lezione viene impostata in maniera da arrivare, lavorando in gruppi, ad una possibile soluzione in pseudo codice per calcolare i cammini Hamiltoniani di grafi pesati. Nel 95% percento dei casi la soluzione costruita insieme agli studenti è basata su euristiche incomplete come *nearest neighbour*. Questo spunto può essere usato per ragionare attraverso esempi concreti sulla differenza tra problemi semplici e difficili e tra soluzioni esatte ed approssimate. Il problema del commesso viaggiatore ben si adatta al formato di LabOrinto che ne rappresenta un'istanza con vincoli reali (tempi, distanze tra le tappe) che stimola ulteriori discussioni in fase di restituzione delle esperienze fatte insieme.

2.3 Transfer di concetti

Per completare le attività in aula, abbiamo individuato alcuni temi in cui è possibile applicare o semplicemente definire collegamenti con i concetti su cui si è ragionato in aula. Un esempio di tema considerato riguarda l'analisi del linguaggio naturale e sui modi in cui i Large Language Model possono essere applicati in ambito didattico. La lezione prevede una sessione interattiva,

in cui gli studenti possono interagire con il tool CPT4CT [3]) che, tramite chiamate a ChatGPT, permette di generare decomposizioni di problemi non banali in sotto-task più semplici. Gli studenti sono invitati a riflettere sia sulla correttezza dei risultati ottenuti, sia su similitudine e differenze rispetto a decomposizioni ottenute senza l'uso del tool.

3 Discussione

Gli stage organizzati nel corso degli anni, da Slow Rogaining a LabOrinto, hanno creato una rete con scuole sul territorio che ha favorito la partecipazione di gruppi di studenti motivati e rispettosi degli ambienti di studio e lavoro in cui devono muoversi.

I principi del pensiero computazionale [5] si sono rivelati fondamentali nella progettazione delle singole attività. Ad esempio la decomposizione dei problemi in sottotask usata nelle lezioni frontali è un mezzo straordinario per arrivare ad una soluzione completa in maniera collaborativa. Questo metodo si è rivelato molto efficace anche in lezione di breve durata (tipicamente 2 ore). Inoltre favorisce attività come la generalizzazione delle soluzioni riformulando sotto task in diversi contesti applicativi (ad es. analisi delle frequenze usata nei sistemi di raccomandazione, crittografia [4], linguaggio naturale, ecc)

Riferimenti bibliografici

- [1] Pierluigi Crescenzi and Linda Pagli. *Problemi, algoritmi e coding. Le magie dell'informatica*. Zanichelli Editore Spa, 2017.
- [2] Giorgio Delzanno, Giovanna Guerrini, Viviana Mascardi, Luca Gelati, Vincenzo Petito, Francesca Vitali, Angelo Ferrando, Davide Ancona, Manuela Chessa, Nicoletta Noceti, and Francesca Odone. Slow rogaining: An innovative teamwork model for computer science education. In Tsvi Kuflik, Ilaria Torre, Robin Burke, and Cristina Gena, editors, *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2020, Genoa, Italy, July 12-18, 2020*, pages 119–126. ACM, 2020.
- [3] Davide Ponzini, Giovanni Adorni, Giorgio Delzanno, and Giovanna Gurerrini. Toward the use of generative AI to develop computational thinking by supporting problem decomposition. In *Ital-IA 2024 conference proceedings*, 2024.
- [4] Simon Singh. *Codici e segreti. La storia affascinante dei messaggi cifrati dall'antico Egitto a Internet*. Rizzoli, 2010.
- [5] Jeannette M. Wing. Computational thinking. *Commun. ACM*, 49(3):33–35, 2006.

“Laboratorio di Informatica” ai Licei Faes di Milano, Parte Seconda: il Livello dei Dati

Fabio Sartori¹, Elisabetta Zanichelli², Miriam Nobile²

¹Università degli Studi di Milano-Bicocca
fabio.sartori@unimib.it

²FAES Milano
{elisabetta.zanichelli, miriam.nobile}@faesmilano.it

Abstract

Questo articolo presenta lo sviluppo del progetto di “Laboratorio di Informatica” svolto come collaborazione tra l’Università degli Studi di Milano-Bicocca e i Licei Faes di Milano. Il progetto ha avuto come obiettivo l’introduzione alle Basi di Dati per gli studenti delle classi terze e quarte dei Licei che lo sceglievano come progetto di potenziamento didattico per il proprio curriculum di studi, seguendo la metodologia già sperimentata con successo l’anno prima durante il progetto pilota.

1 Introduzione

L’insegnamento dei fondamenti teorici e pratici alla base dello sviluppo delle basi di dati è un argomento che può risultare ostico agli studenti delle scuole superiori, specialmente dei licei. Da un lato, gli aspetti matematici alla base dei più noti modelli di riferimento, come l’algebra e il calcolo relazionale, non sono tipicamente noti agli studenti; dall’altro, i linguaggi di programmazione necessari all’implementazione di esempi, anche semplici, siano essi basati su database management systems (DBMS) SQL oriented o NOSQL oriented, non sono tipicamente vicini a quelli già noti agli studenti da precedenti esperienze scolastiche (come ad esempio Scratch).

Proseguendo l’esperienza avviata lo scorso anno [1], il progetto “Laboratorio di Informatica” si è concentrato sull’insegnamento di metodi per la concettualizzazione di una storia e la sua

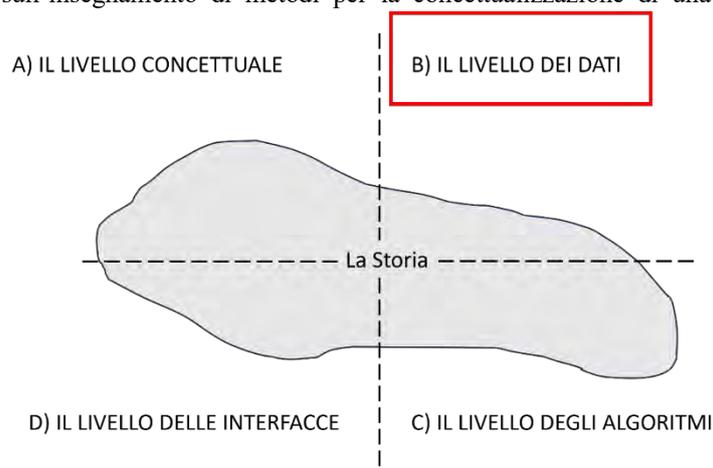


Figura 1: Il livello dei dati, ovvero come rappresentare opportunamente in maniera formale, attraverso linguaggi e strumenti opportuni, come DBMS sia SQL che NoSQL, le informazioni necessarie al funzionamento di un sistema reale.

implementazione in un sistema reale e funzionante (si veda la Figura 1), che permettesse agli studenti partecipanti di riflettere su concetti alla base della teoria e pratica delle basi di dati, come entità, attributo, relazione, chiave primaria, join, tralasciando aspetti troppo difficili da comprendere al loro livello (come la normalizzazione), ma spingendosi anche a tentare un confronto fra approcci diversi (per esempio, tramite la sperimentazione di MongoDB in contrasto a SQL).

2 Metodologia: Storytelling e Learning by Doing per l'apprendimento dell'Informatica

Come lo scorso anno, il progetto è stato pensato come una collezione di unità didattiche in cui gli studenti fossero invitati a riflettere in maniera pratica su come modellare un problema concreto e familiare in maniera comprensibile a un calcolatore, con tecniche di narrazione tipiche dello storytelling. Partendo dal *biglietto di un concerto*, mostrato in Figura 2, si è potuto iniziare a riflettere su come modellare informazioni complesse ed eterogenee, astraendole via in modo che siano facilmente rappresentabili e gestibili da un calcolatore (concetto di dato, relazione tra dati...)



Figura 2: il punto di partenza del progetto. Da qui si è riflettuto su concetti come Informazione, Dato, QR Code, Relazione fra Dati, Chiave nei DBMS.

I 22 ragazzi aderenti al progetto, di terza e quarta superiore, provenienti da liceo scientifico, classico e delle scienze umane sono stati lasciati liberi di aggregarsi in gruppi secondo le loro esigenze e affinità, come già sperimentato lo scorso anno, seguendo la metodologia delle comunità di pratica [2] (CoP), in modo da massimizzare l'impatto positivo degli allievi più ricettivi su quelli più in difficoltà. Il docente ha avuto il ruolo di facilitatore, impostando le lezioni alla spiegazione dei concetti in termini che risultassero chiari dal punto di vista delle conoscenze: da questo punto di vista, la parte più problematica è stata sicuramente la spiegazione dell'operazione di join tra relazioni nel modello ER, esemplificata tramite la teoria degli insiemi.

3 Risultati e Conclusioni

La fase di valutazione è stata pensata come un progetto a gruppi in cui ogni CoP ha analizzato un aspetto correlato al caso del concerto, sorteggiato a sorte: ciascun progetto poteva riguardare la gestione del blog di un cantante, il sistema di vendita dei biglietti, il controllo degli accessi allo stadio il giorno del concerto, la gestione della scaletta delle canzoni del cantante.

Come detto in precedenza, gli studenti hanno formato in maniera autonoma i gruppi (tranne un caso, in cui una studentessa rimasta spaiaata è stata inserita dal docente in un gruppo precedentemente formato). Ogni gruppo ha quindi deciso come sviluppare il progetto, a partire dalla traccia fornita. Alcuni hanno scelto di procedere come “da manuale”, riproducendo fedelmente i passi di progettazione visti durante il corso (modellazione concettuale, modello ER, query SQL). Altri hanno scelto di cominciare “dalla fine”, prediligendo l’aspetto implementativo rispetto a quello modellistico, trovandosi in difficoltà al momento di scrivere le query, ma riuscendo comunque a trovare strade alternative per risolvere il compito assegnato. Un gruppo è riuscito ad approfondire particolarmente dal punto di vista modellistico il dominio, estendendo la traccia proposta dal docente con nuovi attributi e nuove entità, in modo da arricchirlo ulteriormente dal punto di vista semantico. Interessanti le strategie di risoluzione adottate da altri due gruppi: uno di essi si è trovato in grande difficoltà all’inizio, dovuta all’assenza di quello che può essere identificato come il leader. Rientrato questi in gruppo la volta dopo, il recupero è stato immediato e tutti i compagni hanno beneficiato impegnandosi attivamente nella riuscita del progetto; il secondo ha adottato fin da subito una sorta di “strategia a rotazione” in cui ciascun membro periodicamente sostituiva l’altro in senso orario nell’attività che stava facendo (scrivere le query al computer, controllare gli errori, sviluppare il modello concettuale...). In questa maniera, ciascun membro ha svolto esattamente le stesse “funzioni” all’interno del gruppo, caratterizzandosi come parte attiva per la buona riuscita del progetto. In Figura 3 è riportato un esempio di progetto svolto da uno dei gruppi.

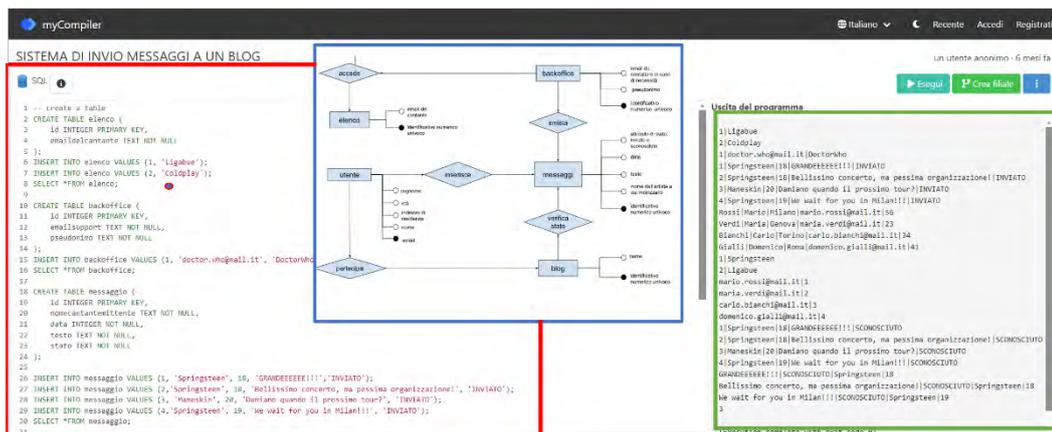


Figura 3: Un esempio di progetto sviluppato da una CoP di studenti: in blu il modello ER, in rosso un estratto del codice SQL, in verde l’output generato dalle query.

I risultati ottenuti sono stati incoraggianti: tutti i gruppi hanno dimostrato impegno e dedizione nello svolgimento del progetto assegnato, riuscendo a sperimentare con successo quanto visto a lezione in un contesto nuovo. Pur essendo il progetto didattico proposto nuovo e piuttosto complicato, il gruppo classe si è dimostrato capace di apprendere i concetti trasmessi e replicarli in maniera originale nell’attività progettuale, riuscendo a produrre dei programmi funzionanti capaci di rappresentare in maniera coerente un dominio reale dal punto di vista della rappresentazione dei dati.

Bibliografia

- [1] F. Sartori, E. Zanichelli e M. Nobile, «Storytelling e Learning by Doing nelle discipline STEM: il caso del "Laboratorio di Informatica" ai Licei Faes di Milano», in *ITADINFO*, Bari, 2023.
- [2] D. Brown, «Knowledge and organization: A social-practice perspective», *Organization science*, 2001.

Due progetti PNRR di approfondimento informatico: VideoGame Academy e Cybersecurity

Salvatore Imperato¹, Silvia Stranieri¹, Vincenzo Schiano Di Cola¹, Giulia Sellitto², Filomena Ferrucci²

¹ Istituto Tecnico Tecnologico “Giordani-Striano”, Napoli, Italia
salvatore.imparato12@scuola.istruzione.it, silvia.stranieri@unina.it,
vincenzo.schianodicola1@scuola.istruzione.it

² Dipartimento di Informatica, Università di Salerno, Salerno, Italia
gisellitto@unisa.it, fferrucci@unisa.it

Sommario

Questo articolo presenta due progetti svolti presso l’Istituto Tecnico Tecnologico Giordani-Striano di Napoli: *VideoGame Academy* e *Hackers: White or Black Hat? Un percorso sulla Cybersecurity*. Questo lavoro ha l’obiettivo di condividere le esperienze didattiche sperimentate, evidenziandone il contesto di riferimento, le metodologie adottate e i contenuti trattati, nonché i risultati ottenuti ed alcune osservazioni conclusive.

1 Introduzione

Lo scopo di questo lavoro è condividere due esperienze significative nell’ambito della didattica dell’informatica. Il contesto di riferimento è l’Istituto Tecnico-Tecnologico Giordani-Striano di Napoli, che coinvolge studenti di diversi quartieri, in particolare Bagnoli-Fuorigrotta, Pianura-Soccavo e Arenella-Vomero. Data la vastità del territorio e l’eterogeneità dei bisogni formativi dell’utenza, l’Istituto propone un’ampia offerta formativa, caratterizzata da un biennio unico e 4 indirizzi di studio per il triennio:

- Informatica
- Chimica, Materiali e Biotecnologia
- Elettronica ed Elettrotecnica
- Meccanica, Meccatronica ed Energia.

Le esperienze didattiche che saranno descritte in questo articolo sono 2 progetti finanziati dal PNRR nell’ambito degli interventi contro la dispersione scolastica (progetto *Ritroviamoci Dopo Scuola*). Le iniziative dovevano rispettare i seguenti vincoli:

1. Le classi dovevano essere composte da almeno 20 studenti (vincolo poi rilassato);
2. Agli studenti era richiesta una frequenza pari almeno al 75% delle ore totali;
3. A ciascuno studente era consentito di partecipare al più ad uno tra i circa 20 progetti proposti dall’intero Istituto.

In origine, i progetti proposti erano 4: *VideoGame Academy*, *Hackers: White or Black Hat? - Un percorso sulla Cybersecurity* (nel seguito abbreviato con *Cybersecurity*), *Linux Academy* e *Python Academy*. Per comprendere gli interessi degli studenti rispetto alle iniziative proposte,

è stata realizzata una fase esplorativa che ha coinvolto l'intero istituto. I 4 progetti sono stati illustrati in ogni classe con una presentazione di 10 minuti avvalendosi di brochure digitali (Figura 1). Agli studenti è stato quindi richiesto di esprimere una sola preferenza, compilando un Google form, in cui hanno specificato anche classe, sezione e indirizzo. Queste informazioni hanno consentito di fare una proiezione sulle possibili formazioni delle classi. Sono state compilate 73 manifestazioni di interesse, così suddivise:

- *VideoGame Academy*: 38 preferenze;
- *Cybersecurity*: 28 preferenze;
- *Linux Academy*: 1 preferenze;
- *Python Academy*: 6 preferenze.

Sono state quindi formate due classi per il corso *VideoGame Academy* e una classe per il corso *Cybersecurity*.

I due corsi sono stati sviluppati sulla base dei principi della Ricerca-Azione [4, 7], ponendoci di partecipare alla ricerca pedagogica e studiare le pratiche educative nell'insegnamento dell'informatica, suggerendo strategie migliorative per aumentare il coinvolgimento degli studenti e arrivando a un lavoro di indagine e partecipazione sul campo.

2 Videogame Academy

La Video Game Academy mira a creare un percorso formativo per gli studenti che vogliono entrare nel mondo dello sviluppo dei videogiochi. Il progetto prevede l'utilizzo di strumenti *open source*, in particolare il *game engine Godot* nella versione 4.0, e tecniche di sviluppo AGILE per rendere l'apprendimento della programmazione più coinvolgente e creativo.

2.1 Contesto

Il progetto *Videogame Academy* ha avuto una durata complessiva di 20 ore, distribuite durante il II quadrimestre su 10 lezioni da 2 ore, in orario extra-curricolare, da svolgersi interamente presso i laboratori informatici dell'istituto.

Il corso, visto il numero di richieste superiori a 30, è stato organizzato in due edizioni:

1. La prima edizione ha coinvolto 15 studenti, di cui il 30% iscritti al quinto anno dell'indirizzo "Informatica", il 60% al quarto anno dello stesso indirizzo, e il restante 10% distribuiti tra terzo anno ("Informatica") e biennio. Gli studenti presentavano un livello di competenze di base medio-alto, con conoscenze di base sufficientemente solide sulla programmazione ad oggetti.
2. La seconda edizione ha avuto una platea più eterogenea, e ha coinvolto 25 studenti di cui il 60% iscritti al triennio ("Informatica") e il 40% iscritti al biennio. Gli studenti hanno mostrato fin da subito una carenza delle competenze di base, nello specifico l'80% non aveva ancora affrontato la programmazione ad oggetti, né a livello curricolare né autonomamente.

Per entrambe le edizioni non sono stati previsti particolari requisiti di ingresso, ma il livello di partecipazione è stato sensibilmente diverso, data la composizione delle classi. Infatti, nella classe della prima edizione, almeno il 50% aveva già sperimentato diversi tool di *videogaming* per passione personale, mentre nella seconda classe questo si è verificato solo per due studenti.

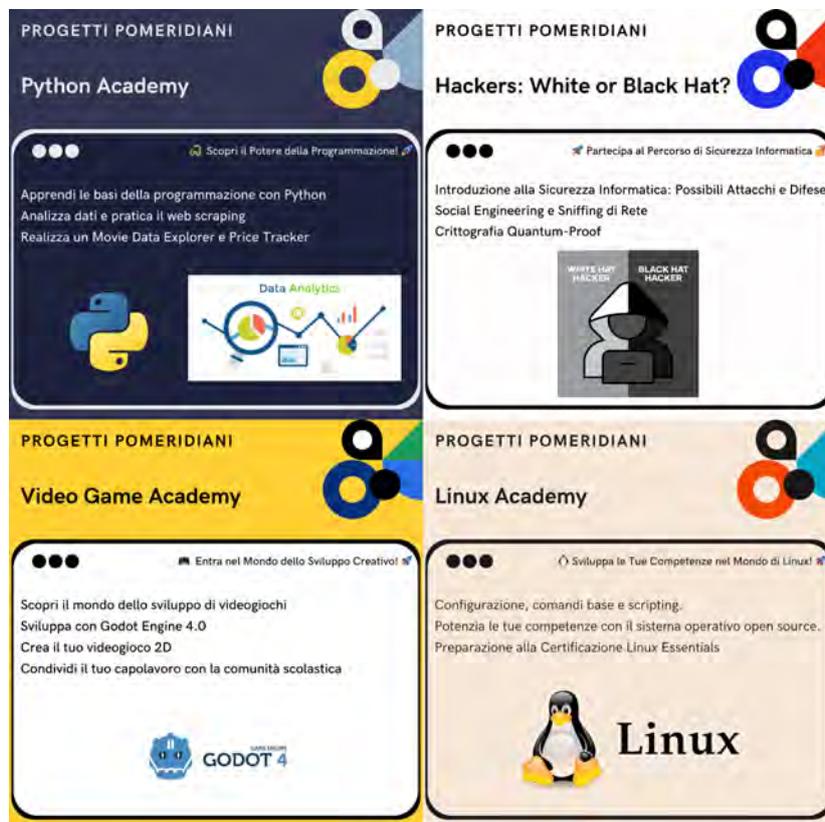


Figura 1: Locandine dei progetti proposti: *VideoGame Academy*, *Hackers: White or Black Hat?*-Un percorso sulla *Cybersecurity*, *Linux Academy* e *Python Academy*.

2.2 Contenuti e Metodologie

Il corso si è basato sull'utilizzo dell'*engine open source Godot 4.0* [2], che ha permesso agli studenti di acquisire le competenze di programmazione in *GDScript*, un linguaggio simile a *Python*, con l'approccio di programmazione orientata agli oggetti per utilizzare tale motore di gioco.

Il corso è stato progettato prevedendo due fasi distinte:

- Nella prima fase, basata su **apprendimento imitativo**, gli studenti dovevano sviluppare un videogioco *top-down* 2D contestualmente alla dimostrazione del docente. Poter da subito sviluppare un videogioco doveva risultare motivante per gli studenti, che potevano vedere immediatamente i risultati tangibili del proprio lavoro. Per questa fase, sono state prodotte delle slide sulla base di materiale *open source*¹ e sono state fornite agli studenti tutte le risorse grafiche e audio per lo sviluppo.
- Nella seconda fase, basata su **Learning by Doing** [13] e **Cooperative Learning** [10], gli studenti dovevano progettare e sviluppare un videogioco interamente in autonomia seguendo una metodologia di sviluppo AGILE, opportunamente introdotta dai docenti.

¹<https://www.youtube.com/c/CrystalBit>

In questa fase, il ruolo dei docenti è di supporto e di guida, in linea con una metodologia di **didattica costruttivista**.

Con la classe della prima edizione è stato possibile attuare interamente la progettazione prevista, con una prima fase della durata di circa 4 lezioni. Al contrario, per la classe della seconda edizione, viste le carenze di competenze di base, è stata necessaria una rimodulazione della programmazione. Infatti, la prima fase si è protratta per circa 7 lezioni. Inoltre, non è stato possibile condurre la seconda fase come ideato: constatata la poca autonomia degli studenti, si è deciso di utilizzare codici *open source* di videogame esistenti, che gli studenti si sono cimentati ad analizzare e modificare.

2.3 Risultati e Osservazioni

La forte affluenza al corso è stata motivata principalmente dalla tematica, che riscuote un forte interesse tra molti studenti nella fascia di età tipica di un istituto superiore. A nostro avviso, è importante sfruttare questa occasione per veicolare conoscenze e competenze di sviluppo informatico in un *setting* didattico orientato alla creazione di qualcosa di ludico e tangibile. Risulta altamente motivante per gli studenti passare da un ruolo di utilizzatori di tecnologia (videogiochi) alla loro creazione.

Durante lo svolgimento delle attività della prima edizione, quasi tutti i gruppi hanno preferito creare una modalità di gioco di tipo *sidescrolling* o misto, diversamente da quanto illustrato a lezione. Inoltre, hanno mostrato una tendenza a partire dagli aspetti più vicini all'utente finale, in contrasto con le indicazioni loro fornite. Questo, a nostro avviso, evidenzia ancora una difficoltà di astrazione, che rende difficile separare la visione *modularizzata* del progetto da quella d'insieme, portandoli a condurre la progettazione immaginandosi mentre giocano.

Di fatto, chi si iscrive a una *Gaming Academy* risulta particolarmente interessato allo *storytelling* del gioco, in quanto si identifica nell'utente finale come giocatore e si cala negli scenari di gioco. In ogni caso, su 5 team della prima edizione, 4 hanno prodotto una demo funzionante e giocabile, come mostrato in Figura 2. Inoltre, molti hanno sperimentato diversi strumenti *open source* utili allo sviluppo, come la creazione di avatar personalizzati, generatori di effetti sonori, ecc. Per la seconda edizione, viste le difficoltà evidenziate, non è stato possibile realizzare uno sviluppo *from scratch*, ma un approccio di tipo **WebQuest** [16] ha comunque consentito di sperimentare concretamente l'attività di analisi e modifica di un gioco esistente.

L'esperienza condotta ci ha fatto concludere che, a seconda del livello di base della classe, sarebbe opportuno utilizzare approcci diversi: in particolare, per chi non possiede conoscenze di base di programmazione ad oggetti, probabilmente non è consigliabile utilizzare una piattaforma come *Godot*, ma potrebbe essere utile orientarsi piuttosto verso piattaforme come *Roblox* [1] e *microStudio*.²

3 Hackers: White or Black Hat?-Un percorso sulla Cybersecurity

Hackers: White or Black Hat? mira sia ad esplorare i concetti di sicurezza informatica e di *ethical hacking* in modo guidato, sia a simulare attacchi specifici, servendosi di opportuni ambienti protetti.

²<https://microstudio.dev/>



Figura 2: Screen di uno dei giochi demo sviluppato dagli studenti con Godot Engine.

3.1 Contesto

Il progetto *Hackers: White or Black Hat?* ha avuto una durata complessiva di 20 ore, distribuite durante il II quadrimestre su 10 lezioni da 2 ore, in orario extra-curricolare, da svolgersi interamente presso i laboratori informatici dell'istituto. Il corso ha coinvolto circa 25 ragazzi, di cui il 70% iscritti al triennio di indirizzo "Informatica", con competenze di programmazione di livello base/intermedio, mentre il restante 30% ha interessato studenti iscritti al primo biennio. Il corso è stato progettato senza assumere particolari requisiti di ingresso, infatti la classe si è dimostrata eterogenea dal punto di vista delle competenze di base, mentre il livello di partecipazione è stato medio-alto per la maggioranza dei partecipanti.

3.2 Contenuti e Metodologie

Per stimolare il senso di responsabilità degli studenti, è stato chiesto loro di firmare un documento che attestava l'intento di utilizzare le conoscenze e le competenze sulla sicurezza informatica, che il corso gli avrebbe fornito, in modo responsabile.

Dopo aver fornito una panoramica sulla storia della sicurezza informatica, i ragazzi sono stati coinvolti in un'attività **didattica meta-cognitiva** [6], ponendo loro due quesiti su cui riflettere [11]:

- Esistono sistemi informatici al mondo che sono al sicuro dagli attacchi hacker?
- Come sarebbe un sistema totalmente sicuro?

Dopo la fase di riflessione, gli studenti hanno esposto le loro considerazioni, seguendo il setting di una **lezione partecipata** [15].

Successivamente, seguendo una metodologia di **gamification** [3], gli studenti sono stati divisi in due gruppi: la squadra rossa (gli attaccanti), la squadra blu (i difensori). Alla squadra blu è stato richiesto di elaborare una strategia per la protezione di un *segreto* (una breve frase dalla letteratura italiana stampata su un foglio di carta), a quella rossa una strategia per scoprirlo. L'elaborazione della strategia è stata condotta attraverso **cooperative learning** [10], stimolando le capacità di comunicazione e negoziazione tra gli appartenenti allo stesso gruppo.

Nelle successive lezioni, sono state analizzate le tecniche di autenticazione e di creazione di una password, stimolando gli studenti, attraverso un **apprendimento per scoperta** [5], ad individuare possibili vulnerabilità e punti di forza nel processo di determinazione di una password robusta. Questo ha consentito di introdurre il concetto di *hash function*. Inoltre gli studenti hanno avuto la possibilità di osservare concretamente il processo di intercettazione di una password sulla piattaforma *cyberskillslive*.³

È stato quindi realizzato il setting dell'ambiente di attacco: gli studenti hanno installato volutamente una versione leggera di Ubuntu, Lubuntu, piuttosto che uno strumento già pronto all'uso come Kali Linux, per poter partecipare attivamente all'installazione successiva dei tool necessari per la messa in atto degli attacchi, sposando pienamente la metodologia **Learning by Doing** (LbD) [13]. Nella seconda fase del corso, gli studenti hanno sperimentato la metodologie **Capture the Flag** [8, 12], attraverso le piattaforme *tryhackme* [14] e *OliCyber*,⁴ e si sono cimentati nelle sfide dell'ambito: Network security (*Wireshark*), Web Security (*SQL-injection* [9]) e le basi della Criptography. Infine, gli studenti hanno implementato nella rete locale sui propri PC attacchi di tipo *ARP spoofing* e *MiM* (Man-in-the-middle). Queste ultime attività sono state portate avanti in modalità-tipo **WebQuest** [16]: infatti, gli studenti hanno seguito le indicazioni di una guida online e hanno cercato di emulare quanto indicato, estendendo con ulteriori ricerche autonome le conoscenze acquisite. Gli studenti hanno lavorato in gruppi di due elementi, ciascun gruppo avanzando alla propria velocità.

3.3 Risultati e Osservazioni

Al termine del corso, sono stati raccolti i feedback diretti degli studenti, in relazione all'attività di Cybersecurity: diversi studenti, in particolare gli iscritti al quarto anno dell'indirizzo informatico, hanno apprezzato l'utilità dei concetti trattati, evidenziandone in particolare l'impatto positivo anche sulle attività curricolari. Inoltre, molti studenti hanno valutato positivamente il chiarimento di alcuni aspetti sulla sicurezza informatica, in particolare, riguardo le VPN, evidenziando una disinformazione sul tema, dovuta ad una pubblicità insistente e spesso fuorviante. Alcuni studenti, invece, soprattutto quelli del biennio, hanno manifestato difficoltà nelle attività riguardanti gli attacchi avanzati, come *SQL-injection*, *ARP spoofing* e *MiM*. Al contrario, alcune eccellenze del biennio hanno superato brillantemente questi ostacoli colmando le proprie lacune, in totale autonomia anche al di fuori del corso. Per questi motivi, continuiamo a ritenere non necessaria l'introduzione di prerequisiti di ingresso, ma riteniamo che sia auspicabile la partecipazione di studenti con una giusta dose di motivazione, che possa compensare l'eventuale carenza di conoscenze di base.

4 Conclusioni

Questo lavoro presenta due progetti didattici di informatica svolti presso l'Istituto Giordani-Striano di Napoli: *VideoGame Academy* e *Hackers: White or Black Hat?-Un percorso sulla Cybersecurity*. Dopo aver illustrato i contenuti e le metodologie dei corsi, sono stati analizzati i risultati ottenuti, evidenziando miglioramenti possibili.

Le tematiche vicine agli studenti e i titoli accattivanti sono stati i punti di forza per *VideoGame Academy* e *Cybersecurity*, rispetto a *Python Academy* e *Linux Academy*. Tuttavia, la partecipazione è stata limitata agli informatici, indicando la necessità di attrarre anche altri indirizzi dell'istituto per promuovere l'alfabetizzazione digitale. È opportuno analizzare i

³<https://cyberskillslive.com/>

⁴<https://olicyber.it/>

requisiti di ingresso per adattare i corsi al gruppo classe. Inoltre, è importante strutturare formalmente l'acquisizione dei feedback degli studenti per valutare il gradimento e l'acquisizione delle competenze. In conclusione, visto il riscontro positivo, condividiamo questa esperienza per una possibile replica in altri istituti. Il materiale didattico è disponibile al link bit.ly/4cCXlph.

Ringraziamenti

Un sentito ringraziamento alla Dirigente Scolastica, Dott.ssa Elena De Gregorio, per il suo continuo supporto e la sua preziosa guida, che hanno reso possibile la realizzazione di questa esperienza sulla didattica.

Riferimenti bibliografici

- [1] Khaled Alhasan, Khawla Alhasan, et al. Roblox in higher education: Opportunities, challenges, and future directions for multimedia learning. *International Journal of Emerging Technologies in Learning (iJET)*, 18(19):32–46, 2023.
- [2] Chris Bradfield. *Godot Engine Game Development projects: Build five cross-platform 2D and 3D games with Godot 3.0*. Packt Publishing Ltd, 2018.
- [3] Ilaria Caponetto, Jeffrey Earp, and Michela Ott. Gamification and education: A literature review. In *European conference on games based learning*, volume 1, page 50. Academic Conferences International Limited, 2014.
- [4] Bart Cunningham. Action research: toward a procedural model. *Human relations*, 29(3):215–238, 1976.
- [5] David Hammer. Discovery learning and discovery teaching. *Cognition and instruction*, 15(4):485–529, 1997.
- [6] Alice Y Kolb and David A Kolb. The learning way: Meta-cognitive aspects of experiential learning. *Simulation & gaming*, 40(3):297–327, 2009.
- [7] Kurt Lewin. Action research and minority problems. *Journal of social issues*, 2(4):34–46, 1946.
- [8] Flores Manuela and Barbara Masucci. Capture The Flag: un nuovo approccio all'apprendimento della Cybersecurity. In " *Atti del I convegno italiano sulla didattica dell'informatica (ITADINFO 2023)*", pages 1–4. Università degli Studi di Bari Aldo Moro, 2023.
- [9] Johannes Schildgen and Jessica Rosin. Game-based Learning of SQL Injections. In *Proceedings of the 1st International Workshop on Data Systems Education, DataEd '22*, pages 22–25, New York, NY, USA, June 2022. Association for Computing Machinery.
- [10] Robert E Slavin. Cooperative learning. *Review of educational research*, 50(2):315–342, 1980.
- [11] Jennifer Joy Smith. *Educators' Perspectives on Cybersecurity Educational Resources*. PhD thesis, Carleton University, 2023.
- [12] Matthew Swann, Joseph Rose, Gueltoum Bendiab, Stavros Shiaeles, and Fudong Li. Open Source and Commercial Capture The Flag Cyber Security Learning Platforms - A Case Study. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 198–205, July 2021.
- [13] Peter Thompson. Learning by doing. *Handbook of the Economics of Innovation*, 1:429–476, 2010.
- [14] Rebeka Toth and László Erdődi. Expanding horizons: The evolving landscape of development opportunities in cybersecurity training platforms. In *Norsk IKT-konferanse for forskning og utdanning*, number 3, 2023.
- [15] Ronald R Yager. A model of participatory learning. *IEEE transactions on systems, man, and cybernetics*, 20(5):1229–1234, 1990.
- [16] Robert Zheng, Bradd Stucky, Matt McAlack, Mike Menchana, and Sue Stoddart. Webquest learning as perceived by higher-education learners. *TechTrends*, 49(4):41–49, 2005.

Come nasce un videogioco: aspetti di progettazione multimediale audio-video ed interfacce

Adriano Mancini^{1,*}, Leonardo Gabrielli^{1,†}, Maura Mengoni^{2,‡}, Gionata Massi^{3,§}, and Federico Robuffo^{4,¶}

¹ Dipartimento di Ingegneria dell'Informazione
Università Politecnica delle Marche Ancona, Italia

`a.mancini@staff.univpm.it`, `l.gabrielli@staff.univpm.it`

² Dipartimento di Ingegneria Industriale e Scienze Matematiche
Università Politecnica delle Marche Ancona, Italia

`m.mengoni@staff.univpm.it`

³ Istituto Istruzione Superiore Savoia Benincasa
Ancona, Italia

`gionata.massi@savoiabencasa.it`

⁴ Istituto Istruzione Superiore Cambi Serrani
Falconara M.ma, AN, Italia

`federicorobuffo@cambiserrani.it`

Abstract

Il corso "Come nasce un videogioco: aspetti di progettazione multimediale audio-video ed interfacce" ha offerto agli studenti delle scuole secondarie di secondo grado una panoramica completa sullo sviluppo di videogiochi. In 15 ore, il corso ha introdotto la progettazione di interfacce utente (UI) e dell'esperienza utente (UX), l'uso di `Inkscape` per la grafica vettoriale, l'acquisizione e il post-processamento dell'audio, e la programmazione di un gioco in Python con il framework `Pygame`. Gli studenti e le studentesse hanno realizzato un gioco simile ad Atari Breakout, comprendendo l'importanza di acquisire competenze tecniche e trasversali, di sviluppare abilità di problem-solving in un'ottica di collaborazione e creatività.

1 Introduzione

Negli ultimi anni, la progettazione di videogiochi è diventata una delle aree più affascinanti e dinamiche del panorama tecnologico. È noto che il mondo dei (video)giochi può alimentare il desiderio di una persona nell'impegnarsi in attività spesso considerate di ridotto interesse [3, 8]. Questo aspetto porta a dover introdurre aspetti anche complessi nello sviluppo di applicazioni multimediali [7]. A tal proposito linguaggi di programmazione come Python consentono di ridurre la complessità nelle fasi di sviluppo di applicazioni [1] grazie anche alla ridotta complessità nell'apprendere concetti della programmazione [4] che spaziano dalle istruzioni condizionali alle strutture dati. Programmare videogame in Python oggi è molto semplice grazie alla disponibilità di diverse librerie e framework che consentono di ridurre anche in questo

*Parte programmazione in python del corso

†Parte audio del corso

‡Parte UI/UX del corso

§docente presso istituto di istruzione superiore: attività di coordinamento e supporto agli studenti/studentesse

¶docente presso istituto di istruzione superiore Liceo Scientifico - Videogame e Realtà Virtuale: attività di coordinamento e supporto agli studenti/studentesse

caso la complessità tipica di questo "mondo" [6, 5]. L'idea (vincente) di insegnare la programmazione attraverso la creazione dei giochi non è nuova ed è stata ampiamente sviluppata negli anni '80, quando le cassette con i videogiochi per gli *home computer* dell'epoca erano accompagnate da riviste che introducevano alla programmazione. In questo articolo si vuole presentare l'esperienza sul campo sviluppata nel contesto del DM 934 del 3 agosto 2022 incentrata sull'introduzione del mondo dei videogame sotto differenti punti di vista che spaziano dalle interfacce al mondo dell'audio e del video.

2 Il progetto

Nell'ambito di un mini-corso intensivo di 15 ore intitolato "Come nasce un videogioco: aspetti di progettazione multimediale audio-video ed interfacce", gli studenti e le studentesse delle scuole secondarie di secondo grado hanno avuto l'opportunità di immergersi nel mondo dello sviluppo di videogiochi. Tale mini-corso è fortemente collegato con le tematiche del corso di laurea triennale recentemente istituito presso l'Università Politecnica delle Marche dal titolo "Ingegneria dell'Informazione per Videogame e Realtà Virtuale". Il mini-corso di 15 ore si colloca all'interno del quadro normativo del DM 934 del 3 agosto 2022, con specifico riferimento ai Corsi di Orientamento (art. 3) [2].

Il mini-corso, strutturato in tre fasi principali, ha offerto una panoramica delle competenze necessarie per creare un videogioco, dalla progettazione dell'interfaccia utente (UI) e dell'esperienza utente (UX), fino agli aspetti di sound design ed elaborazione dell'audio e alla programmazione in Python utilizzando il framework `Pygame`.

2.1 Modulo 1 - Introduzione alla UI e UX con cenni di game design

La prima fase del mini corso si è concentrata sulla UI e UX, elementi fondamentali per il successo di qualsiasi videogioco. Gli studenti hanno appreso come progettare interfacce utente intuitive e coinvolgenti, studiando i principi fondamentali del design delle interfacce e della *user experience*.

Durante questa fase, sono stati introdotti i concetti base del game design, enfatizzando l'importanza di creare esperienze di gioco fluide e gratificanti. Gli studenti hanno scoperto come le decisioni di design possono influenzare l'interazione del giocatore con il gioco e come un buon design possa migliorare l'immersività e l'*engagement* del gioco stesso.

Una parte significativa di questa fase è stata dedicata all'apprendimento base di `Inkscape`, un potente strumento di grafica vettoriale open-source. Gli studenti hanno acquisito competenze pratiche nella creazione di grafica vettoriale, essenziale per la progettazione delle interfacce di gioco e degli elementi visivi. Hanno imparato a disegnare e manipolare forme semplici, creare icone e progettare *layout* di interfacce.

2.2 Modulo 2 - Audio e post-processamento

La seconda fase del corso ha portato gli studenti nel mondo dell'audio per videogiochi, un elemento spesso sottovalutato ma cruciale per l'immersività del gioco. Gli studenti hanno sviluppato una fruizione più consapevole del sonoro nell'industria del cinema e nel videogioco attraverso esperienze di ascolto critico, guidate e discusse in classe, in modo da discriminare elementi musicali ed effetti speciali e riconoscerne l'uso da un punto di vista emotivo o come vettore di informazione e contesto.

Maturate alcune semplici considerazioni, agli studenti è stato chiesto di sonorizzare il *game-play* di un videogioco. Onde evitare di dover affrontare nelle poche ore del corso concetti legati all'elaborazione numerica di segnali o tecniche di editing audio con software professionali, si è preferito lavorare in maniera snella ed efficace.

Gli studenti hanno selezionato un breve clip video tratto da un videogioco, al quale è stato tolto l'audio originale. Per stimolare la creatività e la socializzazione, gli studenti sono poi stati divisi in gruppi, ciascuno dei quali ha effettuato un *brainstorming* per stabilire che genere di suoni creare a partire da semplici oggetti quotidiani o attraverso la vocalizzazione e l'uso del corpo.

Durante questa fase, agli studenti sono state fornite alcune indicazioni fondamentali sulla registrazione in digitale dei suoni precedentemente stabiliti utilizzando semplici registratori portatili e cuffie circumaurali. Il docente ha poi raccolto il materiale ed ha effettuato con la classe un ascolto attento di quello che ciascun gruppo ha prodotto e si è fatto guidare dagli studenti nell'editing creativo dei suoni registrati e nell'inserimento all'interno del clip del videogioco, fino a raggiungere il consenso della classe.

Gli studenti hanno così capito l'importanza dell'audio nel contesto di un videogioco e di come esso apporti un fondamentale contributo nell'esperienza di gioco. Hanno compreso le possibilità fornite dall'elaborazione digitale del segnale e hanno compreso la complessità delle professionalità coinvolte in tale lavoro.

2.3 Modulo 3 - Programmazione di un gioco in Python con Pygame

La fase finale del corso è stata dedicata alla programmazione, utilizzando Python e il *framework* Pygame per creare un gioco simile ad Atari Breakout con un approccio ispirato al *Necessity Learning Design* [4]. Questa fase ha rappresentato il culmine delle conoscenze teoriche e pratiche acquisite nelle fasi precedenti.

Gli studenti hanno iniziato comprendendo le basi della programmazione grafica, partendo da primitive semplici come il disegno di una pallina. Da qui, hanno progredito verso la gestione delle interazioni più complesse, come i rimbalzi della pallina sulle superfici e le interazioni con altri oggetti nel gioco.

Durante lo sviluppo del gioco, gli studenti hanno dovuto affrontare e risolvere vari problemi di programmazione, migliorando le loro capacità di problem-solving e di pensiero logico. Hanno imparato a implementare la fisica di base nel gioco, gestire semplici collisioni (pallina-parete) e integrare gli effetti sonori contestuali sviluppati nella fase precedente.

I docenti hanno fornito esempi pratici e *snippet* di codice per facilitare lo sviluppo del gioco. Questi esempi hanno aiutato gli studenti a comprendere meglio i concetti di programmazione e a applicarli in modo efficace. Gli studenti hanno avuto l'opportunità di lavorare sia individualmente che in gruppo, migliorando le loro competenze collaborative e di comunicazione.

In Fig.1 vengono raffigurati le quattro fasi principali dello sviluppo con complessità crescente.

Le realizzazione del gioco ha visto quattro momenti base. Il primo obiettivo è stato di realizzare una semplice scena introducendo un elemento grafico (cerchio) da muovere con una velocità costante; è stato fatto variare anche il valore di *frame per second* per comprendere l'importanza di tale aspetto. Successivamente è stato introdotto un elemento grafico più complesso (immagine raster con sfondo trasparente) chiedendo di implementare logiche di rimbalzo con l'attivazione dell'audio contestualmente allo stesso. Successivamente si è posto l'accento sull'interazione dell'utente attraverso *arrow keys* spostando una piattaforma nella parte inferiore del gioco, iniziando così ad implementare la parte centrale del gioco. Come ultimo sviluppo è stato chiesto di inserire degli elementi rettangolari a griglia con colori e valori *random*; questa

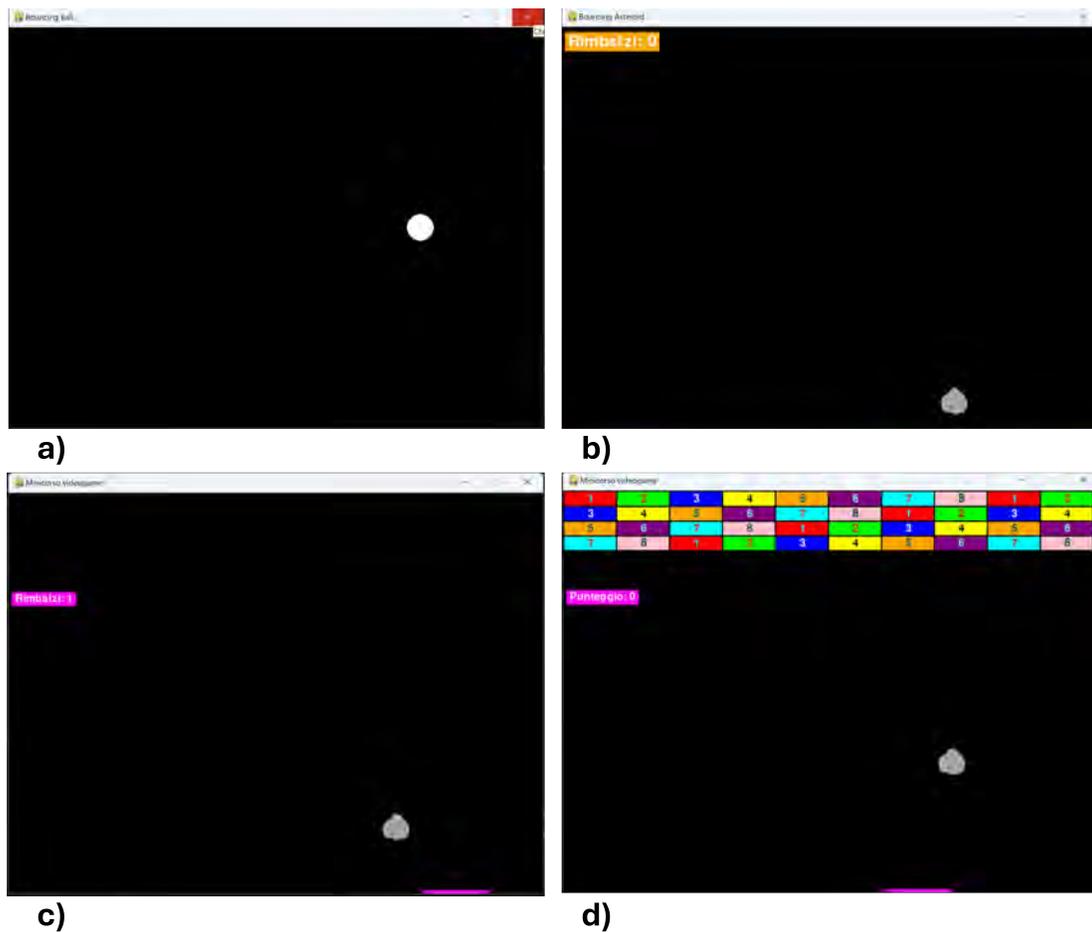


Figure 1: a) animazione di un elemento all'interno della scena. b) rimbalzo di un elemento grafico complesso all'interno di una scena con audio contestuale. c) gestione del rimbalzo su una piattaforma mobile mediante *arrow key*. d) logica completa di gioco con randomizzazione degli elementi (colore e punteggio).

parte è sicuramente quella più complessa ed è stata completata in generale solo da un sottoinsieme di studenti e studentesse con una solida conoscenza di un linguaggio di programmazione.

2.4 Competenze, Abilità e Conoscenze

Il mini-corso è stato progettato nell'ottica di promuovere il potenziamento di competenze tecniche, trasversali e teoriche nell'ambito del mondo dei videogame. Al termine del corso, gli studenti hanno acquisito una panoramica sulle competenze preziose e trasversali, applicabili non solo nella progettazione di videogiochi ma anche in altre aree della tecnologia e del design.

2.4.1 Competenze tecniche

- **Design grafico:** utilizzo di *Inkscape* per creare grafica vettoriale, progettazione di interfacce utente intuitive e visivamente accattivanti.
- **Acquisizione ed elaborazione numerica del suono:** tecniche di registrazione audio e possibilità creative fornite dai software utilizzati nel settore.
- **Programmazione:** conoscenze di base della programmazione in Python, utilizzo del framework *Pygame* per lo sviluppo di giochi, gestione delle collisioni e implementazione della fisica di base nel gioco.

2.4.2 Competenze trasversali

- **Problem-solving:** abilità nel risolvere problemi complessi durante lo sviluppo del gioco, applicazione del pensiero logico e critico.
- **Collaborazione:** capacità di lavorare in gruppo, comunicare efficacemente con i compagni di squadra e condividere idee e soluzioni.
- **Creatività:** sviluppo della creatività attraverso la progettazione di interfacce, la creazione di grafica e la realizzazione di effetti sonori a partire dall'uso del corpo e da oggetti quotidiani.

2.4.3 Conoscenze teoriche

- **Principi di game design:** comprensione delle basi del design di videogiochi e dell'importanza della UI e UX.
- **Fondamenti di audio per videogiochi:** conoscenza degli aspetti chiave dell'audio nel game design e orientamento rispetto alle esistenti tecniche di acquisizione e *post-processing*.
- **Fondamenti di programmazione di videogiochi:** conoscenza di concetti base come rendering ed animazione di una scena, *frame buffer* e *frame per second* anche in ambiente Linux.

3 Conclusioni

In conclusione, il corso "Come nasce un videogioco: aspetti di progettazione multimediale audio-video ed interfacce" ha offerto agli studenti e studentesse un'esperienza formativa che riteniamo essere stata completa e coinvolgente. Attraverso un approccio laboratoriale e interattivo, gli studenti e le studentesse hanno imparato a riconoscere quali sono le competenze tecniche necessarie per affrontare il mondo dei videogame dove più che mai è necessario confrontarsi con sfide multidisciplinari. Sicuramente i tempi previsti, ovvero 15h, si sono rivelati ridotti rispetto alle necessità di uno sviluppo completo ed in autonomia. Il corso è stato erogato a studenti e studentesse che spaziano dal primo al quinto anno; la conoscenza pregressa di un linguaggio di programmazione si è dimostrata di notevole utilità aumentando il grado di raggiungimento degli obiettivi iniziali; vale la pena rimarcare comunque che l'utilizzo di un linguaggio come Python abbia semplificato il processo di scrittura del codice.

Acknowledgements

Gli autori desiderano ringraziare anche la Prof.ssa Patrizia Melli (Istituto di Istruzione Superiore Cambi Serrani) e la Prof.ssa Federica Minni (Istituto Istruzione Superiore Savoia Benincasa) per il loro supporto nell'organizzare le attività di orientamento presso i rispettivi istituti. Si ringrazia inoltre il Prof. Domenico Ursino dell'Università Politecnica delle Marche per aver contribuito all'istituzione del corso di laurea triennale in Ingegneria dell'Informazione per Videogame e Realtà Virtuale.

References

- [1] Santiago Criollo-C, Mario González-Rodríguez, Andrea Guerrero-Arias, Luis F. Urquiza-Aguiar, and Sergio Luján-Mora. A review of emerging technologies and their acceptance in higher education. *Education Sciences*, 14(1), 2024.
- [2] Ministro dell'università e della ricerca. Decreto ministeriale n. 934 del 03-08-2022. <https://www.mur.gov.it/sites/default/files/2022-08/Decreto%20Ministeriale%20n.%20934%20del%2003-08-2022.pdf>, 2022.
- [3] Teemu H. Laine and Renny S. N. Lindberg. Designing engaging games for education: A systematic literature review on game motivators and design principles. *IEEE Transactions on Learning Technologies*, 13(4):804–821, 2020.
- [4] Sbaraglia Marco, Lodi Michael, and Martini Simone. Apprendimento della programmazione guidato dalla necessità: il necessity learning design. In *Convegno Italiano Sulla Didattica dell'Informatica (ITADINFO), 13-15 Ottobre 2023, Bari*, 2023.
- [5] Pygame. python library for game development. <https://www.pygame.org/docs/index.html>, 2024.
- [6] Al Sweigart. *Invent your own computer games with python*. No Starch Press, 2016.
- [7] Cristian Vidal-Silva, Nicolás A. Barriga, Franco Ortega-Cordero, Javiera González-López, Claudia Jiménez-Quintana, Claudia Pezoa-Fuentes, and Iván Veas-González. Developing computing competencies without restrictions. *IEEE Access*, 10:106568–106580, 2022.
- [8] Dan Zhao, Cristina Hava Muntean, Adriana E. Chis, Gregor Rozinaj, and Gabriel-Miro Muntean. Game-based learning: Enhancing student experience, knowledge gain, and usability in higher education programming courses. *IEEE Transactions on Education*, 65(4):502–513, 2022.

Insegnare Informatica con Unreal Engine: l'esperienza di Ragazze Digitali

Diletta Maniglia, Lisa Fregni, and Claudia Canali

Università degli Studi di Modena e Reggio Emilia, Modena, Italy
{`diletta.maniglia,lisa.fregni,claudia.canali`}@unimore.it

Abstract

In un mondo in cui le competenze digitali sono alla base di ogni settore, è cruciale incentivare la conoscenza in questo campo fin dai primi stadi dell'apprendimento. Tuttavia, l'iniziale apprendimento della programmazione attraverso linguaggi tradizionali può risultare noioso e difficile per gli studenti più giovani, portandoli spesso ad abbandonare questo percorso. L'apprendimento della programmazione tramite l'Unreal Engine offre un'esperienza molto più interattiva e coinvolgente per gli studenti. Con Unreal Engine, è possibile imparare a programmare utilizzando i Blueprint, che consentono agli studenti di comprendere concetti fondamentali come la logica di programmazione, i costrutti, la ricorsione in modo più intuitivo e divertente. Il progetto Ragazze Digitali è un summer camp innovativo con l'obiettivo di avvicinare all'informatica le studentesse delle classi terze e quarte delle scuole superiori. In questo progetto si è recentemente sperimentato l'insegnamento della programmazione tramite Blueprint di Unreal Engine. In questo contributo si esplora l'utilizzo di questo strumento e se ne discutono vantaggi e svantaggi.

1 Introduzione

Recenti statistiche confermano come le donne siano ancora fortemente sottorappresentate nelle professioni informatiche e nei relativi programmi universitari [3, 1, 9]. Il divario di genere, unitamente alla crescente domanda di competenze in questo settore, ha spinto istituzioni e scuole a realizzare numerose attività extracurricolari con l'obiettivo di avvicinare giovani studentesse all'informatica [4]. In questi contesti, la scelta del giusto strumento per avvicinare all'informatica è di importanza strategica, perchè l'iniziale apprendimento della programmazione attraverso linguaggi tradizionali può risultare difficile, risultando controproducente in termini di orientamento [8].

Lanciato nel 2014, il progetto Ragazze Digitali [6, 5] offre alle ragazze un'esperienza di apprendimento divertente e innovativa, mirata a rompere gli stereotipi di genere nell'informatica. Per l'edizione 2023 del progetto è stato scelto l'uso di Unreal Engine [7], un software molto potente per la creazione di videogiochi, noto per il suo motore grafico avanzato che offre grafica di alta qualità e rendering realistico [2]. Grazie a questo strumento, gli sviluppatori possono facilmente creare mondi virtuali e immersivi che catturano l'attenzione degli utenti. Una delle caratteristiche distintive di Unreal Engine è l'uso del linguaggio di programmazione C++, che consente di creare codice di gioco altamente performante. Tuttavia, ciò che rende Unreal Engine particolarmente accessibile è l'integrazione del sistema di scripting visuale chiamato Blueprint che consente di creare logiche di gioco e interazioni attraverso nodi e collegamenti grafici, senza la necessità di scrivere codice direttamente. Questo approccio è stato progettato per essere intuitivo e accessibile anche a chi non ha esperienza di programmazione: i principianti hanno la possibilità di imparare come creare e collegare nodi per definire interazioni, animazioni e logiche di gioco in modo visuale e intuitivo. Questo rende il processo di apprendimento della programmazione più accessibile e divertente, consentendo agli studenti di plasmare l'esperienza di gioco



Figure 1: Ragazze Digitali 2023

a proprio piacimento. L'efficacia di Unreal Engine e Blueprint come strumento didattico è stata valutata nel campo di Ragazze Digitali, dove è stato utilizzato con successo per insegnare alle partecipanti i fondamenti della programmazione. Di seguito, esploreremo i vantaggi e gli svantaggi dell'uso di Unreal Engine per imparare a programmare, basandoci sull'esperienza di Ragazze Digitali 2023.

2 Il progetto Ragazze Digitali

Nato nel 2014 a Modena da una collaborazione tra il Dipartimento di Ingegneria 'Enzo Ferrari' dell'Università di Modena e Reggio Emilia e l'associazione femminile EWMD (European Women Management Development), Ragazze Digitali è un progetto di orientamento che mira ad avvicinare le ragazze all'informatica attraverso un approccio creativo e divertente. Costantemente cresciuto nel tempo, il summer camp dal 2018 è stato replicato anche a Reggio Emilia e a Cesena, con la collaborazione dell'Università di Bologna. Dal 2022, Ragazze Digitali viene promosso dalla Regione Emilia-Romagna, in collaborazione con tutti gli Atenei presenti sul territorio regionale, e replicato nelle città della regione. L'esperienza dei summer camp è basata su un approccio learn-by-doing fondato sul lavoro di squadra e rivolto ad un apprendimento del coding contestualizzato ad ambiti applicativi creativi ed innovativi, come la programmazione di videogiochi o la programmazione di robot. Il progetto ha ricevuto importanti riconoscimenti, tra i quali vanno menzionati la citazione nel report She Figures 2021 della Commissione Europea quale iniziativa italiana di successo per il contrasto del digital gender gap e l'inserimento tra i casi di studio dell'OPSI, Observatory of Public Sector Innovation, l'osservatorio della OECD che analizza e monitora le buone pratiche per l'innovazione nella Pubblica Amministrazione dei paesi che ne fanno parte.

3 Utilizzo di Unreal Engine

Durante l'edizione 2023 di Ragazze Digitali è stato sperimentato l'uso di Unreal Engine per la progettazione di videogiochi in 3D. In questa sezione analizziamo vantaggi e svantaggi di questo approccio.

3.1 Vantaggi

L'uso dei Blueprint di Unreal Engine presenta una serie di vantaggi significativi per l'apprendimento della logica di programmazione e lo sviluppo di competenze interdisciplinari. Ecco alcuni punti chiave che illustrano i benefici di questo approccio:

- **Interesse per i giochi:** La possibilità di creazione di un gioco interattivo motiva gli studenti mantenendo alto il loro interesse nell'apprendimento della programmazione.
- **Interattività:** Unreal Engine consente di creare ambienti interattivi, in modo che gli utenti possono esplorare gli spazi in modo immersivo e interagendo con gli elementi circostanti. Ciò è utile per mostrare come gli spazi si relazionano tra loro e come gli utenti possono interagire con essi.
- **Accessibilità:** Blueprint è stato progettato per essere accessibile anche a coloro che non hanno esperienza di programmazione: questo lo rende ideale per studenti che possono non avere familiarità con concetti di programmazione complessi. L'interfaccia visuale e l'assenza di codice testuale possono abbassare la barriera all'ingresso e rendere più facile per i principianti iniziare a creare e capire i concetti fondamentali della programmazione.
- **Approccio visuale:** Un approccio visuale alla programmazione consente agli studenti di creare logiche di gioco e interazioni tramite nodi e collegamenti grafici anziché scrivere codice testuale. Questo approccio può essere più intuitivo per gli studenti, poiché possono creare e comprendere facilmente la logica attraverso rappresentazioni visive anziché attraverso righe di codice.
- **Logica di programmazione:** Con i Blueprints, gli studenti imparano a pensare in modo algoritmico e a comprendere i concetti fondamentali della programmazione, come le strutture di controllo (es. cicli e condizioni) e l'organizzazione dei dati. Anche se non stanno scrivendo codice testuale, stanno sviluppando le competenze di base necessarie per risolvere problemi e creare soluzioni attraverso la logica di programmazione. Questo è particolarmente importante perché la logica di programmazione è una competenza trasversale che può essere applicata in molte altre discipline e situazioni nella vita quotidiana. Insegnare agli studenti a pensare in modo algoritmico li prepara non solo per eventuali carriere nel campo della tecnologia, ma anche per risolvere problemi in altri settori, come la matematica, la scienza o l'ingegneria.
- **Risultati visivi immediati:** Con Blueprints, gli studenti possono vedere immediatamente i risultati delle loro modifiche senza la necessità di ricompilare il codice, con risultati gratificanti e incoraggianti. Questo ciclo di feedback rapido può mantenere alta l'attenzione degli studenti e mantenerle coinvolte nel processo di apprendimento.
- **Collaborazione e apprendimento sociale:** I Blueprints di Unreal Engine offrono l'opportunità di lavorare in team, potendo unire i vari pezzi di progetto facilmente e condividendo progetti: questo consente di sviluppare competenze di comunicazione, problem solving collaborativo e peer-to-peer learning. La collaborazione sociale, il supporto reciproco e l'ampliamento delle prospettive possono arricchire notevolmente l'esperienza di apprendimento degli studenti, preparandoli non solo per il mondo della programmazione, ma anche per il mondo del lavoro e della vita.

In sintesi, l'uso di Unreal Engine e dei Blueprints offre un approccio innovativo e coinvolgente per l'apprendimento della programmazione, che stimola l'interesse degli studenti, favorisce una comprensione profonda dei concetti di base e promuove la collaborazione e il lavoro di squadra.



Figure 2: Blueprints di Unreal Engine

3.2 Svantaggi

Consideriamo ora anche i possibili svantaggi associati all'uso di Unreal Engine per poter valutare attentamente quale approccio sia più adatto alle esigenze e agli obiettivi specifici di apprendimento. Alcuni svantaggi sono:

- **Requisiti hardware:** Unreal Engine notoriamente pesante e richiede computer performanti per funzionare correttamente. Non è sempre facile aggiudicarsi computer performanti in ambito scolastico. In presenza di hardware meno performante, potrebbe essere necessario ridurre la qualità grafica del gioco. Ciò potrebbe includere la diminuzione della risoluzione, la riduzione del numero di oggetti e dettagli nell'ambiente, la diminuzione della qualità delle texture e la riduzione dell'effetti grafici come le ombre dinamiche e gli effetti di illuminazione avanzati. Gli sviluppatori devono trovare un equilibrio tra prestazioni e qualità grafica, cercando di creare un'esperienza soddisfacente anche su hardware meno performante senza compromettere troppo la qualità visiva complessiva del gioco.
- **Assenza di scrittura diretta di codice:** I Blueprints non preparano direttamente gli studenti a scrivere codice reale, il che potrebbe essere importante per coloro che desiderano proseguire in campi più tecnici. L'assenza di pratica nella scrittura diretta di codice potrebbe rendere più difficile per gli studenti adattarsi ai linguaggi di programmazione "reali". Senza l'esperienza diretta nella scrittura di codice, gli studenti potrebbero trovare più difficile imparare la sintassi e le convenzioni dei linguaggi di programmazione effettivi, che spesso richiedono una comprensione dettagliata della struttura del linguaggio e delle regole grammaticali associate.

Sebbene questi svantaggi possano rappresentare delle sfide, è importante considerare che gli strumenti visivi come i Blueprints offrono comunque un modo intuitivo e accessibile per imparare i concetti fondamentali della programmazione e creare giochi interattivi. Tuttavia, per coloro che desiderano una formazione più approfondita o proseguire in campi più tecnici, potrebbe essere necessario integrare l'apprendimento dei Blueprints con l'esperienza nella scrittura diretta di codice.

4 Conclusioni

L'uso di strumenti come Unreal Engine e Blueprints presenta vantaggi e svantaggi nell'ambito dell'insegnamento dell'informatica. L'approccio visuale e intuitivo offerto dai Blueprints rende

più accessibili ai neofiti concetti complessi e stimola l'interesse immediato. D'altro canto, data l'assenza di scrittura diretta di codice gli studenti potrebbero trovare più difficile adattarsi poi ai linguaggi di programmazione convenzionali. Tuttavia, è importante considerare che l'obiettivo dell'insegnamento della programmazione non è solo quello di impartire conoscenze specifiche su un linguaggio di programmazione, ma anche di sviluppare competenze di pensiero critico, risoluzione dei problemi e collaborazione. In questo senso, gli strumenti visivi come i Blueprints possono essere molto utili nel promuovere tali competenze, consentendo agli studenti di concentrarsi sulla logica di programmazione piuttosto che sulla sintassi del codice.

References

- [1] K. Davaki. *The underlying causes of the digital gender gap and possible solutions for enhanced digital inclusion of women and girls*. 2018. URL: [http://www.europarl.europa.eu/RegData/etudes/STUD/2018/604940/IPOL_STU\(2018\)604940_EN.pdf](http://www.europarl.europa.eu/RegData/etudes/STUD/2018/604940/IPOL_STU(2018)604940_EN.pdf).
- [2] Yasin A. M. El-Wajeh, Paul V. Hatton, and Nicholas J. Lee. “Unreal Engine 5 and immersive surgical training: translating advances in gaming technology into extended-reality surgical simulation training programmes”. In: *British Journal of Surgery* 109 (2022). DOI: [10.1093/bjs/znac015](https://doi.org/10.1093/bjs/znac015).
- [3] European Statistics Eurostat. *ICT specialists are predominantly male*. 2019. URL: <https://ec.europa.eu/eurostat/de/web/products-eurostat-news/-/DDN-20190513-1>.
- [4] F. Faenza, C. Canali, and T. Addabbo. “Design and development of tools for the implementation, evaluation and analysis of STEM activities to counteract stereotypes and gender gap”. In: *Journal Name* ().
- [5] F. Faenza, C. Canali, and A. Carbonaro. “ICT Extra-curricular Activities: The “Digital Girls” Case Study for the Development of Human Capital”. In: (2021). Online Event. DOI: [10.1007/978-3-030-84311-3_18](https://doi.org/10.1007/978-3-030-84311-3_18).
- [6] F. Faenza et al. “The digital girls response to pandemic: Impacts of in presence and online extracurricular activities on girls future academic choices”. In: *Education Science Journal* 11 (2021). DOI: [10.3390/educsci11110715](https://doi.org/10.3390/educsci11110715).
- [7] Epic Games. *Unreal Engine: We make the engine. You make it Unreal*. 2024. URL: <https://www.unrealengine.com>.
- [8] Rhys Hanrahan. “Designing a Computer Game to Teach Computer Science Concepts”. In: *Wellington Faculty of Engineering Symposium* (2023). URL: <https://ojs.victoria.ac.nz/wfes/article/view/8405>.
- [9] B. Spieler, L. Oates-Indruchova, and W. Slany. “Female Students in Computer Science Education: Understanding Stereotypes, Negative Impacts, and Positive Motivation”. In: *Journal of Women and Minorities in Science and Engineering* 26.5 (2020), pp. 473–510. DOI: [10.48550/arXiv.1903.01190](https://doi.org/10.48550/arXiv.1903.01190).
- [10] European Eurostat Statistics. *Girls and women under-represented in ICT*. 2017. URL: <https://ec.europa.eu/eurostat/de/web/products-eurostat-news/-/EDN-20170426-1>.

Diagrammi di flusso a supporto dell'apprendimento della programmazione

Gabriele Pozzan, Tullio Vardanega

Università degli studi di Padova
gabriele.pozzan@phd.unipd.it
tullio.vardanega@unipd.it

Abstract

Questo studio presenta i risultati di una sperimentazione didattica mirata a valutare l'impatto di uno strumento di supporto all'apprendimento della programmazione basato su diagrammi di flusso eseguibili. 68 studentesse e studenti del primo anno di un corso di Laurea Triennale in Informatica hanno risolto un problema di programmazione dopo aver affrontato 1 mese di didattica. Metà di questi studenti hanno ricevuto supporto nella forma di diagrammi di flusso eseguibili che rappresentavano soluzioni parziali a esercizi di laboratorio. L'ipotesi sperimentale era che questo supporto, riducendo il carico cognitivo, avrebbe favorito l'apprendimento. I risultati sembrano confermare l'ipotesi e suggerire un impatto positivo di questo approccio sull'apprendimento di concetti base di programmazione in C.

1 Introduzione

Gli insegnamenti universitari di introduzione alla programmazione, anche nei corsi di laurea in Informatica, possono comportare il primo incontro dei e delle partecipanti con:

- il pensiero algoritmico informatico (cioè computabile);
- i modi per tradurre il prodotto in un programma testuale.

Il primo elemento è fondamentale poiché permette di risolvere problemi informatici; il secondo è "accidentale" perché dipendente dal linguaggio utilizzato, dall'architettura su cui deve essere eseguita la soluzione, ecc.

La teoria del **carico cognitivo** [4] mette in guardia contro la presentazione di una eccessiva quantità di concetti "nuovi" nell'arco di brevi intervalli di apprendimento. Questo stile porta a sovraccarico della memoria di lavoro (memoria a breve termine attiva durante l'apprendimento) non solo causando il mancato assorbimento delle nozioni "in eccesso", ma più in generale riducendo la qualità dell'apprendimento.

Un modo per ridurre il carico cognitivo di attività di apprendimento è fornire supporto attivo (*scaffolding*) ai discenti [5]: in questo articolo presentiamo i risultati di una sperimentazione basata sul supporto all'apprendimento della programmazione in linguaggio C tramite diagrammi di flusso eseguibili.

Abbiamo confrontato due gruppi, omogenei a livello di competenze di partenza, dopo aver fornito solo a uno dei due questo supporto per un mese di attività didattiche. La nostra aspettativa era che questo gruppo avrebbe appreso e padroneggiato con più sicurezza i concetti, dimostrandolo nella risoluzione di un problema di programmazione complesso.

2 Sperimentazione

2.1 Intervento

Abbiamo condotto la sperimentazione durante il corso di introduzione alla programmazione offerto al primo anno del corso di Laurea Triennale in Informatica dell'Università di Padova. L'insegnamento è obbligatorio e prevede lezioni teoriche (40 ore) e sessioni pratiche in laboratorio (32 ore).

Abbiamo diviso casualmente la platea in un gruppo **sperimentale** e un gruppo di **controllo**. Durante i primi 4 laboratori del corso (della durata di 2 ore ciascuno e condotti tra marzo e aprile 2024) i due gruppi hanno affrontato la stessa sequenza di esercizi di programmazione in C basati sui concetti di selezione, iterazione, scomposizione in funzioni e sull'uso di array.

Per ogni esercizio, i partecipanti ricevevano il testo (descrizione di obiettivi, esempi di input e output) e test di unità automatici per verificare la correttezza delle proprie soluzioni (programmi in C). Il testo degli esercizi che introducevano un nuovo concetto (p.es, il primo esercizio sul concetto di iterazione) era aumentato con un diagramma di flusso che rappresentava la soluzione algoritmica completa di quell'esercizio. Il diagramma poteva essere eseguito e modificato su una piattaforma online. Questi esercizi invitavano esplicitamente a concentrarsi sulla traduzione in C del diagramma di flusso.

In aggiunta a questo materiale, il gruppo sperimentale riceveva diagrammi di flusso eseguibili rappresentanti soluzioni **parziali** anche per gli esercizi successivi al primo. Per ogni esercizio riguardante un unico tema, i diagrammi di flusso mostrati al gruppo sperimentale erano progressivamente sempre più incompleti, fino a scomparire del tutto, lasciando completa libertà nella soluzione algoritmica dell'esercizio. La Figura 1 esemplifica questo approccio, ispirato dal modello didattico discusso in [5].

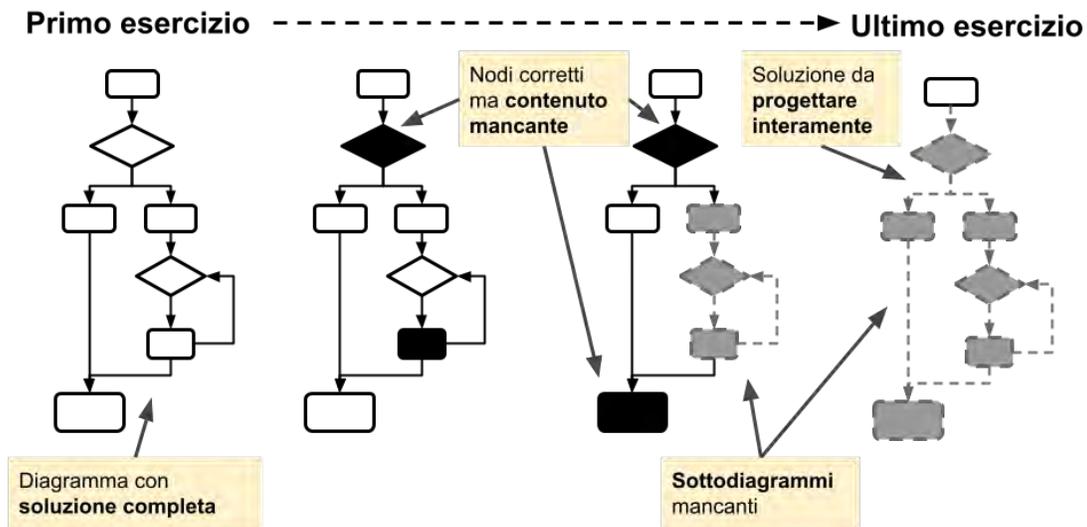


Figura 1: Supporto "a scomparsa" tramite diagrammi di flusso, fornito al gruppo sperimentale

L'obiettivo di questo supporto "a scomparsa" è ridurre il carico cognitivo rimuovendo inizialmente, e reintroducendo progressivamente, il costo legato alla soluzione del problema a livello

algoritmico.

I membri del gruppo di controllo dovevano invece costruire la soluzione algoritmica completa: avevano la possibilità di utilizzare i diagrammi di flusso eseguibili sulla piattaforma online, ma dovevano costruirli completamente per ogni esercizio successivo al primo.

2.2 Raccolta dati

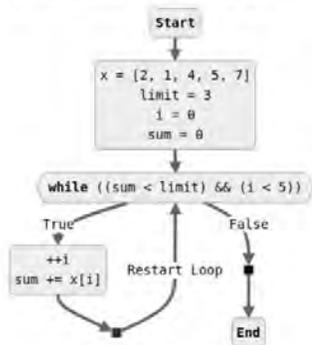
Abbiamo valutato i partecipanti con un Pretest a monte e un Posttest a valle dell'intervento.

Per il primo abbiamo somministrato un questionario che chiedeva di:

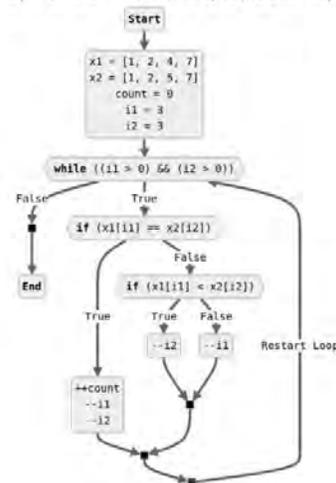
- Segnalare (risposta "sì/no") **esperienze precedenti** con la programmazione a livello 1) di scuole superiori, 2) universitario, 3) professionale. Abbiamo valutato ogni esperienza precedente con 3 punti. Il punteggio di questa sottosezione variava tra 0 e 9.
- Autovalutare la propria **competenza** relativa ai concetti di variabili, selezione, logica Booleana, iterazione, funzioni, ricorsione. Ogni categoria permetteva di selezionare una risposta tra quattro livelli di competenza il cui punteggio variava tra 0 (nessuna esperienza) e 1.5 (ampia esperienza, anche a livello professionale). Anche per questa sottosezione il punteggio variava tra 0 e 9.
- Interpretare 3 **diagrammi di flusso** (cfr. Figura 2), indicando il valore assunto da una particolare variabile dopo l'esecuzione. Questa domanda dava la possibilità di rispondere "non lo so" (0 punti) o di indicare il valore numerico assunto dalla variabile (3 punti per una risposta corretta, -3 punti per una risposta errata). Il punteggio di questa sottosezione variava tra -9 e 9.

Domanda: "Indicare il valore della variabile ... dopo l'esecuzione"

1) variabile **i** (risposta: 2)



2) variabile **count** (risposta: 2)



3) variabile **i** (risposta: 3)

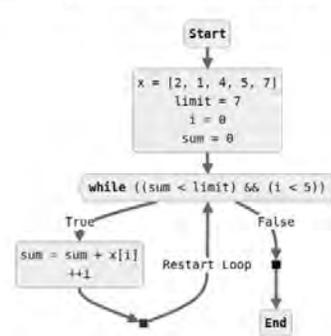


Figura 2: Diagrammi di flusso da interpretare per il Pretest, ispirati da [1]

Sommando il punteggio delle tre sottosezioni il totale poteva quindi variare tra -9 e 27.

Durante il Pretest abbiamo inoltre raccolto i consensi informati per la partecipazione alla sperimentazione e l'uso anonimo dei dati.

Il Posttest chiedeva di programmare in C una soluzione al *Rainfall problem* [3]. Il testo dell'esercizio è il seguente: "Scrivete un programma che, dato un array di $N > 0$ valori interi (una qualsiasi combinazione di numeri interi, cioè maggiori, minori o uguali a zero), legga questi valori uno ad uno fino ad incontrare il valore di guardia 99999. Dopo aver incontrato il valore di guardia il programma deve stampare la media dei valori positivi (maggiori di 0) letti fino a quel momento, escluso il valore di guardia."

Abbiamo condotto il Posttest durante il quinto laboratorio del corso: abbiamo chiesto di lavorare autonomamente e di provare a superare il test con il minor numero possibile di tentativi, senza limite massimo. I partecipanti potevano effettuare più "consegne": ognuna veniva compilata e verificata con test di unità di cui potevano consultare il responso (cfr. Figura 3). I partecipanti non avevano né un limite massimo al numero di tentativi né l'obbligo di completare l'esercizio.

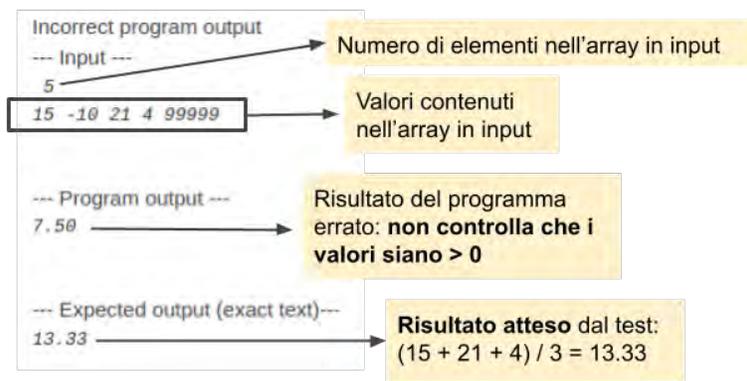


Figura 3: Esempio di responso di un test di unità per una soluzione errata

3 Risultati

3.1 Partecipanti e Pretest

173 studentesse e studenti su 263 totali hanno acconsentito a partecipare alla sperimentazione durante il Pretest. Di questi, 83 hanno partecipato al Posttest: 34 appartenenti al gruppo sperimentale e 49 al gruppo di controllo.

I due gruppi non sono equivalenti a livello di punteggi Pretest: i 34 membri del gruppo sperimentale hanno un risultato medio di 14.13 (deviazione standard, $DS = 6.31$), mentre i 49 membri del gruppo di controllo hanno un risultato medio di 10.64 ($DS = 7.97$). Un t-test conferma la differenza tra i due gruppi con $p = .048$.

Per poter confrontare i risultati del Posttest abbiamo quindi selezionato a posteriori un sottoinsieme del gruppo di controllo equivalente al gruppo sperimentale a livello Pretest. Per ogni partecipante P_s del gruppo sperimentale abbiamo selezionato il partecipante del gruppo di controllo P_c il cui risultato Pretest fosse il più vicino possibile a quello di P_s e lo abbiamo aggiunto al gruppo di controllo selezionato.

In questo modo abbiamo ottenuto un gruppo di controllo selezionato di 34 partecipanti, con punteggio medio Pretest di 13.76 ($DS = 6.71$). Un t-test conferma che è statisticamente confrontabile a Pretest con il gruppo sperimentale ($p = .819$).

La Tabella 1 riassume i risultati del Pretest.

Gruppo	N. studenti	Punteggio medio	Deviazione standard
Sperimentale	34	14.13	6.31
Controllo (completo)	49	10.64	7.97
Controllo (selezionato)	34	13.76	6.71

Tabella 1: Risultati Pretest: media e deviazione standard dei punteggi

3.2 Posttest

La Figura 4 mostra quanti partecipanti hanno consegnato una soluzione che compilasse e superasse tutti i test di unità dopo 1, 2, 3 o più di 3 tentativi. Inoltre, la Figura mostra quanti non sono arrivati a consegnare una soluzione corretta. 17 membri del gruppo sperimentale (50%) e 11 del gruppo di controllo (32%) hanno consegnato una risposta corretta entro i primi tre tentativi.

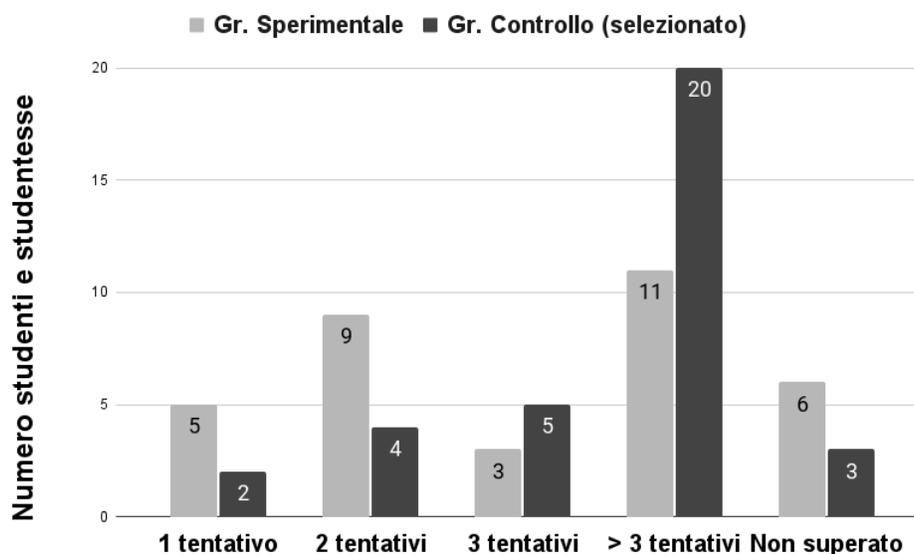


Figura 4: Tentativi per superare il Posttest

La Figura 5 mostra la distribuzione del numero di tentativi necessari per superare il Posttest. Un t-test conferma che le distribuzioni sono diverse con $p = .038$: i partecipanti del gruppo sperimentale che hanno superato l'esercizio lo hanno fatto mediamente con un numero minore di tentativi.



Figura 5: Distribuzione del numero di tentativi per superare il Posttest

4 Discussione

Innanzitutto osserviamo che il *Rainfall problem* risulta difficile dopo un solo mese di percorso nell'insegnamento della programmazione: solo 5 persone del gruppo sperimentale e 2 del gruppo di controllo sono riuscite a superarlo al primo tentativo. Questo è in linea con quanto riportato nella letteratura scientifica [3] e fa riflettere sul fatto che le aspettative didattiche possano spesso essere eccessive, come discusso da [2].

Il gruppo sperimentale è riuscito a risolvere l'esercizio con un numero di tentativi statisticamente minore rispetto al gruppo di controllo: questo risultato emerge sia verificando quanti abbiano risolto l'esercizio entro i primi 3 tentativi, sia considerando la distribuzione generale della Figura 5. Il numero di studenti del gruppo di controllo che ha fornito una soluzione corretta è maggiore (31 contro i 28 del gruppo sperimentale), però gli studenti non avevano alcun obbligo a completare l'esercizio e potrebbero quindi essere entrati in gioco fattori di motivazione.

Questi risultati sembrano suggerire che il gruppo sperimentale abbia tratto giovamento dal supporto dato dalla possibilità di concentrarsi inizialmente sulla **codifica** delle soluzioni, aggiungendo progressivamente aspetti di **ragionamento algoritmico**.

Riferimenti bibliografici

- [1] Raymond Lister, Elizabeth S Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, et al. A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, 36(4):119–150, 2004.
- [2] Andrew Luxton-Reilly. Learning to program is easy. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 284–289, 2016.
- [3] Otto Seppälä, Petri Ihantola, Essi Isohanni, Juha Sorva, and Arto Vihavainen. Do we know how difficult the rainfall problem is? In *Proceedings of the 15th Koli Calling Conference on Computing Education Research*, pages 87–96, 2015.
- [4] John Sweller, Jeroen JG van Merriënboer, and Fred Paas. Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review*, 31:261–292, 2019.
- [5] Jeroen JG Van Merriënboer. Perspectives on problem solving and instruction. *Computers & Education*, 64:153–160, 2013.

Insegnare SQL a Chi Non Ha Mai Programmato: Analisi delle Misconcezioni

Daniele Traversaro¹²

¹ I.I.S.S. EINAUDI-CASAREGIS-GALILEI, Genova
`traversaro.daniele@ecg-genova.edu.it`

² Università di Genova, DIBRIS
`daniele.traversaro@dibris.unige.it`

Sommario

L'insegnamento di basi di dati a studenti di scuola secondaria di secondo grado presenta diverse sfide. Questo articolo esplora le principali misconcezioni su SQL tra gli studenti di un istituto professionale per i servizi commerciali, che non hanno precedenti esperienze di programmazione. Vengono analizzate le cause e proposti metodi per migliorare l'insegnamento e l'apprendimento di SQL.

1 Introduzione

Nell'anno scolastico 2023-2024, il gruppo di lavoro del progetto "Curricolo" afferente al Piano di Miglioramento dell'Istituzione Scolastica dell'Istituto Scolastico Einaudi-Casaregis-Galilei¹ di Genova ha elaborato un nuovo curriculum di informatica per l'indirizzo Servizi Commerciali. Il curriculum, articolato nelle due opzioni - Web Community Manager (WCM) ed Esperto del Turismo Innovativo - ha l'obiettivo di adeguare le competenze informatiche degli studenti alle reali esigenze del mercato del lavoro e di promuovere una progettazione didattica per unità di apprendimento e per assi culturali.

Entrambe le opzioni del curriculum si focalizzano sul web semantico, con particolare attenzione alla SEO per migliorare il posizionamento dei siti nei motori di ricerca, aumentando la visibilità e il traffico organico. Approfondiscono l'uso dell'HTML per sviluppare pagine web ben strutturate e ottimizzate. Infine, sono orientate verso strumenti di analisi web per interpretare i dati di traffico e comprendere il comportamento degli utenti, oltre che verso strumenti per il social media marketing.

Oltre a ciò, l'opzione WCM si focalizza ulteriormente sulla gestione delle comunità online, sul sistema informativo aziendale e sull'analisi e l'etica dei dati. In questo ambito, vengono introdotte diverse tecnologie del sistema informatico, come i database, i data warehouse e i software ERP. In particolare, viene presentato il linguaggio informatico SQL per la gestione e la manipolazione dei dati.

Questo articolo si concentra sulla didattica dei database in particolare sulle misconcezioni del linguaggio SQL emerse durante l'anno scolastico tra gli studenti delle classi quinte, offrendo spunti di riflessione per migliorare l'insegnamento e per rendere l'apprendimento di questo linguaggio più efficace e accessibile a tutti gli studenti, anche a quelli senza precedenti esperienze di programmazione.

2 Revisione della Letteratura

Uno dei principali ostacoli nell'apprendimento del linguaggio SQL è il trasferimento di conoscenze acquisite in altri contesti, come la matematica e la programmazione [2, 5]. Gli studenti

¹<https://www.ecg-genova.edu.it>

tendono ad applicare modelli di pensiero e strutture linguistiche familiari a SQL, generando misconcezioni. Per esempio, la presenza di parole chiave in SQL che assomigliano al linguaggio naturale può portare a trasferimenti linguistici errati e a errori sintattici. Anche l'esperienza con altri linguaggi di programmazione può causare misconcezioni; una delle più frequenti è la confusione tra gli operatori “==” e “=” nelle query SQL. Sebbene l'operatore “==” non faccia parte della sintassi SQL, gli studenti lo utilizzano erroneamente a causa della familiarità con altri linguaggi [2].

Altri esempi di misconcezioni tra gli studenti alle prime esperienze includono l'uso inappropriato degli auto-join per recuperare coppie di record dalla stessa tabella, senza rendersi conto della necessità di referenziare la tabella due volte nella query, rivelando un modello mentale incompleto di SQL [2]. In alcuni casi, gli studenti vedono il sistema di gestione del database come una “scatola nera”, senza comprenderne i principi sottostanti. In particolare, la letteratura evidenzia che le sotto-query, le clausole `GROUP BY` e le `JOIN` sono le aree principali in cui si manifestano più frequentemente errori e misconcezioni [1, 2, 6]. Per quanto riguarda la clausola `GROUP BY`, si riscontrano problemi di collocazione all'interno della query, nonché confusione su quando è necessaria.

Infine, altre misconcezioni che emergono dalla letteratura includono l'erronea convinzione che la clausola `DISTINCT` selezioni la prima tupla della tabella, l'uso inappropriato delle chiavi primarie, errori sintattici come ricordare solo parte di una parola chiave (ad esempio, `GROUP` invece di `GROUP BY`) o usare sinonimi in modo intercambiabile (ad esempio, `SUM` con `COUNT`, `SORT` con `ORDER BY`) [4, 2].

3 Contesto e Popolazione

L'esperienza didattica si è svolta in due classi quinte dell'istituto professionale per i servizi commerciali, opzione WCM, composte rispettivamente da 21 e 25 alunni, di età compresa tra i 18 e i 22 anni. Gli studenti non avevano competenze specifiche in linguaggi di programmazione, ma possedevano conoscenze informatiche di base sull'utilizzo di software di fogli di calcolo e di CMS come Wix e WordPress. Tuttavia, la loro familiarità con i software di fogli di calcolo si è rivelata un punto di forza per facilitare la comprensione di SQL, poiché entrambi gli strumenti condividono alcune caratteristiche concettuali, come la rappresentazione tabulare dei dati e l'uso di funzioni per l'elaborazione delle informazioni.

Per valutare l'apprendimento degli studenti sono stati impiegati diversi metodi, tra cui esercitazioni pratiche in laboratorio, prove orali e scritte, e l'osservazione del docente durante le lezioni, con l'obiettivo di fornire una valutazione formativa continua. Il software PostgreSQL è stato utilizzato in laboratorio per fornire agli studenti un'esperienza pratica con un sistema DBMS.

In entrambe le classi sono stati trattati i concetti base di SQL, come le clausole `SELECT`, `FROM`, `WHERE`, `GROUP BY`, `ORDER BY`, `LIKE`, `BETWEEN`, `LIMIT`, `DISTINCT` e le funzioni di aggregazione `COUNT`, `SUM`, `AVG` e `MIN`. La verifica scritta verteva su questi costrutti e consisteva nel formulare interrogazioni SQL su una singola tabella. Tuttavia, le `JOIN`, le chiavi primarie ed esterne, e i comandi di base di DDL sono stati affrontati solo durante le esercitazioni in laboratorio con PostgreSQL.

Di seguito, sono elencati e analizzati i principali errori e le misconcezioni su SQL emersi nel corso dell'anno. I dati sono stati raccolti attraverso la correzione e la discussione della verifica scritta, eventuali interrogazioni orali di recupero e l'osservazione durante le ore di laboratorio, in cui il docente svolgeva il ruolo di tutor e facilitatore dell'apprendimento.

4 Misconcezioni degli Studenti su SQL

Uno degli errori più comuni tra gli studenti era dimenticare di utilizzare l'uguale tra il nome della colonna e il valore nella clausola `WHERE`, portando a una sintassi non corretta e al fallimento della query. Un altro errore frequente, sia sintattico che concettuale, era omettere il nome della colonna prima del valore. Ad esempio, invece di scrivere `WHERE colonna = valore`, veniva scritto `WHERE valore`. Questo errore si riscontra anche tra gli studenti del primo biennio, quando applicano funzioni di Excel come `SOMMA.SE` o `MEDIA.SE`, dimenticando il riferimento alla cella e indicando solo il valore della condizione. Durante le interrogazioni orali, molti studenti hanno dichiarato di aver dimenticato di specificare la colonna perché pensavano fosse implicita, attribuendo al computer una sorta di intelligenza capace di interpretare le loro intenzioni, evidenziando così la misconcezione del “superbug” [3]. Questo evidenzia una comprensione errata della natura dichiarativa del linguaggio.

Un errore simile riguardava l'omissione della clausola `FROM`, spesso ritenuta implicita dagli studenti. Questo indica un modello mentale di SQL altamente incorretto e il fatto che in classe si svolgevano principalmente query su singola tabella, perdendo di vista la natura relazionale dei database. In laboratorio, dove si lavorava con un database composto da più tabelle in relazione tra loro, questo errore non si verificava. È quindi utile presentare agli studenti scenari diversi, per evitare misconcezioni, come l'idea che le query SQL operino esclusivamente su singole (o multiple) tabelle.

Un altro errore frequente riguardava la clausola `GROUP BY`. Molti studenti riscontravano difficoltà a ragionare in termini di raggruppamenti e a riconoscere quando era necessario applicare la clausola, spesso sostituendola con una semplice `WHERE`, mettendo come condizione una colonna e un solo valore, senza distinguere tra filtraggio dei dati e raggruppamento. L'omissione del nome della colonna nella clausola `ORDER BY` era un altro errore comune. Inoltre, la confusione tra i termini inglesi `DESC` (decrescente) e `ASC` (crescente) portava a ordinamenti indesiderati.

Alcuni studenti confondevano anche la clausola `DISTINCT` con `DESC`, a causa di difficoltà con la lingua inglese. Inoltre, alcuni studenti mostravano difficoltà nell'ordinare i valori in modo decrescente: anziché utilizzare la sintassi `ORDER BY colonna DESC`, tendevano a scrivere `ORDER BY MAX(colonna)`, dimostrando una comprensione sbagliata dell'esecuzione delle query e delle funzioni di aggregazione.

Quando si trattava dell'utilizzo dell'operatore `LIKE`, si riscontravano frequentemente difficoltà nel distinguere tra `%` e `_`, il che portava a errori a causa della complessità della notazione.

Altri errori frequenti includevano l'uso di sinonimi, come `COUNT` al posto di `SUM` e viceversa, con difficoltà nel distinguere tra il conteggio dei record e la somma di valori numerici. Inoltre, alcuni studenti mettevano le colonne del `SELECT` dentro la clausola `FROM`, o le colonne da indicare nel `WHERE` nel `SELECT`, dimenticando anche di includere il nome della colonna nelle funzioni di aggregazione come `AVG` o `SUM`.

Tutti questi errori evidenziano un modello mentale superficiale di SQL e una visione del piano di esecuzione delle query come “scatola nera”. Infatti, molti studenti, pur riuscendo a interpretare visivamente i dati nella tabella, trovavano difficile tradurre la richiesta in una query scritta in SQL.

5 Conclusioni

I risultati ottenuti riguardo alle misconcezioni sul linguaggio SQL rappresentano un punto di partenza per riflessioni e sviluppi futuri. Questi risultati sono utili per comprendere le

principali difficoltà affrontate dagli studenti senza esperienze precedenti in programmazione. Inoltre, offrono l’opportunità di riflettere su come si sviluppa un modello mentale di SQL senza conoscenze pregresse in altri linguaggi di programmazione. Molte delle misconcezioni riscontrate confermano quanto discusso nella letteratura, mentre altre sono ancora più di base e suggeriscono che per chi non ha esperienza di programmazione, imparare SQL può essere particolarmente impegnativo.

Tali problematiche richiedono un approccio didattico che comprenda attività pratiche, esercitazioni guidate e l’uso di strumenti didattici innovativi come SQL Tutor² o SQLVis³, che mostrano passo dopo passo l’esecuzione delle query e la manipolazione dei dati. È fondamentale, quindi, stabilire il livello appropriato di astrazione della macchina nozionale di SQL per gli studenti.

Per esplorare in modo più approfondito le sfide e migliorare l’apprendimento, si suggerisce di condurre future ricerche che includano interviste strutturate con un campione più ampio di studenti. Estendere lo studio anche agli studenti universitari potrebbe consentire di classificare le misconcezioni in base a variabili come tipologia, età e esperienza pregressa con la programmazione.

Riferimenti Bibliografici

- [1] Stefan Brass and Christian Goldberg. Semantic errors in SQL queries: A quite complete list. *Journal of Systems and Software*, 79(5):630–644, 2006.
- [2] Daphne Miedema, Efthimia Aivaloglou, and George Fletcher. Identifying SQL misconceptions of novices: Findings from a think-aloud study. *ACM Inroads*, 13(1):52–65, 2022.
- [3] Roy D. Pea. Language-independent conceptual “bugs” in novice programming. *Journal of Educational Computing Research*, 2(1):25–36, 1986.
- [4] Kai Presler-Marshall, Sarah Heckman, and Kathryn Stolee. SQLRepair: Identifying and repairing mistakes in student-authored SQL queries. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pages 199–210. IEEE, 2021.
- [5] John P. Smith III, Andrea A. diSessa, and Jeremy Roschelle. Misconceptions reconceived: A constructivist analysis of knowledge in transition. *The journal of the learning sciences*, 3(2):115–163, 1994.
- [6] Toni Taipalus and Piia Perälä. What to expect and what to focus on in SQL query teaching. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 198–203, 2019.

²SQL Tutor: <https://cudbg.github.io/sqltutor/>

³SQLVis: <https://github.com/Giraphne/sqlvis>

Indagine su Approccio Cognitivo e Risultati conseguiti dagli Studenti delle Scuole Superiori in Problemi di Programmazione Concorrente

Emanuele Scapin¹ and Nicola Dalla Pozza¹

ITT G. Chilesotti, 36016 Thiene, VI
{escapin,ndallapozza}@chilesotti.it

Abstract

Di recente, abbiamo esplorato la percezione e le sfide della programmazione concorrente nelle scuole superiori con specializzazione in Informatica, dove questo argomento è parte del curriculum. È emerso che gli studenti necessitano di supporti visivi per comprendere meglio il comportamento dei thread e la sincronizzazione, in linea con ricerche che indicano una prevalenza di stili cognitivi visuo-spaziali negli studenti STEM. Con questo lavoro abbiamo investigato il collegamento tra l'apprendimento della programmazione concorrente e gli stili cognitivi degli studenti, valutandone le prestazioni e misurando i loro stili cognitivi con un questionario standard. Ne è risultata una lieve prevalenza di stili cinestetici e uditivi, senza una correlazione significativa tra stili cognitivi individuali e livello di successo o autostima. Infine, gli schemi tracciati degli studenti per rappresentare thread concorrenti sottolineano le difficoltà nel padroneggiare la sincronizzazione.

1 Introduzione

La programmazione concorrente è obbligatoria per gli studenti delle superiori specializzati in Informatica in Italia¹. Tuttavia, si incontrano difficoltà nell'insegnamento e apprendimento, soprattutto per la natura astratta dei concetti alla base del comportamento dei thread e della sincronizzazione [1].

Le nostre precedenti indagini [2] hanno confermato che gli studenti spesso faticano a comprendere questi concetti e hanno rivelato che sentirebbero il bisogno di ausili visivi/grafici per migliorare la loro comprensione. Questo sembra in linea con altri lavori che suggeriscono che gli ausili visivi possono migliorare la comprensione di argomenti complessi [3], e che gli studenti nei campi STEM sono più propensi a sfruttare lo stile cognitivo visivo-spaziale [4].

Il presente studio mira a esplorare le seguenti due domande di ricerca:

1. In che misura le prestazioni degli studenti e la percezione della fiducia in se stessi nei compiti di programmazione concorrente correlano con una prevalenza di uno stile cognitivo visuo-spaziale?
2. Che tipo di annotazioni o schizzi utilizzano spontaneamente gli studenti mentre affrontano semplici compiti di programmazione concorrente?

Sebbene studi recenti abbiano dimostrato che gli stili cognitivi possono influenzare i processi di apprendimento nell'educazione STEM [5, 6], il loro impatto sul successo nei compiti di programmazione è ancora poco esplorato. Da questa prospettiva, questo studio esplorativo offre dati empirici sull'impatto degli stili cognitivi sulle prestazioni nella programmazione concorrente e fornisce indicazioni preliminari sugli approcci visivi spontanei degli studenti nella comprensione dei programmi, utili per sviluppare strategie didattiche più efficaci.

¹<https://www.gazzettaufficiale.it/eli/gu/2012/03/30/76/so/60/sg/pdf>

Il presente lavoro è organizzato come segue. La Sezione 2 presenta il contesto di riferimento, successivamente la Sezione 3 descrive la struttura dell'indagine e la logica alla base della sua progettazione. I principali risultati della nostra indagine esplorativa sono poi delineati nella Sezione 4 e discussi nella Sezione 5. Alla fine, nella Sezione 6 traiamo alcune conclusioni e accenniamo ai possibili sviluppi futuri del presente lavoro.

2 Contesto di riferimento

Sebbene Torbert et al. [7] sostengano l'insegnamento del pensiero algoritmico parallelo e del multi-threading nelle scuole superiori, le difficoltà che gli studenti affrontano con la programmazione concorrente sono riconosciute a tutti i livelli educativi. Per una rassegna completa, i lettori possono fare riferimento all'analisi di Brođanac et al. [1]. Ben-Ari & Kolikant [8] esplorano lo sviluppo degli algoritmi paralleli da parte degli studenti delle scuole superiori e le loro attitudini verso l'elaborazione concorrente. Eccles et al. [9] identificano la sincronizzazione come uno degli aspetti più difficili delle computazioni thread-based. In merito agli ausili per l'apprendimento, Slaviero & Haeusler [10] analizzano come la concorrenza sia gestita nei più diffusi ambienti di programmazione (Scratch, Alice, AgentSheets, NetLogo e Greenfoot) e il loro impatto sull'apprendimento. Numerosi educatori hanno sviluppato strumenti per visualizzare l'esecuzione di programmi concorrenti [11], altri hanno proposto di rendere "udibili" i problemi di sincronizzazione tramite linguaggi musicali [12].

Gli stili di apprendimento cognitivi, che si riferiscono ai diversi modi in cui le persone elaborano e memorizzano le informazioni, sono fondamentali per un'educazione efficace. Felder e Silverman [13] suggeriscono che allineare i metodi didattici con gli stili di apprendimento cognitivi può migliorare la comprensione e le prestazioni accademiche, classificando gli studenti per dimensioni come attivo/riflessivo e visivo/verbale. Tuttavia, i critici sostengono che le prove a supporto dell'efficacia di adattare i metodi di insegnamento agli stili di apprendimento siano inconcludenti [14]. Coffield et al. [15] hanno rilevato incoerenze nelle definizioni e misurazioni degli stili cognitivi, evidenziando le difficoltà pratiche nell'applicarli efficacemente nei contesti educativi. Hattie [16] critica anche il concetto di stile cognitivo, sottolineando che l'impatto dell'adattamento dei metodi di insegnamento agli stili cognitivi è minimo rispetto ad altri interventi educativi. Sebbene gli stili cognitivi possano influenzare le prestazioni, adattamento, esperienza e motivazione sono fattori altrettanto cruciali. I ricercatori raccomandano pratiche di insegnamento basate su evidenze che beneficino tutti gli studenti, piuttosto che adattare l'istruzione a specifici stili cognitivi, considerati inaffidabili come predittori delle prestazioni.

3 Strumento d'indagine

La struttura del test proposto segue il formato presentato nel precedente lavoro [2]², che include domande di comprensione dei programmi e valutazioni della fiducia delle risposte fornite.

Per valutare una possibile connessione tra le prestazioni degli studenti e i loro stili cognitivi, abbiamo somministrato un ulteriore questionario. Nel contesto nazionale, ci sono diversi questionari che esplorano gli stili cognitivi. Tra questi, abbiamo analizzato quelli più comunemente usati e citati: il questionario di Mariani [17] e il sondaggio di Cornoldi et al. [18].

Il test di Mariani è stato scelto in quanto a) gli item sono formulati in modo più concreto (con situazioni reali) e sono più facili da comprendere; b) non identifica solo gli stili cognitivi

²Struttura del test e ulteriori dettagli sono disponibili al link: http://nid.dimi.uniud.it/additional_material/thread_survey/thread_survey.pdf

“classici”, come fa Cornoldi, ma collega anche gli stili cognitivi agli stili di apprendimento; c) considera canali sensoriali (compreso lo stile cinestetico), modalità di elaborazione delle informazioni e metodi di lavoro; d) è più adatto per la ricerca in un contesto scolastico. Le motivazioni dell’autore per lo sviluppo del questionario, principalmente mirato ad aiutare gli studenti a prendere consapevolezza del loro approccio all’apprendimento, sono spiegate in [19].

In aggiunta, in relazione al Task 1.c e al Task 2, abbiamo chiesto agli studenti di fornire, su carta, una breve spiegazione dei passaggi seguiti per giungere al risultato del Task, includendo eventuali disegni o supporti grafici utilizzati a tale scopo. Questo approccio è conforme alla prospettiva di Seel et al. [20]. A tal proposito, Cunningham et al. evidenziano che “la ricerca sulle annotazioni grafiche può portare allo sviluppo di tecniche pedagogiche coinvolgenti e attive” [21].

Nella nostra indagine abbiamo selezionato alcune delle categorie di Cunningham et al.: A) Descrizioni verbali dell’evoluzione (in forma “narrativa”); B) Annotazioni relative ai punti richiesti nella specifica del compito (simili a sottolineature nel testo); C) Tracciamento focalizzato sulla sequenza cronologica delle azioni/istruzioni eseguite; D) Tracciamento focalizzato sul flusso di dati (ad esempio, successione di valori delle variabili, buffer, I/O); E) Allineamento dei flussi operativi di due thread concorrenti (percorsi paralleli, ordinati sulla base della storia); F) Elementi simbolici o grafici per rappresentare le circostanze di sincronizzazione (ad esempio, frecce che collegano le operazioni che si sincronizzano); G) Pagina bianca (niente scritto).

4 Risultati

Il test è stato somministrato a 78 studenti del quinto anno (età compresa tra 18 e 19 anni; 73 ragazzi e 5 ragazze³) frequentanti l’indirizzo di Informatica nel nostro istituto. Gli studenti, che avevano studiato programmazione concorrente nell’anno scolastico precedente, hanno affrontato il compito in una situazione controllata, sotto la supervisione dei loro insegnanti. Si prevedeva che completassero il test entro un’ora.

Le risposte al test sono state registrate e non potevano essere utilizzate per valutazioni formative o sommative. Lo stesso campione di studenti ha compilato anche il questionario sugli stili cognitivi, e i risultati di entrambi sono stati incrociati solo dagli insegnanti delle rispettive classi, evitando ostacoli legati alla normativa sulla privacy⁴.

I risultati dell’indagine confermano essenzialmente quanto emerso nel lavoro precedente, riguardo alla percezione degli studenti, alle loro prestazioni nei compiti e ai livelli di fiducia in sé stessi. Ancora una volta, il 77% degli studenti percepisce la *sincronizzazione* come difficile, il 74% trova difficile gestire gli *stati dei thread*. Ancora una volta, solo le soluzioni per i tasks 1a e 1b erano corrette per una larga maggioranza degli studenti, negli altri casi meno della metà ha avuto successo, con le prestazioni più scarse nei tasks apparentemente semplici 1c e 1d.

Indicazioni nuove emergono, invece, in relazione all’analisi comparativa degli stili cognitivi degli studenti. Gli stili cognitivi degli studenti sono distribuiti, con una prevalenza per gli stili uditivo (**28.8%**) e cinestetico (24.5%), rispetto a visuale non-verbale (visuo-spaziale) (23.6%) e visuale verbale (23.1%). Quanto osservato si discosta da alcuni risultati presenti in letteratura, secondo i quali uno stile visuo-spaziale sarebbe prevalente in ambito STEM [5, 6]. Inoltre, confrontando gli stili cognitivi degli studenti con quelli dei loro insegnanti, si nota che gli insegnanti presentano una distribuzione diversa degli stili: stile uditivo 23.5%, cinestetico 22.1%, visuale

³Troppo poche per considerare un’analisi di genere.

⁴In particolare, poiché nessuna informazione personale è stata condivisa con terze parti e tutte le informazioni sono state elaborate dagli insegnanti stessi, le politiche di ricerca italiane non richiedono l’approvazione da parte di una commissione etica.

Table 1: Correlazione tra le prestazioni complessive degli studenti e i loro punteggi "normalizzati" relativi a diversi stili cognitivi. Se p -value > 0.05 non è statisticamente significativa.

Stile cognitivo	correlazione	p -value
visuale-verbale	-0.024	0.832
visuale-non verbale	-0.023	0.838
uditivo	0.004	0.969
cinestetico	0.078	0.488
analitico	0.001	0.991
globale	0.020	0.862
individuale	0.306	0.006
“di gruppo”	-0.300	0.007

Table 2: Correlazione tra la prestazione degli studenti e la loro confidenza nelle risposte (misurata con una scala Likert 1–4). Se p -value > 0.05 non è statisticamente significativa.

Task	correlazione	p -value
Task 1 (number of correct answers to questions a–d)	0.278	0.0215
Task 2 (4=correct, 2=incorrect, 1=severely incorrect)	0.440	< 0.0002
Task 3 (4=correct, 3=partly correct, 1=incorrect)	0.150	0.2234
Task 4 (4=correct, 2=incorrect, 1=severely incorrect)	0.017	0.8917
Task 1–4 (average score vs. average self-confidence level)	0.365	0.0022

non-verbale (visuo-spaziale) **27.7%** e visuale verbale 26.6%. In particolare uno stile cognitivo visuale non verbale (visuo-spaziale) è molto diffuso fra gli insegnanti laureati in Informatica, con valori compresi tra il 30% e il 36%.

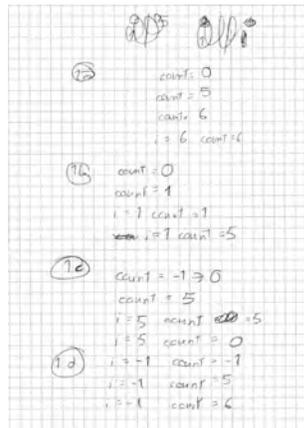
La Tabella 1 riporta le correlazioni tra le prestazioni e i punteggi normalizzati per diversi stili cognitivi. I punteggi sono normalizzati rispetto alla somma dei punteggi per ciascuna delle tre aree separate: visivo/uditivo/cinestetico, analitico/globale, individuale/gruppo. In altre parole, misuriamo l’equilibrio tra gli stili cognitivi relativi a una data area, come percepiti da ciascuno studente, piuttosto che i punteggi assoluti che potrebbero essere influenzati dalla loro propensione a assegnare punteggi più alti o più bassi in generale. Come si può vedere dalla tabella, le correlazioni sono essenzialmente inesistenti, con l’unica eccezione della dicotomia individuale/gruppo, dove le prestazioni correlano debolmente con lo stile cognitivo individuale e in modo simile, ma negativamente, con quello di gruppo — cosa non sorprendente dato che gli studenti dovevano impegnarsi individualmente nei compiti proposti dal test.

La distribuzione degli stili cognitivi per gli studenti che hanno risposto correttamente ai compiti è risultata proporzionale alla distribuzione generale sopra esposta; non evidenziando correlazioni tra performance e stili.

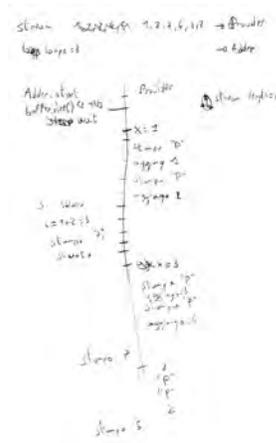
Per approfondire le relazioni tra la qualità delle risposte degli studenti e i livelli percepiti di fiducia in sé stessi, abbiamo valutato le prestazioni sul compito 1 in base al numero di soluzioni corrette alle domande a–d, e le prestazioni sugli altri compiti distinguendo tre livelli di qualità: 1 (= gravemente incorretto), 2 (= incorretto) e 4 (= corretto) per i compiti 2 e 4; 1 (= incorretto), 3 (= parzialmente corretto) e 4 (= completamente corretto) per il compito 3. Correlazioni e relativa significatività statistica sono riassunte nella Tabella 2. Analizzando le categorie di annotazioni, catalogate come descritto nella sezione 3, abbiamo ottenuto i risultati presentati nella Tabella 3, che suggeriscono ulteriori approfondimenti. Gli studenti che hanno utilizzato le categorie di schizzi D, E e F (vedi Figura 1) hanno ottenuto prestazioni medie migliori nei

Table 3: Categorie di Cunningham in relazione con gli stili cognitivi e prestazioni.

Categoria	%	punti	visuale-verbale	visuale-non verbale	individuale	di gruppo
A	17.9	1.85	25.5	22.6	44.8	55.1
B	6.4	2.13	26.9	22.8	49.6	50.4
C	14.1	1.70	22.9	24.7	45.0	55.0
D	16.7	2.60	22.7	23.1	49.4	50.6
E	9.0	2.38	23.7	23.0	43.1	56.9
F	3.8	2.50	17.5	27.0	28.5	71.5
G	32.1	2.08	21.6	23.0	46.5	50.7



(a)



(b)

Figure 1: Categories D (a) and E (b) sketch examples.

compiti. Inoltre, coloro che hanno utilizzato schemi della categoria F hanno dimostrato una chiara preferenza per il lavoro “in gruppo” rispetto al lavoro individuale, e una preferenza per lo stile visuale-non verbale.

Lo studente che ha presentato lo schizzo di Figura 1.a ha risposto correttamente ai task 1.a, 1.b, 2, 4, con stile cognitivo prevalente quello uditivo (37.0%). Mentre lo studente che ha presentato lo schizzo di Figura 1.b ha risposto correttamente ai task 1.a, 1.b, 2, anch’esso con stile cognitivo prevalente quello uditivo (35.0%).

Ancora una volta, gli studenti ritengono che una rappresentazione grafica potrebbe essere efficace per migliorare la loro comprensione della programmazione concorrente (74%), mentre gli strumenti generalmente considerati più utili per la comprensione della programmazione concorrente ricevono poca considerazione: UML (18% di valutazioni positive) e il grafico di attesa – Wait-for graph (8%). Nella risposta aperta finale sono state presentate diverse proposte, ma quella che meglio rappresenta e riassume l’idea generale è quella dello studente che ha dichiarato: “Proporrei di aggiungere più dimostrazioni grafiche, consentendo la visualizzazione concreta del ciclo di vita dei thread, ma anche delle priorità imposte tra di essi”.

5 Discussione

I risultati preliminari indicano che nonostante gli studenti mostrino una leggera preferenza per gli stili di apprendimento cinestetico e uditivo, queste preferenze non correlano significativamente con le loro prestazioni o la fiducia in sé stessi nei compiti di programmazione concorrente.

rente. Inoltre, le rappresentazioni schematiche create dagli studenti evidenziano le difficoltà che incontrano con la sincronizzazione, suggerendo che un supporto didattico mirato sia cruciale.

Riguardo le domande di ricerca sollevate nell'introduzione:

Q1 – In che misura le prestazioni degli studenti e la percezione della fiducia in se stessi nei compiti di programmazione concorrente correlano con una prevalenza di uno stile cognitivo visivo-spaziale? I risultati indicano chiaramente che non esiste alcuna correlazione tra prestazioni/autostima e stili cognitivi degli studenti. Come mostrato nella Tabella 1, non emergono correlazioni significative. Questi risultati supportano lo scetticismo di alcuni autori [14, 16] e contraddicono le conclusioni di altri studi [5, 6].

Q2 – Che tipo di annotazioni o schizzi utilizzano spontaneamente gli studenti mentre affrontano semplici compiti di programmazione concorrente? I risultati mostrano che gli studenti preferiscono descrizioni verbali, tracciamento del flusso dei dati e tracciamento cronologico delle azioni. Tuttavia, le descrizioni verbali risultano scarsamente efficaci ai fini delle prestazioni. Il tracciamento, comune nei corsi introduttivi, è ampiamente utilizzato poiché gli studenti non conoscono strumenti alternativi per la programmazione concorrente e la sincronizzazione [11, 22].

Benché la nostra indagine non si fosse postata l'obiettivo di esplorare questo aspetto, abbiamo osservato che gli studenti con migliori prestazioni in Informatica e Matematica mostrano una combinazione più alta di stili cognitivi visivo-verbale e visivo-non verbale rispetto agli altri. Ulteriori indagini sono necessarie per determinare se gli stili cognitivi influenzino l'apprendimento piuttosto che le prestazioni. Sebbene non ci sia una correlazione tra stili cognitivi e prestazioni/autostima, gli studenti desiderano ausili grafici per comprendere meglio la programmazione concorrente. È quindi consigliabile che gli insegnanti utilizzino rappresentazioni grafiche [23] per chiarire concetti come la sincronizzazione e l'evoluzione dei thread. Tuttavia, ulteriori indagini sono necessarie per determinare il modello grafico più efficace in questo contesto.

6 Conclusioni

In questo lavoro, abbiamo presentato i risultati di un'indagine esplorativa tramite test e questionario, concentrata su percezioni, competenze e fiducia degli studenti delle scuole superiori nella programmazione concorrente in relazione agli stili cognitivi. Le implicazioni principali richiedono un equilibrio tra l'elevata sfida cognitiva dell'argomento [1] e l'identificazione di strumenti efficaci per facilitare la comprensione. Sarebbe interessante verificare se strategie basate sullo stile cognitivo visivo-non verbale possano migliorare le capacità degli studenti. Ulteriori ricerche potrebbero valutare e confrontare l'efficacia di diversi strumenti grafici per migliorare la comprensione dei concetti legati ai thread.

References

- [1] P. Brodanac, J. Novak, and I. Boljat. Has the time come to teach parallel programming to secondary school students? *Heliyon*, 8(1):e08662, January 2022.
- [2] E. Scapin, N. Dalla Pozza, and C. Mirolo. An exploratory investigation on high-school students' understanding of threads, 2023.
- [3] J. Sweller, P. Ayres, and S. Kalyuga. *Cognitive Load Theory*. Explorations in the Learning Sciences, Instructional Systems and Performance Technologies. Springer New York, 2011.

- [4] R.E. Mayer. *Multimedia Learning*. Cambridge University Press, 2 edition, 2009.
- [5] J. Buckley, N. Seery, and D. Canty. Spatial cognition in engineering education: Developing a spatial ability framework to support the translation of theory into practice. *European Journal of Engineering Education*, 44(1-2):164–178, 2019.
- [6] K. Atit, J.R. Power, N. Veurink, D.H. Uttal, S. Sorby, G. Panther, C. Msall, L. Fiorella, and M. Carr. Examining the role of spatial skills and mathematics motivation on middle school mathematics achievement. *International Journal of STEM Education*, 7(1):1–13, 2020.
- [7] S. Torbert, U. Vishkin, R. Tzur, and D.J. Ellison. Is teaching parallel algorithmic thinking to high school students possible? One teacher’s experience. In *Proc. of the 41st ACM Technical Symposium on Computer Science Education*, pages 290–294, New York, USA, 2010. ACM.
- [8] M. Ben-Ari and Y. Ben-David Kolikant. Thinking parallel: The process of learning concurrency. *SIGCSE Bull.*, 31(3):13–16, June 1999.
- [9] R. Eccles, B. Nonneck, and D.A. Stacey. Exploring parallel programming knowledge in the novice. In *19th International Symposium on High Performance Computing Systems and Applications (HPCS’05)*, pages 97–102, 2005.
- [10] C. Slaviero and E. Hermann Haeusler. Computational thinking tools: Analyzing concurrency and its representations. *SBC Journal on Interactive Systems*, 9(1):40–52, 2018.
- [11] J.C. Adams, P.A. Crain, C.P. Dilley, C.D. Hazlett, S.M. Koning, E.R. and Nelesen, J.B. Unger, and M.B. Vande Stel. Tsgl: A tool for visualizing multithreaded behavior. *Journal of Parallel and Distributed Computing*, 118:233–246, 2018.
- [12] G. Delzanno, G. Guerrini, and D. Traversaro. Exploring student misconceptions about concurrency using the domain-specific programming language "sonic pi". In *Proc. of the 55th ACM Technical Symposium on Computer Science Education V. 2, SIGCSE 2024*, pages 1622–1623, New York, USA, 2024. Association for Computing Machinery.
- [13] R.M. Felder and L.K. Silverman. Learning and teaching styles in engineering education. *Journal of Engineering Education - Washington-*, 78:674–681, 01 1988.
- [14] H. Pashler, M.A. McDaniel, D. Rohrer, and R.A. Bjork. Learning styles. *Psychological Science in the Public Interest*, 9:105 – 119, 2008.
- [15] F. Coffield, K. Ecclestone, D. Moseley, and E. Hall. *Learning styles and pedagogy in post 16 education: a critical and systematic review*. Learning and Skills Research Centre, 01 2004.
- [16] J. Hattie. *Visible learning for teachers: Maximizing impact on learning*. Routledge, 2012.
- [17] L. Mariani. *Portfolio. Strumenti per documentare e valutare cosa si impara e come si impara*. Zanichelli, Bologna, Italy, 09 2000.
- [18] C. Cornoldi and R. De Beni. *Imparare a studiare 2. Strategie, stili cognitivi, metacognizione e atteggiamenti nello studio*. Materiali per l’educazione. Centro Studi Erickson, 2001.
- [19] L. Mariani. Investigating learning styles. *Perspectives*, XXI(2)/XXII(1):35–49, Spring 1996.
- [20] N.M. Seel, D. Ifenthaler, and P. Pirnay-Dummer. *Mental Models and Problem Solving: Technological Solutions for Measurement and Assessment of the Development of Expertise*, pages 17–40. Brill, Leiden, The Netherlands, 2009.
- [21] K. Cunningham, S. Blanchard, B. Ericson, and M. Guzdial. Using tracing and sketching to solve programming problems: Replicating and extending an analysis of what students draw. In *Proc. of the 2017 ACM Conference on International Computing Education Research, ICER ’17*, pages 164–172, New York, USA, 2017. ACM.
- [22] F. Strömbäck, L. Mannila, and M. Kamkar. Pilot study of progvis: A visualization tool for object graphs and concurrency via shared memory. In *Proc. of the 24th Australasian Computing Education Conference, ACE ’22*, pages 123–132, New York, USA, 2022. ACM.
- [23] K.B. Bruce, A. Danyluk, and T. Murtagh. Introducing concurrency in CS1. In *Proc. of the 41st ACM Technical Symposium on Computer Science Education, SIGCSE ’10*, pages 224–228, New York, USA, 2010. ACM.

Le analogie visive nel processo di insegnamento e apprendimento di Sistemi e Reti

Virginia Beato¹, Luca Pinet²

¹ Istituzione scolastica di Istruzione Liceale, Tecnica e Professionale (I.S.I.L.T.P.)
v.beato@mail.scuole.vda.it

² Istituzione scolastica di Istruzione Liceale, Tecnica e Professionale (I.S.I.L.T.P.)
l.pinet@mail.scuole.vda.it

Abstract

Le scienze e le tecnologie informatiche sono note per essere discipline astratte e talvolta complesse per gli studenti. Risultano quindi particolarmente importanti le analogie visive per la comprensione dei contenuti, che favoriscono l'apprendimento degli stessi per l'intero gruppo classe rispetto a metodi più tradizionali, oltre a favorire l'inclusione di alunni con bisogni educativi speciali.

1 Analogie visive

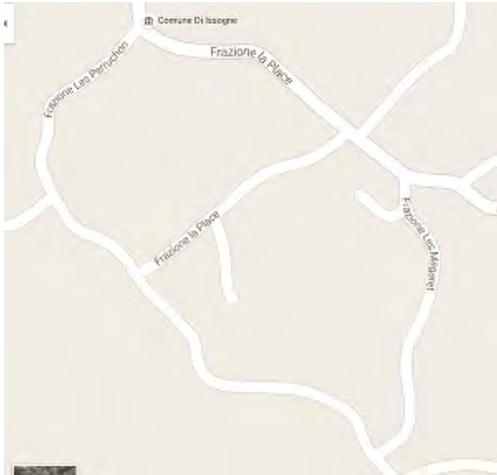
Un'analogia visiva si crea identificando somiglianze tra due concetti. Grazie alle analogie visive, le idee possono essere trasferite da un concetto familiare ad uno meno noto. Il concetto familiare è chiamato analogo, mentre quello meno noto è il target. Sia l'analogo che il target hanno delle caratteristiche, chiamate attributi. Se analogo e target condividono attributi comuni o simili, è possibile stabilire un'analogia tra di essi.

Le analogie rendono il processo di insegnamento motivante per gli studenti e portano ad un ambiente di apprendimento attivo. Gli insegnanti le utilizzano spesso per favorire una migliore comprensione della teoria e per rendere il materiale didattico più interessante.

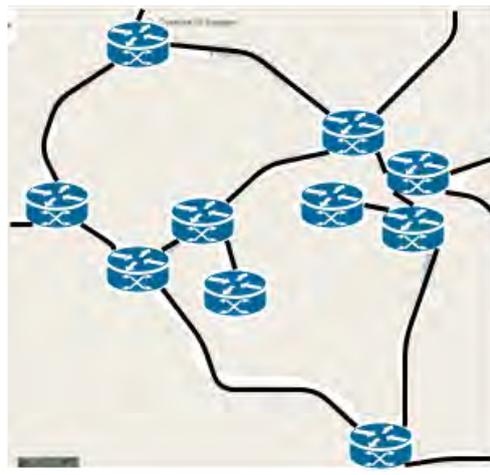
Un ulteriore vantaggio consiste nella capacità degli studenti di demistificare i concetti precedentemente ritenuti ostici. Inoltre, si osserva un incremento della profondità di ragionamento, con un maggior numero di studenti in grado di seguire la discussione tecnica e alcuni di essi in grado di proporre scenari e spiegazioni che denotano una comprensione più approfondita. In aggiunta, l'utilizzo di analogie ha portato ad un aumento dell'efficienza delle lezioni.

2 Il livello di rete: il routing e il forwarding

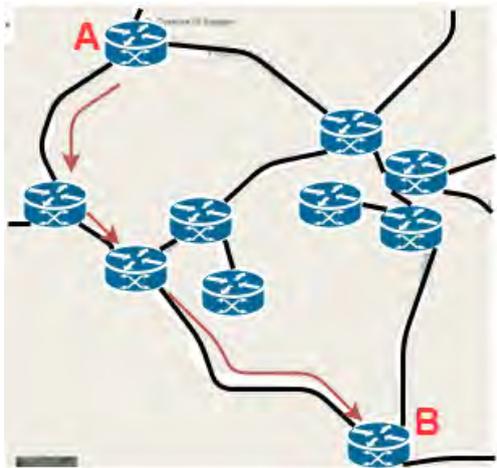
Per permettere agli studenti di comprendere il concetto di rete, viene inserita l'analogia visiva della strada: i tratti di strada sono i cavi, gli incroci sono i router, le macchine sono i pacchetti inviati sulla rete, i segnali stradali sono le tabelle di routing.



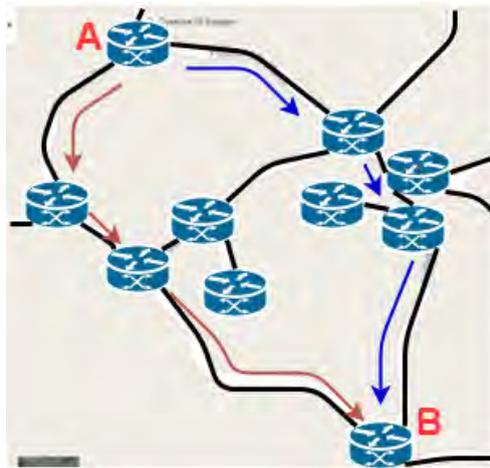
L'immagine riporta le strade di una frazione di Issogne



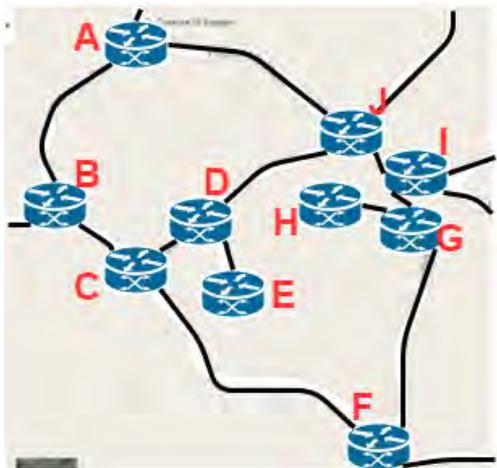
L'immagine riporta un'analogia tra le strade dell'immagine precedente e una rete.



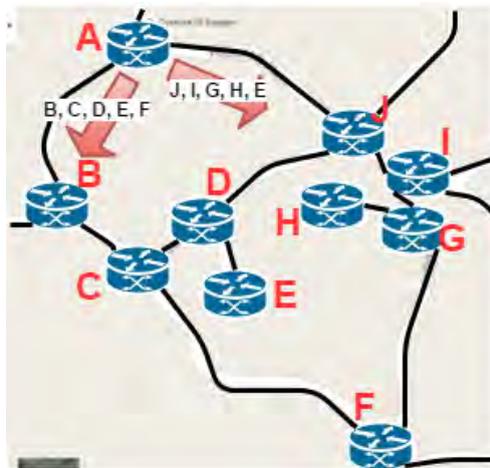
L'immagine evidenzia un possibile percorso dal router A al router B.



L'immagine evidenzia un possibile percorso alternativo dal router A al router B



L'immagine riporta uno schema logico della rete



L'immagine riporta l'esempio di due segnali stradali: la tabella di instradamento

Per chiarire i concetti di routing e forwarding è stata utilizzata una semplice analogia visiva, che sfrutta le caratteristiche del territorio in cui vivono gli studenti della classe in esame. La Valle d'Aosta è composta da numerose vallate, e la principale è quella che collega il paese di Pont-Saint-Martin, il primo paese che si incontra entrando dal Piemonte, con il capoluogo di regione, ovvero Aosta. Se una persona vuole partire da Pont-Saint-Martin per raggiungere Aosta con una vettura, può scegliere due strade: la strada statale o l'autostrada.

Considerando solo il tempo di percorrenza, la scelta dell'autostrada sembrerebbe la più conveniente per raggiungere Aosta, poiché consente di completare il viaggio in soli 35 minuti, rispetto ai circa 65 minuti necessari se si percorre la strada statale. Tuttavia, va considerato anche il costo del pedaggio autostradale, che ammonta a circa 10 euro. Tale cifra potrebbe essere ritenuta eccessivamente alta, specialmente considerando che la distanza totale del tragitto non supera i 50 chilometri. Tutte queste informazioni sono ben chiare agli studenti, i quali sono prossimi alla maggiore età, quindi al conseguimento della patente di guida, e conoscono bene la conformità del territorio in cui vivono.

Alla classe viene quindi posta la domanda: "Se dovete andare da Pont-Saint-Martin ad Aosta, prendete la strada statale o l'autostrada?". Gli studenti cominciano quindi a confrontarsi, individuando diverse strategie in base alla disponibilità di tempo e di denaro.

Dopo aver terminato questa discussione con gli studenti, si evidenzia che questo scenario rappresenta il primo compito del livello di rete: il routing. Infatti, un algoritmo di routing deve selezionare la strada ottimale per connettere mittente e destinatario, considerando diversi criteri, come i costi espressi in termini di tempo, denaro, lunghezza del tragitto o altri fattori pertinenti.

Terminata l'introduzione dell'operazione di routing, si passa alla seconda analogia. Si pone quindi alla classe la seguente domanda: "Partendo da Verrès e volendo raggiungere Aosta attraverso la strada statale, quale percorso fareste?". Gli studenti iniziano quindi a discutere su quale uscita prendere ad ogni rotonda e dove svoltare ad ogni incrocio. Dopo un momento di discussione, viene introdotta la seconda funzionalità del livello di rete, il forwarding, definito come l'operazione locale eseguita da un router per determinare su quale interfaccia inoltrare un pacchetto.

Si noti che Verrès è il paese dove è situata la scuola, ed è stato utilizzato questo paese come punto di partenza perché è più familiare agli studenti, che percorrono quelle strade ogni giorno.

Le due analogie visive si sono dimostrate strumenti molto efficaci per la comprensione delle due funzionalità del livello di rete. Anche in base alle esperienze degli anni precedenti, è emerso che gli studenti, grazie a queste analogie visive, sono stati in grado di comprendere e assimilare più rapidamente questi due concetti.

3 Il motivo di più sottolivelli MAC del livello Data Link

Per fornire una spiegazione più dettagliata sulla motivazione dell'utilizzo di più sottolivelli MAC, si è ricorso a una semplice analogia visiva. Si immagina una grande multinazionale con sedi in diversi stati, in ognuno dei quali si parla una lingua diversa. Talvolta, è necessario che i dirigenti delle diverse sedi comunichino tra di loro, ad esempio per coordinare lo sviluppo di un progetto in più sedi. Tuttavia, a causa delle diverse lingue parlate dai dirigenti, la barriera linguistica impedisce una comunicazione fluida tra le varie componenti. Ad esempio, il dirigente cinese parla cinese, mentre il dirigente italiano parla italiano: senza un mezzo comune, i due non possono comunicare.

In questo contesto, entra in gioco l'inglese, utilizzato come lingua internazionale per comunicare con le diverse sedi. Quindi, ciascun dirigente pensa a ciò che deve dire nella sua lingua madre, ma comunica in inglese con gli altri partecipanti. Questo rappresenta esattamente il ruolo dei sottolivelli MAC: raccolgono le informazioni provenienti dai diversi mezzi trasmissivi, che possono essere segnali elettrici con Ethernet, segnali luminosi con la fibra ottica o onde radio

con il WiFi, e traducono questi segnali completamente diversi in bit, la lingua internazionale utilizzata per comunicare all'interno dei computer.

4 Il processo di incapsulamento e decapsulamento

Per chiarire agli studenti i concetti di incapsulamento e decapsulamento sono state utilizzate delle buste di carta di dimensioni diverse, ciascuna corrispondente ad un header diverso dello stack TCP/IP.

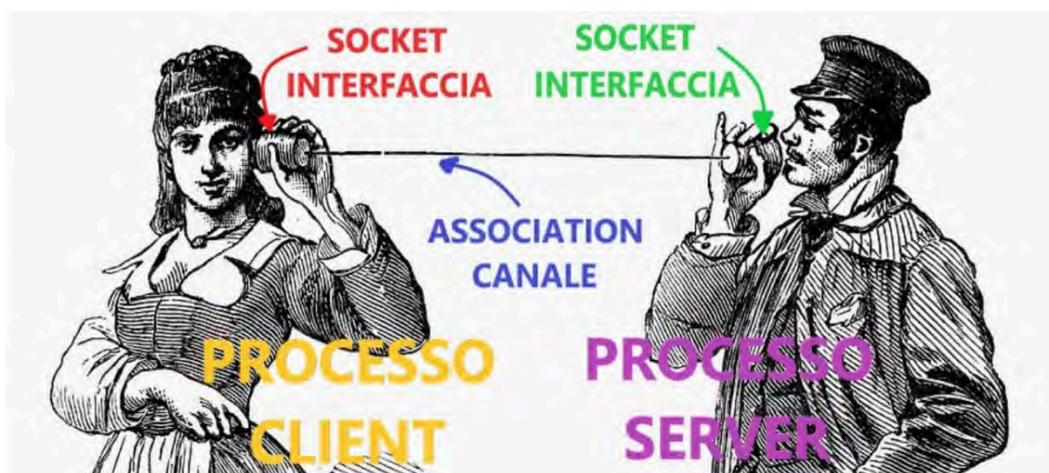
Inizialmente, il mittente compone il messaggio da inviare e simula l'incapsulamento inserendo il messaggio nella busta che rappresenta l'header del livello di applicazione. Successivamente, colloca tale busta all'interno di quella che identifica l'header del livello di trasporto, e così via. Sopra ogni busta vengono inserite le informazioni che verrebbero aggiunte dal relativo livello. Ad esempio, all'interno dell'header del livello di collegamento vengono aggiunti gli indirizzi MAC mittente e destinatario. Analogamente, sopra la busta che rappresenta l'header del livello di rete vengono aggiunti gli indirizzi IP mittente e destinatario.

Raggiunto l'header del livello di collegamento, il frame può essere inserito nell'ultima busta, corrispondente al livello fisico. È stato chiarito agli studenti che il livello fisico non aggiunge informazioni, bensì si occupa di trasformare i bit in un segnale che può essere trasmesso nel mezzo trasmissivo appropriato, in questo caso, un cavo Ethernet. La busta associata al livello fisico rappresenta quindi la conversione dei bit in un segnale elettrico, idoneo per la trasmissione sul cavo.

Per simulare il decapsulamento viene eseguita l'operazione inversa: il destinatario scarta una busta per volta, e ad ogni iterazione analizza il contenuto dell'header che sta eliminando.

5 Il significato di socket

Per chiarire meglio il concetto di socket, ovvero l'insieme di indirizzo IP e porta logica associati ad un processo, che rappresenta l'interfaccia del processo stesso, e per chiarire anche il concetto di association (o quintupla), ovvero l'insieme dei socket del mittente e del destinatario e il protocollo utilizzato a livello di trasporto, viene adottata come analogia visiva quella del telefono meccanico, meglio conosciuto come telefono a barattolo oppure telefono a spago.



Bibliografia

- [1] Tesi del Corso di Laurea magistrale in Informatica “*Metodologie didattiche innovative per l'insegnamento di Reti: un'esperienza in classe*” di Beato Virginia, presso Università degli studi di Torino (anno accademico 2023/2024)
- [2] Shawn M. Glynn. «*Teaching Science With Analogy: A Strategy for Teachers and Textbook Authors*». In: *Proceedings of the Learning Science in the Schools: Research Reforming Practice* (1994), pp. 247–273. url: <https://eric.ed.gov/?id=ED373306>
- [3] Allan G. Harrison e David F. Treagust. «*Teaching and Learning with Analogies*». In: *Metaphor and Analogy in Science Education*. A cura di Peter J. Aebischer, Allan G. Harrison e Stephen M. Ritchie. Dordrecht: Springer Netherlands, 2006, pp. 11–24. isbn: 978-1-4020-3830-3. doi: 10.1007/1-4020-3830-5_2
- [4] Lin Tang. «*Study on Teaching Methods of Computer Network Course Based on Analogy and Problem-based Approach*». In: *DEStech Transactions on Social Science, Education and Human Science* (nov. 2017). doi: 10.12783/dtssehs/iced2017/15137
- [5] Sridhar Iyer e Sahana Murthy. «*Demystifying Networking: Teaching Non-Majors via Analogical Problem-Solving*». In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (2013), pp. 77–82. doi: 10.1145/2445196.2445223
- [6] Anders Berglund. «*What is good teaching of computer networks?*» In: vol. 3. Dic. 2003, S2D–13. isbn: 0-7803-7961-6. doi: 10.1109/FIE.2003.1265954
- [7] Ilkka Niiniluoto. «*ANALOGY AND SIMILARITY IN SCIENTIFIC REASONING*». In: 1988. url: <https://api.semanticscholar.org/CorpusID:118910194>

LABORATORI PER LA PRIMARIA

Mostri, pesci e virus: un laboratorio per la comprensione dell'apprendimento automatico.

Matteo Baldoni¹, Cristina Baroglio¹, Monica Bucciarelli²,
Sara Capecchi^{1,3}, Elena Gandolfi², Francesco Iani², Elisa Marengo¹,
Roberto Micalizio¹

¹Università di Torino, Dipartimento di Informatica

²Università di Torino, Dipartimento di Psicologia

³Laboratorio Informatica e Scuola CINI

nome.cognome@unito.it

In questo laboratorio presentiamo delle attività frutto di una collaborazione tra psicologi e informatici. Lo studio nell'ambito del quale sono state progettate le attività fa parte del progetto di ricerca T3-AI-LEAP (<http://www.ai-leap.it/>) che affronta il problema di come supportare la comprensione dei concetti di apprendimento automatico da parte dei bambini attraverso l'identificazione e l'addestramento di strategie di ragionamento e attraverso l'identificazione e l'uso di compiti che richiedono la conoscenza di questi concetti.

Destinatari

Scuola primaria e secondaria di primo grado.

Obiettivi di apprendimento

Comprensione dei concetti di apprendimento automatico attraverso i) l'identificazione e l'addestramento di strategie di ragionamento e ii) l'identificazione e l'uso di compiti che richiedono la conoscenza di questi concetti.

Attività svolte

Utilizziamo un approccio unplugged, che non prevede l'utilizzo di strumenti e quindi ha l'obiettivo più ampio di addestrare non solo il pensiero funzionale per la comprensione dei concetti di apprendimento automatico, ma anche ragionamento e di prendere decisioni in contesti quotidiani. In primo luogo, abbiamo identificato tre strategie di riflessione per migliorare la comprensione degli algoritmi di apprendimento automatico: i) prendersi il tempo necessario per pensare senza affidarsi al pensiero rapido, ii) pensare ai casi più rari, iii) applicare un pensiero insolito, cioè ragionare sulla risposta corretta senza farsi distrarre da stereotipi o modelli. Per addestrarli, abbiamo sviluppato tre attività unplugged che possono essere svolte in classe. Per valutare l'effetto di queste attività, abbiamo sviluppato una serie di esercizi che valutano la comprensione dell'apprendimento automatico. Successivamente, abbiamo identificato quattro categorie di compiti che costituiscono il nucleo dei compiti di apprendimento automatico (identificare il miglior modello da un insieme dato, costruzione di un modello applicando alcune regole di generalizzazione, associare istanze a modelli, identificare le caratteristiche significative di un modello) e per ognuna di esse abbiamo sviluppato una serie di esercizi. Durante il laboratorio illustreremo alcuni esempi delle attività sopra descritte.

Ilognatter Angesid: un laboratorio sulla rappresentazione delle informazioni tramite codifiche digitali

Carlo Bellettini, Marta Crippa, Sofia Gavanelli, Luca Lamanna, Violetta Lonati, Lucia Mellini, Mattia Monga, and Anna Morpurgo

Università degli Studi di Milano
aladdin.info@di.unimi.it

Il laboratorio Ilognatter Angesid¹ si colloca nell'area "Dati e informazione" della Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella scuola dell'obbligo² e in particolare affronta il tema della rappresentazione delle informazioni tramite codifiche digitali.

Obiettivi di apprendimento L'attività si presta a promuovere vari traguardi e obiettivi di apprendimento previsti nella suddetta proposta quali: esplorare la possibilità di rappresentare dati di varia natura mediante formati diversi, anche arbitrariamente scelti; definire l'interpretazione degli oggetti utilizzati per rappresentare l'informazione (legenda); utilizzare combinazioni di simboli per rappresentare informazioni familiari complesse o informazioni semplici strutturate.

Durata e destinatari L'attività ha una durata compresa tra i 60 ed i 90 minuti e può essere adattata a vari livelli scolastici cambiando il sistema di codifica. In questa descrizione facciamo riferimento alla versione pensata per alunni/e della scuola primaria.

Attività I/le partecipanti devono scoprire come funziona un *sistema di codifica*, partendo da esempi. Nello specifico, il sistema di codifica permette di dare una descrizione simbolica di una immagine composta da riquadri colorati e ai partecipanti si chiede di scoprire ed esplicitare le regole di codifica, ovvero il modo in cui vengono usati i simboli per rappresentare l'immagine. L'attività si sviluppa secondo la metodologia didattica chiamata *algomotricità*³, sviluppata dal gruppo ALaDDIN, che applica strategie costruttiviste ad argomenti informatici e permette ai partecipanti di esplorare il tema indagandolo in prima persona, costruendo modelli interpretativi e facendo ipotesi che possano essere messe alla prova nel contesto guidato dell'attività. Uno strumento *software* sviluppato *ad hoc* permette di fare esperimenti e osservazioni applicando il metodo scientifico per scoprire come sono rappresentate simbolicamente le informazioni rilevanti. Naturalmente l'obiettivo non è imparare questo specifico sistema di codifica, ma comprendere come sia possibile rappresentare informazioni anche complesse attraverso l'uso di simboli e regole precise applicate rigorosamente.

Al termine del laboratorio verrà condotta una riflessione dialogata sui temi informatici pertinenti l'attività stessa e alcuni aspetti didattici significativi.

Strumentazione e materiali Ogni partecipante dovrà avere a disposizione un dispositivo elettronico (PC o tablet). Inoltre si useranno i seguenti materiali: post-it, scheda didattica predisposta, matite/penne, lavagna con pennarelli (ove possibile).

¹L'attività è descritta in dettaglio, col titolo "Rettangoli", in C. Bellettini, V. Lonati, D. Malchiodi M. Monga, A. Morpurgo, "Algomotricità: manipolare i fondamenti dell'informatica", capitolo del libro "Coding e oltre: l'informatica nella scuola" (a cura di Enrico Nardelli, ed. Lisciani Scuola, 2020).

²<https://www.consortio-cini.it/index.php/it/component/attachments/download/745>

³<https://aladdin.di.unimi.it/algomotricita.html>

Le grandi idee della crittografia alla scuola primaria

Maria Cristina Carrisi^{1,3}, Michael Lodi^{2,3}, Simone Martini^{2,3}

¹ Dipartimento di Matematica e Informatica
Università degli Studi di Cagliari
mariacri.carrisi@unica.it

² Dipartimento di Informatica - Scienza e Ingegneria
Alma Mater Studiorum - Università di Bologna
{michael.lodi, simone.martini}@unibo.it

³ Laboratorio CINI Informatica e Scuola

Destinatari Insegnanti di Scuola Primaria (ma anche insegnanti di Scuola Secondaria di Primo Grado)

Obiettivi di apprendimento L'obiettivo di questo laboratorio è presentare agli insegnanti di scuola primaria (e secondaria di primo grado) alcune attività, già sperimentate in classe, sulle grandi idee della crittografia classica e moderna.

Si propone un percorso formativo che consenta di introdurre un nuovo crittosistema attraverso l'analisi delle limitazioni del precedente e, per questo motivo si, utilizzeranno anche semplici tecniche di crittoanalisi.

Mettendosi nei panni degli studenti, gli insegnanti sperimenteranno le attività ma anche le difficoltà nella realizzazione e nella progettazione, facilitando l'eventuale adozione delle attività.

Gli insegnanti potranno anche riflettere su quali aspetti tenere in considerazione nella scelta del percorso e dei cifrari adottati.

Strumentazione eventualmente necessaria

- Tempere di 6 colori diversi
- 6 Bottiglie di plastica da 1,5 o 2 litri per miscelare le tempere con acqua
- Bicchieri di plastica (anche da caffè, meglio se rigati per dosare più facilmente)
- Pennarelli neri punta fine
- Scatoloni di cartone larghi almeno 40 cm (anche senza coperchio, un lato deve comunque restare aperto)
- Scotch di carta (o per pacchi, o nastro adesivo o un qualsiasi nastro adesivo colorato)
- Fogli bianchi

Attività svolte Il laboratorio durerà 3 ore (due sessioni da 1h30). Due degli autori fungeranno da facilitatori.

Gli insegnanti lavoreranno in gruppo, replicando quanto si è svolto in classe durante la sperimentazione: attività unplugged per imparare concetti di crittografia e matematica, come il cifrario di Cesare e la ciclicità del resto, le reti a sostituzione e permutazione e lo scambio di chiavi di Diffie-Hellman.

Le attività, quasi esclusivamente unplugged, vogliono massimizzare il coinvolgimento, anche fisico, degli studenti.

Si utilizzeranno alcuni ambienti personalizzati e sviluppati in Snap!, sia per ottimizzare i tempi in alcune parti dell'attività, sia per mostrare una formalizzazione rigorosa (ed effettivamente eseguibile) di alcuni algoritmi crittografici sperimentati.

È prevista una fase finale di riflessione (circa 30 min) e discussione.

LABORATORI PER LA SECONDARIA DI PRIMO GRADO

Trovare il lupo: cassetta degli attrezzi per insegnare il debugging a scuola

Gabriele Pozzan, Tullio Vardanega

Università degli studi di Padova
gabriele.pozzan@phd.unipd.it
tullio.vardanega@unipd.it

1 Informazioni

Il laboratorio è costruito intorno al classico approccio sistematico all'individuazione e correzione di errori di programmazione (*debugging*) chiamato *Wolf Fence Algorithm*. Vuole dare la possibilità di sperimentare concretamente i suoi benefici tramite attività che riproducono la metafora della "ricerca del lupo" per poi approfondirne le componenti.

È rivolto a insegnanti della scuola secondaria di primo grado ma può essere adatto anche a livello di scuola primaria e secondaria di secondo grado. Non richiede competenze pregresse sul tema né dispositivi specifici.

Gli obiettivi di apprendimento principali sono:

- Ragionare su come gli errori di programmazione possono essere individuati e corretti, sull'insieme di competenze necessarie e sulle complessità del loro apprendimento.
- Conoscere il *Wolf Fence Algorithm* come approccio sistematico alla correzione di errori e sperimentare alcune attività mirate al suo insegnamento.

2 Attività

2.1 Parte 1 - Alla ricerca del lupo (30 minuti)

Questa parte del laboratorio introduce il *debugging* e la gerarchia di abilità che lo costituiscono. Il primo obiettivo è ragionare sull'individuazione e correzione di errori come competenze scomponibili in sottocompetenze che possono essere affrontate separatamente in ambito educativo. Il secondo obiettivo è presentare il *Wolf Fence Algorithm* e sperimentarlo tramite un'attività in cui, letteralmente, si "cerca il lupo" su una mappa digitale.

2.2 Parte 2 - Ipotesi e ululati (30 minuti)

Questa parte del laboratorio è dedicata al tema della ricerca degli errori (ascoltare gli ululati del lupo). Vengono affrontati e discussi esercizi volti ad allenare le competenze necessarie per ipotizzare e in seguito verificare la presenza di un errore.

2.3 Parte 3 - Test e barriere (30 minuti)

Questa parte del laboratorio è dedicata a strumenti utili per valutare la correttezza di sottoprogrammi (costruire barriere per isolare il lupo). Viene affrontato e discusso un esercizio dedicato alla creazione di test automatici per mettere alla prova la correttezza dell'implementazione di un algoritmo.

Laboratorio di Intelligenza Artificiale con la Programmazione a Blocchi e la Robotica Educativa

Ersilia Pagano
IIS Nitti Portici
ersilia.pagano@gmail.com

1 Destinatari:

Docenti della Secondaria di Primo Grado e del Biennio della Secondaria di Secondo Grado

2 Obiettivi di Apprendimento

Il laboratorio si propone di fornire ai docenti le competenze necessarie per integrare l'Intelligenza Artificiale (AI) e la Robotica nelle loro attività didattiche. Gli obiettivi specifici includono:

- Comprendere i fondamenti dell'Intelligenza Artificiale e la sua applicazione nell'istruzione.
- Apprendere i concetti base della programmazione a blocchi e come utilizzarli per programmare robot educativi.
- Sviluppare competenze pratiche nell'integrazione di AI e Robotica in attività didattiche.
- Esplorare le implicazioni etiche dell'AI e promuovere una cittadinanza digitale responsabile.
- Migliorare le competenze trasversali come Creatività, Pensiero Critico, Collaborazione e Comunicazione.

3 Strumentazione Necessaria:

È richiesto l'uso di PC o tablet per ogni partecipante.

4 Attività Svolte:

Il laboratorio prevede una serie di attività pratiche che gli insegnanti svolgeranno in prima persona, eventualmente in gruppi, sotto la guida del docente. Le attività includono:

- Rilevamento delle emozioni e riconoscimento delle immagini utilizzando AI.
- Programmazione di robot interattivi tramite programmazione a blocchi.
- Sviluppo e utilizzo di chatbot e altre applicazioni AI rilevanti per il contesto didattico.
- Esplorazione di progetti replicabili nelle ore di Educazione Civica, promuovendo un approccio interdisciplinare e una cittadinanza digitale consapevole.

Laboratorio formativo per insegnanti

“Welcome to NIMROD”

Claudio Mirolo

Dip. di Scienze Matematiche, Informatiche e Fisiche dell'Università di Udine
claudio.mirolo@uniud.it

Il percorso proposto prende spunto da un episodio della storia dell'informatica e si articola attorno alla logica di un pionieristico computer dimostrativo, *NIMROD*, in grado di competere con successo nel gioco del *nim*. Le attività si integrano con argomenti di matematica e scienze, consentono ulteriori agganci interdisciplinari, in particolare richiamando il ruolo del contesto storico sugli sviluppi scientifico-tecnologici, ed eventualmente possono offrire l'opportunità di discutere le riflessioni filosofiche di A. Turing su macchine e intelligenza.

Destinatari

Il laboratorio è proposto principalmente agli insegnanti della scuola secondaria di primo grado, ma è adatto anche al biennio della secondaria di secondo grado.

Obiettivi di apprendimento

Obiettivi generali di apprendimento:

- Stimolare l'interesse di allieve e allievi per il pensiero computazionale;
- Sviluppare il pensiero astratto.

Obiettivi di apprendimento specifici dell'ambito informatico:

- Identificare i dati rilevanti per perseguire un determinato obiettivo e utilizzare strumenti diversi per rappresentarli e manipolarli;
- Riflettere sulle proprietà operative di una rappresentazione;
- Sviluppare o approfondire il concetto di algoritmo;
- Ripensare dati e algoritmi a un livello più astratto, analizzandoli da diverse prospettive.

Strumentazione necessaria

Possibilmente i partecipanti dovrebbero avere a disposizione forbici, matita e stick di colla; può inoltre essere utile (ma non indispensabile) un PC personale.

Attività svolte

Il laboratorio si articola in due sessioni della durata di circa 90 minuti, in ciascuna delle quali i partecipanti si cimentano, individualmente o in piccoli gruppi, nello svolgimento delle attività rivolte agli allievi, sperimentandole direttamente.

1. Introduzione dell'episodio storico; confronto fra le tecnologie dell'epoca e quelle attuali; familiarizzazione con il gioco del *nim*; caratterizzazione della strategia di gioco di Bouton; breve illustrazione delle diverse unità didattiche che possono essere proposte.
2. Attività “unplugged” (con carta, cartoncino, matite colorate, forbici e colla) per modellare la strategia di *NIMROD* a diversi livelli di dettaglio; simulazione manuale e automatica dei passi dell'algoritmo; attività “unplugged” per capire in che termini la strategia di *NIMROD* persegue l'obiettivo di scegliere la (una) mossa migliore; indicazioni su ulteriori potenziali sviluppi delle attività.

Per partecipare al laboratorio non sono richieste conoscenze informatiche.

Maggiori dettagli sulle attività e sugli strumenti utilizzati sono disponibili online attraverso link:

http://nid.dimi.uniud.it/projects/pls_10_17.html#nimrod

LABORATORI PER LA SECONDARIA DI SECONDO GRADO

Ascoltiamo gli errori più comuni nella programmazione concorrente con Sonic Pi

Giorgio Delzanno, Giovanna Guerrini e Daniele Traversaro

DIBRIS, Università degli Studi di Genova, Italy
{giorgio.delzanno, giovanna.guerrini}@unige.it
daniele.traversaro@dibris.unige.it

Il laboratorio è basato sull'uso del linguaggio Sonic Pi, nato per live coding, per introdurre i concetti di base della programmazione concorrente e provare a risolvere le misconcezioni più tipiche di questo argomento facendo suonare gli errori più comuni come data race e drifting.

Destinatari

Il laboratorio è proposto principalmente agli insegnanti della scuola secondaria di secondo grado che affrontano la programmazione concorrente nei loro insegnamenti, ma è adatto a tutti gli interessati a linguaggi di programmazione per il dominio musicale. È possibile partecipare al laboratorio anche in assenza di conoscenze specifiche sulla programmazione concorrente.

Obiettivi di apprendimento

- Avvicinarsi ai linguaggi di programmazione specifici per il dominio musicale e per live coding.
- Stimolare l'interesse per le problematiche della concorrenza con esempi e artefatti concreti dal dominio musicale.
- Migliorare la comprensione degli aspetti cruciali della programmazione concorrente aiutandosi con suoni e musica.

Strumentazione necessaria

Possibilmente i partecipanti dovrebbero avere a disposizione un laptop personale Laptop con installato Sonic Pi e delle cuffie. Alcuni PC verranno messi a disposizione dagli organizzatori.

Attività svolte

Il laboratorio si articola in una sessione della durata di 90 minuti, in cui i partecipanti affrontano direttamente, individualmente o in piccoli gruppi, le attività che potranno essere proposte agli allievi. In particolare le tematiche affrontate sono le seguenti:

1. Introduzione al linguaggio Sonic Pi e realizzazione di un semplice brano musicale.
2. Focus su concorrenza: identificazione dei concetti di concorrenza e meccanismi di sincronizzazione presenti in Sonic Pi; analisi di alcune misconcezioni tipiche della concorrenza.
3. Realizzazione in Sonic Pi di semplici programmi che possano permettere di "ascoltare" i problemi di sincronizzazione legati a tale misconcezioni; discussione e indicazioni su ulteriori possibili sviluppi delle attività.

Maggiori dettagli sulle attività e sugli strumenti utilizzati sono disponibili online al link: <https://github.com/MaterialeInformatica/SonicPiPoster>

Pytch – Un ponte tra Scratch e Python

Sara Fiori¹, Ben North¹, Glenn Strong¹

¹Trinity College Dublin, The University of Dublin
info@pytch.org

1 Pytch

Il progetto di ricerca Pytch, svolto al Trinity College Dublin con collaboratori in TU Dublin, mira a promuovere e supportare interesse e coinvolgimento nell'apprendimento dell'Informatica in scuola Secondaria. Come parte del progetto, il team di ricerca ha sviluppato Pytch, un ambiente web di programmazione gratuito, co-progettato con docenti e studenti per aiutare quest'ultimi a passare da programmazione a blocchi (Scratch) a programmazione testuale (Python). Con Pytch, gli studenti mantengono tutte le conoscenze, l'intuizione e le competenze acquisite con Scratch e possono concentrarsi sull'imparare Python. Questo grazie a concetti come sprite e script basati su eventi, grafica e suoni, introducendo allo stesso tempo l'idea di scrivere codice Python testuale anziché trascinare e impilare blocchi. Per maggiori informazioni: <https://pytch.scss.tcd.ie/pytch-ita/>

1.1 Laboratorio formativo

Pytch è stato sviluppato per supportare docenti e studenti della scuola Secondaria 2' ma attività più semplificate per introdurre la programmazione Python possono essere proposte anche a ragazzi di scuola Secondaria 1'. In questo laboratorio informativo i partecipanti potranno esplorare Pytch e il suo ambiente di sviluppo [Figura 1], scoprendo diversi dettagli e funzionalità che lo rendono una soluzione molto apprezzata per introdurre Python agli studenti. Inoltre verranno anche presentati e discussi i materiali di supporto all'insegnamento sviluppati dal team (tra cui materiali per lezioni e tutorial passo passo). In questo modo i partecipanti potranno sperimentare Pytch e adattare l'attività di apprendimento proposta, nel laboratorio formativo, per i loro studenti di scuola secondaria. Il team rimarrà a disposizione per eventuale supporto nella creazione di attività specifiche per la classe. Pytch è basato sul web e non richiede alcuna procedura di installazione, ed è utilizzabile sia su computer che su tablet (suggerita una tastiera esterna). Al momento Pytch non richiede un account o login, per rendere l'accesso ancora più semplificato in un contesto scolastico.

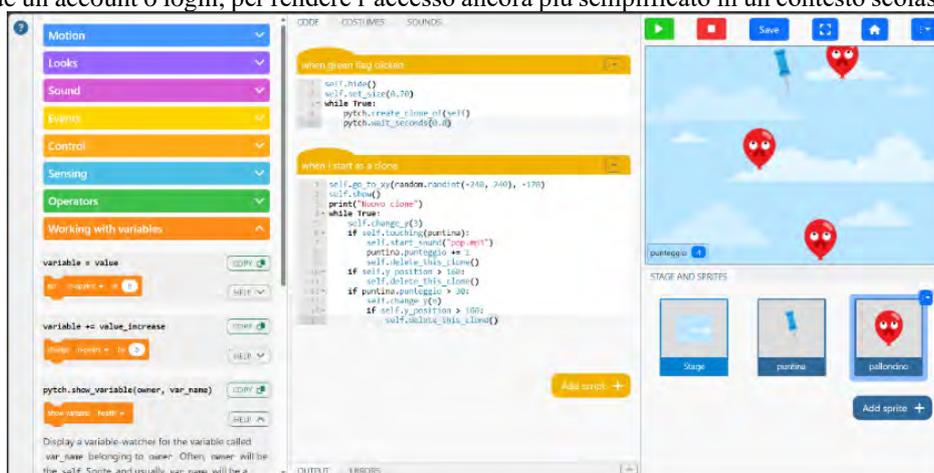


Figura 1: Ambiente di programmazione Pytch. Programma adattato e modificato a partire da un progetto di uno studente del Liceo L. Lanfranconi (Genova) guidato dal docente Marco Canesi.

Introduzione alla realizzazione di ambienti tridimensionali in realtà virtuale e aumentata

Manuela Chessa¹, Giorgio Delzanno¹, Lorenzo Gerini¹
Università degli Studi di Genova, DIBRIS
manuela.chessa@unige.it

Abstract

Il laboratorio introduce alla creazione di ambienti tridimensionali (3D) in realtà virtuale (VR) e aumentata (AR) utilizzando l'ambiente di sviluppo visuale CoSpaces Edu e il linguaggio a blocchi CoBlocks. CoSpaces Edu (<https://www.cospaces.io/>) offre strumenti per la progettazione e manipolazione di ambienti 3D interattivi, adatti a studenti dalla scuola primaria alle secondarie. Durante il laboratorio verrà mostrato come creare un semplice ambiente 3D, gestire le relazioni spaziali, realizzare animazioni e interazioni con CoBlocks, e visualizzare l'ambiente in modalità virtuale non immersiva, immersiva (con visore VR) o aumentata (tablet).

1 Organizzazione e descrizione delle attività

Il laboratorio, della durata di 90 minuti, si svolge in modalità interattiva, alternando spiegazioni del docente e attività pratiche dei partecipanti su PC. Le attività includono:

- Introduzione ai principi della modellazione 3D, con cenni sui sistemi di riferimento 3D e le trasformazioni (traslazioni, rotazioni, deformazioni) degli oggetti. Gestione delle relazioni spaziali tra gli oggetti con analogie con il mondo 3D reale e brevi cenni alle proprietà fisiche degli oggetti.
- Introduzione alla telecamera virtuale per visualizzare la scena 3D in VR o AR, evidenziando analogie con l'occhio umano come osservatore del mondo reale.
- Introduzione al linguaggio CoBlocks e alla programmazione di ambienti 3D per creare animazioni e interazioni di base. Creazione di scene complesse mediante la combinazione di blocchi semplici.
- Realizzazione di scene 3D interattive composte da blocchi e script CoBlocks, basate su quanto appreso nei punti precedenti (attività proposta ai docenti).
- Visualizzazione degli ambienti realizzati in modalità virtuale non immersiva, immersiva (con visore VR) o aumentata (con tablet).

2 Obiettivi di apprendimento

- Creazione di scenari virtuali tridimensionali, posizionamento e manipolazione di oggetti 3D e loro visualizzazione in VR e AR

Principi di coding applicati in questo contesto con le difficoltà del gestire scene in 3D.

3 Destinatari e strumentazione necessaria

Il laboratorio è dedicato ai docenti della Scuola Primaria e della Scuola Secondaria di primo e secondo grado. Le attrezzature necessarie comprendono: un proiettore o LIM, computer portatili con accesso a internet e un account CoSpaces (gratuito), un tablet e un visore Meta Quest 2.

Ordinamento distribuito nel mondo reale

Matteo Dell'Amico

Università di Genova

AVVERTENZE

- Consiglio a chi vuole partecipare all'attività di laboratorio di non leggere oltre, perché l'attività è concepita per persone che arrivano senza sapere cosa succederà.
- Ho inviato una versione estesa di questo documento su [ArXiv](https://arxiv.org/search/cs?searchtype=author&query=Dell%27Amico,+M), e dovrebbe apparire all'URL <https://arxiv.org/search/cs?searchtype=author&query=Dell%27Amico,+M>

1 Destinatari

Scuola secondaria di secondo grado.

2 Obiettivi di apprendimento

Mostrare che si può fare leva sull'intuizione per capire e risolvere problemi algoritmici anche complessi.

3 Strumentazione necessaria

Per i partecipanti: nessuna.

Io stesso porterò all'evento mazzi di carte numerate.

Servirà una lavagna, un monitor o un proiettore per riassumere i risultati.

4 Attività svolte

I partecipanti verranno divisi in squadre; ad ognuna di esse verrà dato un mazzo di carte numerate e verrà chiesto di restituirlo ordinato nel più breve tempo possibile.

Discuteremo di come possa servire fare leva sull'intuizione dei ragazzi per aiutarli a capire algoritmi complessi. In questo caso, le analogie tra l'ordinamento distribuito come problema informatico e l'attività proposta sono molte e, secondo la mia esperienza, i ragazzi trovano facilmente delle soluzioni efficienti.

ALTRI CONTRIBUTI

Reti Unplugged

L'apprendimento cinestetico per imparare i concetti di incapsulamento, instradamento e indirizzamento attraverso la gamification

Virginia Beato¹, Luca Pinet²

¹ Istituzione scolastica di Istruzione Liceale, Tecnica e Professionale (I.S.I.L.T.P.)
v.beato@mail.scuole.vda.it

² Istituzione scolastica di Istruzione Liceale, Tecnica e Professionale (I.S.I.L.T.P.)
l.pinet@mail.scuole.vda.it

Abstract

Insegnare i principali concetti relativi alle reti di calcolatori, quali incapsulamento, indirizzamento, instradamento e il ruolo dei principali dispositivi di rete, attraverso l'utilizzo della gamification, con un approccio cinestetico alla didattica, favorisce l'apprendimento degli stessi per l'intero gruppo classe rispetto a metodi più tradizionali, oltre a favorire l'inclusione di alunni con bisogni educativi speciali.

1 Gamification

La **gamification** è un approccio didattico che utilizza elementi tipici dei giochi per coinvolgere gli studenti nel processo di apprendimento. Essa mira a rendere le lezioni più coinvolgenti, interattive e motivanti, sfruttando dinamiche ludiche per trasmettere concetti complessi in modo accessibile. La gamification può assumere diverse forme, tra cui l'uso di giochi didattici, competizioni e sfide, contribuendo a creare un ambiente più stimolante per gli studenti.

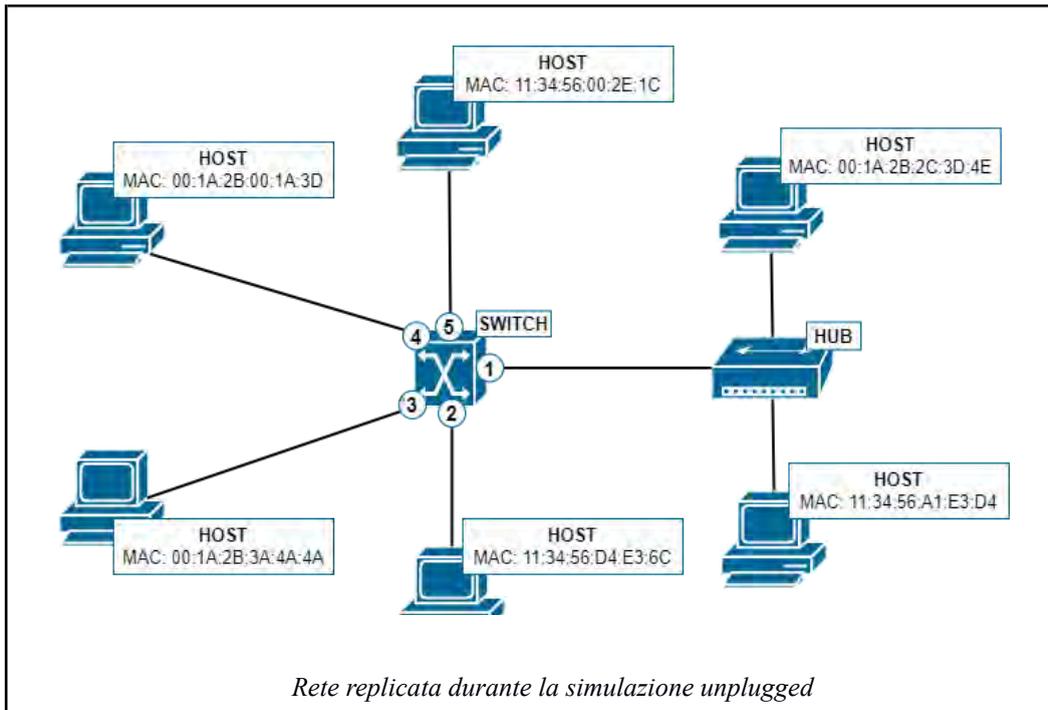
L'apprendimento attraverso il gioco è un approccio di successo. Questo metodo offre un ambiente di apprendimento più rilassato, che può portare ad un miglioramento delle performance degli studenti.

Al fine di integrare la gamification nell'apprendimento delle reti, sono state create diverse **simulazioni unplugged**. Una simulazione unplugged è un'esperienza pratica che coinvolge gli studenti in attività che non prevedono l'ausilio di dispositivi tecnologici. Ad esempio, in questo contesto, sono state create delle reti con fili di spago e buste di carta, al fine di simulare le reti e lo scambio di pacchetti senza l'ausilio dei computer.

Queste attività hanno permesso di semplificare i concetti che per gli studenti risultano più ostici da comprendere, come ad esempio il significato di incapsulamento. Questo approccio si è rivelato particolarmente vantaggioso per gli studenti orientati all'apprendimento visivo e soprattutto all'apprendimento cinestetico, oltre che per coloro che affrontano barriere linguistiche.

2 Rete Unplugged

Nel corso di una delle attività unplugged proposte alla classe, gli studenti sono stati coinvolti in una simulazione pratica per apprendere il funzionamento di uno switch e il concetto di incapsulamento. Per farlo, è stata creata una rappresentazione tangibile di una rete locale attraverso materiali semplici, come spaghi e buste di carta.



2.1 Creazione della rete unplugged

La simulazione comprende sei **host**, uno **switch** e un **hub**. Ogni dispositivo di rete è personificato da uno studente, collegato agli altri mediante uno spago. A turno, gli host si scambiano pacchetti di dati, rappresentati da buste di carta. Il ruolo dello switch e dell'hub, invece, è quello di inoltrare i pacchetti sulla base delle proprie regole di inoltro. I rettangoli con la riga blu, in realtà, sono cappelli di carta utilizzati dai cuochi in cucina. Ciascun cappello assume il ruolo di un dispositivo di rete differente, e ogni host è distintamente identificato dal proprio indirizzo MAC, annotato sul relativo cappello.

I fili bianchi delineano i **collegamenti** fisici tra gli apparati di rete, e sono composti da uno spago con due mollette fissate alle estremità. Gli studenti hanno fissato le mollette alle proprie maglie (o ai passanti per la cintura dei propri pantaloni) per simulare l'infrastruttura fisica. Alcuni cavi sono anche dotati di un Post-It, rappresentante il numero dell'interfaccia dello switch, alloggiato su una delle due estremità. Le mollette sono state utilizzate per ragioni di sicurezza: legare direttamente gli studenti avrebbe comportato un grave rischio in una potenziale evacuazione in situazioni di emergenza.

Un altro aspetto chiave della simulazione è il ruolo delle buste di carta: per comprendere a pieno il concetto di incapsulamento, sono state utilizzate varie buste, ciascuna rappresentante un header di un particolare livello dello stack TCP/IP. Gli studenti, per comunicare all'interno della rete, devono eseguire le operazioni di incapsulamento e decapsulamento, rispettivamente inserendo o estraendo le buste una dentro l'altra. Sono state utilizzate cinque buste di diverse dimensioni per illustrare il concetto di incapsulamento, che rappresentano ciascuna l'**header** di un livello diverso dello stack TCP/IP.



Nell'immagine è visibile un telefono sulla cattedra, utilizzato per impostare il timer necessario alla cancellazione periodica delle righe nella tabella di inoltro dello switch. Inoltre, è stata fotografata la lavagna su cui è disegnata la tabella di inoltro.

2.2 Invio di un pacchetto alla volta all'interno della rete

Inizialmente, il mittente compone il messaggio da inviare e simula l'**incapsulamento** inserendo il messaggio nella busta che rappresenta l'header del livello di applicazione. Successivamente, colloca tale busta all'interno di quella che identifica l'header del livello di trasporto, e così via. Raggiunto l'header del livello di collegamento, il mittente include il suo indirizzo MAC e quello del destinatario, quindi il frame può essere inserito nell'ultima busta, corrispondente al livello fisico. È stato chiarito agli studenti che il livello fisico non aggiunge informazioni, bensì si occupa di trasformare i bit in un segnale che può essere trasmesso nel mezzo trasmissivo appropriato, in questo caso, un cavo Ethernet. La busta associata al livello fisico rappresenta quindi la conversione dei bit in un segnale elettrico, idoneo per la trasmissione sul cavo.

Il mittente procede quindi con l'invio del pacchetto, il quale potrebbe essere ricevuto da uno switch o da un hub. Nel caso di ricezione da parte dell'hub, quest'ultimo duplica il pacchetto su tutte le interfacce, escludendo quella di provenienza. Nel caso di ricezione da parte dello switch, è richiesta la scelta dell'azione da compiere tra flooding, filtering o forwarding sulla base della tabella di inoltro attuale. Per gestire efficacemente la duplicazione dei pacchetti, sono stati designati due studenti a rappresentare uno switch e due un hub, definiti rispettivamente "aiutante dello switch" e "aiutante dell'hub". In questa fase della simulazione, essi agiscono in collaborazione con hub e switch per duplicare i pacchetti e determinare la destinazione.

Si è evidenziato che, a questo punto, un host all'interno della rete potrebbe ricevere il messaggio anche se non è il destinatario effettivo. In tal caso, l'host se ne rende conto durante il processo di decapsulamento, quando scopre che l'indirizzo MAC destinatario non corrisponde al proprio. Quindi, procede con l'eliminazione del pacchetto. Al momento della ricezione del pacchetto da parte del destinatario, è richiesto il processo di lettura del messaggio interno. A tale scopo, il destinatario deve rimuovere gli header uno alla volta, simulando il **decapsulamento**.



Ogni studente ha agito da mittente almeno una volta, e coloro che incontravano difficoltà hanno avuto l'opportunità di svolgere nuovamente il ruolo di mittente. Tale procedura è stata ripetuta fino a quando non si sono verificati più errori, indicando che tutti gli studenti hanno raggiunto il livello di comprensione desiderato riguardo all'incapsulamento.

2.3 Invio di più pacchetti alla volta all'interno della rete

La terza fase dell'attività è dedicata all'inoltro di più pacchetti contemporaneamente all'interno della rete.

In questa fase, periodicamente e sottovoce, il docente assegna a un host il compito di inviare un messaggio ad un altro host. Tale modalità garantisce che solamente il mittente sia a conoscenza del destinatario, mentre gli altri host della rete si comportano in modo analogo a come agiscono i dispositivi di rete nella realtà: gestiscono i pacchetti in entrata basandosi unicamente sulle informazioni contenute negli header.

L'introduzione di più pacchetti simultanei nella rete ha generato **collisioni**, offrendo, tra le altre cose, l'opportunità di esaminare nuovamente il comportamento di CSMA/CD. Lo snodo di questa fase consiste nell'abilità di tutti gli studenti di gestire in modo autonomo i compiti a loro assegnati. L'indicatore utilizzato per valutare tale competenza si è fondato sull'osservazione degli eventi all'interno della rete in ogni istante. La simulazione è terminata dopo che tutti gli studenti sono riusciti a gestire in modo autonomo i pacchetti in entrata e in uscita.

3. Conclusioni

L'attività presentata è stata condotta all'interno di una classe quarta di un istituto di istruzione secondaria di secondo grado, in particolare nell'istituto tecnico del settore tecnologico dell'indirizzo "Informatica e telecomunicazioni" nell'articolazione Informatica.

Il Dipartimento di informatica dell'Istituzione scolastica di istruzione liceale, tecnica e professionale di Verrès, in Valle d'Aosta (AO), sperimenta da ormai più anni scolastici l'utilizzo dell'algomotricità e dell'apprendimento cinestetico (vedi "L'apprendimento cinestetico degli algoritmi di ordinamento e ricerca su un vettore attraverso la danza" di Pinet Luca e Laura Frasson, ITADINFO 2023).

In base all'esperienza didattica significativa riguardante l'apprendimento cinestetico, dopo una lunga sperimentazione in tal senso, ci sentiamo di poter dire che, in classi a medio-alta complessità, il processo di insegnamento/apprendimento cinestetico sia davvero uno strumento efficace per poter rendere la didattica accessibile a tutti gli studenti, soprattutto se combinato ad una moltitudine di altri stili di apprendimento e insegnamento.

Questa attività didattica e i suoi risultati soddisfacenti per quanto concerne gli obiettivi raggiunti dagli studenti, in termini sia di competenze, sia di conoscenze, sia di abilità, mostra l'importanza dell'impiego di metodologie didattiche innovative nell'insegnamento della materia Sistemi e Reti, la quale si caratterizza per la sua complessità e astrattezza.

In conclusione, l'attività, che è stata presentata in maniera sintetica, può essere ulteriormente implementata e migliorata: è possibile modificare la rete aggiungendo progressivamente i dispositivi router e il concetto di indirizzamento di rete, oltre che la descrizione di alcuni tra i principali protocolli, quali ad esempio NAT/PAT.

Bibliografia

- [1] Tesi del Corso di Laurea magistrale in Informatica “*Metodologie didattiche innovative per l'insegnamento di Reti: un'esperienza in classe*” di Beato Virginia, presso Università degli studi di Torino (anno accademico 2023/2024).
- [2] Constantinos Hilas e Anastasios Politis. «*Motivating students' participation in a computer networks course by means of magic, drama and games*». In: SpringerPlus 3.362 (2014). doi: 10.1186/2193-1801-3-362.
- [3] Luma da Rocha Seixas, Alex Sandro Gomes e Ivanildo José de Melo Filho. «*Effectiveness of gamification in the engagement of students*». In: Computers in Human Behavior 58 (2016), pp. 48–63. issn: 0747-5632. doi: <https://doi.org/10.1016/j.chb.2015.11.021>.
- [4] Luigi Catuogno e Alfredo De Santis. «*An internet role-game for the laboratory of network security course*». In: SIGCSE Bull. 40.3 (giu. 2008), pp. 240–244. issn: 0097-8418. doi: 10.1145/1597849.1384336.
- [5] Craig Jordan, Matt Knapp, Dan Mitchell, Mark Claypool e Kathi Fisler. «*CounterMeasures: A game for teaching computer security*». In: 2011 10th Annual Workshop on Network and Systems Support for Games. 2011, pp. 1–6. doi: 10.1109/NetGames.2011.6080983.
- [6] Evangelos Katsadouros, Dimitris Kogias, Lazaros Toumanidis, Christos Chatzi-georgiou e Charalampos Z. Patrikakis. «*Teaching network security through a scavenger hunt game*». In: 2017 IEEE Global Engineering Education Conference (EDUCON). 2017, pp. 1802–1805. doi: 10.1109/EDUCON.2017.7943094.

First Lego League: la robotica educativa per avvicinare gli studenti all'informatica

Giorgio Gasparini¹, Maria Elena Lai², Elisabetta Pezzati³, Marco Sorrentino⁴

¹ IC Cornigliano

giorgio.gasparini@iccornigliano.edu.it

² IC Cornigliano

mariaelenalai@iccornigliano.edu.it

³ IC Cornigliano

elisabetta.pezzati@iccornigliano.edu.it

⁴ IC Cornigliano

marco.sorrentino@iccornigliano.edu.it

Abstract

First Lego League, una sfida mondiale di scienza e robotica per ragazzi in cui gli studenti progettano, costruiscono e programmano robot autonomi per affrontare problemi reali, con l'obiettivo di avvicinarsi alle discipline STEM, in particolare all'informatica, attraverso l'uso dei robot Lego. Il progetto, attraverso un apprendimento pratico e cooperativo, favorisce lo sviluppo di competenze scientifiche e tecnologiche e prepara gli studenti per futuri percorsi educativi e professionali.

1 Introduzione

Il nostro Istituto ormai da anni partecipa alla competizione internazionale First Lego League (FLL) con due squadre di venti studenti della scuola secondaria di primo grado. FLL è una sfida mondiale di scienza e robotica tra squadre di ragazzi dai 9 ai 16 anni che progettano, costruiscono e programmano robot autonomi, applicandoli a problemi reali di grande interesse per cercare soluzioni innovative. La manifestazione richiede inoltre ai partecipanti di effettuare una ricerca con i criteri caratteristici del metodo scientifico su una problematica attuale. Oltre ad appassionarsi alla scienza divertendosi, i ragazzi acquisiscono conoscenze e competenze utili al loro futuro lavorativo e si avvicinano in modo concreto a carriere in ambito scientifico e ingegneristico che richiedono indispensabili competenze digitali, con particolare riferimento all'informatica tramite la programmazione di robot Lego Mindstorm EV3 [1] [2] [3].

2 Robot Lego Mindstorm EV3 e software

Per la gara di robotica sono stati utilizzati robot Lego Mindstorm EV3 che sono strumenti all'avanguardia nella didattica delle STEM. Il robot è uno strumento di apprendimento progettato per avvicinare gli studenti ai concetti fondamentali della programmazione, del data logging e hardware. L'elemento principale del robot è rappresentato dal "brick", il corpo centrale

programmabile, che permette di controllare i motori e i sensori ad esso collegati. L'interfaccia di programmazione utilizzata nel nostro percorso è il software Lego Mindstorm education EV3 basato su una programmazione a blocchi. Il software scelto è idoneo per introdurre studenti dagli 11 ai 14 anni al pensiero computazionale: la programmazione risulta visiva ed intuitiva; è applicata su un oggetto fisico che restituisce subito un feedback allo studente.

L'ambiente di programmazione EV3 è formato da un'area di programmazione suddivisa in vari blocchi tra cui i principali sono:

- il blocco azione che permette di controllare il movimento dei motori;
- il blocco flusso che consente di avviare, interrompere e controllare tempi e cicli delle azioni;
- blocchi sensore che controllano i sensori collegati al brick.

3 Fasi di lavoro

Il lavoro è stato svolto durante tutto il primo quadrimestre in orario extracurricolare coinvolgendo un gruppo di 20 alunni delle classi seconde e terze scelti dalle diverse sezioni dell'istituto in base alle loro personali attitudini e interessi. Sono stati divisi in due squadre da 10 partecipanti in base al regolamento della competizione; il progetto si è articolato in diverse attività utili alla programmazione del robot per eseguire le missioni della gara FLL.

3.1 Costruzione del robot

In questa fase gli alunni hanno seguito le istruzioni per montare la struttura base del Mindstorm EV3. In questo modo hanno potuto acquisire consapevolezza degli elementi fondamentali che costituiscono la macchina e che successivamente dovranno controllare tramite la programmazione. La costruzione iniziale del robot ha rappresentato un punto di partenza per apportare le successive modifiche alla struttura di base che influenzano la programmazione del robot in fase di risoluzione delle missioni.

3.2 Introduzione al software Mindstorm

Nella seconda fase gli studenti sono stati introdotti al software con piccole esercitazioni (traiettorie lineari e piccoli percorsi) al fine di acquisire competenze di programmazione e una sempre maggiore capacità di utilizzare il robot nelle sue componenti più complesse come sensore di colore e ultrasuoni [4]. Queste esercitazioni sono risultate fondamentali per introdurre gli studenti alla comprensione del funzionamento meccanico del robot e di conseguenza delle variabili in gioco, come ad esempio numero di rotazioni del motore e relativa potenza e modalità di rotazione del robot, cognizione dei tempi di risposta e di esecuzione della macchina.

3.3 Missioni First Lego League

Dopo le fasi introduttive di costruzione del robot e programmazione, gli studenti, con la guida dei docenti, hanno inizialmente analizzato le varie missioni della competizione, ragionando sulle possibili soluzioni per risolverle con i robot. Le missioni sono state studiate attraverso il video ufficiale della Lego League e la guida allegata che esplicita regole, obiettivi da raggiungere, ostacoli da evitare e elementi da tenere in considerazione durante la programmazione. Nella fase di costruzione e programmazione definitiva dei robot, finalizzata al superamento delle missioni, gli studenti sono stati liberi di sperimentare per prove ed errori. In questa fase i docenti hanno assunto una funzione di facilitatori, rendendo gli studenti protagonisti attivi del processo di apprendimento per una costruzione autonoma delle competenze. All'interno del gruppo ogni studente ha assunto un ruolo ben definito. In particolare alcuni alunni prediligono la parte meccanica: in un primo momento aggiungendo al robot gli elementi opzionali ritenuti utili al superamento delle missioni (leve, sensori, aste, ecc.) e in una seconda fase verificando che gli elementi aggiunti siano

compatibili con la programmazione del robot, apportando modifiche in itinere. Altri componenti della squadra invece si sono focalizzati sulla programmazione vera e propria con il software attraverso numerosi tentativi verificati progressivamente sul campo di gara. Le attività delle singole squadre sono state intervallate da momenti di scambio e condivisione tra studenti e docenti esaminando come ogni gruppo aveva programmato la stessa missione e valutando le differenze e le analogie di programmazione delle due squadre per giungere ad una soluzione condivisa della missione da svolgere. Questa fase di riflessione metacognitiva ha contribuito a far acquisire competenze di programmazione finalizzate al raggiungimento di un obiettivo. Per la programmazione gli studenti hanno lavorato principalmente sui blocchi di azione e di flusso, sviluppando competenze logico-matematiche in quanto hanno dovuto riflettere sui seguenti fattori:

- numero di rotazione da assegnare al motore per completare una determinata distanza;
- gradi di rotazione del motore per far ruotare il robot in un punto preciso del campo di gara;
- angolazione e traiettoria di partenza del robot;
- impostazione dei tempi di pausa ed esecuzione delle varie azioni programmate;
- calibrazione dei sensori (colore) considerando anche l'influenza dell'intensità luminosa dell'ambiente circostante.

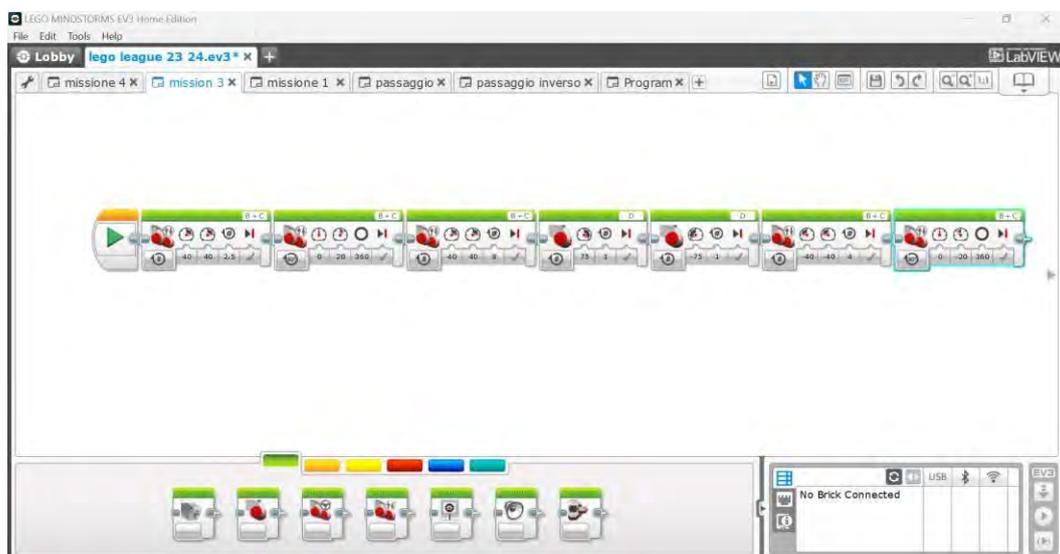


Figura 1 Esempio di programmazione svolta dagli alunni

4 Conclusioni

Introdurre l'informatica nella scuola secondaria di primo grado è una mission del nostro istituto già da diversi anni, anche in ottica di orientamento alle scuole superiori e di inclusione di un sempre maggior numero di studentesse alle discipline STEM. Per raggiungere però questi obiettivi, considerata la fascia di età del target (11-14 anni), è stata fondamentale la strutturazione del progetto secondo metodologie didattiche cooperative e di learning by doing, per rendere gli studenti e le studentesse reali protagonisti e costruttori del processo di apprendimento. Gli studenti hanno mostrato in questo lavoro anche delle ottime competenze di collaborazione e spirito di iniziativa; in particolare gli alunni delle classi terze, più esperti nella programmazione, hanno svolto ruolo di tutor per quelli più piccoli e alla prima esperienza nel progetto. Al termine del percorso agli studenti viene somministrato un questionario di autovalutazione, focalizzando le domande sui seguenti

aspetti: difficoltà incontrate nel percorso sia dal punto di vista personale che nel lavoro di gruppo, livello di soddisfazione personale al termine del progetto, punti di forza del proprio percorso, autovalutazione sulle competenze di programmazione e di problem solving raggiunte, coinvolgimento e motivazione nella partecipazione alle attività. Dalle risposte emerge che gli alunni sentono di aver acquisito competenze in campo informatico e di aver sviluppato buone capacità di lavorare in gruppo. Altro aspetto importante che è emerso dalle riflessioni degli studenti riguarda la soddisfazione nel poter “toccare con mano” i risultati di un lavoro di gruppo e la possibilità che questo percorso ha dato loro per acquisire competenze spendibili nella realtà di tutti i giorni. I risultati di questo progetto possono essere esemplificati dalle parole di uno degli studenti che già da due anni partecipa al percorso: “ho imparato ad aprire la mente e a pensare oltre!”

Considerata la complessità della disciplina informatica e della vasta gamma dei nuovi istituti tecnici e informatici sul nostro territorio, l'avviamento allo studio dell'informatica attraverso un progetto strutturato nel PTOF dell'Istituto da anni, sta consentendo di rendere gli studenti maggiormente consapevoli nella scelta dell'Istituto tecnico informatico, per poter affrontare il percorso di studi superiori con una competenza di base dell'informatica.

Bibliografia

- [1] <https://www.firstlegoleague.org/>
- [2] Gasparini G., Lai M.E.; Pezzati E.; Raffo N.; Sorrentino M. First Lego League *STEAMITALIA 2022*; ISBN 979-12-210-2579-8 <https://www.steamitalia.it/steam-italia-2022/>
- [3] https://drive.google.com/file/d/1hStDloMmKNfO_GCogopOLjwA-hUU1DnZP/view?usp=sharing
- [4] <https://www.lego.com/it-it/themes/mindstorms/learntoprogram>

Creazione di videogiochi VR per la didattica dell'informatica nella Scuola Superiore

Alessandro Forgiarini¹, Sonia Guerci², Eva Venturini², Fabio Buttussi¹

¹ HCI Lab, Dipartimento di Scienze Matematiche, Informatiche e Fisiche,
Università degli Studi di Udine

alessandro.forgiarini@uniud.it, fabio.buttussi@uniud.it

² Istituto Tecnico Antonio Zanon di Udine

sonia.guerci@itzanon.it, eva.venturini@itzanon.it

Abstract

La creazione di esperienze videoludiche in realtà virtuale è un ambito di interesse sia per il mondo commerciale che per quello accademico.

Con lo scopo di far applicare agli studenti, in un progetto concreto, le potenzialità dei diversi argomenti trattati durante il percorso scolastico, è stato progettato e proposto un corso intensivo in cui gli studenti hanno imparato a creare un videogioco utilizzando strumenti professionali.

Gli studenti hanno sperimentato tutte le fasi del processo di sviluppo: dalla modellazione della meccanica di gioco al design delle ambientazioni, dall'implementazione del codice fino al test del gioco sia con un visore per la realtà virtuale che con uno smartphone.

L'esperienza è stata non solo stimolante e formativa, perché gli studenti sono riusciti a mettere in pratica la teoria studiata in classe e a testare il prodotto realizzato in prima persona, ma anche utile in un'ottica di orientamento, grazie alla collaborazione tra Scuola Superiore e Università.

1 Introduzione

L'Istituto IT A. ZANON di Udine ha colto l'opportunità di realizzare percorsi didattici, formativi e di orientamento per promuovere contenuti, finanziati dal DM 65, per sviluppare le competenze STEM, digitali e di innovazione, e garantire attività di orientamento STEM.

In tale contesto, focalizzandosi sulle competenze che potevano avere anche ricadute in ambito professionale e analizzando gli interessi degli studenti, ci siamo chiesti quali fossero le azioni formative più rispondenti agli indirizzi scolastici proposti dall'Istituto e a chi rivolgere i suddetti percorsi.

Negli ultimi anni la realtà virtuale (VR), i.e., un ambiente digitale generato al computer che può essere vissuto e con cui è possibile interagire come se l'ambiente fosse reale [1], ha ricevuto crescente attenzione sia dal mondo accademico che dal mondo aziendale, per esempio nella creazione di esperienze immersive per l'addestramento delle persone sulla sicurezza sul lavoro o nelle emergenze [2]. Questi ambienti digitali possono essere fruiti attraverso un comune schermo per pc o smartphone (VR non-immersiva) [3] oppure attraverso un visore per la realtà virtuale che permette agli utenti di essere completamente circondati nell'ambiente virtuale (VR immersiva) [4].

Per rendere le esperienze formative virtuali più efficaci e coinvolgenti, molti ricercatori stanno sperimentando l'inserimento di meccaniche di gioco che vadano oltre alla semplice assegnazione di punti o al raggiungimento di livelli (gamification), creando dei veri e propri videogiochi

finalizzati a scopi seri, chiamati Serious Games [5]. Per questo motivo, abbiamo proposto un percorso formativo per insegnare agli studenti come sviluppare un gioco in VR immersiva e non-immersiva.

Nel curriculum scolastico dell'Istituto, la parte di programmazione orientata agli oggetti è proposta prevalentemente a livello teorico, mentre manca l'utilizzo di strumenti per sviluppatori professionali complessi come Unity [6] e Visual Studio [7]. Inoltre, il settore dei videogiochi offre opportunità di lavoro diversificate: a partire dall'idea iniziale fino al rilascio del videogioco, le figure professionali coinvolte sono numerose e diverse tra loro [5], per cui lo sono anche le opportunità offerte. Considerato che l'avvento dei Game Engine ha permesso anche agli appassionati di avvicinarsi allo sviluppo dei videogame, abbiamo pensato di proporre un percorso per far sperimentare agli studenti l'attività di uno sviluppatore di videogiochi professionista.

Pertanto, è stato progettato un apposito percorso didattico, formativo e di orientamento rivolto agli studenti delle classi quinte del percorso Sistemi Informativi Aziendali (SIA). Per promuovere la valorizzazione del merito e incentivare la motivazione e la passione degli studenti più meritevoli, abbiamo selezionato gli studenti con la valutazione media più alta.

Grazie all'accordo siglato con l'Università degli Studi di Udine abbiamo proposto alle eccellenze delle classi quinte un percorso di dieci ore per creare esperienze in realtà virtuale. Il corso è stato preparato dal co-autore Forgiarini, con la supervisione del co-autore Buttussi, ed ha accompagnato i ragazzi a scoprire Unity al fine di popolare un ambiente tridimensionale e renderlo fruibile attraverso un visore e uno smartphone Android.

2 Strumenti utilizzati

Il laboratorio scolastico utilizzato dispone di 20 postazioni studente e due postazioni docente (Sistema operativo Windows 11 64bit, Processore AMD Ryzen 5 PRO 3400G 3.70 GHz, RAM 16 GB). Vista la limitata potenza dei calcolatori per lo sviluppo di applicazioni in VR immersiva, abbiamo chiesto ai ragazzi di utilizzare i propri smartphone per provare in prima persona ciò che avrebbero implementato e solo alla fine il progetto è stato caricato su un visore Meta Quest 3.

Il Game Engine scelto per il corso è Unity vista la sua diffusione per lo sviluppo di applicazioni professionali e videogiochi. Esso consente la creazione di ambienti tridimensionali fruibili attraverso diverse piattaforme. Inoltre, permette di scrivere codice che modella, per esempio, le diverse meccaniche di gioco, le interazioni tra gli oggetti virtuali e la gestione dell'input.

Con lo scopo di riprendere gli strumenti già noti agli studenti, è stato scelto di utilizzare l'ambiente di sviluppo Visual Studio, già visto nel biennio del corso SIA per lo sviluppo con il linguaggio C++, poiché supporta anche C# utilizzato da Unity.

Le postazioni sono state quindi predisposte per l'attività installando Unity 2022.3 LTS (con licenza education), compresi i moduli aggiuntivi per lo sviluppo su Android, e Visual Studio Community 2022 con il supporto allo sviluppo di giochi con Unity e al linguaggio C#.

3 Attività svolta

Il corso è tenuto dal co-autore Forgiarini con il supporto delle co-autrici Venturini e Guerci presso l'Istituto Zanon nel mese di marzo 2024.

Il gioco proposto ha lo scopo di far scoppiare i palloncini che compaiono a schermo per guadagnare punti.

L'attività è stata strutturata in modo da fornire un riscontro immediato dopo ogni fase di sviluppo. Sono state svolte due lezioni in laboratorio per complessive dieci ore in cui il docente prima spiegava i diversi concetti e poi i ragazzi, a piccoli gruppi, li applicavano al proprio progetto, con il supporto del docente quando necessario.

In quest'ottica, si è scelto di iniziare con la descrizione di Unity Hub, le versioni di Unity con le relative interfacce grafiche e l'integrazione dei diversi tipi di asset (modelli 3D e file audio) all'interno del progetto in modo da poter popolare l'ambiente virtuale.

Successivamente si è gestito l'input dell'utente, con la creazione di uno script dedicato al tocco su smartphone e l'utilizzo del giroscopio per poter osservare l'ambiente di gioco. In Figura 1 è possibile leggere il codice sviluppato per il tocco.

```
// Creo un raggio che parte dalla posizione del tocco e prosegue in quella direzione
Ray ray = mainCamera.ScreenPointToRay(↵ position);

// Determino se il raggio colpisce qualsiasi cosa
if (Physics.Raycast(ray, out RaycastHit hit, maxDistance:100, layerMask:-0, QueryTriggerInteraction.UseGlobal))
{
    GameObject hitGameObject = hit.transform.gameObject;
    Debug.Log($"HIT: {hitGameObject.name}");

    // Se l'oggetto colpito è un Palloncino
    Balloon balloon = hitGameObject.GetComponentInParent<Balloon>();
    if (balloon != null)
    {
        // Eseguo la logica per far scoppiare il palloncino
        balloon.BalloonHit();
    }
}
}
```

Figura 1 Codice sviluppato per la gestione del tocco su smartphone.

In seguito, sono state implementate le diverse meccaniche di gioco, la gestione della fisica, i materiali per il rendering, la creazione di semplici animazioni, le interfacce grafiche, e la definizione dei diversi stati del gioco (preparazione della partita, gioco, fine della partita) con la creazione della classe "Game Manager" che si occupa di orchestrare tutte le componenti oltre che la durata della partita e la quantità di punti assegnati. In Figura 2 è possibile vedere la schermata conclusiva con le diverse parti della gerarchia e i parametri del gioco.

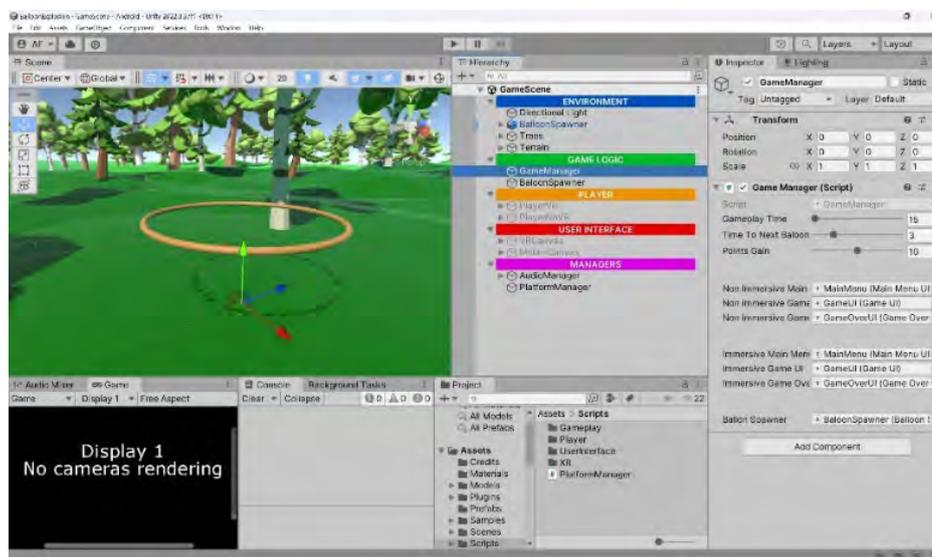


Figura 2 Schermata del Game Engine a conclusione del progetto.

Successivamente si è passati all'introduzione e integrazione di XR Framework all'interno del progetto per poter utilizzare il visore per la VR immersiva come dispositivo di input. In Figura 3 viene mostrato il docente mentre spiega come integrare il supporto nel progetto di XR utilizzando il visore e le postazioni utilizzate dagli studenti.



Figura 3 Docente mentre spiega l'integrazione di XR framework.

Infine, il corso si è concluso con la prova del gioco sul visore e la creazione della build per lo smartphone con seguente installazione su di esso. In questo modo, dopo una prima presentazione sugli strumenti utilizzati, gli studenti hanno potuto simulare tutte le fasi di sviluppo dei game 3D, dall'inserimento delle risorse nel Game Engine fino all'installazione sui dispositivi.

4 Conclusioni

L'esperienza è stata stimolante e formativa poiché gli alunni sono riusciti ad applicare la teoria studiata in classe durante i precedenti anni del loro percorso ad un progetto concreto. Hanno avuto l'opportunità di testare il prodotto sviluppato in prima persona utilizzando sia i loro smartphone sia un visore per la VR immersiva.

Inoltre, gli studenti hanno potuto sperimentare l'utilizzo di altri paradigmi e linguaggi di programmazione, nonché di altri strumenti dedicati agli sviluppatori. Gli studenti sono riusciti a seguire le spiegazioni e raggiungere gli obiettivi dell'attività proposta, nonostante alcune nozioni di programmazione fossero completamente sconosciute e molti dei concetti affrontati durante il breve percorso siano oggetto di studio della laurea in Informatica.

Il corso è stato dunque utile anche in un'ottica orientativa per gli studenti in q hanno avuto l'opportunità di confrontarsi con l'ambiente universitario.

Acquisite le basi e potendo contare su un maggior numero di ore a disposizione, un interessante sviluppo futuro potrebbe consistere nel far proseguire l'attività con un secondo progetto che faccia lavorare gli studenti in piccoli gruppi, con i ruoli e gli strumenti per lo sviluppo di software in team, dando così anche un assaggio di applicazione reale dei concetti di ingegneria del software.

Bibliografia

- [1] J. Jerald, *The VR Book: Human-Centered Design for Virtual Reality*, ACM Press, 2015.
- [2] D. Scorgie, Z. Feng, D. Paes, F. Parisi, T. Yiu and R. Lovreglio, "Virtual reality for safety training: A systematic literature review and meta-analysis," *Safety Science*, vol. 171, p. 106372, 01 03 2024.
- [3] G. Robertson, S. Card and J. Mackinlay, "Three views of virtual reality: nonimmersive virtual reality," *Computer*, vol. 26, no. 2, pp. 81-, 1993.
- [4] R. Pausch, "Three views of virtual reality: an overview," *Computer*, vol. 26, no. 2, pp. 79-80, 1993.
- [5] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, pp. 25-32, 2005.
- [6] Unity, "Unity Real-Time Development Platform | 3D, 2D VR & AR Engine," [Online]. Available: <https://unity.com/>. [Accessed 09 07 2024].
- [7] Microsoft, "Visual Studio: IDE and Code Editor for Software Developers and Teams," Microsoft, [Online]. Available: <https://visualstudio.microsoft.com>. [Accessed 09 07 2024].

Progetto App me Up: creatività, autonomia e peer tutoring

Nicoletta Rossi¹

¹ITE G. B. Bodoni - Parma

nicolet.rossi61@gmail.com

Abstract

Questo lavoro descrive una sperimentazione condotta in un Istituto Tecnico Economico consistita nella creazione da parte degli studenti di uno strumento didattico finalizzato all'utilizzo in una classe inferiore del medesimo istituto. Scopi del progetto sono state la rimotivazione all'apprendimento, l'acquisizione della consapevolezza delle proprie capacità e di una metodologia di lavoro. Gli scopi prefissati sono stati raggiunti al di là delle aspettative, con ricadute positive anche nel prosieguo degli studi.

1 Introduzione

Il progetto App me Up nasce per tre scopi principali: sviluppare le capacità di problem solving e pensiero computazionale degli studenti, stimolare la loro creatività e, attraverso questo, trovare una strada per accendere l'interesse e la motivazione anche in una classe considerata estremamente problematica.

Durante la sperimentazione sono state coinvolte classi di un Istituto Tecnico Economico, in particolare una classe 4° Sistemi Informativi Aziendali ed una classe 2°.

2 Svolgimento dell'attività

Il progetto si è articolato in diverse fasi, la prima delle quali, svolta in una modalità che richiama la classe capovolta, è stata di ice breaking. E' stato chiesto agli studenti di individuare una modalità per proporre settimanalmente un quesito ad una classe del biennio allo scopo di incentivare lo studio dell'informatica, strutturando la proposta come una competizione che fosse tanto stimolante quanto formativa.

Successivamente, e' stato richiesto loro di individuare gli strumenti necessari per implementare l'idea, rispettando alcune caratteristiche: dovevano essere gratuiti e adatti a realizzare due piattaforme; una per l'inserimento dei quesiti ed una per la loro soluzione e la memorizzazione delle risposte degli studenti. Sono stati scelti MIT App Inventor, per la creazione delle interfacce in modo che fossero fruibili su cellulare, e mySQL su piattaforma Altvista per contenere il Database dei quesiti e la classifica.

Le conoscenze informatiche di partenza degli studenti erano limitate alla programmazione di base HTML, JavaScript ed SQL; si è quindi reso necessario introdurre alcuni argomenti chiave prima di procedere con la fase implementativa.

Innanzitutto, è stato necessario presentare la programmazione client-server, che è abitualmente parte del programma della classe quinta. Per far comprendere la necessità di una programmazione differente, si è usata una modalità di gioco "unplugged": gli studenti sono divisi in due squadre e

ciascuna squadra in tre gruppi. Gli studenti del primo gruppo, che simulano il server, hanno in dotazione un vocabolario ed una calcolatrice; il secondo, che simula il client, una biro ed un insieme di foto, il terzo infine, che simula l'operatore, un elenco di nomi di animali non banali ed operazioni matematiche difficilmente risolvibili senza l'aiuto di una calcolatrice.

Scopo dell'attività è che l'operatore ottenga, per ogni elemento dell'elenco, un foglio contenente la soluzione dell'operazione, la traduzione del nome dell'animale e la sua foto. Per ottenere ciò, dovrà passare il foglio al client che, dopo aver inoltrato la richiesta al server (unico che abbia il dizionario e la calcolatrice), scriverà sul foglio i risultati ricevuti dal server, incollerà la foto e passerà il risultato finale all'operatore. Per interessare maggiormente i ragazzi al gioco, i fogli dovevano essere passati attraverso delle "teleferiche" autoprodotte con materiali di fortuna, come corde, mollette e nastro adesivo. Risultava vincitore la squadra che avesse completato per prima il lavoro. Lo scopo di questa attività era far comprendere i diversi ruoli di client/server/operatore all'interno del processo.

Il concetto è stato recepito in questo modo senza grandi difficoltà, più agevolmente che con una spiegazione tecnica del docente.

Per l'apprendimento del linguaggio PHP, necessario per la programmazione lato server, si è lavorato in modalità flipped: sono stati consegnati agli studenti i sorgenti di due file HTML e PHP contenenti il codice per visualizzare il contenuto di una tabella ed è stato richiesto di individuare similitudini e differenze fra JavaScript e PHP, e di ricercare il significato e la funzionalità delle istruzioni non conosciute. Nonostante questa fosse concettualmente la parte più impegnativa, ed abbia richiesto parecchio tempo e un'attenta revisione da parte del docente, i ragazzi, dopo un primo momento di sconcerto, hanno trovato stimolante anche questa attività che ha spinto anche i meno motivati a ricercare interpretazioni e spiegazioni. Particolarmente utile e gradito è stato il momento della restituzione, durante il quale gli studenti hanno spesso "gareggiato" per vedere chi avesse trovato più novità o più usi diversi per le nuove funzioni.

Più semplice è stato l'approccio con MIT App Inventor, in quanto la possibilità di programmare usando i blocchi è molto intuitiva e non presenta difficoltà per studenti abituati alla programmazione.

Altro punto critico è stato individuare una modalità di interazione fra un database online e MIT App Inventor, poiché non era possibile utilizzare né TinyDB né CloudDB; infatti, questi strumenti non permettono tutte le funzionalità necessarie e vi era scarsa documentazione di altre soluzioni. Si è scelto quindi di utilizzare il componente Web utilizzando il metodo GET per inviare la query alla pagina PHP di Altvista, che esegue fisicamente la query sul suo database MySQL e restituisce il risultato della esecuzione mediante la funzione echo(). Questa soluzione ha permesso di avere a disposizione tutte le funzionalità dei database MySQL e le potenzialità del linguaggio PHP.

Per la realizzazione delle App è stata utilizzata la metodologia Agile, consistente nel suddividere il compito in sprint autoconsistenti la cui efficacia è immediatamente verificabile. In questo modo il software viene sviluppato e consegnato in piccoli incrementi, permettendo agli studenti di vedere il progresso e di utilizzare le funzionalità man mano che vengono completate. Questo metodo, che era già stato testato in altri progetti, porta molteplici vantaggi, quali maggiore coinvolgimento di tutti gli studenti nelle fasi di testing, l'identificazione precoce dei problemi, una migliore qualità del software dovuta all'attenzione continua ai test e al feedback ed una maggiore motivazione e produttività del gruppo classe grazie all'autonomia e alla responsabilizzazione dei membri.

Gli studenti sono stati divisi in due gruppi: il primo gruppo si è occupato di realizzare l'App per inserire i quesiti nel Database, strutturati in forma di domande a risposta chiusa, e temporizzandoli in modo da renderli disponibili per le risposte alla mezzanotte di domenica di ogni settimana; il secondo gruppo, invece, doveva realizzare l'App che proponeva il quesito settimanale e consentiva ad ogni studente di rispondere una sola volta ad ogni domanda, memorizzando il punteggio acquisito nella misura di 3 punti al primo solutore, 2 al secondo e 1 a tutti gli altri.

Conclusa l'attività di creazione delle App, è iniziata quella di rilascio e test: gli studenti della classe 2° hanno iniziato la competizione, mentre gli studenti della classe 4° si sono dedicati all'inserimento dei quesiti ed alla gestione degli inevitabili errori, creando un gruppo di supporto tecnico incaricato di rispondere nel giro di una giornata alle richieste di assistenza e gestire/risolvere le anomalie.

Occorre notare che i quesiti dovevano essere di tipo logico-deduttivo, ossia che richiedessero più pensiero computazionale e creativo che conoscenze specifiche. Per questo motivo è stato consentito agli studenti di prendere ispirazione da quesiti già presenti sul web o su altri testi, purché venissero personalizzati e fossero poi in grado di fornire e spiegare la soluzione corretta.

Alcune delle fonti utilizzate sono state:

- Testi dei giochi Bebras (Vedi figura),
- Testi di verifiche assegnate (Da una verifica, in cui si chiedeva di determinare con una query i nomi degli studenti inglesi e francesi di una scuola, è scaturito il seguente quesito: Con che connettivo logico si traduce la condizione “tutti gli studenti francesi e tedeschi”?

Risposte possibili: and-or-xor-not),

- Giochi della settimana enigmistica (es. Tre matematici vanno al ristorante, il cameriere chiede “tutti desiderate l'antipasto?” Il primo risponde “non lo so”, il secondo risponde “non lo so” il terzo risponde “no”. Quanti antipasti porterà il cameriere? Risposte possibili: 0-1-3-non può saperlo senza chiedere ancora).



Screen di quesito

Questo ha permesso ai ragazzi di formare loro stessi, prima di formare i compagni, sulla logica e sui meccanismi dell' IT.

Inoltre, per ogni gruppo, si è cercato anche di stimolare particolari soft skills: i creatori dei quesiti dovevano sviluppare interesse, creatività, ma anche precisione, capacità espositiva e rispetto dei tempi; il gruppo dei programmatori, invece, ha dovuto coltivare le proprie capacità di problem solving, creatività, capacità di accettare le critiche, e capacità comunicative; dall'altro lato, la classe che ha utilizzato l'App per rispondere ai quesiti doveva mostrare precisione, rispetto dei tempi, ma anche incrementare l'interesse per l'informatica e la propria autostima.

3 Analisi dei risultati

Il risultato alla fine della sperimentazione è stato pienamente raggiunto, soprattutto per la classe che ha curato la codifica e la produzione dei quesiti.

La classe 4° SIA, infatti, si presentava problematica per via della presenza di studenti di recente immigrazione, studenti con difficoltà di apprendimento e studenti con carenze motivazionali, facendo sì che i pochi elementi di spicco faticavano a trovare il proprio ruolo di leader. In generale, ne risultava una classe disunita e rassegnata al ruolo di “classe scarsa e poco motivata”. Sono quindi rimasti spiazzati dall'aver ricevuto fiducia e autonomia da parte dell'insegnante per lo sviluppo del progetto, e questo ha agito positivamente, spronando tutti a dare il meglio di sé. La divisione in gruppi e la varietà dei ruoli ha consentito a ciascuno studente di trovare in se stesso qualità che non si riconosceva. Tutto ciò ha anche permesso alla classe di riconoscersi come gruppo con uno scopo comune, ha notevolmente migliorato il profitto scolastico e tale risultato è stato mantenuto fino all'ultimo anno, portando non solo ad un migliore andamento scolastico, ma anche un incentivo a proseguire gli studi dopo il diploma.

Al raggiungimento di questo risultato hanno contribuito sia la scelta di far decidere agli studenti come strutturare il prodotto finito, sia aver realizzato qualcosa che è stato testato nel suo reale funzionamento, non solo un progettino fine a se stesso.

Inoltre ha giovato aver dovuto cercare, sia pure con il sostegno della docente, le soluzioni appropriate, invece di limitarsi ad applicare tecniche spiegate in classe.

Al buon risultato del progetto ha contribuito anche la classe 2° affrontando la sfida con entusiasmo e serietà, calandosi nel ruolo e trattando i ragazzi la classe 4° non come compagni, ma come veri e propri mentori, pregustando lo scambio dei ruoli auspicato quando fossero arrivati alla classe 4°.

Infine si è realizzato che le librerie approntate per interfacciare il Database con MIT App Inventor sono talmente generali e ben progettate, da poter essere rese disponibili per altre scuole/classi che desiderassero replicare il progetto, anche classi della secondaria inferiore, in quanto dotate di un'interfaccia standard che non richiede, da parte di chi sviluppa l'App, alcuna conoscenza della programmazione Database o del linguaggio SQL.

4 Conclusione

Il metodo utilizzato dimostra che incentivare la creatività e l'entusiasmo nei nostri studenti è la chiave vincente per il loro progresso; occorre però per farle emergere, fare assumere agli studenti il ruolo di protagonista della propria formazione, aiutandoli a creare il proprio metodo di lavoro con la guida ed il supporto del docente.

Parafrasando la frase di Confucio: "Se ascolto dimentico, se vedo ricordo, se faccio capisco e se invento un modo nuovo per farlo, cresco".

Bibliografia

- [1] M. Missiroli, D. Russo, P. Ciancarini, "Una didattica Agile per la programmazione" [Online]. Available: https://www.researchgate.net/publication/309576737_Una_didattica_Agile_per_la_programmazione.
- [2] M. Maglioni, "Classe capovolta: ecco quali sono i pilastri" . [Online]. Available: <https://www.tuttoscuola.com/classe-capovolta-quali-pilastri>.

“L’intelligenza artificiale in aiuto allo studio”: un breve corso Moodle

Giuliana Barberis

Liceo Scientifico M. Curie di Pinerolo
giuliana.barberis@gmail.com

Abstract

In questo mio contributo descrivo un’ “esperienza sul campo”: durante la giornata di autogestione della mia scuola, il gruppo degli studenti organizzatore mi ha chiesto di proporre una lezione sull’Intelligenza Artificiale, e io ho pensato di proporre una lezione sull’uso degli strumenti più che sul loro funzionamento dal punto di vista informatico.

In questo momento storico è necessario che noi docenti si tenga sotto controllo l’evoluzione di queste tecnologie, imparando ad utilizzarle e utilizzandole anche al fine di considerarle criticamente.

A mio avviso, il nostro approccio all’insegnamento, alla preparazione degli studenti nella nostra disciplina, alla richiesta di elaborati personali e all’adempimento delle attività didattiche deve necessariamente evolversi in risposta allo sviluppo di queste nuove tecnologie.

Ho quindi creato un corso Moodle, liberamente accessibile entrando come ospite:
<https://www.barberisg.it/moodle4/course/view.php?id=27>

progettato per fornire agli studenti una panoramica sull’Intelligenza Artificiale e sul suo utilizzo consapevole.

Il corso è strutturato in modo da essere fruibile sia in autonomia che con il supporto di un docente. Gli studenti possono accedere a materiali didattici, esercitazioni interattive e video esplicativi. Il corso è inoltre arricchito da contributi di esperti del settore e da riflessioni sui rischi e sulle implicazioni etiche dell’AI.

Gli obiettivi del corso sono quelli di sensibilizzare gli studenti sull’importanza di un uso etico e responsabile dell’AI, svilupparne le competenze necessarie per utilizzarla in modo efficace e creativo per il loro apprendimento e promuovere il dibattito critico sui rischi e sulle opportunità dell’AI per la società.

1 Introduzione

Il corso Moodle che propongo è stato progettato per fornire agli studenti spunti di riflessione e strumenti per guidarli nell’uso consapevole delle tecnologie di Intelligenza Artificiale attualmente disponibili.

Poiché questi strumenti sono in costante aggiornamento e l’evoluzione è talmente veloce che quello che si scrive oggi può essere già obsoleto domani, ho deciso di realizzare un corso Moodle disponibile a tutti per poterlo tenere aggiornato con informazioni, materiali e nuovi strumenti con i quali posso venire a contatto.

Il corso si trova all’indirizzo:

<https://www.barberisg.it/moodle4/course/view.php?id=27>

è accessibile agli ospiti e comprende una serie di argomenti che spiego nel dettaglio nei prossimi capitoli.



Figura 1 - QRcode per accedere al corso

L'indirizzo di questo corso può anche essere fornito agli studenti senza un'introduzione, ma è stato progettato per essere introdotto e spiegato in tutte le sue parti in una lezione in laboratorio (di 2/3 ore circa) in modo da permettere la sperimentazione degli strumenti illustrati, per poi essere lasciato in libera consultazione.

2 Raccomandazione Unesco sull'uso etico dell'AI

Un qualsiasi testo che abbia come argomento l'Intelligenza Artificiale non può non iniziare con le opportune raccomandazioni sull'uso opportuno che si deve fare di questo strumento, senz'altro molto potente e utile ma anche molto pericoloso e eticamente controverso, in quanto è richiesta una riflessione profonda sui limiti e sulle responsabilità di chi lo utilizza e lo sviluppa.

Innanzitutto, spendiamo qualche parola riguardo alle questioni etiche sullo sviluppo dell'AI, sono questioni molto delicate che meritano senz'altro di essere dibattute.

Nel maggio 2023 l'UNESCO ha pubblicato il suo 'The UNESCO Recommendation on The Ethics of AI: Shaping the Future of Our Societies' [1], la Commissione Nazionale Italiana per l'Unesco ne ha curato la traduzione in italiano: "L'etica dell'IA: modellare il futuro delle nostre società" [2], in questa brochure possiamo trovare molti spunti per stimolare un dibattito con i nostri colleghi e con i nostri studenti.



Figura 2 - Brochure dell'Unesco

Il testo si concentra sui principi fondamentali che dovrebbero governare lo sviluppo e l'implementazione delle tecnologie di AI, con particolare attenzione ai diritti umani, all'inclusività, alla sostenibilità e alla trasparenza. Inoltre, il documento fornisce raccomandazioni per i governi, le aziende e le istituzioni accademiche su come integrare questi principi nelle loro politiche e pratiche. Si sottolinea l'importanza della cooperazione internazionale e della necessità di quadri normativi che proteggano gli individui e le società dagli eventuali rischi associati all'AI.

3 AI act

La Commissione Europea ha proposto il primo quadro giuridico sull'AI, che affronta i rischi dell'AI.

- Il 9 dicembre 2023 il Parlamento europeo e il Consiglio hanno raggiunto un accordo politico sulla legge sull'AI.
- Il 13 marzo 2024 ha approvato la versione finale dell'European Artificial Intelligence Act (EU AI Act) [3], primo testo normativo al mondo a regolamentare i sistemi di IA e il loro utilizzo al fine di garantire il rispetto dei diritti umani e civili delle persone.

Un aspetto particolarmente rilevante del AI Act è la classificazione delle applicazioni di intelligenza artificiale, che le posiziona all'interno di una piramide di rischio. Questa categorizzazione differenzia i requisiti e le precauzioni da adottare in base alla fascia di rischio in cui l'applicazione di AI si colloca.



Figura 3 - Piramide del rischio per l'AI Act

Rischio inaccettabile: Sono vietati i sistemi di AI che presentano un rischio inaccettabile per la sicurezza, la salute, i diritti fondamentali o la dignità umana. Esempi di sistemi di AI in questa categoria includono sistemi di IA che utilizzano il punteggio sociale da parte dei governi, il riconoscimento facciale per la sorveglianza di massa, la manipolazione emotiva per scopi dannosi.

Rischio alto: I sistemi di AI che presentano un rischio alto per la sicurezza, la salute, i diritti fondamentali o la dignità umana devono essere conformi a requisiti rigorosi per la loro progettazione, sviluppo e utilizzo. Esempi di sistemi di AI in questa categoria includono sistemi di IA utilizzati in contesti critici come la diagnostica medica, il controllo del traffico aereo o la guida autonoma, sistemi di IA utilizzati per la rilevazione di frodi o la valutazione del credito o per la personalizzazione di contenuti o pubblicità.

Rischio limitato: I sistemi di AI che presentano un rischio limitato per la sicurezza, la salute, i diritti fondamentali o la dignità umana devono comunque conformarsi a determinati obblighi di trasparenza e vigilanza. Esempi di sistemi di AI in questa categoria includono Chatbot, Software di riconoscimento vocale, Sistemi di raccomandazione.

Rischio minimo: I sistemi di AI che presentano un rischio minimo per la sicurezza, la salute, i diritti fondamentali o la dignità umana non sono soggetti a requisiti specifici ai sensi dell'AI Act. Esempi di sistemi di AI in questa categoria includono Videogiochi e Filtri antispam.

4 AI generativa

Gli strumenti di intelligenza artificiale attualmente più utilizzati appartengono alla categoria dell'AI generativa. Questa tipologia di intelligenza artificiale si concentra sulla creazione autonoma

di contenuti, come testi, immagini o suoni, mediante l'impiego di modelli complessi di apprendimento automatico che generano output originali basati sui dati con cui sono stati addestrati.

Gli esempi di AI generativa includono la creazione di testi tramite sistemi come ChatGPT e Gemini, o la generazione di immagini a partire da descrizioni testuali attraverso piattaforme come DALL-E di Bing Image Creator e Lucid di ChatGPT.



Figura 4 - logo di chatGPT

È fondamentale ricordare che i risultati prodotti da questi strumenti devono essere sottoposti a un'attenta verifica e non possono essere accettati senza un esame critico, in quanto l'intelligenza artificiale si basa su dati non sempre verificati e potrebbe quindi fornire risposte errate.

Pertanto, è imperativo che tali tecnologie non vengano affidate agli studenti senza una guida che li renda consapevoli dei rischi o senza il filtro di una persona qualificata.

Esempi di AI generativa possono essere:

- ChatGPT
- Gemini
- Perplexity
- Copilot in Microsoft Bing
- Claude.

5 L'importanza di progettare bene il prompt

Per ottenere risposte accurate da un'applicazione di intelligenza artificiale, è fondamentale formulare la domanda, denominata "prompt", in modo corretto.

La tecnica di composizione di un prompt è diventata oggetto di studio approfondito. A tal proposito, si segnala un articolo rilevante sull'ingegneria dei prompt, scritto dal Prof. Alessandro Fiori del Politecnico di Torino, disponibile al link [4] nella bibliografia.

In sostanza, è importante specificare nel prompt il comando da impartire all'AI, il contesto in cui si opera, i dati di input utili a facilitare la risposta, e il formato desiderato per la risposta stessa. È raro riuscire a formulare immediatamente il prompt corretto; spesso è necessario riformulare la richiesta più volte per ottenere la risposta desiderata, un esempio di prompt efficace è illustrato in figura.



Figura 5 - un esempio di Prompt

Questo argomento è talmente importante che sono nate delle “librerie di prompt”, che li forniscono già belli pronti, un esempio si può trovare al link [5] nella bibliografia.

6 Copiare durante le verifiche

Le tecnologie di AI generativa sono fonte di grandi aspettative per gli studenti che vedono un modo semplice e veloce per risolvere le consegne dei compiti assegnati per casa o addirittura per prendere “spunto” nel momento delle verifiche in classe o in laboratorio.

D'altra parte sono anche fonte di grande preoccupazione per i docenti che temono che assegnare compiti per casa possa diventare del tutto inutile e si immaginano che la sorveglianza alle verifiche possa diventare molto impegnativa se non inefficace.

In qualità di docente, non posso che concordare con tali osservazioni, ed è indubbio che il nostro approccio all'insegnamento debba adattarsi immediatamente per tenerne conto. Sarà necessario impiegare tutte le nostre risorse creative e la nostra esperienza per trasformare questa significativa sfida in una grande opportunità, sia per noi che per i nostri studenti.

Nulla possiamo dire ai nostri studenti per dissuaderli da queste pratiche, se le raccomandazioni all'onestà sono inutili e se non serve ricordare loro che è necessario investire tempo ed energia nello studio e nella preparazione, per ottenere risultati che riflettono realmente le loro capacità, l'unica cosa che rimane è avvertirli sui rischi che corrono utilizzando questi strumenti senza conoscerli a fondo.

Possiamo metterli in guardia in merito alla risposta che potrebbero ottenere da un'AI generativa se non sono in grado di leggerla in maniera critica e non si può considerare in maniera critica una risposta se non la si conosce già, se non si è quindi già studiato l'argomento.



Figura 6 - Le allucinazioni

Infatti l'AI potrebbe dar loro risposte eccessivamente accurate, è in grado di generare risposte molto dettagliate e articolate che potrebbero sembrare perfette, tuttavia, risposte che appaiono troppo sofisticate rispetto allo stile abituale dello studente possono destare facilmente sospetti.

Oppure potrebbe formulare risposte insufficienti, superficiali o incomplete, che non rispondono adeguatamente alle domande poste. Affidarsi a queste risposte può compromettere la qualità del loro lavoro e riflettersi negativamente sul loro rendimento scolastico.

Un caso che non si sospetta è che l'AI può dare risposte infarcite di contenuti privi di logica, può generare informazioni inesatte o completamente errate, note come "allucinazioni". Questi errori possono essere difficili da individuare e, se non verificati, possono portare alla presentazione di risposte fuorvianti o imprecise, danneggiando ulteriormente la propria credibilità.

7 AI per studiare

Molto più significativo è dare spunti e idee ai nostri studenti in modo che usino l'AI per accrescere proprie possibilità e per trovare sistemi di ausilio allo studio; in questa sezione del corso si trovano esempi di buone pratiche per stimolare l'uso positivo dell'AI:

- Per ottenere sintesi di argomenti.
- Per ottenere esempi applicativi di un argomento che stanno studiando.
- Per controllare le fonti di una risposta dell'AI e verificarne la veridicità (Perplexity cita le fonti e quasi sempre anche Gemini).
- Per farsi correggere elaborati e anche farsi dare una valutazione in base a una griglia di correzione.
- Per farsi dare degli esercizi e confrontare la propria soluzione con la soluzione proposta dall'AI.
- Per farsi disegnare schemi concettuali per facilitare l'identificazione e la memorizzazione dei concetti principali di un argomento.

8 Generatori di immagini

L'uso di generatori di immagini come ausilio allo studio può interagire con diversi aspetti:

- per creare immagini uniche per illustrare concetti per presentazioni e progetti

- per generare rappresentazioni visive di concetti astratti per migliorarne la comprensione.
- per sviluppare la propria creatività sperimentando con diverse idee visive rapidamente.
- per accrescere le proprie competenze digitali sperimentando tecnologie all'avanguardia.
- per esplorare nuove tecniche e stili artistici.
- per creare infografiche e rappresentazioni visive di dati complessi.

Ho voluto quindi riservare una sezione anche a questo tipo di AI generativa, anche se al momento non è ancora a livelli ottimali.

9 Presentazioni

L'intelligenza artificiale sta trasformando il modo in cui creiamo e gestiamo le presentazioni, offrendo nuove opportunità agli studenti per migliorare la qualità e l'efficacia delle loro esposizioni.

Molte volte può capitare di avere una relazione e di volerla trasformare in presentazione, avendo a disposizione un testo è quasi un procedimento meccanico ricavarne delle slide, quindi potremmo pensare di delegarne lo svolgimento ad un'AI.

In realtà potremmo chiedere all'AI di fare una presentazione da zero, solo fornendo l'argomento e alcune precisazioni, in questo caso però la risposta dell'AI va controllata molto bene e confrontata con altre fonti.

In questa sezione del corso do alcuni strumenti per poter ottenere la base per una presentazione che andrà poi migliorata e adattata alle proprie esigenze.

10 Dietro all'AI

Questo corso è progettato per studenti che non hanno competenze specifiche di informatica, ho voluto quindi anche riservare una sezione del corso ad un'attività di laboratorio per mostrare loro come vengono addestrate le applicazioni di machine learning.

L'obiettivo è far toccare loro con mano che l'intelligenza artificiale impara dai dati che le vengono forniti dagli umani e dunque la bontà del suo prodotto è direttamente proporzionale all'esattezza dei dati forniti e la risposta ha sempre una data probabilità di essere vera, che raramente è del 100%.

In questa sezione è compreso un piccolo tutorial per eseguire un'attività con la “teachable machine” [6], una piattaforma didattica per l'addestramento di strumenti di machine learning.

Nella sezione c'è anche il link a una lezione del Prof. Zecchina [7] sui diversi tipi di addestramento.

11 Video

Per stimolare ulteriormente la discussione, soprattutto sui rischi che l'AI generativa possa generare immagini o video fake, ho incluso un mio video generato con la piattaforma HeyGen [8], dove un breve video viene convertito in diverse lingue generando di fatto dei fake video, innocui nel video presentato ma l'idea è quella di introdurre in modo sottile una riflessione sull'argomento.

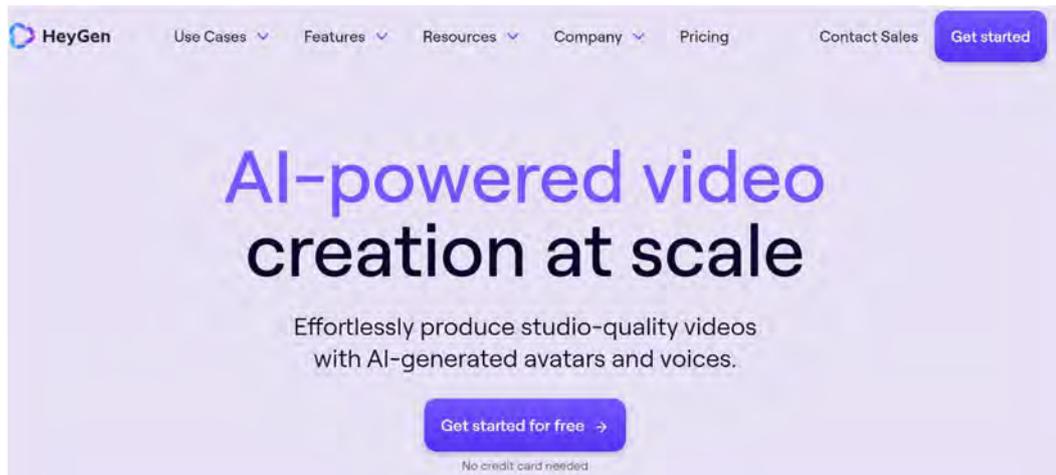


Figura 7 - HeyGen

Bibliografia

- [1] UNESCO, «The UNESCO Recommendation on The Ethics of AI: Shaping the Future of Our Societies,» 2023. [Online]. Available: <https://www.unesco.nl/sites/default/files/inline-files/Unesco%20AI%20Brochure.pdf>.
- [2] UNESCO, «L'etica dell'IA: modellare il futuro delle nostre società,» 2023. [Online]. Available: <https://www.unesco.it/wp-content/uploads/2023/11/Brochure-su-Raccomandazione-UNESCO-sullIntelligenza-Artificiale-1.pdf>.
- [3] Euronews, «Intelligenza artificiale: il Parlamento Ue approva il testo finale dell'AI Act,» 2024. [Online]. Available: <https://youtu.be/goJDQq4tQ8I>.
- [4] A. Fiori, «Ingegneria Dei Prompt,» 2023. [Online]. Available: <https://flowygo.com/blog/ingegneria-dei-prompt/>.
- [5] PuntoInformatico, «Come utilizzare il nuovo Prompt Store Claude 3 AI di Anthropic,» 2024. [Online]. Available: <https://www.punto-informatico.it/come-utilizzare-il-nuovo-prompt-store-claude-3-ai-di-anthropic/>.
- [6] T. Machine, «The teachable machine,» [Online]. Available: <https://teachablemachine.withgoogle.com/>.
- [7] P. R. Zecchina, «AIMOOC 4.4 - Reti neurali artificiali - Riccardo Zecchina,» 2021. [Online]. Available: <https://www.youtube.com/watch?v=-h4F-4wutFE>.
- [8] «HeyGen,» [Online]. Available: <https://www.heygen.com/>.
- [9] 2023. [Online].

Intelligenza Artificiale Generativa nella Didattica: Un'Analisi dei dati raccolti nelle attività formative

Andrea Canesi¹

¹Nuvie APS

andrea.canesi@nuvie.it

Abstract

Al fine di valutare l'impatto della formazione sull'intelligenza artificiale generativa tra i docenti della scuola italiana, sono stati analizzati i dati raccolti in occasione di una serie di attività formative attraverso questionari somministrati all'inizio e alla fine dei corsi. I corsi, della durata di 6-9 ore, sono stati offerti sia in presenza che online e hanno coperto vari aspetti delle IA generative, dal funzionamento teorico all'applicazione pratica nella didattica. Questo studio confronta le percezioni e l'utilizzo delle IA generative tra i docenti prima e dopo la formazione.

1 Introduzione

Con l'avanzare delle tecnologie dell'intelligenza artificiale (IA), è diventato essenziale per i docenti di tutti gli ordini e gradi scolastici comprendere e integrare queste tecnologie nella didattica. Questo studio analizza l'impatto di corsi di formazione sulle IA generative, esplorando come le percezioni e l'utilizzo di queste tecnologie siano cambiati tra i docenti che hanno partecipato ai corsi.

2 I corsi di formazione sulle IA generative

I corsi hanno seguito un programma strutturato in quattro moduli:

1. Teoria dell'IA: Introduzione all'IA, machine learning (ML), modelli di addestramento (supervisionato, non supervisionato e rinforzato), deep learning, reti neurali artificiali, algoritmi discriminativi e generativi, e pericoli legati all'utilizzo delle IA.
2. Utilizzo Pratico delle IA Generative: Applicazioni delle IA generative nella creazione di testo, immagini, musica e video. Esempi di utilizzo nella produzione di materiale didattico e non didattico.
3. IA Generative e Studenti: Utilizzo delle IA generative da parte degli studenti per compiti scolastici come riassunti, parafrasi, traduzioni e risoluzione di problemi matematici.
4. Formazione per Docenti della Scuola Secondaria: Metodologie per insegnare agli studenti a utilizzare le IA generative per migliorare lo studio.

3 Metodologia

Sono stati somministrati due questionari: uno introduttivo e uno finale. Il questionario introduttivo è stato compilato prima dell'inizio del corso, mentre il questionario finale è stato compilato al termine del corso. I dati raccolti includono la frequenza di utilizzo delle IA generative, la percezione della formazione disponibile, il livello di preparazione individuale e le opinioni sui potenziali vantaggi e sfide legate all'uso delle IA generative nella didattica.

4 Analisi delle risposte

I due questionari contenevano le stesse domande, suddivise nei seguenti gruppi:

1. Conoscenza e Uso dell'IA
2. Percezione dell'Uso da parte degli studenti dell'IA
3. Percezione dell'impatto delle IA Generative in generale e sulla Didattica

Ad oggi sono pervenute 115 risposte al questionario introduttivo e 66 al questionario finale.

I docenti provengono dalla maggior parte delle regioni italiane con una particolare concentrazione in Liguria dove sono stati svolti tre corsi in presenza.

Alla formazione hanno partecipato docenti di tutti gli ordini e gradi di scuola, con una leggera prevalenza della scuola secondaria di II grado (circa 41%).

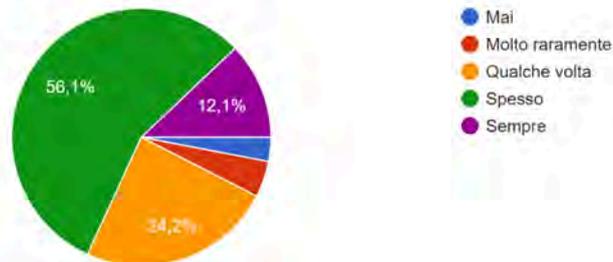
4.1 Conoscenza e Uso dell'IA

- Conoscenza delle IA Generative: Prima del corso, molti docenti dichiaravano di avere una conoscenza limitata o nulla delle IA generative. Dopo il corso, si è osservato un aumento nella dichiarazione di conoscenza moderata o avanzata delle IA generative.
- Uso delle IA Generative: La frequenza di utilizzo delle IA generative nella didattica era bassa prima del corso, con la maggior parte dei docenti che dichiarava di non utilizzarle mai o raramente. Dopo il corso, c'è stato un incremento significativo nell'uso delle IA generative, con più docenti che hanno iniziato a utilizzarle frequentemente nelle loro attività didattiche.



Quanto frequentemente utilizzi tecnologie basate sull'intelligenza artificiale in generale?

66 risposte

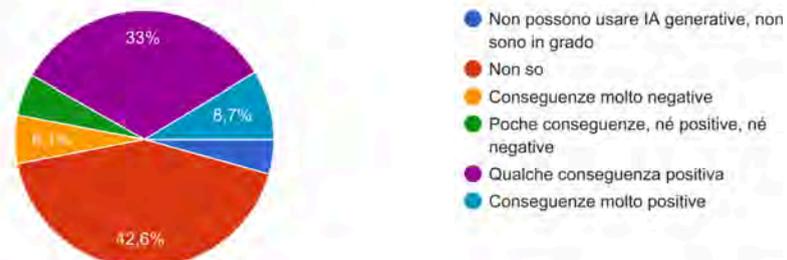


4.2 Percezione dell'Uso da parte degli Studenti dell'IA

- Percezione Pre-Corso: Prima del corso, molti docenti erano preoccupati che gli studenti potessero utilizzare le IA generative come scorciatoia per evitare il lavoro scolastico.
- Percezione Post-Corso: Dopo il corso, i docenti hanno iniziato a vedere le IA generative come strumenti potenziali per migliorare l'apprendimento degli studenti. Hanno riconosciuto che, se usate correttamente, queste tecnologie possono aiutare gli studenti a comprendere meglio i contenuti, a fare esercizi di autovalutazione e a ricevere spiegazioni dettagliate su argomenti complessi.

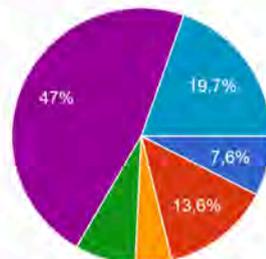
Se i tuoi studenti facessero uso dell'IA generativa per studiare o fare i compiti che conseguenze pensi avrebbe sulla loro educazione?

115 risposte



Se i tuoi studenti facessero uso dell'IA generativa per studiare o fare i compiti che conseguenze pensi avrebbe sulla loro educazione?

66 risposte



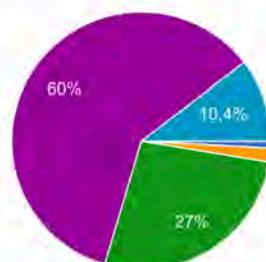
- Non possono usare IA generative, non sono in grado
- Non so
- Conseguenze molto negative
- Poche conseguenze, né positive, né negative
- Qualche conseguenza positiva
- Conseguenze molto positive

4.3 Percezione dell'Impatto delle IA Generative in generale e sulla didattica

- Impatto Generale delle IA Generative: Prima del corso, le opinioni sui benefici delle IA generative erano contrastanti, con alcuni docenti che vedevano il potenziale per migliorare l'insegnamento e altri che erano scettici riguardo ai benefici effettivi.
- Impatto sulla Didattica: Dopo il corso, c'è stato un cambiamento positivo nella percezione dell'impatto delle IA generative sulla didattica. I docenti hanno riconosciuto che le IA generative possono personalizzare l'apprendimento, migliorare la qualità dei materiali didattici, automatizzare la valutazione e supportare la creatività degli insegnanti.

Quale impatto pensi che le IA generative possano avere sul tuo lavoro, considerando l'insegnamento e le attività non didattiche (compilazione documenta...ione con scuola, docenti, studenti, genitori, ecc.)

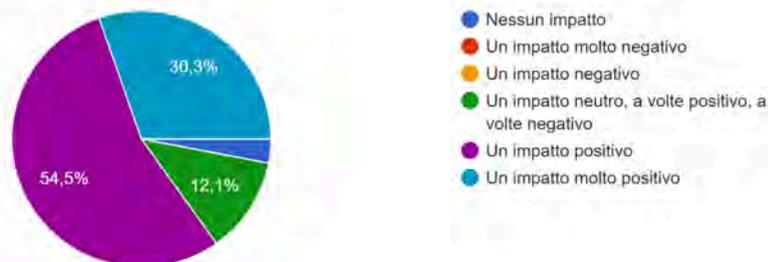
115 risposte



- Nessun impatto
- Un impatto molto negativo
- Un impatto negativo
- Un impatto neutro, a volte positivo, a volte negativo
- Un impatto positivo
- Un impatto molto positivo

Quale impatto pensi che le IA generative possano avere sul tuo lavoro, considerando l'insegnamento e le attività non didattiche (compilazione con scuola, docenti, studenti, genitori, ecc.)

66 risposte



4.4 Analisi delle risposte aperte

Le risposte aperte mostrano un'evoluzione nelle opinioni dei docenti riguardo alle opportunità e alle sfide delle IA generative. Prima del corso, molti erano incerti o avevano idee limitate sui benefici delle IA. Alcuni esempi:

- *“...Sfide: Dipendenza dalla tecnologia, Privacy e sicurezza dei dati, Disuguaglianza digitale, Formazione degli insegnanti”*
- *“speriamo che non si sostituisca gli insegnanti”*
- *“...Dipendenza dall'IA: È importante garantire che gli studenti non diventino troppo dipendenti dall'IA generativa e che sviluppino ancora le capacità di pensiero critico e di risoluzione dei problemi.”*
- *“È importante notare che l'IA generativa è ancora una tecnologia in via di sviluppo e il suo impatto sull'istruzione è ancora in gran parte inesplorato. È necessaria una ricerca ulteriore per comprendere appieno i potenziali benefici e rischi dell'utilizzo dell'IA generativa nell'istruzione.”*

Il parere generale era comunque positivo ed ottimistico.

Dopo il corso, emerge una maggiore comprensione delle potenzialità delle IA generative, come la personalizzazione dell'apprendimento, il supporto alla creatività e la semplificazione di compiti amministrativi. Tuttavia, sono state ancora espresse preoccupazioni riguardo alla privacy dei dati e all'equità nell'accesso alle tecnologie. Esempi:

- *“L'IA generativa è uno strumento potente che può essere utilizzato per migliorare l'apprendimento di tutti gli studenti. Se utilizzata in modo responsabile e consapevole, l'IA generativa può contribuire a creare un futuro più equo e prospero per tutti.”*
- *“Ho scoperto che le IA generative forniscono davvero enormi opportunità per la didattica (e non solo), però ritengo che, solo se saremo in grado di conoscerle, gestirle e utilizzarle in maniera adeguata e consapevole, potranno rappresentare davvero dei vantaggi. Questa è la grande sfida.”*
- *“cambierà il modo di preparare il materiale e le lezioni e il modo di studiare ed apprendere degli studenti. Come insegnanti dovremo educare gli studenti al corretto utilizzo di questi strumenti. Questi strumenti facilitano la ricerca delle informazioni, e producono delle risposte, in un processo che accelera in modo veloce; le nuove generazioni devono imparare ad adattare il proprio pensiero critico”*

5 Risultati

I risultati mostrano un aumento significativo nell'apprezzamento delle IA generative tra i docenti dopo aver partecipato ai corsi. La percezione della disponibilità di formazione sull'IA generativa è migliorata, così come il livello di preparazione individuale dichiarato dai docenti.

I dati raccolti evidenziano un cambiamento positivo nelle percezioni e nell'utilizzo delle IA generative. I docenti hanno riconosciuto le potenzialità delle IA generative nel personalizzare l'apprendimento, sviluppare contenuti educativi avanzati e supportare la creatività. Tuttavia, emergono anche preoccupazioni etiche e sociali, come la privacy dei dati degli studenti e l'equità nell'accesso alle tecnologie.

6 Conclusioni

I corsi di formazione hanno avuto un impatto positivo sul livello di preparazione e sull'adozione delle IA generative tra i docenti. Il monitoraggio continuo e ulteriori formazioni saranno cruciali per affrontare le sfide emerse e per sfruttare appieno le opportunità offerte dalle IA generative nella didattica.

7 Futuri sviluppi

La ricerca proseguirà sottoponendo i questionari in occasione delle prossime attività formative e inviando nuovamente lo stesso questionario qualche mese dopo l'inizio del prossimo anno scolastico per analizzare il cambio di percezione e di utilizzo di questi strumenti e con interviste personalizzate ai docenti che hanno dato disponibilità.

Living Syllabus per l'informatica per la cittadinanza digitale e per le digital humanities: un approccio di sviluppo collaborativo

Giovanni Adorni

DIBRIS - Università degli Studi di Genova

Viale Francesco Causa 13 - Genova

giovanni.adorni@unige.it

Abstract

Questo articolo presenta il progetto di sviluppo di un "living syllabus" per l'informatica per la cittadinanza digitale e per le Digital Humanities, secondo un approccio collaborativo attraverso un portale web, con fasi di consultazione, discussione, affinamento e votazione. Il processo mira a creare un documento flessibile e aggiornabile, adatto alle rapide evoluzioni tecnologiche. Il syllabus copre le cinque aree di competenza digitale previste dal framework DigComp 2.2 e verrà rilasciato con licenza Creative Commons. L'obiettivo è fornire uno strumento efficace e condiviso per la progettazione di percorsi formativi in ambito informatico.

1 Introduzione

L'informatica e le competenze digitali sono diventate pilastri fondamentali della società contemporanea, influenzando profondamente ogni aspetto della vita quotidiana e professionale. La rapida evoluzione tecnologica richiede una continua adattabilità e un aggiornamento costante delle conoscenze, con implicazioni significative per cittadini, educatori e futuri professionisti. In questo scenario di trasformazione digitale, emerge la necessità critica di un approccio formativo flessibile e aggiornabile.

Il framework europeo **DigComp 2.2** si afferma come punto di riferimento cruciale, offrendo una struttura completa per identificare e sviluppare le competenze digitali essenziali [1]. Basandosi su questo framework, il presente articolo propone un **living syllabus**, concepito come guida dinamica e condivisa per la progettazione di percorsi formativi sulle competenze digitali. Questo syllabus si rivolge a tre target specifici, ciascuno con esigenze e livelli di padronanza distinti:

- *Cittadini digitali*: individui che aspirano a esercitare una cittadinanza digitale attiva, mirando a un livello di padronanza 4 secondo DigComp 2.2. Questo livello implica la capacità di affrontare compiti ben definiti ma non sistematici, con autonomia e comprensione dei processi.
- *Docenti digitali*: insegnanti che devono acquisire competenze avanzate per trasmetterle efficacemente agli studenti, puntando a un livello di padronanza 6. Questo comporta la gestione di compiti complessi e l'adattamento a contesti mutevoli.
- *Studenti digitali*: studenti universitari, principalmente nell'ambito di percorsi di studio non tecnologici e delle Digital Humanities, che necessitano di una prospettiva avanzata sulle competenze informatiche, anch'essi mirando al livello 6 di padronanza.

Tali livelli, allineati con il vocabolario dell'**EQF - European Qualification Framework** [2], riflettono la necessità di competenze differenziate per ciascun gruppo target. Il syllabus, facendo riferimento alla tassonomia di Bloom [3] e a EQF, definisce per ciascun livello la complessità dei compiti, il grado di autonomia e il dominio cognitivo richiesto.

L'approccio del "*living syllabus*" risponde all'esigenza di un documento in continua evoluzione, capace di adattarsi al rapido progresso tecnologico. Lo sviluppo collaborativo di questo syllabus, che coinvolge esperti del settore in un processo continuo di revisione e aggiornamento attraverso un portale pubblicamente accessibile, garantisce la sua rilevanza e efficacia nel tempo.

Questo progetto riconosce il ruolo cruciale della scuola e dell'università nella formazione digitale. Gli insegnanti, in particolare, necessitano di una solida preparazione per formare adeguatamente gli studenti, futuri lavoratori in un mondo sempre più digitalizzato. Allo stesso tempo, il syllabus mira a preparare i cittadini per una partecipazione attiva nella società digitale e gli studenti universitari per le sfide specifiche delle Digital Humanities. In sintesi, questo living syllabus si propone come strumento essenziale per affrontare le sfide della digitalizzazione, offrendo un approccio flessibile e mirato per lo sviluppo delle competenze digitali fondamentali in diversi contesti formativi e professionali, e nel seguito vengono fornite indicazioni precise su come partecipare allo sviluppo collaborativo di questo syllabus.

2 Syllabus e processo di sviluppo collaborativo

Per avviare un processo di sviluppo collaborativo e condiviso di un syllabus efficace, è fondamentale partire da una proposta di riferimento concreta. Questa base iniziale serve come punto di partenza per stimolare la discussione, raccogliere feedback e facilitare il perfezionamento iterativo del contenuto. La proposta qui presentata si basa sulle cinque aree di competenza digitale definite nel framework DigComp 2.2, e per ciascuna di queste aree, il syllabus dettaglia i contenuti di conoscenza e le abilità richieste. Data la limitazione di spazio, si rimanda però ad un portale web dedicato la declinazione di tutte le voci del syllabus.

L'accesso al portale è tramite SPID o CIE, per garantire così la sicurezza e l'autenticità dei contributi. L'indirizzo è <https://scuoladigitale.aulaweb.unige.it/>, selezionando poi il bottone "Informatica per la Cittadinanza Digitale e per le Digital Humanities". In **Figura 1** sono comunque riportati a grandi linee i livelli di padronanza attesi per i docenti digitali e per gli studenti digitali, per ogni competenza chiave di ognuna delle 5 aree, tramite verbi di azione (essere in grado di spiegare, valutare, scegliere, ...), rimandando anche in questo caso, per limitazioni di spazio, al portale web sopra citato per una descrizione di maggior dettaglio, e per la descrizione dei livelli di padronanza per i cittadini digitali.

Il cuore di questo progetto è costituito da: approccio collaborativo e natura evolutiva, elementi essenziali per creare un syllabus che sia veramente rappresentativo e sempre attuale. Per facilitare questo processo è stato predisposto il portale web di cui sopra. Il processo di sviluppo proposto si articola in diverse fasi, le cui tempistiche saranno riportate sul portale stesso. Le fasi sono:

1. Consultazione iniziale - Il portale offre agli utenti la possibilità di interagire direttamente con il contenuto proposto. Attraverso chat dedicate per ciascuna delle cinque aree di competenza del syllabus, i partecipanti possono lasciare commenti di carattere generale, proporre modifiche specifiche, aggiungere note o suggerire revisioni. Questa fase è cruciale per raccogliere una vasta gamma di prospettive e expertise.
2. Integrazione e discussione - Al termine della fase di consultazione, tutte le proposte ricevute verranno accuratamente analizzate e integrate. Successivamente, si terrà una riunione online a cui saranno invitati tutti coloro che hanno contribuito attivamente. Questo momento di confronto diretto permetterà di discutere in dettaglio le varie proposte, favorendo un dialogo costruttivo tra esperti del settore.
3. Pubblicazione versione beta e affinamento - Sulla base di quanto emerso durante le fasi precedenti, verrà pubblicata una versione beta del syllabus. Questa fase intermedia consente ulteriori perfezionamenti e aggiustamenti, assicurando che il documento rifletta accuratamente il consenso emergente della comunità.
4. Votazione e rilascio ufficiale - Il syllabus finale sarà sottoposto a votazione all'interno della comunità di tutti coloro che hanno contribuito attivamente. Una volta approvato, verrà rilasciato sotto licenza Creative Commons CC BY-SA 4.0, riconoscendo esplicitamente il contributo di tutti i partecipanti al processo di sviluppo.

- 1 - **Comprendere l'universo digitale.**
 - 1.1 - **Navigare, ricercare e filtrare dati, informazioni e contenuti digitali:** valutare i fabbisogni informativi, adeguare la sua strategia di ricerca per trovare i dati, le informazioni e i contenuti più adatti all'interno di ambienti digitali, spiegare come accedere ai dati, alle informazioni e ai contenuti più adatti e navigare al loro interno, variare le strategie di ricerca personali.
 - 1.2 - **Valutare dati, informazioni e contenuti digitali:** valutare in maniera critica credibilità e affidabilità delle fonti dei dati, informazioni e contenuti digitali; valutare in maniera critica i dati, le informazioni e i contenuti digitali.
 - 1.3 - **Gestire dati, informazioni e contenuti digitali:** adeguare la gestione di informazioni, dati e contenuti affinché vengano recuperati e archiviati nel modo più facile e opportuno; adeguarsi affinché vengano organizzati ed elaborati nell'ambiente strutturato più adatto.
- 2 - **Comunicare e collaborare nell'era digitale.**
 - 2.1 - **Interagire con gli altri attraverso le tecnologie:** adeguare una varietà di tecnologie digitali per l'interazione più appropriata; adeguare i mezzi di comunicazione più appropriati per un determinato contesto.
 - 2.2 - **Condividere informazioni attraverso le tecnologie digitali:** valutare le tecnologie digitali più appropriate per condividere informazioni e contenuti; adeguare il mio ruolo di intermediario; variare l'utilizzo delle prassi di riferimento e di attribuzione più appropriate.
 - 2.3 - **Esercitare la cittadinanza attraverso le tecnologie digitali:** variare l'utilizzo dei servizi digitali più opportuni per partecipare alla vita sociale; variare l'utilizzo delle tecnologie digitali più adeguate per potenziare le mie capacità personali e professionali e partecipare come cittadino alla vita sociale.
 - 2.4 - **Collaborare attraverso le tecnologie digitali:** variare l'utilizzo degli strumenti e delle tecnologie digitali più appropriati per i processi collaborativi; scegliere strumenti e tecnologie digitali più appropriati per co-costruire e co-creare dati, risorse e know-how.
 - 2.5 - **Netiquette:** adattare norme comportamentali e know-how più appropriati per l'utilizzo delle tecnologie digitali e l'interazione con gli ambienti digitali; adattare le strategie di comunicazione più appropriate negli ambienti digitali a un pubblico; applicare i diversi aspetti della diversità culturale e generazionale negli ambienti digitali.
 - 2.6 - **Gestire l'identità digitale:** distinguere molteplici identità digitali; spiegare le modalità più appropriate per tutelare la propria reputazione; cambiare i dati prodotti attraverso vari strumenti, ambienti o servizi digitali.
- 3 - **Creazione multimediale, pubblicazione responsabile e introduzione alla programmazione.**
 - 3.1 - **Sviluppare contenuti digitali:** modificare contenuti digitali utilizzando i formati più appropriati; adattare i miei atti espressivi attraverso la creazione di materiali digitali più opportuni.
 - 3.2 - **Integrare e rielaborare contenuti digitali:** valutare le modalità più appropriate per modificare, affinare, migliorare e integrare nuovi contenuti e informazioni specifici per crearne di nuovi e originali.
 - 3.3 - **Copyright e licenze:** scegliere le regole più appropriate che applicano copyright e licenze a dati, informazioni e contenuti digitali.
 - 3.4 - **Programmazione:** stabilire le istruzioni più appropriate per un sistema informatico per risolvere un determinato problema o svolgere compiti specifici.
- 4 - **Sicurezza digitale, della salute e dell'ambiente.**
 - 4.1 - **Proteggere i dispositivi:** scegliere la protezione più adeguata per dispositivi e contenuti digitali; distinguere i rischi e le minacce negli ambienti digitali; scegliere le misure di sicurezza più appropriate; individuare le modalità più opportune per tenere in debita considerazione l'affidabilità e la privacy.
 - 4.2 - **Proteggere i dati personali e la privacy:** scegliere le modalità più appropriate per proteggere i miei dati personali e la privacy negli ambienti digitali; valutare le modalità più appropriate per utilizzare e condividere informazioni personali proteggendo me stesso e gli altri da danni; valutare l'adeguatezza delle clausole della politica sulla privacy inerenti le modalità di utilizzo dei dati personali.
 - 4.3 - **Proteggere la salute e il benessere:** spiegare modalità per evitare minacce alla mia salute psico-fisica collegate all'utilizzo della tecnologia; scegliere modalità per proteggere me stesso e gli altri da pericoli negli ambienti digitali; discutere delle tecnologie digitali per il benessere sociale e l'inclusione.
 - 4.4 - **Proteggere l'ambiente:** scegliere le soluzioni più appropriate per proteggere l'ambiente dall'impatto delle tecnologie digitali e del loro utilizzo.
- 5 - **Risoluzione di problemi digitali e pensiero critico.**
 - 5.1 - **Risolvere problemi tecnici:** valutare i problemi tecnici nell'utilizzo di dispositivi e ambienti digitali; risolverli con le soluzioni più adeguate.
 - 5.2 - **Individuare bisogni e risposte tecnologiche:** valutare le esigenze, scegliere gli strumenti digitali più appropriati e le possibili risposte tecnologiche per soddisfarli; decidere le modalità più appropriate per adattare e personalizzare gli ambienti digitali alle esigenze personali.
 - 5.3 - **Utilizzare in modo creativo le tecnologie digitali:** adattare gli strumenti e le tecnologie digitali più appropriati per creare know-how e innovare processi e prodotti; risolvere individualmente e collettivamente problemi concettuali e situazioni problematiche negli ambienti digitali.
 - 5.4 - **Individuare i divari di competenze digitali:** decidere quali sono le modalità più appropriate per migliorare o aggiornare i fabbisogni di competenze digitali di ciascuno; valutare lo sviluppo delle competenze digitali altrui; scegliere le opportunità più appropriate per la crescita personale e per rimanere al passo con i nuovi sviluppi.

Figura 1: Livelli di padronanza attesi per i docenti digitali e per gli studenti digitali, per ogni competenza chiave di ognuna delle 5 aree di competenze DigComp 2.2

Un aspetto importante di questo approccio è che il portale rimarrà attivo anche dopo il rilascio del syllabus avvenuto al termine della fase 4, al fine di poter continuare a raccogliere contributi di nuovi partecipanti alla comunità. Questa caratteristica permetterà revisioni periodiche, trasformando il syllabus in un vero e proprio "*living document*". L'idea di un "*living syllabus*" risponde alla necessità di mantenere il documento costantemente aggiornato in un campo in rapida evoluzione come quello delle competenze digitali.

Questo approccio dinamico e collaborativo assicura che il syllabus rimanga sempre rilevante, adattandosi alle nuove tecnologie, metodologie e esigenze formative che emergeranno nel tempo. In questo modo, il syllabus diventa uno strumento flessibile e sempre attuale per la progettazione di percorsi formativi efficaci nel campo delle competenze digitali.

3 Conclusioni

L'approccio di sviluppo collaborativo e condiviso del syllabus "Informatica per la cittadinanza digitale e per le digital humanities digitali" offre numerosi vantaggi potenziali. Questo metodo promette di integrare prospettive diverse e esperienze variegata, garantendo un syllabus completo e attuale in un campo in rapida evoluzione come quello delle tecnologie digitali. Il syllabus di partenza, che sarà oggetto di questo processo collaborativo, ha già dimostrato la sua efficacia pratica durante l'anno scolastico/accademico 2023-2024 in due importanti iniziative formative:

- Docenti digitali, nell'ambito del progetto PNRR C-KIDD con l'obiettivo di costituire una Comunità di esperti certificati per la formazione dei docenti nella scuola digitale, e la costruzione di kit innovativi da usare nelle classi per la didattica delle discipline tramite l'uso del coding e della realtà aumentata e virtuale, da mettere a disposizione tramite corsi sul portale ministeriale Scuola Futura (<https://www.liceogiorgione.edu.it/pnrr-c-kidd/>).
- Studenti digitali, nell'ambito dell'insegnamento di "Tecnologie e linguaggi per le digital humanities" del corso di laurea triennale in "Media, Comunicazione e Società" e del corso di laurea magistrale in "Digital Humanities" dell'Università di Genova. Al termine del corso agli studenti è stato conferito – oltre la certificazione dei crediti formativi con il superamento dell'esame – una certificazione tramite open-badge quale micro-credenziale di attestazione di competenza definita "Certificazione Cittadinanza Digitale - DigComp 2.2" (<https://tinyurl.com/BadgeCittadinanzaDigitale>).

Queste esperienze forniscono una solida base di partenza per il processo di revisione collaborativa che sta per iniziare. L'approccio di "*living syllabus*" proposto mira a capitalizzare su questi successi iniziali, permettendo continui aggiornamenti e miglioramenti. Questo processo collaborativo promette di trasformare il syllabus in uno strumento ancora più efficace e adattabile per la formazione digitale a vari livelli, rispondendo alle esigenze in continua evoluzione del panorama digitale.

Bibliografia

- [1] Dipartimento per la trasformazione digitale, DigComp 2.2 - Quadro europeo per lo sviluppo delle Competenze Digitali per i cittadini – Trad. italiana. [Online]. Available: <https://repubblicadigitale.gov.it/portale/-/da-oggi-il-digcomp-2.2-parla-italiano>.
- [2] EUROPASS – EU, The European Qualification Framework. [Online]. Available: <https://europass.europa.eu/en/europass-digital-tools/european-qualifications-framework>.
- [3] UNC Charlotte, Bloom's Taxonomy of Educational Objectives. [Online]. Available: <https://teaching.charlotte.edu/services-programs/teaching-guides/course-design/blooms-educational-objectives>

Coding e Realtà Virtuale: un'esperienza laboratoriale interattiva nella scuola primaria

Manuela Chessa¹, Giorgio Delzanno¹, Lorenzo Gerini¹, Giovanna Guerrini¹

Marco Barbagelata², Barbara Gaggero²

¹Università degli Studi di Genova, DIBRIS
manuela.chessa@unige.it

²Istituto Comprensivo Molassana e Prato, Genova
barbara.gaggero@icmolassanaeprato.edu.it

Abstract

Le attività di introduzione al pensiero computazionale e al coding sono ormai diffuse, mediante l'utilizzo di linguaggi visuali a blocchi quali Scratch o ambienti e portali quali code.org. Parallelamente, si stanno affermando le esperienze basate su Realtà Virtuale, anche rivolte a bambini della primaria, in particolare per la fruizione di contenuti immersivi, quali le esperienze museali.

In questo lavoro, descriviamo un'esperienza laboratoriale volta a introdurre i concetti principali della realtà virtuale tridimensionale interattiva, utilizzando un ambiente semplice ma completo di modellazione di contenuti tridimensionali, programmabile mediante un semplice linguaggio a blocchi.

1 Introduzione

La crescente diffusione e pervasività dell'informatica si è espansa rapidamente nella società moderna [1]. Nel frattempo, molti paesi nel mondo hanno incorporato la programmazione informatica come materia obbligatoria nell'istruzione primaria (K-9) [2,3]. La programmazione con linguaggi quale Scratch, un linguaggio di programmazione grafico basato su blocchi, è particolarmente popolare in questa fascia di età, fornendo così un contesto potenzialmente di grande impatto per la ricerca educativa. Tuttavia, diversi studiosi considerano l'educazione alla programmazione non come fine a sé stessa ma essenziale – sebbene non esclusiva – per favorire il pensiero computazionale (Computational Thinking - CT) (cioè supportare i compiti cognitivi in esso coinvolti). Infatti, CT è un termine generico che incarna una base intellettuale necessaria per comprendere il mondo computazionale e impiegare capacità multidimensionali di risoluzione dei problemi all'interno e attraverso le discipline [4].

Parallelamente, si assiste a un crescente interesse nei confronti dell'utilizzo delle tecnologie di Realtà Virtuale (VR) e Realtà Aumentata (AR) nel campo della didattica, ad esempio per l'insegnamento della matematica [5], sfruttando la possibilità di visualizzare informazioni in tre dimensioni (3D) e di sfruttare l'immersione in ambienti simulati. Sebbene gli studi siano ancora limitati in numero e in metodologia, alcuni lavori concludono che le tecnologie VR possono influenzare positivamente i risultati dell'apprendimento nell'istruzione, migliorare i risultati dell'apprendimento in tutto il mondo tranne che in Europa e facilitare i risultati dell'apprendimento a diversi livelli educativi, sebbene con limitazioni per quanto riguarda la scuola primaria [6].

In questo lavoro, proponiamo la descrizione di un'esperienza laboratoriale, condotta in una classe terza della Scuola Primaria in Italia, in cui si sono combinati gli approcci basati sul coding con un linguaggio di programmazione grafico basato su blocchi e dell'utilizzo della VR, non come strumento aggiuntivo per proporre nuovi contenuti, ma come strumento per la creazione di esperienze tridimensionali e interattive.

Nello specifico si è introdotto l'ambiente di sviluppo visuale CoSpaces (Sezione 2) per la creazione di semplici scenari interattivi tridimensionali rappresentanti, come caso di studio l'ambiente montagna e le sue trasformazioni (antropizzazione), argomento precedentemente affrontato dalla classe in oggetto durante l'insegnamento della geografia (Sezione 3). I risultati osservati mostrano come il coinvolgimento della classe sia stato elevato, l'esperienza valutata positivamente dagli alunni e i risultati potenzialmente di interesse per l'introduzione di semplici tecniche di realizzazione di ambienti VR e non solo della loro fruizione passiva a livello della scuola primaria. Infine, lo strumento analizzato ha le potenzialità per essere usato anche nei successivi gradi di istruzione, quali le Scuole Secondarie di Primo e Secondo grado.

2 Sviluppo di esperienze virtuali interattive in ambiente Cospaces

CoSpaces Edu (<https://www.cospaces.io/>) è una webapp di creazione ampiamente utilizzata nelle scuole di tutto il mondo e che consente a scolari di varie età di creare facilmente i propri contenuti virtuali. Ha un sistema di licenza che prevede una versione gratuita, limitata nelle funzionalità, e una licenza a pagamento, con prezzi calcolati in base al numero degli utenti.

Oltre alla progettazione degli ambienti tridimensionali mediante browser, mette a disposizione un'applicazione per dispositivi mobili e tablet, consentendo agli studenti di creare, programmare ed esplorare le proprie creazioni sia in VR sia in AR, su desktop o dispositivi mobile. Infatti, CoSpaces Edu permette inoltre la fruizione dei contenuti in VR utilizzando uno smartphone inserito in dispositivi quali la Google Cardboard, oppure utilizzando visori per la realtà virtuale immersiva, quali i ben noti e diffusi Meta Quest 3, Quest Pro, Quest 2 e Quest (<https://www.meta.com/>). La visualizzazione in realtà aumentata è possibile usando dispositivi mobile che supportano le librerie ARCore (<https://developers.google.com/ar?hl=it>) e ARKit (<https://developer.apple.com/augmented-reality/arkit/>).

La creazione in CoSpaces Edu è un semplice processo di trascinamento della selezione che utilizza una varietà di funzioni di creazione tra cui oggetti 3D, blocchi predefiniti, codifica basata su blocchi e molto altro. CoBlocks, il linguaggio di codifica visiva di CoSpaces Edu, è ideale per i giovani programmatori e rappresenta un'ottima introduzione al pensiero computazionale. Discuteremo il processo di creazione dei contenuti tridimensionali, spiegato a bambini della scuola Primaria, nella sezione seguente. Cospaces Edu consente anche la creazione di codice mediante script Python, tuttavia questa funzionalità non verrà affrontata in questo articolo.

CoSpaces Edu è inoltre progettato per consentire agli insegnanti di avere pieno accesso e controllo sul flusso di lavoro. Dall'interfaccia web su PC, gli insegnanti possono creare e condividere compiti con la classe, assegnare compiti individuali agli studenti o creare compiti di gruppo collaborativi e monitorare il lavoro degli studenti da remoto, anche seguendo il loro processo in tempo reale.

CoSpaces Edu può essere utilizzato in qualsiasi classe e per qualsiasi livello o materia parte del curriculum, per svolgere i seguenti compiti:

- Per la creazione di ambienti tridimensionali, anche interattivi e/o animati
- Per la creazione di esperienze storytelling
- Per la creazione di infografiche tridimensionali
- Per la creazione di esperienze museali, o la ricostruzione di ambientazioni geografiche, storiche, scientifiche

3 Esperienza laboratoriale

L'esperienza laboratoriale qui descritta ha avuto luogo in una classe terza della Scuola Primaria, situata nel comune di Genova, specificatamente il Plesso Doria dell'Istituto Comprensivo Molassana e Prato.

La classe era composta da 18 bambini, eterogenei per familiarità con gli strumenti digitali, tuttavia con pregresse esperienze, fornite dai docenti del plesso, di attività di coding (es. mediante piattaforma code.org o scratch).

L'ambiente CoSpaces è stato introdotto utilizzando analogie con il mondo reale che circonda i bambini, ad esempio:

- La classe che li circonda è composta da oggetti (banchi e sedie), ognuno caratterizzato da una sua posizione nello spazio e da posizioni relative tra di essi. Questo esempio è un modo semplice di introdurre il concetto di “riferimento oggetto” e “riferimento mondo” specifici della catena di visualizzazione della Computer Graphics.
- La classe è visibile agli occhi di un osservatore, che può muoversi in essa, quindi osservare la classe da diversi punti di vista, e che ha bisogno di una fonte di illuminazione (naturale o artificiale) per poter vedere gli oggetti. Questa semplice descrizione introduce i 2 elementi fondamentali di una scena CoSpaces: la telecamera, caratterizzata da una posizione 3D nello spazio che può essere modificata, e da un'illuminazione.

È importante notare che bambini della classe terza primaria possono non avere appreso pienamente il concetto di geometria tridimensionale, ma l'analogia con il mondo (naturalmente tridimensionale) in cui vivono è certamente comprensibile.

Successivamente è stato introdotto il concetto di “primitiva”, concetto alla base della Computer Graphics, e presente in motori per lo sviluppo di esperienze tridimensionali complesse, utilizzando l'analogia con i mattoncini da costruzione, noti e usati dai bambini di quell'età.

A partire da queste poche informazioni è stato possibile per gli alunni cominciare a inserire oggetti nella scena tridimensionale e a manipolarli, cambiandone:

- Posizione 3D
- Colore
- Forma, agendo sulle scale lungo i 3 assi

A seguire è stato mostrato come inserire oggetti più complessi, presenti nell'ambiente CoSpaces, analoghi agli assets o modelli 3D aggiuntivi che possono essere importati in Unity 3D, ma visti dagli alunni in maniera semplice, come “mattoncini” aggiuntivi per la creazione di una scena.

Infine, è stata mostrata la possibilità di aggiungere uno sfondo alla scena, ovvero un'immagine che appare sempre dietro a ogni oggetto inserito.

Dopo la creazione di semplici scene 3D, è stato introdotto il linguaggio a blocchi, specificatamente per la realizzazione di semplici forme di interazione, quali l'animazione degli oggetti inseriti o

l'implementazione di dialoghi tra i personaggi. Essendo il linguaggio CoBlocks analogo ai molti già esistenti, non forniamo qui ulteriori dettagli.

Infine, è stato proposto agli alunni, suddivisi in gruppi di 2, di realizzare in maniera autonoma il seguente progetto, ispirato da un concetto precedentemente visto nelle ore curriculari:

Realizzazione di un ambiente montano che mostri le successive fasi di antropizzazione.

Gli alunni potevano utilizzare tutte le risorse messe a disposizione dalla versione licenziata di CoSpaces, nello specifico si suggeriva la creazione delle seguenti 3 scene:

- Ambiente montano senza alcun segno della presenza dell'uomo
- Ambiente montano con minimi segni della presenza dell'uomo
- Ambiente montano con forti segni di presenza dell'uomo (strade, automobili, ...)

4 Risultati e conclusioni

La Figura 1 mostra alcune schermate degli ambienti 3D realizzati dagli alunni della classe terza dell'Istituto Comprensivo Molassana e Prato di Genova, plesso Doria, nell'anno scolastico 2022-23.

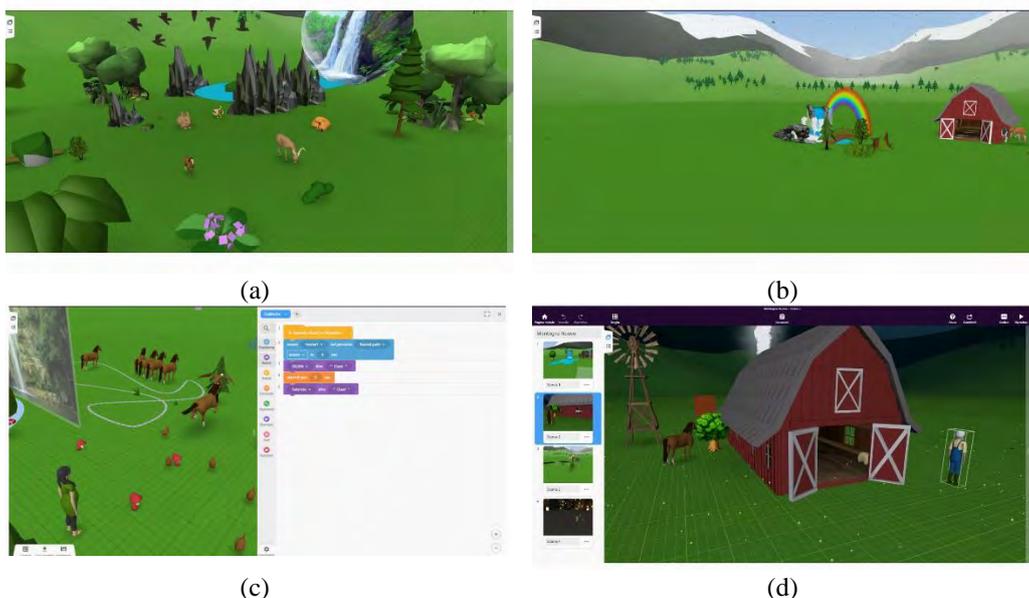


Figura 1. (a) Esempio di scena 3D di ambiente montano non antropizzato (b) Esempio di scena 3D di ambiente montano antropizzato (c) Esempio di coding con linguaggio CoBlocks (d) Intero progetto composto da diverse scene 3D.

Ulteriori dettagli sono forniti nel video <https://youtu.be/ylyp9OjNdNY>, in cui viene mostrata anche la fruizione dei contenuti creati in VR e AR.

Principale sviluppo futuro sarà utilizzare l'approccio qui descritto in maniera preliminare per verificare il grado di apprendimento delle basi di pensiero computazionale applicato alla realizzazione e manipolazione di scenari 3D, da parte degli alunni della scuola primaria.

Bibliografia

- [1] Aho, A. V. (2012). Computation and computational thinking. *The computer journal*, 55(7), 832-835.
- [2] F. Heintz, L. Mannila and T. Färnqvist, A review of models for introducing computational thinking, computer science and computing in K–12 education. 2016 IEEE Frontiers in Education Conference. IEEE. 2016, pp. 1–9.
- [3] L. Mannila, V. Dagiene, B. Demo, N. Grgurina, C. Mirolo, L. Rolandsson and A. Settle, Computational thinking in K–9 education. Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference. New York, NY: ACM, 2014, pp. 1–29.
- [4] Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12-28.
- [5] Vakaliuk, T. A., Shevchuk, L. D., & Shevchuk, B. V. (2020). Possibilities of using AR and VR technologies in teaching mathematics to high school students. *Universal Journal of Educational Research*, 8(11B), 6280-6288.
- [6] Yu, Z., & Xu, W. (2022). A meta-analysis and systematic review of the effect of virtual reality technology on users' learning outcomes. *Computer applications in engineering education*, 30(5), 1470-1484.

Un curriculum verticale per comprendere l'IA: dalle Reti Neurali alla Citizen Science

Laura Cesaro¹, Giovanni Dodero²

¹ Università degli Studi di Padova
laura.cesaro@phd.unipd.it

² Liceo statale scientifico e classico "Martin Luther King", Genova - EFT Liguria
giovanni.dodero@gmail.com

Abstract

Questo contributo presenta tre proposte didattiche relative alla comprensione e all'utilizzo di reti neurali, pensate per un curriculum verticale e strutturate a livelli di complessità crescente. L'obiettivo è preparare gli studenti a interagire con l'Intelligenza Artificiale in modo consapevole ed etico, promuovendo una conoscenza critica del suo funzionamento e delle implicazioni etiche e sociali.

1 Introduzione

Nel dicembre 2022 il rilascio di ChatGPT, l'ormai notissimo chatbot dotato di una considerevole capacità di elaborazione del linguaggio, ha catalizzato un'attenzione senza precedenti verso le potenzialità e le implicazioni dell'Intelligenza Artificiale (IA) nella vita quotidiana. Questo fenomeno ha reso evidente quanto l'introduzione di modelli di IA avanzata sia pervasiva e determinante nelle nostre interazioni digitali, influenzando vari settori, dall'informazione all'intrattenimento, e ponendo nuove sfide e opportunità.

Nel contesto educativo, questa presa di coscienza stimola una riflessione profonda, supportata anche da diversi documenti istituzionali: come possiamo preparare i giovani a vivere e interagire con un mondo sempre più permeato dall'IA? Come promuovere una conoscenza critica e informata, capace di interagire con queste tecnologie in modo etico e consapevole? Quali attività didattiche proporre per aiutare gli studenti non solo a sperimentarne l'uso attraverso un'interazione consapevole [1], ma anche a comprendere il funzionamento [2-6]?

L'orizzonte in cui si colloca questo contributo è quello di far progredire le proposte didattiche verso una gamma sempre più ampia di attività, che consentano agli studenti di intuire quali meccanismi si trovano alla base dei più diffusi modelli di IA, contribuendo a rafforzare le iniziative già in essere [7, 8] e proseguendo idealmente un percorso iniziato con lo sviluppo di proposte di Machine Learning (ML) applicate alla didattica curricolare [9].

2 Oltre ChatGPT: proposte didattiche per sperimentare le Reti Neurali Artificiali

Il presente lavoro propone tre differenti attività didattiche incentrate sulle reti neurali strutturate secondo un criterio di complessità crescente, che potrebbero quindi contribuire allo sviluppo di un curriculum verticale sull'IA nei diversi ordini di scuole.

In particolare, la prima attività propone di scoprire il funzionamento di una rete neurale che scrive la didascalia di un'immagine attraverso un gioco unplugged inserito dal Massachusetts Institute of Technology (MIT) all'interno di un curriculum sull'Intelligenza Artificiale. Questa esperienza consente ai partecipanti di avere una comprensione pratica del funzionamento di una semplice rete neurale, e di scoprire come i neuroni artificiali possono interagire all'interno di una rete, come le reti possono essere utilizzate per risolvere problemi come il riconoscimento di modelli o la classificazione di dati, e di riflettere sul funzionamento dei meccanismi dell'IA, sui potenziali rischi e possibili benefici del suo uso.

La seconda esperienza si focalizza sulla trasparenza e spiegabilità dei processi decisionali di una rete neurale artificiale smontandone la percezione di "scatola nera". In Open Roberta Lab gli studenti definiscono e testano una rete neurale, esplorando come essa consenta a un robot (simulato o reale) di muoversi autonomamente nell'ambiente, tramite l'elaborazione dei dati ricevuti dai sensori e la restituzione delle istruzioni agli attuatori del robot usando una piattaforma di programmazione visuale a blocchi. La visualizzazione e l'analisi delle connessioni e dei processi che portano a una determinata decisione migliorano la comprensione della logica sottostante e permettono lo sviluppo di competenze logiche e di programmazione.

La terza proposta prevede che gli studenti sviluppino un'applicazione mobile utilizzando AppInventor del MIT, e collega le reti neurali alla citizen science. I sensori dei dispositivi mobili vengono utilizzati per raccogliere dati ambientali che vengono poi interpretati e categorizzati in modo da poter essere associati a descrizioni che rappresentano una percezione o un'emozione umana, creando un sistema che traduce in tempo reale le informazioni raccolte dai sensori in risposte interpretabili facilitando una comunicazione più intuitiva tra l'utente e la tecnologia, dato che questo processo di associazione permette al sistema di tradurre dati numerici grezzi in concetti comprensibili e rilevanti per l'utente. Questa attività più complessa integra la conoscenza delle reti neurali con lo sviluppo di strumenti pratici, permettendo agli studenti di applicare le loro competenze in un contesto reale e condivisibile socialmente.

Alla fine di ogni proposta sono esplicitati gli obiettivi didattici e le competenze che possono essere attivate e acquisite secondo il framework europeo DigComp 2.2.

2.1 Neural Network Unplugged

A partire dal 2018 il MIT ha sviluppato il programma RAISE (Responsible AI for Social Empowerment and Education) [10] che comprende diverse piattaforme di alfabetizzazione all'Intelligenza artificiale rivolto agli studenti del segmento K-12, composto da bambini e ragazzi fino a 18 anni di età [11]. Tra i materiali elaborati all'interno delle iniziative dedicate all'Educazione all'IA troviamo il curriculum DAILY, rilasciato con licenza Creative Commons CC-BY-NC. Ideato da ricercatori del MIT e da facilitatori esperti, raccoglie sia esperienze pratiche che basate su computer, e permette di esplorare i concetti chiave dell'IA, le implicazioni etiche del suo utilizzo, le potenzialità creative ad essa associate e il suo impatto sul futuro.

Abbiamo trovato l'introduzione alle reti neurali significativa per il suo approccio che non richiede esperienze pregresse, ed è articolata come un gioco unplugged in cui gli studenti diventano i nodi di una rete: suddivisi fra stato di input, livello nascosto e stato di output, hanno il compito di imparare a generare la didascalia di un'immagine.

Attraverso attività guidate, gli studenti vengono introdotti a concetti di base come Rete Neurale Artificiale, Feed forward, Valutazione, Back Propagation e Tuning sperimentando con l'esperienza come l'addestramento di una rete neurale sia un processo che richiede più fasi di ottimizzazione e come il training renda la rete in grado di imparare a eseguire meglio in un compito.

Per partecipare a questo gioco non è necessaria nessuna conoscenza pregressa sulle Reti Neurali, e l'attività diventa propedeutica ad un successivo passaggio ad applicazioni computazionali.

La proposta ha i seguenti obiettivi didattici:

- sviluppare una comprensione di base su come funziona una rete neurale artificiale;-
- riflettere sui potenziali benefici e rischi associati all'uso delle reti neurali e dell'intelligenza artificiale.



Figura 1: Neural Networks Unplugged - fonte: raise.mit.edu

Le dimensioni del DigComp 2.2 rilevanti per questa attività sono:

1.1 Alfabetizzazione su informazioni e dati: riconoscere come le reti neurali elaborano e interpretano i dati per risolvere problemi complessi (ad esempio il riconoscimento di modelli).

5.2 Individuare bisogni e risposte tecnologiche: capire come l'IA può rispondere a bisogni specifici attraverso la classificazione di dati e il riconoscimento di schemi.

2.2 Primi passi nelle Reti Neurali Artificiali

Open Roberta Lab [12] nasce in Germania nel 2015 come evoluzione del progetto di ricerca promosso dal Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) nel 2002, chiamato Roberta, con l'obiettivo di incoraggiare le ragazze ad avvicinarsi alle materie scientifiche e tecnologiche, in particolare alla robotica, un campo tradizionalmente dominato dai maschi.

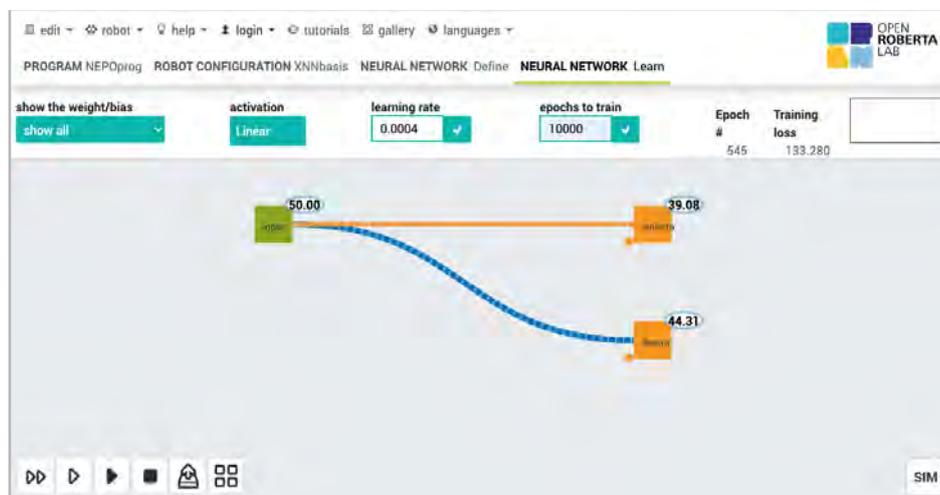


Figura 2: l'ambiente di addestramento delle reti neurali in Open Roberta Lab

La piattaforma, open source e disponibile online, offre un ambiente di programmazione visuale a blocchi che consente di sviluppare software per un'ampia gamma di microcontrollori e robot e di simularne il funzionamento, o di trasferirne il codice su una scheda o su un robot fisico per testarne la corretta esecuzione. A partire dal 2022, tra le funzionalità disponibili è presente quella di definire, configurare e addestrare una Rete Neurale xNN (explainable Neural Network) in modo da scoprire come il sistema apprende e quali passaggi siano necessari per permettergli di arrivare a risolvere un compito.

In Open Roberta Lab si familiarizza con i concetti e la terminologia di base e si sperimenta la configurazione di una rete neurale, attraverso l'organizzazione dei neuroni di input e di output, la quantità e la composizione degli Hidden Layer, la definizione delle relazioni tra connessioni inserendo il peso e il bias e di visualizzare la formula di calcolo risultante per ogni neurone, rendendo comprensibile il processo di feed forward attraverso la funzione di propagazione. È possibile inoltre scegliere la modalità di elaborazione delle informazioni attraverso la funzione di attivazione, e visualizzarne il funzionamento passo passo.

Una volta configurata la rete, è necessario fornire un set etichettato di dati di input e di output e procedere all'addestramento finalizzato a minimizzare gli errori: lo scopo della rete infatti è calcolare i dati di output corretti a partire dai dati di input e di elaborare un modello che potrà essere applicato a dati nuovi e sconosciuti per effettuare previsioni. La rete può quindi essere usata nella programmazione dei robot verificandone le prestazioni sia utilizzando il simulatore integrato che trasferendo il software in un robot fisico.

Un semplice esempio di progettazione e di addestramento di una rete neurale per controllare un robot segui-linea può mostrare agli studenti come l'uso della rete neurale renda più fluido e preciso il movimento del robot rispetto ad una programmazione tradizionale.

La proposta ha i seguenti obiettivi didattici:

- comprendere come una rete neurale elabora i dati sensoriali e prende decisioni autonome per il movimento di un robot.
- scomporre e analizzare le connessioni e i processi decisionali per smontare la percezione della rete neurale come "scatola nera".

La dimensione del DigComp 2.2 rilevante per questa attività è la dimensione 1.2 Valutare dati, informazioni e contenuti digitali: analizzare e valutare in maniera critica come i dati vengono elaborati e utilizzati da una rete neurale per prendere decisioni.



Figura 3: l'ambiente di programmazione di Open Roberta Lab

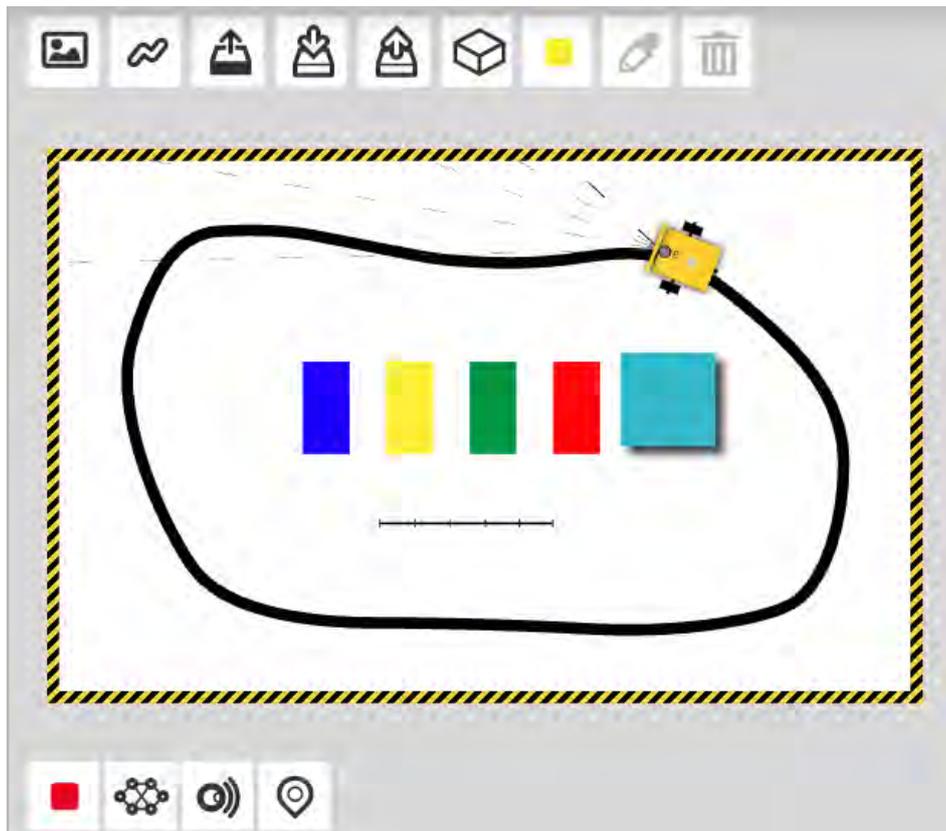


Figura 4: la simulazione del funzionamento del robot segui-linea

2.3 Dare parole al “sentire”: Reti Neurali e Citizen Science

Nel segmento dei cosiddetti Teen l’uso della IA (generativa e non) è consistente, ma d’altro canto la consapevolezza è molto poca. Nell’ottica di rendere comprensibile e rivelare molti dei meccanismi nascosti dell’IA, e i principi su cui si basa, proponiamo un percorso che con pochi passi e strumenti alla portata degli studenti la renda molto meno magica, e subito utilizzabile, con tutta la sua potenza. Il fine ultimo è la comprensione delle basi di questa nuova tecnologia in modo che diventi sia patrimonio culturale collettivo, sia il fondamento dell’essere cittadino digitale cosciente. Il percorso parte da idee sviluppate dal gruppo di lavoro nazionale Sentire-AI delle Équipes Formative Territoriali di cui uno degli autori dell’articolo ha fatto parte, e da approfondimenti successivi sviluppati in corsi di formazione per docenti e confronti con il gruppo Associazione Italiana Fisica - Piano Lauree Scientifiche di Genova.

La proposta nasce dall’idea che i sensori di grandezze fisiche acquisiti dagli strumenti usuali di cui il cittadino digitale dispone possano essere associati ad un “sentire” condivisibile con il proprio mondo sociale. Questo afferisce al mondo dell’IOT e dei Big Data, ma, a partire da esperienze didattiche già pronte (vedi Arduino ed Edge[15], Cisco in “Sentire-AI” by EFT[16]) si sono anche volute creare esperienze e strumenti in cui, con semplici mezzi presenti sulla rete, facili da usare e non “chiusi,” si potessero replicare svariati processi utilizzati abitualmente nel ML e per l’addestramento di IA.

In particolare, si è proposta la creazione di una app (con AppInventor sviluppato dal MIT[17]) che, acquisito un dato ambientale (suono, luce, accelerazione o qualunque dato rilevabile dal sensore integrato nel cellulare), possa funzionare in due modi:

1. associando uno stato (una “parola”) al dato o all’insieme di dati (ad esempio: fastidio, piacere)
2. mostrando la “parola” corrispondente al dato o all’insieme di dati che i sensori del cellulare rilevano in quel momento.

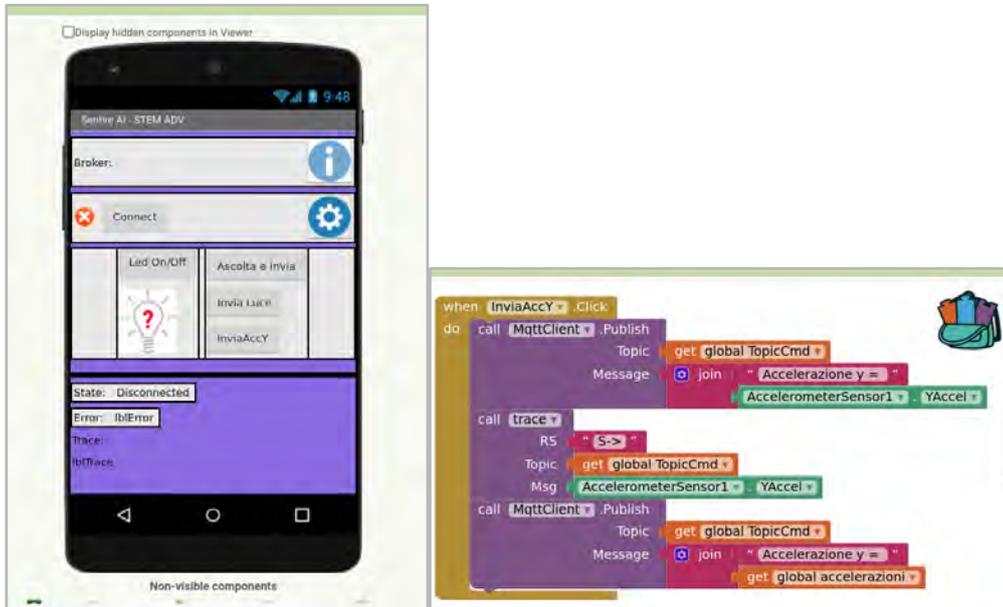


Figura 5: L'applicazione base con AppInventor

Ad esempio, in un certo luogo, uno studente potrebbe associare un “fastidio” a certi livelli di intensità sonora, di luce, temperatura o altro. Un altro studente invece potrebbe solo voler vedere come gli altri si sentono in quella situazione.

Nello specifico, AppInventor trasmette e riceve dati da server MQTT [18] e l'app scritta a blocchi può essere modificata a piacimento.

```

+ Codice + Testo
# Ottieni il nome del file dell'immagine caricata
image_path2 = next(iter(uploaded))

# Caricare il file CSV
data = pd.read_csv(image_path2, header=None)

# Estrarre le etichette
etichettetest = data[0].tolist()
print(etichettetest)
# Estrarre le serie temporali
serietest = data.iloc[:,1:].values.tolist()
print(serietest)
# Controllare la lunghezza delle serie temporali
for serie in serietest:
    #print(len(serie))
    if len(serie) != 100:
        raise ValueError("Le serie temporali devono avere la stessa lunghezza")

#sfoglia... testdata2.csv
testdata2.csv(text/csv) - 2396 bytes, last modified: n/a - 100% done
Saving testdata2.csv to testdata2.csv
['movimento intenso', 'movimento lento', 'tutto fermo', 'movimento intenso', 'movimento lento', 'tutto fermo', 'movimento intenso']
[[9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111]]

y_eval, p_eval = classifier.predict(serietest, return_winning_probability=True)
print(y_eval, p_eval)
print(etichettetest)

/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:464: UserWarning: torch.utils.checkpoint: the use_reentrant parameter should
warnings.warn(
['movimento intenso', 'movimento lento', 'tutto fermo', 'movimento intenso']
['movimento lento', 'tutto fermo', 'movimento intenso'] [0.97399974 0.9952003 0.9996592 0.58696467 0.9975811 0.9998266
0.99945794]
['movimento_intenso', 'movimento_lento', 'tutto_fermo', 'movimento_intenso', 'movimento_lento', 'tutto_fermo', 'movimento_intenso']

```

Figura 6: Uso della Rete addestrata in Colab

La parte IA è scritta in Python e gira su Jupyter Notebook (per semplicità si è usato Colab [19]). Questi script da un lato addestrano la rete neurale di classificazione in modo da affinare i modelli previsionali, dall'altro restituiscono le previsioni se vengono solo interrogati. Sono stati usati due modelli: uno più specifico per problemi di sequenze temporali [20] e l'altro più generico per insiemi di dati non correlati [21].

Questo percorso, anche se non sperimentato ancora su scale di grandi dimensioni, svela parecchi fondamenti sul funzionamento dell'IA, dell'IOT e dei Big Data e fa ben comprendere come tutti i nostri dati digitali possano essere utilizzati da altri e come anche noi come cittadini potremmo consapevolmente contribuire ad una società smart che migliori il benessere di tutti.

La proposta ha come obiettivo didattico l'integrazione della tecnologia AI nella citizen science, permettendo l'applicazione delle competenze in un contesto reale e condivisibile socialmente.

Le dimensioni del DigComp 2.2 rilevanti per questa attività sono:

2.3 Esercitare la cittadinanza attraverso le tecnologie digitali:

- partecipare alla vita sociale attraverso l'utilizzo di servizi digitali pubblici e privati;
- trovare opportunità di self-empowerment e cittadinanza partecipativa attraverso le tecnologie digitali più appropriate.

5.2 Individuare bisogni e risposte tecnologiche: verificare le esigenze e individuare, valutare, scegliere e utilizzare gli strumenti digitali e le possibili risposte tecnologiche per risolverle.

5.3 Utilizzare in modo creativo le tecnologie digitali: utilizzare gli strumenti e le tecnologie digitali per creare conoscenza e innovare processi e prodotti.

3 Conclusione

La complessità crescente delle esperienze presentate ben si prestano a contribuire ad un curriculum verticale mirato all'apertura della black box del funzionamento dell'IA, spaziando dalla scoperta di com'è strutturata una rete neurale artificiale e quali sono i meccanismi di funzionamento a come si può addestrare per l'esecuzione di un compito semplice, per arrivare a mostrare come la presenza dell'IA nella società sia pervasiva ma allo stesso tempo sia anche gestibile e aperta allo sviluppo di applicazioni di "citizen science" che possono contribuire a rendere il mondo più intelligente e sostenibile.

Bibliografia

- [1] Joint Research Centre (European Commission) - Vuorikari, R., Kluzer, S., Punie, Y., "DigComp 2.2, The Digital Competence framework for citizens: with new examples of knowledge, skills and attitudes", Publications Office of the European Union, 2022
- [2] European Commission, Directorate-General for Education, Youth, Sport and Culture, "Digital education action plan 2021-2027 – Improving the provision of digital skills in education and training", Publications Office of the European Union, 2023 <https://data.europa.eu/doi/10.2766/149764> [Accessed January 2024]
- [3] European Commission, Directorate-General for Education, Youth, Sport and Culture, "Ethical guidelines on the use of artificial intelligence (AI) and data in teaching and learning for educators", Publications Office of the European Union (2022) <https://data.europa.eu/doi/10.2766/153756> [Accessed January 2024]

- [4] United Nations Children's Fund, "Policy guidance on AI for children", United Nations Children's Fund (UNICEF), November 2021. [Online]. Available: <https://www.unicef.org/globalinsight/media/2356/file/UNICEF-Global-Insight-policy-guidance-AI-children-2.0-2021.pdf> [Accessed January 2024]
- [5] UNESCO - Miao, F., Holmes, W., Ronghuai H., Hui Z., "AI and education: guidance for policy-makers", 2021. [Online]. Available: <https://unesdoc.unesco.org/ark:/48223/pf0000376709>. [Accessed January 2024].
- [6] Indire, "European Digital Education Hub - Intelligenza Artificiale," [Online]. Available: www.indire.it/progetto/european-digital-education-hub-intelligenzaartificiale/ [Accessed July 2024]
- [7] Code.org. [Online]. Available: <https://code.org/ai> [Accessed July 2024]
- [8] M. Baldoni, C. Baroglio, M. Bucciarelli et al., "EmpAI: l'Intelligenza Artificiale imparata in modo naturale", Proceedings of the DIDAMATICA 2021, Artificial Intelligence for Education
- [9] L. Cesaro, G. Dodero, "Ri-conoscere l'intelligenza artificiale" in Atti Convegno Italiano sulla Didattica dell'Informatica, Università degli studi di Bari, 2023
- [10] Massachusetts Institute of Technology, RAISE - Initiative to Innovate Learning and Education in the Era of AI [Online] <https://www.mitraise.org/> [Accessed July 2024]
- [11] Day of AI, [Online] Available: <https://dayofai.org/> [Accessed July 2024]
- [12] Open Roberta Lab [Online] Available: <https://lab.open-roberta.org/> [Accessed July 2024]
- [13] M. Martin, T. Deneux, M. Chevalier, "From Automaton to AI Robot: the Added Value for Learning", EasyChair Preprint № 13047, 2024. Available: https://easychair.org/publications/preprint_open/6CgmT [Accessed July 2024]
- [14] Learning Robots [Online] Available: <https://learningrobots.ai/resources/?lang=en>
- [15] <https://cloud.arduino.cc/> e <https://edgeimpulse.com/> [Accessed July 2024]
- [16] https://scuolafutura.pubblica.istruzione.it/mooc-innovamenti_intelligenza-artificiale [Accessed July 2024]
- [17] <http://appinventor.mit.edu/> [Accessed July 2024]
- [18] Mosquito o Hivemq
- [19] <https://colab.research.google.com> [Accessed July 2024]
- [20] RNN Recurrent Neural Network
<https://www.datacamp.com/tutorial/tutorial-for-recurrent-neural-network>
[Accessed July 2024]
- [21] TabPFN è un automl (Auto Machine Learning) per tabelle:
<https://www.automl.org/automl-for-x/tabular-data/> paper: <https://arxiv.org/abs/2207.01848>

PepperCare

un umanoide a servizio della didattica

Luca Spandre¹, Laura Camplani², Celestina Zandonai³

¹Istituto Superiore I. Piana - Lovere
luca.spandre@ispiana.edu.it

²Istituto Superiore I. Piana - Lovere
laura.camplani@ispiana.edu.it

³Istituto Superiore I. Piana - Lovere
bgis00700q@istruzione.it

Abstract

Un robot umanoide a servizio della didattica? Tecnologia e scienze umane possono avere una comunione d'intenti? L'AI a scuola può avere un ruolo didattico? Scuola e aziende possono davvero collaborare per formare futuri professionisti? Queste sono le domande che hanno dato vita al progetto PepperCare coinvolgendo gli studenti dell'Istituto Superiore Ivan Piana nella programmazione di un robot umanoide che avesse l'obiettivo di realizzare attività cognitive e riabilitative presso le RSA del territorio.

1 Introduzione

Che cosa può unire indirizzi di studio completamente differenti per profilo d'utenza, discipline specifiche e sbocchi professionali?

Sono punti in comune i consueti obiettivi trasversali, come la capacità di risoluzione di problemi, il pensiero critico, la capacità di cooperare, la creatività, il pensiero computazionale e l'autoregolamentazione. L'innovazione è stata coinvolgere e fondere in un unicum menti pragmatiche e rigorose con altre più sensibili, idealiste ed empatiche. Un'idea in grado di coinvolgere e appassionare tanto uno studente incline al sociale, all'assistenza verso i suoi simili, con uno studente tecnologico incuriosito dal funzionamento di dispositivi robotici e di componenti elettroniche.

La risposta per noi è stata Pepper e il progetto che ha catalizzato l'interesse degli studenti del nostro istituto superiore per quasi l'intero anno scolastico.



Figura 1 Il robot Pepper

2 Il robot Pepper

Pepper è un robot umanoide dotato di torso, braccia, mani e testa snodati che lo rendono in grado di compiere diversi movimenti molto simili a quelli umani e di spostarsi in tutte le direzioni benché non abbia gli arti inferiori, sostituiti con ruote girevoli^[2].

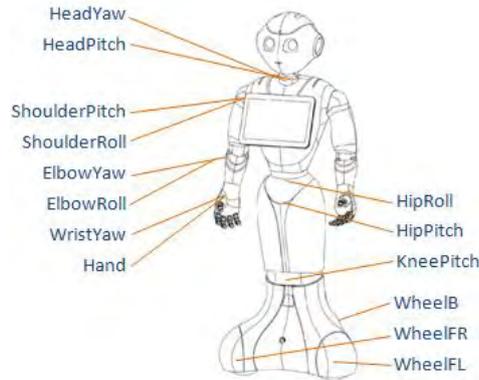


Figura 2 Motori del robot

Pepper è completamente programmabile e può interagire con le persone instaurando conversazioni, riconoscendo le espressioni facciali nonché le emozioni che traspaiono da esse.

Il robot Pepper, creato nel 2014 dagli ingegneri della Aldebaran Robotics a Parigi, azienda poi assorbita dalla Softbank Group e giunta a distribuire il prodotto come Softbank Robotics, grazie alle sue versatili capacità, può essere utilizzato in diversi contesti e per le più svariate finalità. In Giappone, per esempio, è stato utilizzato nel 2020 per fare da cheerleader alla squadra professionale di baseball giapponese^[1] e, in particolare a Tokyo, è stato usato per gestire un negozio di telefonia.

Pepper è stato impiegato anche per svolgere il ruolo di direttore d'orchestra e di personal trainer. Nondimeno, durante un'udienza parlamentare davanti all'Education Select Committee della Camera dei Comuni britannica si è prestato come testimone.

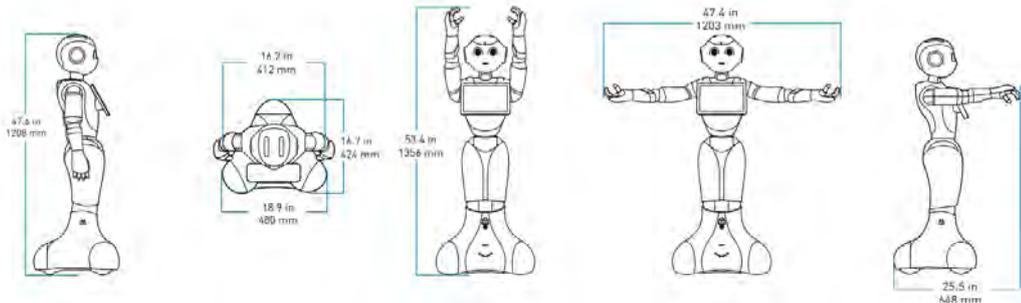


Figura 3 Dimensioni del robot

Nel mondo già più di 2.000 imprese hanno deciso di assumere il robot utilizzandolo come strumento di ricerca nei laboratori di robotica oppure come supporter per assistere le aziende, i centri commerciali e gli aeroporti nell'orientare, accogliere o informare i clienti.

Ma se questo umanoide è così poliedrico, perché non metterlo a servizio della didattica?

3 Il progetto didattico

L'Istituto Superiore Ivan Piana di Lovere, dopo aver analizzato Pepper e le sue caratteristiche peculiari, ha pensato di poter sfruttare la versatilità del robot per unificare in un unico progetto i tre Istituti che compongono il polo scolastico, ovvero l'indirizzo Tecnico Economico (AFM Sistemi Informativi Aziendali), quello Tecnologico nelle due articolazioni Meccanica e Meccatronica ed Elettronica ed Elettrotecnica, e il professionale Socio Sanitario.

Da quest'idea è nato il progetto PepperCare composto da elementi distinti, ma inevitabilmente intrecciati:

- gli studenti con le proprie competenze e peculiarità;
- la scuola e la didattica;
- il territorio e le sue esigenze;
- il presente e il futuro della tecnologia.

Lo studio del robot è stato affidato all'indirizzo Tecnologico, lo sviluppo e la programmazione informatica all'indirizzo Economico (articolazione SIA) mentre la pianificazione delle attività del robot, con e per l'utenza, all'indirizzo Socio Sanitario.

Nella prima fase, gli studenti che si sono occupati dello sviluppo hanno analizzato i vari componenti del robot, le possibilità di interazione e le modalità di programmazione, mentre quelli dell'indirizzo socio sanitario si sono concentrati sul 'progetto sociale' ovvero si sono preoccupati di esplicitare gli obiettivi, i contenuti, i metodi, i tempi e le risorse necessarie all'intervento in RSA.

Successivamente si sono così suddivisi i compiti:

- un gruppo ha approfondito il software utilizzando i vari blocchi già predisposti^[3], studiandone le funzioni, le variabili per permetterne l'interazione;
- un gruppo si è occupato del coding, costruendo nuovi blocchi di codice aggiungendo funzioni personalizzate che hanno reso necessario anche lo studio del Python^[4], il linguaggio di programmazione accettato dall'IDE e conoscenza oggi sempre più richiesta dalle software house, sebbene non fosse previsto dalla programmazione didattica curricolare;
- un gruppo ha creato i dialoghi, prevenendo le varie possibilità con cui l'umano potesse interfacciarsi, ponendo particolare attenzione all'ambiente e all'utenza del progetto;
- un gruppo, con la consulenza degli operatori delle RSA coinvolte, ha pensato alle attività che più si confaccessero alla stimolazione cognitiva o alla riattivazione della mobilità residua degli ospiti.

3.1 Dettaglio attività di coding

Inizialmente sono stati esaminati tutti i blocchi predefiniti dell'ambiente di sviluppo Choregraphe testandone le funzionalità disponibili.

La prima applicazione realizzata è stata una auto-presentazione dell'umanoide in cui Pepper esegue, a richiesta dell'utente, l'imitazione di animali o la simulazione di azioni umane come suonare uno strumento, ballare, passare l'aspirapolvere, guidare un'auto e perfino il decollo dello shuttle, con i rispettivi effetti sonori.

Analizzando il codice dei blocchi già in dotazione si è riusciti a comprendere il significato delle molteplici istruzioni, i vari metodi e le diverse funzioni tanto da comporre nuovi blocchi utilizzando il linguaggio Python.

Poi è stata approfondita la comunicazione con l'utente utilizzando l'apposito linguaggio e rivoluzionando così il *blocco Say*. Tramite le istruzioni del modulo *QiChat* e le diverse opzioni si sono implementate le strutture condizionali dimostrando agli studenti quanto le basi

dell'informatica e i fondamenti teorici restino sempre validi e utili anche quando si parla di robotica e innovazione tecnologica. L'informatica potrà quindi vedere nei prossimi anni nuovi mezzi con cui trasmettere le conoscenze ma le strutture fondamentali di programmazione rimarranno cardini imprescindibili.

Una ulteriore fase ha previsto la realizzazione di una applicazione che funzionasse da calcolatrice, capace di svolgere le operazioni aritmetiche fondamentali. Questo risultato è stato raggiunto a partire dal solo *blocco Sum*, duplicando e modificando opportunamente le istruzioni in esso contenute. Le funzioni così ottenute sono state integrate nel *blocco Dialog* in modo da permettere al robot di comprendere dalla voce dell'utente i numeri e le operazioni da svolgere e di pronunciarne, a seguire, il risultato calcolato.

Uno degli obiettivi futuri sarà quello di ampliare le abilità di calcolo e riuscire a rispondere riducendo o annullando gli inevitabili tempi di attesa.

In ultimo, il team di sviluppo si è dedicato all'approfondimento e alla ricerca delle funzioni più complesse, testando il riconoscimento di alcune emozioni e dei tratti del viso fino a riconoscere le persone.

Questo *modus operandi* ha permesso a tutti i partecipanti di ritagliarsi un posto adeguato alle proprie competenze nella programmazione del robot, pur stimolando le diverse curiosità e soddisfacendo le ambizioni di ciascuno.

Il lavoro svolto fino a qui ha posto le basi per il proseguimento del progetto didattico dei prossimi mesi.

4 Conclusioni

L'ultimo step del progetto, quasi al termine dell'anno scolastico, ha visto la prova sul campo. Il robot Pepper è stato testato nelle tre residenze sanitarie assistenziali locali coinvolte già dalla fase iniziale di pianificazione. Una quarta struttura, venuta a conoscenza dell'entusiasmo degli ospiti e delle relative famiglie e della ricaduta riconosciuta dagli operatori sanitari, ha chiesto all'Istituto di poter far parte della rete degli enti coinvolti.

Osservare la reazione delle persone alle prese con una macchina dalle sembianze umane, che parla e che si muove apparentemente di vita propria, è stato emozionante e coinvolgente, a tratti esilarante, ma sempre appagante. Gli studenti si sono avvicinati a una realtà spesso dimenticata ma in continua espansione, lasciandosi trascinare dall'entusiasmo degli ospiti, consapevoli di aver contribuito a permettere loro di vivere una giornata diversa dalle altre e ad aver loro offerto l'occasione di interfacciarsi con le moderne innovazioni tecnologiche. Dal canto loro, gli anziani, hanno risposto con vivo interesse, divertiti da questa invasione nel loro quotidiano, incuriositi dalle possibilità offerte dal robot e perfettamente a loro agio nell'interagire con un nuovo amico-intrattenitore.



Figura 4 Pepper con un ospite durante il test in RSA

Il progetto si è rivelato utile per sviluppare la creatività dei ragazzi e per permettere loro di concretizzare le conoscenze acquisite durante le ore di lezione curricolari ed è stata anche l'occasione perfetta per unire due diverse generazioni, distanti ma inevitabilmente susseguenti. Pepper è riuscito ad avvicinare giovani e anziani facendo vivere un'esperienza immersiva di tecnologica quotidianità ancora poco esplorata anche dagli stessi ragazzi.

Uno dei prossimi obiettivi sarà quello di rendere il dialogo con Pepper il più naturale possibile e ampliare le sue possibilità d'uso.

In conclusione, il progetto ha richiesto un intero anno scolastico di lavoro modificando in parte la programmazione disciplinare prevista per Informatica e impegnando gli studenti per diversi pomeriggi in orario extracurricolare. L'aver un obiettivo concreto e una deadline dettata dall'aver assunto impegni con enti esterni ha reso gli studenti più responsabili e proattivi.

Sitografia

- [1] <https://www.aldebaran.com/de/node/643>
- [2] <https://unitedrobotics.group/en/robots/pepper>
- [3] <https://www.aldebaran.com/en/support/nao-6/downloads-softwares>
- [4] <https://www.python.org>

“Siete connessi?”

Educazione alla Cittadinanza digitale consapevole

Claudia Incerti

Istituto Comprensivo Bogliasco-Pieve Ligure-Sori
clainc.ci@gmail.com

Abstract

Nel presente contributo viene descritta un'attività proposta in una classe Quarta Primaria, in cui sono stati utilizzati i materiali pubblicati da *Programma il Futuro* per un percorso di apprendimento finalizzato all'acquisizione della consapevolezza nell'utilizzo della rete, promuovendo diverse tipologie di prodotti da parte degli studenti per accogliere diverse modalità espressive.

1 Introduzione

Nel dibattito sull'uso dei dispositivi digitali nella didattica, il presente contributo va nella direzione della profonda convinzione di chi scrive che sia dovere imprescindibile della scuola occuparsi delle competenze digitali delle nuove generazioni, soprattutto relativamente all'uso consapevole dei dispositivi e della rete. È un compito formativo per cui i docenti hanno necessità di essere a loro volta formati e di trovare supporto. Per l'attività che descriviamo, questo supporto consiste nella piattaforma di *Programma il Futuro* [1], che propone un percorso di *Cittadinanza digitale consapevole* per ogni ordine di scuola. Nello specifico, sono stati utilizzati alcuni materiali delle *Lezioni per la scuola primaria* [2], in un percorso di apprendimento proposto con la metodologia della *Flipped Classroom* [3], per l'anticipazione cognitiva dei contenuti, che sono poi stati rielaborati in classe, in gruppi cooperativi, per un compito autentico che ha visto la realizzazione di tre diverse tipologie di prodotto, in modo da permettere ad ogni studente di esprimersi con diverse modalità, secondo il proprio modo di apprendere.

2 Il contesto

L'attività è stata proposta in una classe Quarta Primaria, composta da 21 alunni, di cui 14 femmine e 7 maschi (un'alunna con certificazione ex L.104, due alunni con Piano Didattico Personalizzato per bisogni educativi speciali). La scuola fa parte di un Istituto Comprensivo della Liguria. Il contesto familiare della maggior parte degli studenti è benestante, il retroterra culturale è abbastanza vario.

3 I risultati attesi

L'attività è stata proposta con un approccio interdisciplinare, che ha interessato Italiano, Educazione civica e Tecnologia e Informatica. Sono inoltre stati considerati traguardi di competenza relativi alle competenze digitali e trasversali.

3.1 Obiettivi di apprendimento disciplinari

Per il raggiungimento delle competenze disciplinari, sono stati presi in considerazione i seguenti obiettivi di apprendimento:

- **Italiano:** produrre testi regolativi; esprimersi oralmente in modo chiaro e puntuale; utilizzare il lessico della rete.
- **Tecnologia e Informatica:** scoprire e applicare le regole per navigare in Internet in modo sicuro, per comunicare online con rispetto e consapevolezza, per affrontare situazioni di cyberbullismo; utilizzare la programmazione visuale a blocchi per creare un semplice progetto di storytelling.
- **Educazione civica:** assumere comportamenti responsabili nell'uso della rete e degli strumenti digitali.

3.2 Competenze digitali e trasversali

Le competenze digitali di riferimento sono quelle indicate nel framework europeo *DigComp2.2. Il Quadro delle Competenze Digitali per i Cittadini* [4], in particolare quelle relative alle Aree 2 *Comunicazione e collaborazione*, 3 *Sviluppare contenuti digitali* e 4 *Sicurezza*.

Le competenze trasversali sono riferite alle *Competenze chiave per l'apprendimento permanente* del Consiglio Europeo del 2018 [5] e riguardano la *Competenza alfabetica funzionale*, la *Competenza in scienze, tecnologie e ingegneria*, la *Competenza personale, sociale e capacità di imparare a imparare*, la *Competenza in materia di cittadinanza*, la *Competenza imprenditoriale*, la *Competenza in materia di consapevolezza ed espressione culturali*.

4 Le fasi operative

Il percorso di apprendimento proposto si è articolato sostanzialmente in 6 fasi operative, descritte di seguito:

4.1 Fase 1: valutazione diagnostica

La classe viene divisa in 4 gruppi da 4 e uno da 5 (in cui è inserita l'alunna con certificazione) e viene proposto un brainstorming con la struttura cooperativa *Talking Chips* (Gettoni per parlare) [6]: l'insegnante pone la domanda: "Cosa vuol dire 'Cittadinanza digitale'?"; gli alunni, "spendendo" i propri gettoni, esprimono le proprie idee sull'argomento. Per l'insegnante questo momento rappresenta un'evidenza importante sulle pre-conoscenze degli studenti e regola la successiva predisposizione delle risorse per il *flipped learning*. Al termine, viene assegnata la "lezione capovolta", differenziata per ciascun gruppo [3].

4.2 Fase 2: anticipazione cognitiva

L'insegnante realizza con la webapp *Genially* [7] un'immagine interattiva [8], in cui inserisce 5 video (uno per ogni gruppo), scelti tra quelli proposti da *Programma il Futuro* nel percorso per la Scuola Primaria [2]; quindi la condivide nella classe virtuale di *Microsoft Teams* [9], piattaforma in uso nella scuola, assegnandola nella sezione *Attività*, accompagnata da una consegna precisa e dettagliata (Figura 1). Ogni studente, a casa, ha una settimana di tempo per guardare il video assegnato al proprio gruppo ed eseguire quanto indicato nella consegna, personalizzando i tempi e i modi del proprio apprendimento.

SIETE CONNESSI?

Scade il 25 ottobre 2023 08:00

Tag: INFORMATICA

Ecco il compito sulla CITTADINANZA DIGITALE per mercoledì 25 ottobre:

Cliccando sul link, si aprirà una bella immagine di Genially con dei pulsanti numerati. Cliccate sul pulsante con il numero del vostro gruppo e guardate attentamente il video, anche più volte, cercando di capirlo bene (vi ricordo che il numero del vostro gruppo è quello del vostro tavolo). Poi scrivete sul quaderno di italiano **almeno 2 regole** che avete capito e che pensate siano importanti. Annotatevi anche se avete delle domande o dei dubbi da chiarire, in classe ci confronteremo insieme.

Spero sia divertente! Buon lavoro! 🇮🇹 🧑🏫

<https://view.genial.ly/652ab26c1bbd4600112086a4>

Figura 1: la consegna in classe virtuale

4.3 Fase 3: compito autentico 1

Tornati in classe, i gruppi si riuniscono. L'insegnante consegna il materiale di lavoro cartaceo e digitale, che consiste in: indicazioni di lavoro precise e dettagliate per ogni fase (Figura 2), checklist per il monitoraggio e l'autovalutazione che contiene i comportamenti attesi e i criteri di qualità relativi al prodotto richiesto, un'infografica digitale *"Come si scrive un testo regolativo"* realizzata dalla docente. Come di consueto, i membri di ciascun gruppo si dividono i ruoli e i compiti per il lavoro in *Cooperative Learning* [10]. Inizialmente, si confrontano riflettendo sugli insegnamenti ricavati dal video assegnato. Successivamente, come indicato, redigono, prima sul quaderno, poi su un cartellone, un *Regolamento del Buon Cittadino Digitale* con almeno tre regole esplicitate, rispettando le caratteristiche del testo regolativo indicate nell'infografica. Durante lo svolgersi del lavoro, l'insegnante supervisiona, gira tra i gruppi, verifica la comprensione, supporta le dinamiche cooperative, guida gli studenti a trovare soluzioni condivise, fornisce continui feedback per il miglioramento dell'attività. Alla fine, i cartelloni vengono esposti alle pareti dell'aula (Figura 3).

<p>SIETE CONNESSI?</p> <p>Dopo la lezione capovolta, eccoli pronti per il lavoro di gruppo:</p> <p>Dal video che avete visto a casa, relativo al vostro gruppo, avete capito che ci sono delle regole da rispettare per essere buoni cittadini digitali, e avete potuto confrontarle con le idee che avevate espresso la volta scorsa.</p> <p>1 - La prima richiesta è questa:</p> <p>Dovete confrontarvi sulle vostre riflessioni, poi concordare e scrivere su un cartellone almeno tre regole che avete ricavato dalla visione del video; le vostre regole, unite a quelle pensate dagli altri gruppi, andranno a costruire il REGOLAMENTO DEL BUON CITTADINO DIGITALE che esporremo in classe e, se risulta ben fatto, potrà essere pubblicato sul sito della scuola e proposto a tutti gli alunni.</p> <p>Ricordatevi che il regolamento va scritto rispettando le caratteristiche del testo regolativo, quindi consultate con attenzione l'infografica che trovate sul Team:</p>	<ul style="list-style-type: none"> <input type="checkbox"/> le regole vanno scritte con un elenco puntato o numerato, <input type="checkbox"/> le voci verbali devono avere tutte la stessa forma (stesso tempo, stessa persona), <input type="checkbox"/> le regole devono essere in positivo e non in negativo (cosa si può o si deve fare, invece di cosa non si può fare), <input type="checkbox"/> devono contenere le parole specifiche dell'informatica. <p>2 - La seconda richiesta è questa:</p> <p>Dovete preparare una presentazione creativa (una scenetta, un'intervista, uno spettacolo, una filastrocca, una canzone, ecc.) per spiegare le regole in modo piacevole ai vostri compagni di classe affinché se le ricordino.</p> <p>La vostra rappresentazione dovrà:</p> <ul style="list-style-type: none"> <input type="checkbox"/> catturare l'attenzione dei compagni (cioè essere divertente e non noiosa), <input type="checkbox"/> essere chiara e ordinata, <input type="checkbox"/> mettere bene in evidenza le regole del vostro cartellone. 	<p>Per essere certi di rispettare tutti i punti, potete mettere una crocetta nel quadretto via via che vi sembra di aver svolto ciascun punto.</p> <p>E adesso... BUON LAVORO!!!!</p> <hr/> <p>Alla fine della vostra presentazione, rivedete la lista e attribuite il punto per ogni richiesta rispettata, sarà più facile compilare la tabella di autovalutazione del compito.</p> <p>Fate la somma dei punti:</p> <p>7-6 punti: il compito è andato molto bene!</p> <p>5-4: il compito è andato abbastanza bene</p> <p>3-0 punti: il compito poteva andare meglio...</p>
---	---	---

Figura 2: la consegna per i gruppi in classe

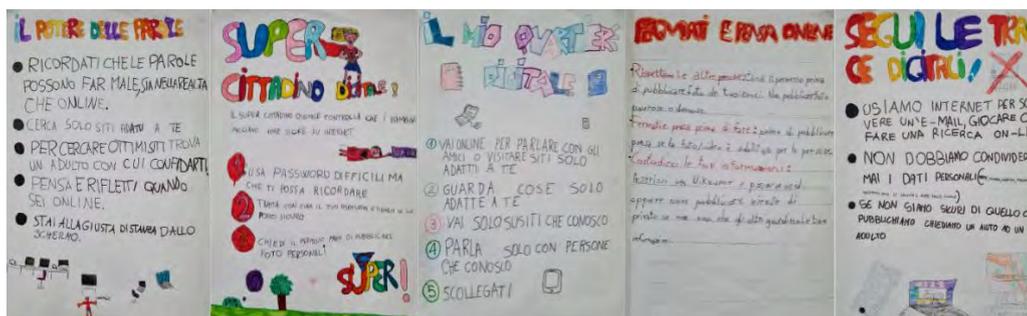


Figura 3: i cartelloni

4.4 Fase 4: compito autentico 2

Per condividere con i pari il proprio *Regolamento del Buon Cittadino Digitale*, nella lezione successiva i gruppi concordano e preparano una performance creativa (una scenetta teatrale, un'intervista, uno spettacolo, una filastrocca, una canzone...): dividendosi ruoli e compiti, stendono una scaletta/un copione/un canovaccio cartaceo, provano le varie parti e preparano eventuali materiali di scena. L'insegnante, anche in questo caso, supervisiona e supporta il lavoro collaborativo, dando continui feedback che regolano lo svolgersi dell'attività.

4.5 Fase 5: condivisione e valutazione

Infine, nell'Aula magna della scuola, ogni gruppo presenta la propria performance ai pari e all'insegnante, utilizzando il Regolamento realizzato. La docente consegna ad ogni studente la scheda per la valutazione tra pari del prodotto, con gli stessi criteri indicati per l'autovalutazione. Al termine della propria rappresentazione, i membri del gruppo svolgono l'autovalutazione, compilando insieme la check-list fornita all'inizio; la condividono poi con l'insegnante, che procede a validarla. Nel contempo, si svolge la valutazione tra pari: ogni studente compila la scheda per la valutazione del prodotto via via che si svolgono le performance.

Al termine dell'attività, ciascun alunno compila la scheda per l'autovalutazione individuale, per riflettere su tutto il lavoro svolto. L'insegnante effettua la valutazione formativa sulla base di tutte le evidenze raccolte e restituisce ad ogni studente un feedback scritto personalizzato.

4.6 Fase 6: il coding

Per comunicare ad un pubblico più vasto le regole del Buon Cittadino Digitale, alla classe viene proposto di condividerle nella Community di *Scratch* [11]. Ogni gruppo si divide in due coppie; ciascuna coppia realizza un progetto *Scratch* di storytelling per illustrare almeno una delle regole elaborate. L'attività mira a consolidare la competenza nell'utilizzo dei messaggi per regolare le interazioni tra gli *sprite*. La scuola è dotata di un Regolamento BYOD (Bring Your Own Device) [12], quindi ogni studente utilizza il proprio tablet personale. La classe conosce la programmazione visuale a blocchi per aver fruito fin dalla classe Prima delle Lezioni di *Code.org* [13] e conosce l'ambiente di *Scratch* [11], che è stato proposto in classe Terza nelle funzioni di base (movimento, aspetto, situazioni). L'insegnante ha attivato da tempo un Account Docente e ha creato le credenziali per ogni alunno, per cui ciascuno procede al Login dal proprio tablet ormai senza difficoltà. Le coppie (più un trio in cui è inserita l'alunna con certificazione) si organizzano con una distribuzione cooperativa del lavoro, come imparato sulla piattaforma di *Code.org*: "conducente" (manovra il mouse o il touch) e "navigatore" (suggerisce i blocchi da utilizzare), che si scambiano ogni sequenza di tre o quattro blocchi. L'insegnante, come sempre, supervisiona il lavoro, fornisce supporto tecnico sulla programmazione, guida al confronto costruttivo e a scelte condivise. Al termine, ogni coppia mostra il proprio progetto alla classe sullo schermo interattivo,

illustrando brevemente il codice realizzato. La docente riunisce i progetti in una *Gallery* [14] su *Scratch*, il cui link viene pubblicato sulla piattaforma regionale dell'*Osservatorio dei Progetti Innovativi* [15] insieme alla documentazione dell'intera attività, completa di tutti i materiali [16].

5 Conclusioni

L'attività, già proposta ad una precedente Classe Quarta nell'a.s. 2018/2019, è stata ripresa e adeguata al contesto e alla classe attuale, avendo avuto esito decisamente positivo. In entrambe le occasioni, i bambini e le bambine hanno potuto riflettere concretamente sul tema della consapevolezza digitale, interiorizzando gli atteggiamenti suggeriti dalle regole costruite insieme. Hanno avuto inoltre la possibilità di esprimersi attraverso diversi canali, allenando differenti stili cognitivi e trovando ciascuno la modalità principe con cui manifestare il proprio percorso formativo.

Bibliografia e sitografia

- [1] MIUR - CINI, «Programma il Futuro,» [Online]. Available: <https://programmmailfuturo.it/>. [Consultato il giorno 11 Luglio 2024].
- [2] MIUR - CINI, «Cittadinanza digitale consapevole. Lezioni per la scuola primaria: Programma il Futuro,» [Online]. Available: <https://programmmailfuturo.it/come/cittadinanza-digitale/cittadinanza-digitale-primaria/introduzione>. [Consultato il giorno 11 Luglio 2024].
- [3] A. Benzi e C. Incerti, *Didattica capovolta alla scuola primaria*, Trento: Casa editrice Erickson, 2022.
- [4] Commissione europea, «DigComp 2.2 Il Quadro delle Competenze Digitali per il Cittadino: rebubblicadigitale.gov.it,» Dicembre 2022. [Online]. Available: <https://repubblicadigitale.gov.it/portale/documents/20122/967347/DigComp+2.2+Italiano+marzo.pdf/3be86d9b-cc83-6d29-c55b-2489c27edb33?t=1679493691038>. [Consultato il giorno 11 Luglio 2024].
- [5] Consiglio dell'Unione Europea, «RACCOMANDAZIONE DEL CONSIGLIO del 22 maggio 2018 relativa alle competenze chiave per l'apprendimento permanente: European Union,» 2018. [Online]. Available: [https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32018H0604\(01\)](https://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32018H0604(01)). [Consultato il giorno 14 Luglio 2024].
- [6] A. Benzi e C. Incerti, «Gettoni per parlare (o talking chips),» in *Didattica capovolta alla scuola primaria*, Trento, Casa editrice Erickson, 2022, p. 28.
- [7] «Genially,» [Online]. Available: <https://genially.com/it/>. [Consultato il giorno 14 Luglio 2024].
- [8] C. Incerti, «immagine interattiva Siete connessi?: Genially,» Ottobre 2023. [Online]. Available: <https://view.genially.com/652ab26c1bbd4600112086a4>. [Consultato il giorno 14 Luglio 2024].
- [9] Microsoft, «Microsoft Teams,» [Online]. Available: <https://www.microsoft.com/it-it/microsoft-teams/group-chat-software/>. [Consultato il giorno 14 Luglio 2024].
- [10] A. Benzi e C. Incerti, «Il Cooperative Learning,» in *Didattica capovolta alla scuola primaria*, Trento, Casa editrice Erickson, 2022, pp. 16-17.
- [11] MIT Boston, «Scratch,» [Online]. Available: <https://scratch.mit.edu/>. [Consultato il giorno 14 Luglio 2024].

- [12] Ministero dell'Istruzione e del Merito, «Azione #6-Linee guida per le politiche attive di BYOD (Bring Your Own Device): scuoladigitale.istruzione.it,» [Online]. Available: <https://scuoladigitale.istruzione.it/pnsd/ambiti/ambienti-e-strumenti/azione-6-linee-guida-per-politiche-attive-di-byod-bring-your-own-device/>. [Consultato il giorno 14 Luglio 2024].
- [13] Code.org, «Code.org,» [Online]. Available: <https://code.org/>. [Consultato il giorno 14 Luglio 2024].
- [14] C. Incerti, «Cittadinanza digitale: Scratch,» 2023. [Online]. Available: <https://scratch.mit.edu/studios/34879520>. [Consultato il giorno 14 Luglio 2024].
- [15] Scuola Digitale Liguria, «Osservatorio dei progetti innovativi: Scuola Digitale Liguria,» [Online]. Available: <https://www.scuoladigitaleliguria.it/osservatorio.html>. [Consultato il giorno 14 Luglio 2024].
- [16] C. Incerti, «U.d.A. Siete connessi?: Genially,» 2024. [Online]. Available: <https://view.genially.com/65ecb5593133fa0014e1195f>. [Consultato il giorno 14 Luglio 2024].
- [17] C. Incerti, «Come si scrive un testo regolativo?: Genially,» Ottobre 2023. [Online]. Available: <https://view.genially.com/6537e6aa0a936300115bacba>. [Consultato il giorno 14 Luglio 2024].
- [18] C. Incerti, «valutazione_tra_pari-SieteConnessi: Google Drive,» Ottobre 2023. [Online]. Available: https://drive.google.com/file/d/1veFs0V2k-ohY0098x5iycmoXR9CFBT75/view?usp=drive_link. [Consultato il giorno 14 Luglio 2024].
- [19] C. Incerti, «SieteConnessi-Consegna_anticipazione: Google Drive,» Ottobre 2023. [Online]. Available: <https://drive.google.com/file/d/1yMan1qjaqjWLWbyN87UGBDD7pbVfdzTU/view?usp=sharing>. [Consultato il giorno 14 Luglio 2024].
- [20] C. Incerti, «Consegna-SieteConnessi: Google Drive,» Ottobre 2023. [Online]. Available: <https://drive.google.com/file/d/1ZTJ-Tf5VznQAZKWVfy618acKxGDmrOho/view>. [Consultato il giorno 14 Luglio 2024].
- [21] C. Incerti, «check-list_gruppo-SieteConnessi: Google Drive,» Ottobre 2023. [Online]. Available: https://drive.google.com/file/d/1gxxzAepUGc5Qe9jkBBpbiiQ_BdIttbBe/view. [Consultato il giorno 14 Luglio 2024].
- [22] C. Incerti, «Autovalutazione_individuale-SieteConnessi: Google Drive,» Ottobre 2023. [Online]. Available: https://drive.google.com/file/d/1HhL4K3QM8ow5SEqqM88uU_lgSla02EK7/view?usp=drive_link. [Consultato il giorno 14 Luglio 2024].

#IA: GiochIAmo e #Scenari Futuri con VR, AR e IoT: gare on line per imparare giocando

Luca Basteris¹, Maria Cristina Daperno²

¹ Liceo Classico e Scientifico Statale “Silvio Pellico – Giuseppe Peano”
luca.basteris@liceocuneo.it

² Liceo Classico e Scientifico Statale “Silvio Pellico – Giuseppe Peano”
cristina.daperno@liceocuneo.it

Abstract

In questo articolo vengono presentate le gare on-line “#IA: giochIAmo” e “#Scenari Futuri con VR, AR e IoT” proposte dal Liceo Classico e Scientifico Statale “S. Pellico – G. Peano” di Cuneo all’interno del progetto “Making, Tinkering, Realtà Aumentata e IoT a servizio della Scuola 4.0”. Il progetto ha previsto attività formative per docenti e attività di didattica innovativa per studenti. Nel progetto sono state proposte gare a squadre a studenti di scuole di ogni ordine e grado su tutto il territorio nazionale. Le gare hanno avuto come obiettivo principale quello di calare le tematiche dell’intelligenza artificiale e dell’IoT all’interno dei curricula della scuola italiana con la metodologia didattica del gaming, avendo consapevolezza che non è sufficiente un approccio a questi temi sotto il profilo semplicemente “tecnologico”, ma che è fondamentale promuovere una riflessione critica da parte della scuola sugli aspetti sociali, etici e relazionali che questi temi solleveranno nel futuro a breve, medio e lungo termine.

1 Introduzione

Il Liceo Classico e Scientifico Statale “S. Pellico – G. Peano” di Cuneo è stata individuata come scuola polo territoriale per potenziare le competenze digitali di insegnamento e apprendimento, attraverso la realizzazione di progetti nazionali per lo sviluppo di modelli innovativi di didattica digitale e di curricula per l’educazione digitale e per la diffusione delle azioni del PNRR, relative alla didattica digitale integrata e alla didattica innovativa nelle scuole, con un progetto dal titolo: “Making, Tinkering, Realtà Aumentata e IoT a servizio della Scuola 4.0”. Il progetto ha previsto attività formative per docenti e attività di didattica innovativa per studenti. All’interno delle attività per studenti si sono organizzate le gare: #IA: giochIAmo (logo gara in figura 1) e #Scenari Futuri con VR, AR e IoT (logo gara in figura 5).

2 Gara #IA: giochIAmo

Questa gara ha come obiettivo principale quello di calare le tematiche dell'intelligenza artificiale all'interno dei curricoli, in collaborazione con l'Ufficio I – area 4 Scuola digitale dell'Ufficio Scolastico Regionale per il Piemonte, e consiste in gare online a squadre, rivolte a gruppi di 4 studenti, per sensibilizzarli all'impatto che l'Intelligenza Artificiale ha e potrà avere nella nostra vita. La sfida ha carattere ludico, ma intende sensibilizzare gli studenti alle tematiche dell'intelligenza artificiale, pur nel contesto competitivo e ludico della competizione. In questa edizione hanno partecipato 37 squadre per un totale di 148 studenti [1].



Figura 1: Logo gara #IA: giochIAmo

2.1 Esempi di domande Gara #IA: giochIAmo

Riportiamo alcuni esempi di domande assegnate nelle gare per spiegare il “taglio” ludico e nello stesso tempo di sensibilizzazione degli studenti su alcuni aspetti fondamentali proprio nel momento del gioco. Le domande hanno riguardato per tutti gli ordini di scuola sia domande generali con taglio teorico, che esercizi pratici con strumenti disponibili on-line, in modo da rendere la gara con un aspetto maggiormente di gaming.

Codice domanda	A2S1
Titolo domanda	Conoscenze tecniche: apprendimento supervisionato
Domanda	Quale delle seguenti affermazioni riguardanti l'apprendimento supervisionato è corretta?
Risposta	A) I dati di addestramento devono essere generati con intelligenza artificiale B) I dati di addestramento devono essere inventati dall'uomo C) I dati di addestramento devono essere reali e rappresentativi del contesto che si vuole studiare D) I dati di addestramento devono essere solamente verosimili
Soluzione	C) I dati di addestramento devono essere reali e rappresentativi del contesto che si vuole studiare
Competenze sviluppate	Acquisire le principali conoscenze relative all'AI
Punteggio massimo	5

Figura 2: Domanda su conoscenze tecniche gara #IA: giochIAmo

Codice domanda	A3S11
Titolo domanda	Utilizzo dell'AI: Deepfake o reale?
Domanda	In allegato potete trovare due video del famoso attore Tom Cruise che potrebbero essere stati generati tramite Intelligenza Artificiale. Selezionare l'opzione corretta:
Risposta	A) Il video A3S11a è reale, mentre il video A3S11b è deepfake B) Il video A3S11a è deepfake, mentre il video A3S11b è reale C) Nessuna delle precedenti
Soluzione	A) Il video A3S11a è reale, mentre il video A3S11b è deepfake
Competenze sviluppate	Uso corretto e consapevole dell'AI
Punteggio massimo	10

<https://prod-files-secure.s3.us-west-2.amazonaws.com/b25392f9-6400-4583-974e-3345635c2f36/4ffc12a4-57c6-4a90-be8e-72fb8808cfa0/Untitled.mp4>

<https://prod-files-secure.s3.us-west-2.amazonaws.com/b25392f9-6400-4583-974e-3345635c2f36/7523d6d4-c8c4-4348-a3d4-f21b62d98ec7/Untitled.mp4>

Figura 3: Domanda sull'utilizzo dell'IA gara #IA: giochIAmo

Domanda	Alcune lingue come il Cinese e l'Ungherese non posseggono il genere, ciò significa che all'interno di una frase non è possibile distinguere il maschile dal femminile. Recatevi sul sito https://translate.google.com/ e provate a tradurre la frase "Lei è un medico" in ungherese. Successivamente (con le freccette al centro) effettuate la traduzione inversa dall'ungherese all'italiano. Fatte le opportune osservazioni ripetete l'esperimento, questa volta con la frase "Lui è un infermiere". Cosa potete notare? Quale problema presenta l'algoritmo? Che effetti potrebbe avere?(N.B. Massimo 3 righe)
Risposta	
Soluzione	L'algoritmo presenta uno stereotipo. In particolare il lavoro del medico viene associato al "maschile" mentre il lavoro dell'infermiere al "femminile". Questo può diventare pericoloso nei sistemi che scannerizzano i CV nelle aziende: associare un certo genere ad una certa professione potrebbe portare all'esclusione di alcuni individui dai colloqui lavorativi solamente per il genere d'appartenenza.
Competenze sviluppate	Contestualizzare l'AI nel mondo reale
Punteggio massimo	10

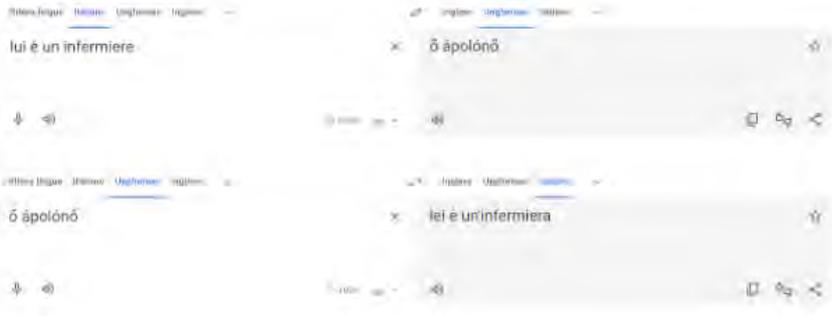


Figura 4: Domanda sulla contestualizzazione dell'IA gara #IA: giochiAmo

3 Gara #Scenari Futuri con VR, AR e IoT

La presente gara ha come obiettivo principale quello di calare le tematiche dell'intelligenza artificiale all'interno dei curricoli, in collaborazione con l'Ufficio I – area 4 Scuola digitale dell'Ufficio Scolastico Regionale per il Piemonte, e consiste in gare online a squadre, rivolte a gruppi di 4 studenti, per sensibilizzarli all'impatto che realtà virtuale, realtà aumentata, internet delle cose ed oggetti "smart" hanno e potranno avere nella nostra vita. La sfida ha carattere ludico e competitivo, ma contemporaneamente di informazione, sensibilizzazione e riflessione. In questa edizione hanno partecipato 10 squadre per un totale di 40 studenti [2].



Figura 5: Logo gara #Scenari Futuri con VR, AR e IoT

3.1 Esempi di domande Gara #Scenari Futuri con VR, AR e IoT

Riportiamo alcuni esempi di domande assegnate nelle gare per spiegare il “taglio” ludico e nello stesso tempo rivolto a sensibilizzare gli studenti su alcuni aspetti fondamentali proprio nel momento del gioco. Le domande hanno riguardato per tutti gli ordini di scuola sia domande generali con taglio teorico, che esercizi pratici con strumenti disponibili on-line, in modo da rendere la gara con un aspetto maggiormente di gaming.

Codice domanda	V2S3
Titolo domanda	Conoscenze tecniche: immersività e presenza
Domanda	Nell'immagine in allegato si possono osservare alcuni utenti che sperimentano un'esperienza VR. A partire dalle informazioni contenute in questa foto, cosa si può dire di questa esperienza di Realtà Virtuale?
Risposta	A) L'esperienza garantisce l'immersione, ma non la presenza B) L'esperienza garantisce la presenza, ma non l'immersione C) L'esperienza garantisce sia la presenza che l'immersione D) L'esperienza non garantisce né la presenza, né l'immersione
Soluzione	D) L'esperienza non garantisce né la presenza, né l'immersione
Competenze sviluppate	Acquisizione delle conoscenze base su AR e VR
Punteggio massimo	5



Figura 6: Domanda sulle conoscenze tecniche gara #Scenari Futuri con VR, AR e IoT

Codice domanda	V2S9
Titolo domanda	BigData: emissioni CO2 (4/5)
Domanda	Al seguente link: https://ourworldindata.org/grapher/co-emissions-by-sector sono mostrati i dati relativi alla quantità di CO2 prodotta ogni anno in base al settore. Quale settore ha subito una variazione significativa dal 2010 al 2011?
Risposta	A) Electricity and heat B) Transport C) Buildings D) Land-use change and forestry
Soluzione	D) Land-use change and forestry
Competenze sviluppate	Analisi di grafici e dati
Punteggio massimo	10

Figura 7: Domanda sui Big Data gara #Scenari Futuri con VR, AR e IoT

Codice domanda	V0S18
Titolo domanda	Applicazioni della VR: NASA JPL
Domanda	Attraverso il seguente link: https://www.jpl.nasa.gov/virtual-tour/ accedere all'applicativo in realtà virtuale del NASA JPL (Jet Propulsion Laboratory), in California. Avviare l'applicazione e, dopo averne preso confidenza, dirigersi nel Mission Control (Centro di controllo delle missioni). Per ognuna delle postazioni nella sala di controllo l'applicazione fornisce una descrizione completa. Per quanto riguarda la postazione "Radio Science", cosa viene detto a proposito della variazione dei segnali radio?
Risposta	A) Servono a sondare gli anelli di Saturno B) Rivelano la struttura interna di pianeti e lune C) Servono a testare la teoria della relatività D) Tutte le precedenti E) Nessuna delle precedenti
Soluzione	D) Tutte le precedenti
Competenze sviluppate	Comprensione delle principali applicazioni della VR
Punteggio massimo	10

Figura 8: Domanda sulle applicazioni della VR gara #Scenari Futuri con VR, AR e IoT

4 Premiazione Gare #IA: giochIAmo e #Scenari Futuri con VR, AR e IoT

Giovedì 22 maggio 2024, alle ore 16.30 on-line, si sono svolte le premiazioni delle GARE #IA: giochIAmo alla sua Terza Edizione e #Scenari Futuri con VR, AR e IoT alla sua Prima Edizione, per studenti delle scuole di ogni ordine e grado statali e paritarie, alla presenza della Dott.ssa Anna Massa dell'USR e del Dott. Alessandro Parola, Dirigente Scolastico del Liceo [3].

Il programma della cerimonia di premiazione ha ospitato alcuni interventi di tipo formativo: l'intervento del giornalista della RAI Dott. Fabio De Ponte, che ha illustrato ai ragazzi come l'IA generativa nel mondo della comunicazione e della scuola possa o meno essere considerata una risorsa; l'intervento di Matteo Tedde, ex studente della scuola e laureando in Ingegneria Informatica, curatore delle domande delle gare, che ha presentato alcune delle domande e indicato la ratio nella creazione e correzione delle domande.

Al termine degli interventi formativi, è stata svelata la classifica, con un breve intervento di commento delle squadre vincitrici e dei loro docenti responsabili e l'assegnazione dei premi previsti.

Bibliografia

- [1] <https://liceocuneo.it/pon/polo-didattica-digitale-cuneo/attivita-studenti-progetto-making-tinkering-realta-aumentata-e-iot-a-servizio-della-scuola-4-0/gara-studenti-ai-giochiamo-2024/>.
- [2] <https://liceocuneo.it/pon/polo-didattica-digitale-cuneo/attivita-studenti-progetto-making-tinkering-realta-aumentata-e-iot-a-servizio-della-scuola-4-0/gara-studenti-vr-ar-e-iot-verso-nuovi-scenari-futuri/>
- [3] <https://liceocuneo.it/pon/polo-didattica-digitale-cuneo/attivita-studenti-progetto-making-tinkering-realta-aumentata-e-iot-a-servizio-della-scuola-4-0/premiazione-gare-iagiochiamo-e-scenari-futuri-con-vr-ar-e-iot/>

Virtual Coding Tour

Giovanni Silvestro ¹, Marta Sanz Manzanedo ²

¹EFT Toscana - Liceo Francesco Cecioni di Livorno (Formatore esterno)

`giovanni.silvestro1@scuola.istruzione.it`

²EFT Toscana - Liceo Francesco Cecioni di Livorno

`marta.sanzmanzanedo@scuola.istruzione.it`

Abstract

Il presente articolo racconta l'esperienza di un corso STEM basato sul coding e finanziato con i fondi europei del PNRR DM65. Il corso si è svolto nel mese di Luglio 2024 nel Liceo Statale Francesco Cecioni di Livorno e ha coinvolto studenti di diverse tipologie di licei con l'obiettivo di potenziare le competenze STEM e incentivare la partecipazione femminile. Le attività hanno incluso l'uso del coding, della realtà virtuale e dell'intelligenza artificiale, e l'implementazione di lingue straniere per migliorare le competenze linguistiche e interculturali degli studenti. Durante il corso, gli studenti hanno utilizzato applicazioni di intelligenza artificiale generativa per raccogliere informazioni, attuare *fact-checking* e tradurre contenuti. L'applicazione *Cospaces Edu* è stata centrale per il *coding* e la creazione di mondi virtuali, stimolando creatività e competenze di *problem-solving*. Divisi in gruppi eterogenei, gli studenti hanno collaborato a progetti come la simulazione di un porto con variabili fisiche e la creazione di pacchetti turistici virtuali, migliorando significativamente le loro abilità tecniche e digitali, come evidenziato dai questionari di autovalutazione finali.

1 Introduzione

Il Liceo Statale Francesco Cecioni di Livorno, noto per il suo indirizzo STEM, si caratterizza per sviluppare le competenze degli studenti in queste aree cruciali, preparandoli per carriere e studi universitari legati alla scienza e alla tecnologia.

Tuttavia questo Liceo ha anche altri percorsi liceali dove normalmente non vengono promosse particolarmente le nuove tecnologie (programmazione, robotica e intelligenza artificiale). Utilizzare strumenti digitali per analizzare dati, creare modelli e risolvere problemi sono competenze sempre più richieste nel mondo del lavoro e sono fondamentali per l'innovazione e lo sviluppo tecnologico. Per questo motivo il corso estivo di coding è stato esteso anche agli studenti degli altri percorsi liceali.

Il corso oggetto di questo articolo, denominato *Virtual Coding Tour*, ha visto la partecipazione di studenti dei licei scientifici ad indirizzo STEM, delle scienze umane e linguistico. Gli obiettivi del corso hanno coinvolto non solo il potenziamento competenze STEM degli studenti come l'utilizzo del coding e della realtà virtuale ma anche l'incentivazione della partecipazione delle ragazze, spesso sottorappresentate in queste aree, e il potenziamento delle *life skills* [1]. Inoltre, in conformità con le linee guida del *DigComp2.2* [2] diverse attività hanno avuto lo scopo di preparare gli studenti all'interazione con l'intelligenza artificiale, sempre più presente nella vita quotidiana e nel mondo del lavoro, e a migliorare le competenze linguistiche attraverso l'uso delle lingue straniere durante le attività.

2 Attività svolte nel Virtual Coding Tour

Durante il corso estivo innovativo denominato *STEAM Virtual Coding Tour*, finanziato dai fondi europei del PNRR DM65 [3], gli studenti del Liceo Cecioni hanno esplorato il mondo del coding e della realtà virtuale. Questo percorso intensivo, della durata di 12 ore, ha adottato un approccio pratico e multidisciplinare per illustrare le applicazioni concrete delle competenze STEM nel mondo reale. Le attività di *coding* hanno permesso agli studenti di utilizzare strumenti digitali per analizzare dati, creare modelli e risolvere problemi complessi, favorendo un'interazione diretta con l'intelligenza artificiale. Inoltre, l'uso delle lingue straniere durante le attività di *coding*, ha migliorato le loro competenze linguistiche e sottolineato l'importanza della comunicazione interculturale.

Le lingue straniere rivestono un ruolo cruciale nella formazione degli studenti, soprattutto in un contesto sempre più globalizzato e tecnologicamente avanzato dove sono sempre più necessarie. La padronanza di più lingue non solo facilita la comunicazione tra persone di diverse nazionalità, ma apre anche numerose opportunità professionali e accademiche. Durante il progetto, l'uso dell'inglese come lingua principale, ha permesso agli studenti di familiarizzare con il linguaggio tecnico del *coding*, rendendoli più competitivi nel mercato del lavoro internazionale ma anche l'uso di altre lingue come lo spagnolo ha reso il progetto più "multilingue" e sicuramente ha motivato molto alcuni ragazzi di altri indirizzi come il Liceo Linguistico. Inoltre, l'implementazione dell'utilizzo dell'intelligenza artificiale per l'apprendimento delle lingue straniere è stato un elemento innovatore di questo progetto perché è nell'ambito delle lingue straniere dove la IA ha più potenzialità dato che la combinazione dell'intelligenza artificiale e delle lingue straniere sta rivoluzionando l'apprendimento, offrendo strumenti personalizzati e interattivi che migliorano l'acquisizione linguistica.

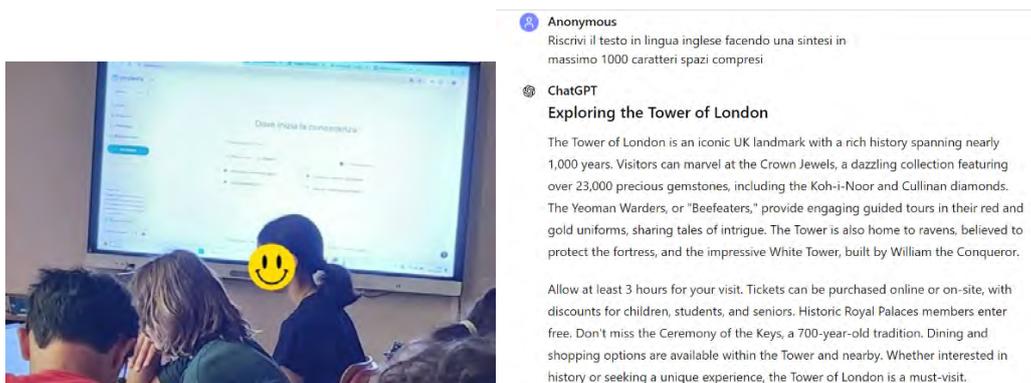


Fig-1 Autovalutazione sulla scrittura di prompt per applicazioni di intelligenza artificiale in ingresso

Per poter progettare il tour virtuale velocemente, sono state utilizzate diverse applicazioni di intelligenza artificiale che i corsisti hanno imparato ad utilizzare in modo corretto. Ad esempio per cercare informazioni sulle evidenze artistiche e storiche delle città visitate virtualmente, da far narrare ai personaggi virtuali in *Cospaces Edu* [4] o per generare le domande dell'*escape room*, hanno utilizzato *Perplexity.ai* [5]. Con questa applicazione, potendo eliminare le fonti non attendibili, gli alunni hanno potuto attuare correttamente il "fact-checking" che è una dei principali obiettivi della Commissione europea in base a precise indicazioni che ha fornito ai docenti ed educatori per contrastare la disinformazione dilagante sui social in particolare [1]. *Chatgpt* [6] di *OpenAI* è stato utilizzato per sintetizzare e/o tradurre in una lingua straniera i contenuti già verificati con *Perplexity*.

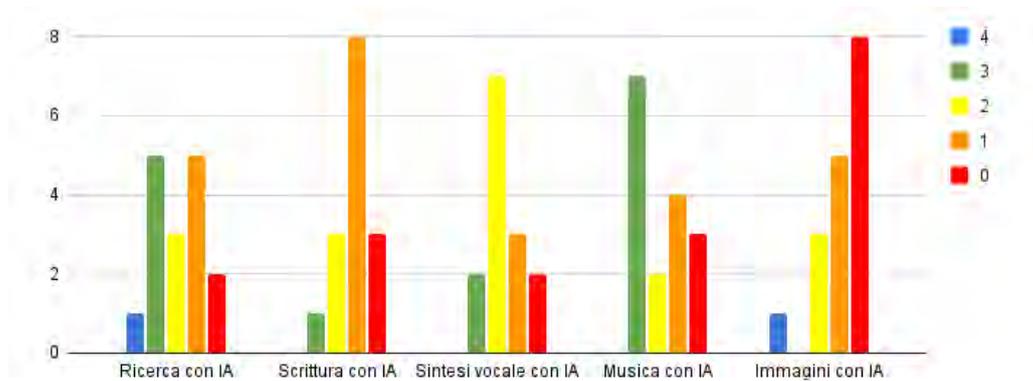


Fig.2- Scarto di apprendimento tra l'autovalutazione in uscita e in ingresso sull'uso di applicazioni di IA generativa

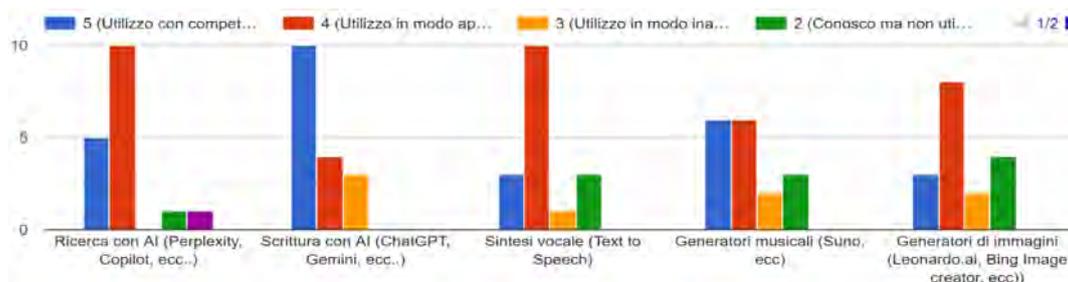


Fig.3 Autovalutazione sull'uso di applicazioni di intelligenza artificiale in uscita

Analizzando i grafici sullo scarto di apprendimento sull'uso delle applicazioni di intelligenza artificiale richiesti con moduli di autovalutazione in ingresso e in uscita, risulta evidente come la maggior parte degli studenti abbia percepito un miglioramento con il corso proposto. Solo con i generatori di immagine non hanno percepito miglioramenti sostanziali. Il motivo è stata la mancanza di tempo per approfondire anche questo aspetto.

Per quanto riguarda il coding, è stato utilizzato una web app basata sulla realtà virtuale: *Cospaces Edu* [4]. Questa applicazione ha permesso anche agli studenti dell'area linguistica, di costruire storie, personaggi, ambientazioni in 3D utilizzando strumenti di creazione intuitivi. La progettazione di mondi virtuali, oggetti e scenari, ha stimolato la loro creatività incidendo su questa precisa "life skills" [1] molto ricercata anche nel mondo del lavoro. Tuttavia la principale caratteristica di *CoSpaces Edu* molto utile per questo corso STEM con alunni di diversi percorsi liceali, è stata la possibilità di aggiungere interazioni ai propri elementi virtuali tridimensionali, utilizzando codice a blocchi o *scripting* avanzato. Questo approccio graduale alla programmazione ha permesso agli studenti di sviluppare competenze di coding, fondamentali nel contesto STEM. Attraverso l'uso di codice a blocchi, i principianti degli indirizzi Linguistico e delle Scienze umane, hanno iniziato a capire la logica di programmazione, mentre gli studenti del Liceo scientifico ad indirizzo STEM sono riusciti esplorare blocchi logici più complessi, migliorando le loro abilità di *problem solving* e pensiero computazionale.

Infine la possibilità di inserire in *Cospaces Edu*, delle immagini 360 gradi scaricate *Google Maps* [7] mediante l'applicazione "Street View download 360" [8] ha permesso un'esperienza coinvolgente e realistica aumentando la comprensione e l'engagement soprattutto degli alunni del liceo linguistico.



Fig-4 Virtual coding tour dalle immagini 360°gelocalizzate all'ambiente di coding in VR di Cospaces.io

Per svolgere questa attività gli alunni sono stati divisi in quattro gruppi eterogenei. Ad ogni alunno è stato affidato un compito che, nelle prime sei ore di corso (due giornate) hanno svolto a rotazione. I ruoli individuati per realizzare *l'European Virtual Tour* [9] sono stati: il responsabile delle scene 360° e della grafica 3D, l'esperto di *coding*, l'esperto di ricerca con AI e lo *storyteller* che si è preoccupato di svolgere il ruolo di regista mettendo insieme i vari elementi realizzati dagli altri componenti del gruppo e coordinando di fatto l'intero gruppo.

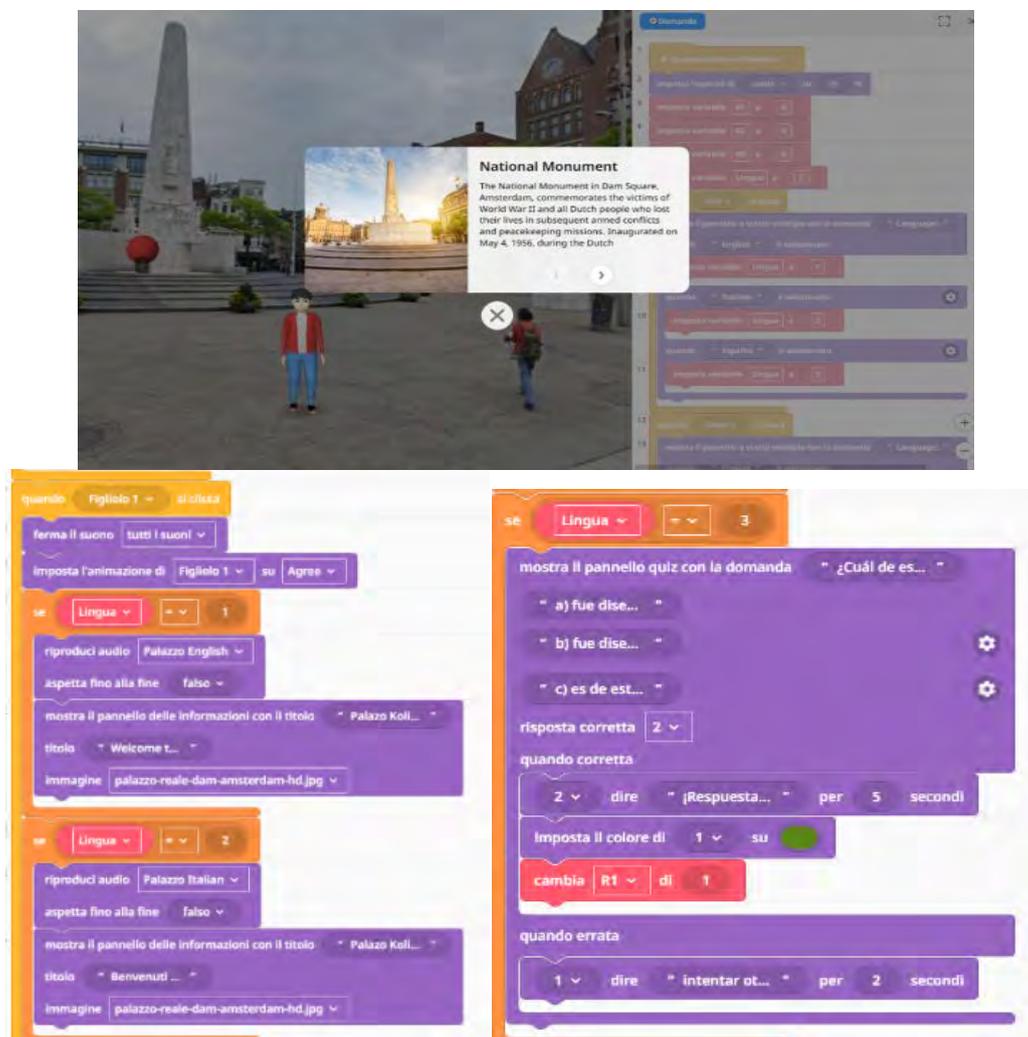


Fig-5 Esempi di coding su Cospaces.io dei personaggi interattivi in realtà virtuale

Negli ultimi due giorni, invece, gli alunni sono stati divisi in 4 gruppi omogenei. I ragazzi del liceo scientifico con indirizzo STEM, sono stati divisi in due gruppi. Il primo, di livello base, si è occupato di realizzare un gioco con blocchi di fisica ambientato nello scenario della foto 360°. L'altro gruppo dell'indirizzo STEM, invece, si è occupato di progettare e realizzare il *videogame* "Physic boat" [10] dove poter regolare grandezze fisiche per simulare il passaggio di una nave in un porto. Hanno simulato la presenza dei venti con blocchi che regolavano il vento da una rosa dei venti virtuale. Inoltre hanno potuto verificare come l'effetto dell'attrito, della velocità e della massa potevano incidere sull'inerzia della nave e quindi sulla difficoltà di portare a termine la missione col massimo punteggio e nei tempi prestabiliti. Oltre a costruire il porto con la grafica 3d, hanno impostato diverse variabili:

- il punteggio che diminuisce ogni volta che la barca urta il porto;
- il tempo che diminuisce grazie ad un uso saggio del blocco "attendi per un secondo" ;
- le variabili fisiche come l'attrito, la velocità e la massa che cambiano al variare dei rispettivi blocchi di codice già presenti sulla *web app Cospaces Edu*.

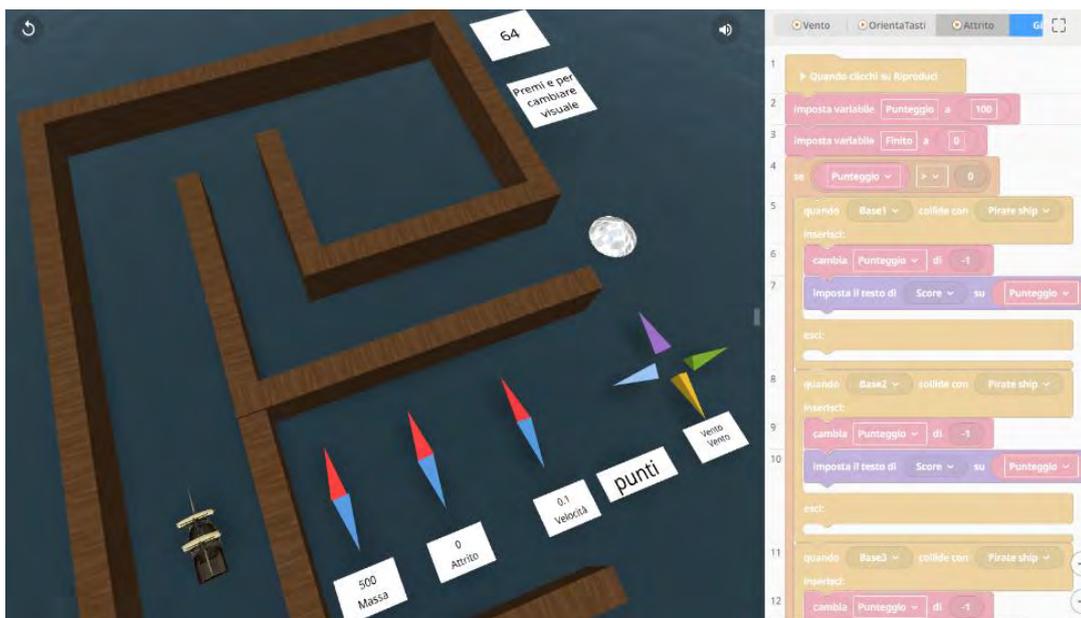


Fig.6 - Esempio di gioco con simulazione di fisica dell'ingresso della barca in un porto realizzato con Cospaces.io

Un aspetto fondamentale di questo lavoro ha riguardato la ricerca degli errori di programmazione che ha permesso con il lavoro di squadra l'individuazione di ogni singolo "bug" la cui eliminazione è stata affidata a ciascun alunno dall'alunno coordinatore del gruppo (Fig.6).



Fig.7- Gruppo cooperativo che si è occupato di in un porto realizzato con Cospaces.io

I due gruppi rimanenti di area umanistica, invece, hanno simulato la realizzazione di un pacchetto turistico sulla città di Livorno mettendo in gioco diverse competenze digitali acquisite durante il corso.

Per aumentare la motivazione, gli alunni più meritevoli, ad ogni traguardo raggiunto, hanno ricevuto un badge che poi è stato stampato sull'attestato finale (Fig.8).



Fig.8- Badge di gamification per aumentare il coinvolgimento dei corsisti nelle attività STEM con il Coding in VR

Dai questionari finali, è emerso che la maggioranza degli alunni ha percepito miglioramenti sull'uso di blocchi di coding specialmente finalizzata alla realizzazione di giochi in realtà virtuale. La parte interessante delle risposte rappresentate nell'istogramma (Fig.9) riguarda la personale autovalutazione che ha visto la maggioranza degli alunni ritenere i migliori progressi soprattutto nei blocchi di coding STEM avanzati che in *Cospaces* si manifestano con le funzioni logico-matematiche in cui sono già impostati parametri che simulano grandezze fisiche.

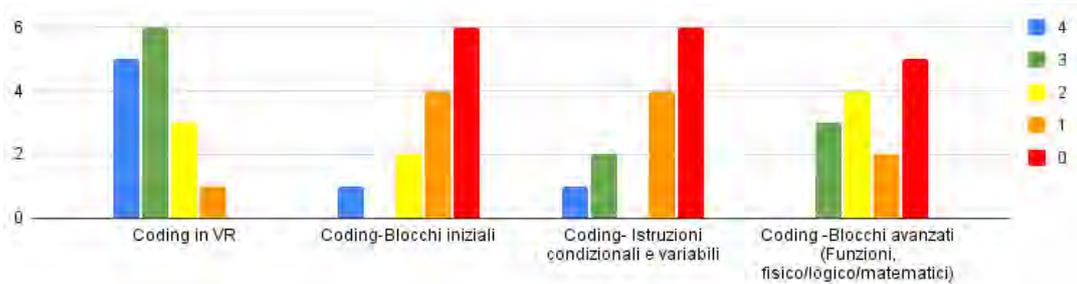


Fig.9 -Scarto di apprendimento tra l'autovalutazione in uscita e in ingresso sulle competenze di coding

3 Conclusioni

Nel questionario finale [11] gli alunni hanno risposto di aver percepito non solo dei miglioramenti nelle discipline STEM ma anche di aver potenziato diverse *life skills*. Dai grafici pubblicati sui risultati sull'autovalutazione del corso si notano particolarmente i progressi percepiti negli alunni nel *problem solving*, nell'analizzare informazioni e nel saper utilizzare la propria creatività. Buoni risultati si evidenziano anche sulla capacità di comunicare in modo efficace e sulla gestione delle relazioni tra pari.

In particolare dalle risposte alla domanda "Pensi che il tuo futuro lavoro coinvolgerà in qualche modo anche le discipline STEM?" (Fig.10) risulta evidente come la percentuale di chi non era propenso a svolgere lavori inerenti le discipline STEM, sia diminuita come conseguenza diretta del corso.

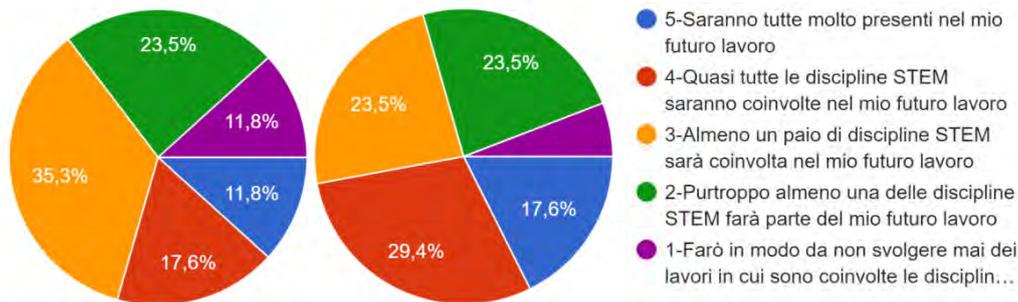


Fig.10 - Risposte alla domanda sull'orientamento lavorativo verso le discipline STEM prima e dopo il corso

In generale, dai risultati pubblicati sul gradimento del corso da parte degli studenti, le attività proposte sembrano aver ottenuto un alto gradimento in merito agli obiettivi programmati in fase di progettazione.

Bibliografia

- [1] S. Arianna, P. Yves, G. Vladimir e C. G. Marcelino, «LifeComp: The European Framework for Personal, Social and Learning to Learn Key Competence,» *Publications Office of the European Union*, 2020.
- [2] J. R. C. V. R. K. S. P. Y. European Commission, «DigComp 2.2, The Digital Competence framework for citizens : with new examples of knowledge, skills and attitudes,» 2022. [Online]. Available: <https://data.europa.eu/doi/10.2760/115376>.
- [3] MIM, «NUOVE COMPETENZE E NUOVI LINGUAGGI (DM 65/2023),» 2023. [Online]. Available: <https://pnrr.istruzione.it/avviso/dm-65-nuove-competenze-e-nuovi-linguaggi-riparto-delle-risorse/>.
- [4] WS Datenschutz GmbH, «Cospaces EDU,» [Online]. Available: <https://edu.cospaces.io/>.
- [5] Perplexity AI, Inc., «Perplexity AI,» [Online]. Available: <https://www.perplexity.ai/>.
- [6] OpenAI, «ChatGPT (Versione 3.5),» 2021. [Online]. Available: <https://openai.com/>.
- [7] Google LLC, «Google Maps,» 2024. [Online]. Available: <https://www.google.it/maps/>.
- [8] Google LLC, «Street View Download 360,» 2024. [Online]. Available: <https://svd360.istreetview.com/>.
- [9] G. Silvestro, «European Virtual Tour,» 2024. [Online]. Available: <https://edu.cospaces.io/JXD-MMD>.
- [10] G. Silvestro, «Physic Boat,» 2024. [Online]. Available: <https://edu.cospaces.io/TJK-ZXY>.
- [11] G. Silvestro, «Risultati dei questionari del corso Virtual Coding Tour,» 2024. [Online]. Available: https://drive.google.com/file/d/1OnuxF1lagG6P-AMmDv5_GbIB9uL5PcCV/view?usp=sharing.
- [12] European Commission, Directorate-General for Education, Youth, Sport and Culture, «Guidelines for teachers and educators on tackling disinformation and promoting digital literacy through education and training,» 2022. [Online]. Available: <https://data.europa.eu/doi/10.2766/28248>.

Un primo contatto con gli algoritmi di Intelligenza Artificiale: verso un'opinione informata

Matteo Moro^{1,2}, Nicoletta Noceti^{1,2}, Francesca Odone^{1,2}, Silvia Villa^{2,3}

¹Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS),
Università di Genova, Genova, Italia

²Machine Learning Genoa (MaL.Ga) Center, Genova, Italia

³Dipartimento di Matematica (DIMIA), Università di Genova, Genova, Italia

{[matteo.moro](mailto:matteo.moro@unige.it), [nicoletta.noceti](mailto:nicoletta.noceti@unige.it), [francesca.odone](mailto:francesca.odone@unige.it),
[silvia.villa](mailto:silvia.villa@unige.it)}@unige.it

Abstract

Negli ultimi anni, l'intelligenza artificiale (IA) è diventata una componente fondamentale della tecnologia e dei servizi, influenzando numerosi aspetti della vita quotidiana, in modo forse imprevisto. Come conseguenza è molto facile imbattersi in articoli o contenuti sull'Intelligenza Artificiale che suonano o troppo entusiasti o eccessivamente preoccupanti creando un divario tra la percezione comune e la realtà di questa tecnologia. E' nostra convinzione che questo sia in parte dovuto all'(apparente) distanza dell'IA dalle conoscenze dell'utente medio non esperto e dalle materie comunemente studiate a scuola. Partendo da tale assunto, questo contributo esplora l'importanza di introdurre nelle scuole i concetti su cui l'IA si fonda, e ragiona sul fatto che alcuni elementi base possono essere introdotti con molta naturalezza, senza l'utilizzo di concetti matematici o informatici avanzati, e anzi legandoli a concetti affrontati comunemente nel percorso della scuola secondaria superiore. In aggiunta all'introduzione di alcune competenze di base, il nostro contributo si pone l'ulteriore l'obiettivo di fornire gli strumenti necessari per comprendere e valutare criticamente i rischi e le opportunità dell'IA. Come casi di studio, presenteremo due strumenti che possono essere utilizzati come punto di partenza per introdurre i principi e gli algoritmi dell'IA nelle scuole: *Teachable Machine*, un tool online per che permette di toccare con mano gli elementi fondamentali del machine learning, il cuore dell'Intelligenza Artificiale moderna; *MeetAI*, un laboratorio interattivo organizzato dal Machine Learning Genoa center (MaL.Ga) durante il Festival della Scienza di Genova, che ci ha permesso di ragionare sul come si possano descrivere in modo giocoso i concetti matematici che si trovano nel cuore dell'IA.

1 Introduzione

L'Intelligenza Artificiale (IA) è diventata un argomento di grande rilevanza, grazie ai progressi tecnologici e alla sua crescente applicazione in vari settori [1]. Tuttavia, la comprensione di come funzionano gli algoritmi di IA rimane spesso limitata a una cerchia ristretta di esperti. Mentre stanno aumentando i contenuti autorevoli per la divulgazione dell'intelligenza artificiale intesa come scienza (citiamo come esempio un articolo apparso di recente su *Frontiers for Young Minds*, nato da un'intervista a Yann LeCun, uno dei padri del Deep Learning [2] o uno dei libri di Nello Cristianini, per esempio il recentissimo [3]), sono ancora

molte le informazioni poco accurate associate alla materia. Con l'aumento dell'interesse nella materia stanno anche aumentando contenuti divulgativi di varia natura che comprendono articoli giornalistici, TED¹ o PodCast sull'argomento². L'Intelligenza Artificiale non è solo una scienza: il suo recente impatto sulla vita di tutti noi la rende una disciplina da conoscere in un percorso di cittadinanza digitale consapevole [4,5]. Per questo motivo, è essenziale introdurre i fondamenti di IA nell'età scolare. In un'epoca in cui l'IA è al centro di molti dibattiti pubblici e scientifici, è infatti fondamentale che le nuove generazioni siano in grado di capirne i principi, le potenzialità, ma anche le sfide ed i problemi. Un primo contatto con questi concetti può avvenire attraverso strumenti didattici e laboratori interattivi che rendano l'apprendimento accessibile e coinvolgente. Negli ultimi anni, sono stati sviluppati strumenti disponibili online che consentono di avvicinarsi al mondo dell'IA fin dall'età della scuola primaria. Ad esempio, Hour of Code (L'Orchestra del Codice)³ è un progetto sviluppato a livello mondiale che fornisce un'introduzione alla programmazione e al problem solving con modalità adatte alle diverse fasce d'età. La celebrazione del 2024, oltre che sulla programmazione, si focalizza sull'Intelligenza Artificiale, con risorse disponibili per bambini e insegnanti. Come altro esempio, Machine Learning for Kids⁴ è uno strumento disponibile online che introduce ai concetti fondamentali del Machine Learning, cuore dell'Intelligenza Artificiale moderna, in modo intuitivo e divertente. In entrambi i casi, esiste la possibilità di utilizzare un linguaggio di programmazione a blocchi come Scratch [6], rendendo l'esperienza ampiamente accessibile anche ai giovanissimi.

Conoscere le basi degli algoritmi di IA permette agli studenti non solo di acquisire competenze tecniche, ma anche di sviluppare un pensiero critico riguardo alle implicazioni sociali ed etiche di queste tecnologie e formarsi un'opinione informata. Capire meglio le debolezze dell'IA moderna consente di formarsi un'opinione su quali siano i benefici ed i rischi del suo ampio utilizzo nella nostra vita quotidiana.

2 Casi di studio

2.1 Teachable machine

Teachable Machine⁵ è un tool online sviluppato da Google che consente a chiunque di creare modelli di Machine Learning senza la necessità di saper programmare. Gli utenti possono addestrare modelli di classificazione (o riconoscimento) di immagini, suoni e pose del corpo umano, semplicemente utilizzando la propria webcam o il microfono del computer. Questo strumento rende l'Intelligenza Artificiale accessibile e comprensibile. Gli studenti possono sperimentare direttamente con il Machine Learning, comprendendo come i dati vengono utilizzati per addestrare un modello e come questo modello può essere applicato a problemi reali. Attraverso Teachable Machine è possibile ragionare sulla formulazione di un problema di classificazione, sulle criticità della raccolta dati in un contesto di "apprendimento da esempi", sulla generalizzazione di un modello allenato rispetto a nuovi dati.

Teachable Machine rappresenta un esempio concreto di come sia possibile introdurre concetti complessi in modo semplice e intuitivo, stimolando l'interesse e la curiosità degli studenti. Abbiamo utilizzato Teachable machine in svariati momenti di incontro con studenti del triennio

¹ Per un esempio molto ispirato ed efficace si può far riferimento ai TED Talk di Fei Fei Li sulla Computer Vision e l'IA

² Un ottimo contenuto prodotto dall'università di Padova: <https://ilbolive.unipd.it/it/podcast/bytebite-lai-piccoli-bocconi-macchine-ingannevoli>
Per una selezione di PodCast sul tema IA in lingua italiana <https://blog.pigro.ai/it/migliori-podcast-ai-intelligenza-artificiale>

³ <https://hourofcode.com/it/ai>

⁴ <https://machinelearningforkids.co.uk/#!/about>

⁵ <https://teachablemachine.withgoogle.com/>

delle scuole secondarie superiori, in occasione di attività di orientamento. Lo strumento si presta a piccoli laboratori di 30 minuti, ma può anche essere adattato ad attività più lunghe.

2.2 MeetAI

MeetAI è un laboratorio organizzato dal Machine Learning Genoa Center (MaLGA) nell'ambito del Festival della Scienza, basato sulla convinzione che sia possibile introdurre concetti complessi in modo semplice e intuitivo, stimolando l'interesse e la curiosità degli studenti. Abbiamo utilizzato Teachable machine in svariati momenti di incontro con studenti del triennio delle scuole secondarie superiori, in occasione di attività di orientamento. Tenutosi per due anni consecutivi, questo laboratorio ha riscosso grande successo tra giovani e adulti. L'obiettivo di MeetAI è sensibilizzare il pubblico riguardo agli algoritmi alla base del processo di apprendimento automatico, in particolare gli algoritmi di ottimizzazione, ad esempio il *gradient descent* e sue varianti [7]. Durante il laboratorio, i partecipanti hanno avuto l'opportunità di comprendere alcuni elementi che stanno alla base della discesa del gradiente, grazie ad una serie di attività che hanno descritto in maniera “allegorica” concetti di per sé complessi quali il costo computazionale nello svolgere calcoli, la valutazione degli errori attraverso l'uso di funzioni costo, la differenza tra approcci deterministici e randomizzati.

MeetAI rappresenta un esempio di come sia possibile coinvolgere il pubblico in maniera attiva, rendendo comprensibili concetti complessi e dimostrando l'impatto concreto degli algoritmi di IA sulla vita quotidiana. Il concetto di “costo legato ai calcoli” è occasione di discutere l'impatto che l'addestramento di grandi modelli di Machine Learning disponibili oggi ha sull'ambiente, e di ragionare su come un uso consapevole delle risorse di calcolo possa aiutare a preservare la salute del nostro pianeta⁶.

3 Conclusioni

L'introduzione di concetti di Intelligenza Artificiale nelle scuole è diventata un'esigenza per preparare le future generazioni alle sfide tecnologiche che ci attendono. Strumenti come Teachable Machine e laboratori interattivi come MeetAI dimostrano che è possibile rendere accessibili e comprensibili queste tecnologie, stimolando l'interesse e la curiosità degli utenti. La diffusione della conoscenza sugli algoritmi di IA non solo arricchisce il bagaglio educativo dei giovani, ma contribuisce anche a costruire una società più consapevole e informata. Investire nell'educazione all'IA significa investire nel futuro, garantendo che le nuove generazioni siano pronte a sfruttare al meglio le opportunità offerte da queste tecnologie e a confrontarsi con le sfide etiche e pratiche che esse comportano.

I contenuti a cui facciamo riferimento nelle attività menzionate si riferiscono principalmente al segmento di algoritmi “discriminativi”, dove l'elemento essenziale è la capacità di distinguere (classificare) elementi diversi. Gran parte di quello che oggi viene chiamato IA dai non esperti è legato ad algoritmi “generativi”, tra i quali comprendiamo i large language models che stanno avendo un grande impatto anche sul percorso educativo degli studenti di oggi⁷. Per questo motivo sarebbe ancora più importante inserirli all'interno del percorso di studi, non solo come strumento da utilizzare, ma anche come algoritmo da comprendere. In conclusione al contributo presenteremo suggerimenti su come intraprendere questa strada.

⁶ <https://www.ilpost.it/2024/06/28/energia-consumata-intelligenza-artificiale/>

⁷ AI in education: where we are and what happens next, Oxford University Press <https://corp.oup.com/feature/ai-in-education-where-we-are-and-what-happens-next/>

Acknowledgement

Questo contributo è stato realizzato nell'ambito di due progetti:

- 1) "TraDE-OPT" è un progetto finanziato dalla Commissione Europea attraverso il programma Horizon 2020 Research and Innovation: Marie Skłodowska-Curie Actions (grant No. 861137).
- 2) "RAISE—Robotics and AI for Socio-economic Empowerment" è un progetto finanziato dalla Commissione Europea—NextGenerationEU e dal Ministero dell'Università e della Ricerca (MUR) attraverso il Piano Nazionale di Ripresa e Resilienza (PNRR) (grant No. ECS00000035)

Bibliografia

- [1] Yin, Jiamin, Kee Yuan Ngiam, and Hock Hai Teo. "Role of artificial intelligence applications in real-life clinical practice: systematic review." *Journal of medical Internet research* 23.4 (2021): e25759.
- [2] LeCun, Yann. "[Will Learning Machines Take Over the World?](#)" *Frontiers for young minds* (May 2024)
- [3] Cristianini, Nello. "Machina Sapiens. L'algoritmo che ci ha rubato il segreto della conoscenza" *Il Mulino* (2023)
- [4] Zawacki-Richter, Olaf, et al. "Systematic review of research on artificial intelligence applications in higher education—where are the educators?." *International Journal of Educational Technology in Higher Education* 16.1 (2019): 1-27.
- [5] Chen, Xieling, et al. "Application and theory gaps during the rise of artificial intelligence in education." *Computers and Education: Artificial Intelligence* 1 (2020): 100002.
- [6] Maloney, John, et al. "The scratch programming language and environment." *ACM Transactions on Computing Education (TOCE)* 10.4 (2010): 1-15.
- [7] Amari, Shun-ichi. "Backpropagation and stochastic gradient descent method." *Neurocomputing* 5.4-5 (1993): 185-196.

Esperienze di Apprendimento Attivo e Collaborativo nel Progetto “Secoli Bui? Il Medioevo tra Reale e Virtuale”

Maria Concetta Brocato¹, Nicola Zangrando², Biagio Tomasetig²,
Letizia Donnini¹, Marzia Toso¹, Fabio Buttussi²

¹ ISIS Arturo Malignani, Udine, Italia

mariaconcetta.brocato@malignani.ud.it,
letizia.donnini@malignani.ud.it, marzia.toso@malignani.ud.it

² Università degli Studi di Udine

nicola.zangrando@uniud.it, tomasetig.biagio@spes.uniud.it,
fabio.buttussi@uniud.it

Abstract

L'articolo illustra alcune esperienze di apprendimento collaborativo realizzate all'interno del progetto "Secoli bui? Il Medioevo tra reale e virtuale" che ha coinvolto due classi quarte del Liceo delle Scienze Applicate dell'ISIS "A. Malignani" di Udine e l'Università degli Studi di Udine. Propone modalità innovative per la costruzione di competenze informatiche in un'ottica orientativa, valorizzando, nel contempo, quelle umanistiche e matematiche, partendo dall'osservazione del reale per arrivare alla realtà virtuale.

1 Introduzione

L'attività è stata realizzata in un Liceo delle Scienze Applicate (LSA) ed ha coinvolto due classi quarte, utilizzando modalità innovative per l'ambito educativo. Ci si riferisce all'uso dell'interdisciplinarietà che ha unito le materie STEM alle HUMANITIES ed all'impiego di metodologie [1] [2] quali: Peer Tutoring [3] [4] [5], Collaborative Learning [6] [7], Outdoor Education [8] [9], Technology Enhanced Active Learning [10] [11], Making Learning and Thinking Visible [12] [13] con focus orientativo, legato alla maturazione di competenze, incentrato sulla laboratorialità e basato su contesti di realtà. Si è applicato il livello “Redefinition” del modello tassonomico SAMR [14].

Relativamente all'informatica è stata introdotta la Object Oriented Programming (OOP), utilizzando il linguaggio C# e il motore di gioco Unity 3D per realizzare un museo virtuale, ambientato all'interno del Duomo di Siena, ed offrire un'esperienza immersiva ludica (Figura 1).

La scelta è collegata al fatto che la OOP, seppure prevista nelle Linee Guida Ministeriali, non è un argomento facile per gli studenti [15] [16]. L'importanza della tematica unita alla volontà di realizzare una efficace attività di orientamento [17] alle professioni informatiche, ha portato al coinvolgimento dell'HCI Lab dell'Università degli Studi di Udine conosciuto attraverso il Piano Lauree Scientifiche. L'attività afferisce al progetto Ministeriale "Metodologie Didattiche Innovative" – PNSD Avviso 20769/2019.

L'esperienza ha considerato l'informatica come volano in grado di valorizzare sia le soft skills, sia altre competenze, quali le abilità di:

- modellazione matematica per realizzare simulazioni della realtà (Figura 2);
- misura con strumenti topografici per rilevare spazi esterni;

- modellazione CAD per riportare misure e riprodurre parti della città di Siena;
- stampa 3D per la realizzazione di modelli in scala di parti del Duomo.



Figura 1: Schermata iniziale del museo virtuale

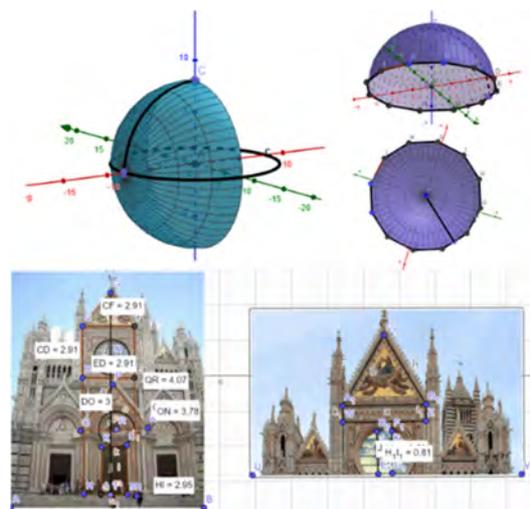


Figura 2: Costruzione della cupola, coniche e numero aureo.

Le attività sono state realizzate collaborativamente, con assegnazione di approfondimenti diversi, condivisione e discussione tra gruppi e classi (metodologia JIGSAW [18]). A titolo di esempio, partendo da approfondimenti disciplinari storici, filosofici e letterari sul Medioevo e sull'architettura del Duomo di Siena, alcuni gruppi hanno studiato fregi, rosoni e tassellazioni del piano e ricostruito elementi tipici delle cattedrali gotiche mediante i software di geometria dinamica, altri hanno ricercato proporzioni auree ed applicazioni nella realtà delle coniche e della geometria analitica, altri ancora hanno realizzato artefatti digitali e prototipi.

Gli artefatti sono stati completamente progettati dagli studenti, con il supporto dei docenti che hanno guidato il processo di acquisizione delle conoscenze e incoraggiato forme attive di apprendimento. La progettazione e lo sviluppo dell'applicazione del museo ha coinvolto una delle due classi ma i risultati sono stati condivisi con l'altra.

2 Metodologia

La sfida orientativa posta agli studenti è stata di non essere fruitori di software, ma progettisti e programmatori di un'applicazione originale, in realtà virtuale, con uso di visori 3D.

Per lo sviluppo del museo, la classe è stata suddivisa in 10 coppie (Figura 3). Le stesse hanno collaborato attraverso brainstorming per definire una trama comune e le modalità di fruizione, interazione e navigazione all'interno del museo.

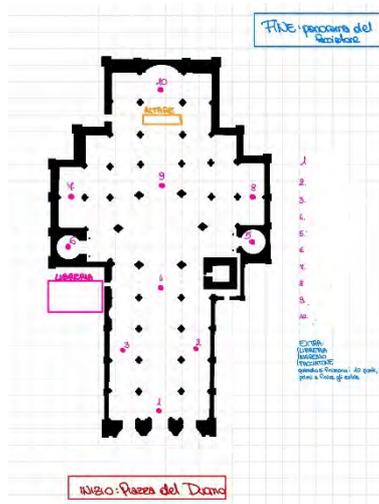


Figura 3: Pianta del Duomo e suddivisione gruppi di lavoro.

A ciascuna coppia è stato assegnato un settore del Duomo, per il quale dovevano ideare contenuti interattivi e informativi che permettessero ai visitatori di apprendere in modo ludico la sua storia. Ogni gruppo ha creato il codice per la gestione delle attività interattive, scritto testi originali (Figura 4) e registrato l'audio e le animazioni utilizzati dai personaggi virtuali che avrebbero guidato i visitatori lungo il percorso nel museo.

Zona: Entrata del Duomo, 1

Frase recitata dal personaggio: "Ti do il benvenuto, nobile ospite, all'entrata di questa sontuosa cattedrale, imponente simbolo di fede e maestosità, che si erge con impavido splendore nel cuore dell'antico borgo di Siena. L'insieme delle raffigurazioni che puoi vedere riportano alla mente il ramificato percorso che l'uomo intraprende alla ricerca della sapienza divina. Esse ti accompagneranno alla scoperta di tutta la cattedrale."

Frase di presentazione dell'enigma: "Ma prima, per proseguire e scoprire il significato nascosto del duomo di Siena, dovrai essere in grado di riuscire a trovare l'uscita attraverso questo confuso percorso. Fai attenzione ai vari ostacoli."

Nome enigma: "Il labirinto"

Descrizione enigma: Si tratta di un labirinto, la quale l'utente deve trovare l'uscita. Non è molto lungo, ma ci sono all'interno dei piccoli tranelli quali: muri invisibili, due personaggi che ti pongono dei quesiti, porte segrete e trappole. Inoltre, all'interno del labirinto ci saranno molti vasi e uno solo contiene una mappa del labirinto.

Soluzione: trovare l'uscita



Figura 4: Esempio di scrittura collaborativa di un testo, la guida virtuale all'interno del museo.

2.1 Formazione iniziale

Gli studenti sono stati introdotti al motore di gioco Unity relativamente all'uso dell'interfaccia e delle sue funzioni fondamentali. In seguito, hanno approfondito la programmazione del movimento degli oggetti, le collisioni, l'elaborazione degli input dell'utente, la gestione dell'audio, dei materiali e della fisica. Gli argomenti trattati sono stati affiancati da esercizi applicativi, consentendo agli studenti di mettere subito in pratica le conoscenze acquisite.

2.2 Attività Outdoor: Visita al Duomo di Siena

L'attività di Outdoor, propedeutica alla realizzazione del museo, è stata progettata con la collaborazione dell'avv. Verdoliva dell'Opera Duomo Siena. In questo contesto, l'arch. De

Benedetti, ha accolto gli studenti e presentato alcune delle caratteristiche del Duomo permettendo agli stessi di acquisire maggiore consapevolezza dell'importanza del manufatto. Il percorso di visita ha offerto una panoramica dall'alto del Duomo, permettendo agli studenti di ammirare ciò che avevano analizzato in aula: il "facciato", la libreria Piccolomini, il soffitto, la cupola, la vetrata, le cappelle musicali e il pavimento con le importanti tarsie marmoree. Il ruolo dei docenti e degli esperti è stato quello di porre stimoli, fornire chiavi interpretative e facilitare il processo di comprensione e condivisione.

La visita ha coinvolto gli studenti nella preparazione dei materiali da utilizzare nel museo virtuale: sono state scattate, utilizzando una macchina fotografica a 360° (Figura 5), foto dei singoli settori del Duomo, sono state raccolte impressioni personali, registrati video, descritti gli ambienti. Inoltre, sono state prese delle misure di interni ed esterni per poter realizzare ulteriori studi matematici e architettonici, analizzando i dati raccolti e mettendoli in relazione ai materiali messi a disposizione dall'Opera Duomo di Siena.



Figura 5: Esempio di foto a 360°.

2.3 Progettazione

Dopo la formazione iniziale, gli studenti hanno avuto l'opportunità di mettere in gioco la propria creatività, proponendo idee per attività ludiche interattive collegate agli interni del Duomo di Siena. Il docente ha guidato la discussione, valutato la fattibilità e fornito feedback, garantendo che le proposte fossero realizzabili in base alle competenze sviluppate.

La fase di modellazione dei dati e progettazione del software è stata condotta in modo metodico e strutturato. Gli studenti hanno suddiviso il Duomo in diverse aree tematiche, assegnando a ciascun gruppo la responsabilità di una specifica sezione. Ogni gruppo ha condotto una ricerca approfondita per identificare le informazioni storiche, artistiche e culturali rilevanti da illustrare.

Sulla base dei risultati della ricerca, gli studenti hanno progettato attività interattive strettamente correlate ai temi emersi. Ad esempio, per una zona con mosaici raffiguranti le sibille, è stato ideato un gioco di carte in stile Memory. In questa attività, ogni volta che veniva trovata una coppia di carte corrispondenti, una guida virtuale forniva una breve narrazione sulla storia e il significato della sibilla rappresentata.

Conclusa la progettazione, gli studenti sono passati all'implementazione delle attività (Figura 6). In questa fase, il docente ha continuato a offrire supporto e consigli, permettendo però agli studenti di gestire in autonomia il proprio lavoro e affrontare le sfide che si presentavano.

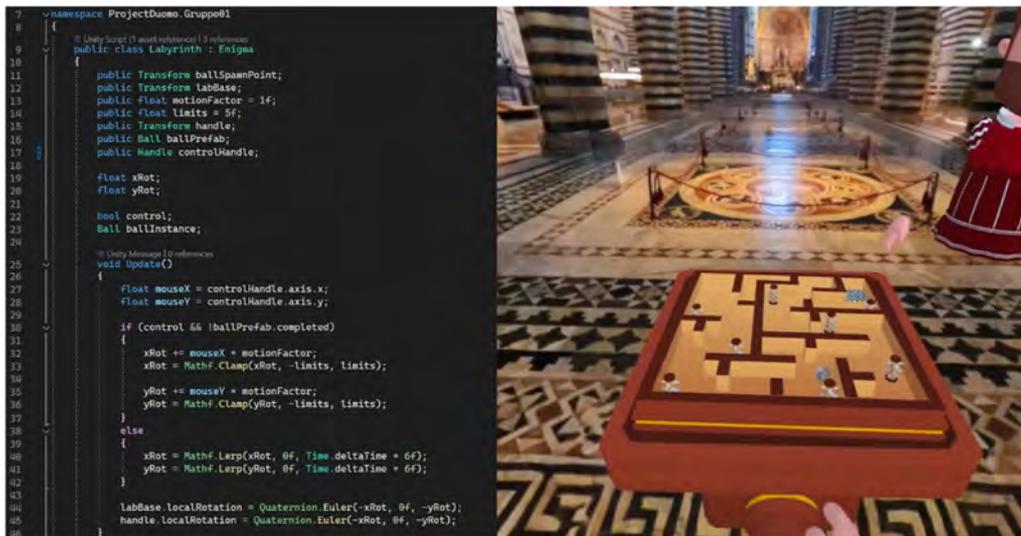


Figure 6: Esempio di implementazione di un'attività interattiva: codice in C# e risultato in Unity.

3 Risultati

L'applicazione realizzata offre un'esperienza interattiva e coinvolgente centrata sul Duomo di Siena. All'inizio dell'esperienza, il visitatore è accolto da un personaggio virtuale guida che spiega lo scopo dell'applicazione e fornisce istruzioni su come interagire con l'applicazione. Terminata la spiegazione, il visitatore viene trasportato nell'area principale dell'esperienza, dove trova una riproduzione in rilievo della mappa del Duomo di Siena (Figura 7), con diverse "sfere immersive" disposte su di essa. Selezionando una sfera, il visitatore viene immerso in una visualizzazione a 360° del settore corrispondente del Duomo [19]. All'interno di ogni ambiente, un personaggio virtuale offre una breve presentazione, seguita da una delle attività interattive realizzate dagli studenti, tematicamente legata al settore specifico (Figura 8).

Alla conclusione di ogni attività, una chiave animata appare e sblocca simbolicamente un lucchetto in una sfera centrale sulla mappa, tracciando i progressi e incentivando il completamento delle attività. Una volta visitati tutti i settori, è possibile accedere all'ultima sfera, dove è possibile osservare Siena dall'alto e concludere l'esperienza.

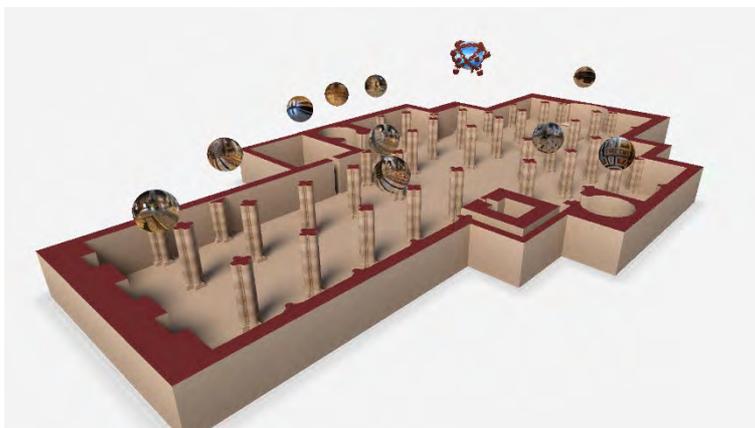


Figure 7: Percorso museale e "sfere immersive".

Questa struttura “gamificata” [20] combina elementi educativi, esplorazione e sfide interattive, creando un'esperienza coinvolgente che permette ai visitatori di scoprire e apprezzare il Duomo di Siena in modo innovativo e memorabile.

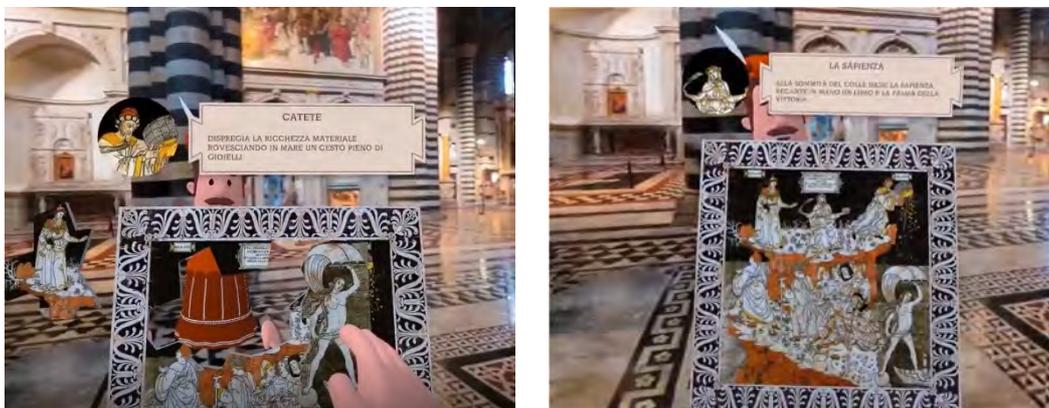


Figura 8: Esempio, di attività interattiva, composizione di un puzzle con trascinnamento di pezzi.

4 Conclusioni

In una società complessa, è cruciale una didattica innovativa che possa incidere efficacemente nella formazione. La scuola moderna dovrebbe adottare la didattica per competenze come obiettivo primario per affrontare le sfide attuali. Abilità come la comprensione, la verifica e la gestione dei dati, insieme alla capacità di sintetizzare e valutare testi e fonti di varia natura, sono cruciali e non scontate.

È fondamentale una didattica attiva che sviluppi la capacità di lavorare in gruppo, confrontarsi tra pari, accettare posizioni diverse e scambiarsi opinioni utilizzando l'argomentazione critica e documentata. La metodologia adottata, che ha combinato attività laboratoriali con ambienti digitali e strumenti tecnologici, può integrare queste competenze nelle dinamiche della modernità, gestendo anche aspetti di correttezza formale ed etica.

Questa didattica è necessaria non solo per generare entusiasmo negli studenti, che può non essere duraturo, ma per sviluppare abilità che permettono di affrontare il mondo, proprio come la scuola ha sempre voluto fare, adattandosi ai tempi e agli strumenti moderni.

Bibliografia

- [1] INDIRE - Avanguardie Innovative, "Il Manifesto delle Avanguardie educative," 2014. [Online]. Available: <https://innovazione.indire.it/avanguardieeducative/il-manifesto>. [Accessed luglio 2024].
- [2] L. Dierking, "Learning Theory and Learning Styles: An Overview.," *Journal of Museum Education*, vol. 16, no. 1, pp. 4-6, 1991.
- [3] INDIRE - Avanguardie Educative, "Apprendimento autonomo e tutoring," 2014. [Online]. Available: <https://innovazione.indire.it/avanguardieeducative/apprendimento-autonomo-tutoring>. [Accessed luglio 2024].
- [4] K. J. Topping, "The effectiveness of peer tutoring in further and higher education: A typology and review of the literature.," *Higher education*, vol. 32, no. 3, pp. 321-345, 1996.

- [5] S. Goodlad and B. Hirst, "Peer Tutoring. A Guide to Learning by Teaching.," 1989.
- [6] T. H. Laine and W. Lee, "Collaborative Virtual Reality in Higher Education: Students' Perceptions on Presence, Challenges, Affordances, and Potential," *IEEE Transactions on Learning Technologies*, vol. 17, pp. 280-293, 2024.
- [7] T. J. Nokes-Malach, J. E. Richey and S. Gadgil, "When is it better to learn together? Insights from research on collaborative learning," *Educational Psychology Review*, vol. 27, pp. 645-656, 2015.
- [8] INDIRE - Avanguardie Educative, "Outdoor education," 2014. [Online]. Available: <https://innovazione.indire.it/avanguardieeducative/outdoor-education>. [Accessed luglio 2024].
- [9] E. Iversen and K. B. Remmen, "A Scoping Review of Research on School-Based Outdoor Education in the Nordic Countries.," *Journal of Adventure Education and Outdoor Learning*, vol. 23, no. 4, pp. 433-451, 2022.
- [10] INDIRE - Avanguardie Educative, "TEAL (Tecnologie per l'apprendimento attivo)," 2014. [Online]. Available: <https://innovazione.indire.it/avanguardieeducative/teal>. [Accessed luglio 2024].
- [11] A. Kirkwood and L. Price, "Technology-Enhanced Learning and Teaching in Higher Education: What Is 'Enhanced' and How Do We Know? A Critical Literature Review.," *Learning, Media and Technology*, vol. 39, no. 1, pp. 6-36, 2013.
- [12] INDIRE - Avanguardie Educative, "MLTV - Rendere visibili pensiero e apprendimento," 2014. [Online]. Available: <https://innovazione.indire.it/avanguardieeducative/integrazione-mltv>. [Accessed luglio 2024].
- [13] M. Krechevsky and D. Wilson, "Making Learning Visible | Project Zero," 1997. [Online]. Available: <https://pz.harvard.edu/projects/making-learning-visible>. [Accessed luglio 2024].
- [14] C. N. Blundell, M. Mukherjee and S. Nykvist, "A scoping review of the application of the SAMR model in research.," *Computers and Education Open*, vol. 3, 2022.
- [15] K. Børte, K. Nesje and S. Lillejord, "Barriers to Student Active Learning in Higher Education.," *Teaching in Higher Education*, vol. 28, no. 3, pp. 597-615, 2020.
- [16] P. C. Flores, I. S. Rivas and J. G. Torres, "Difficulties in Object-Oriented Design and its relationship with Abstraction: A Systematic Review of Literature," *Proceedings of the 4th European Symposium on Software Engineering*, pp. 1-13, 2023.
- [17] Ministero dell'Istruzione e del Merito, "Linee guida per l'orientamento," 2023. [Online]. Available: https://miur.gov.it/documents/20182/0/linee_guida_orientamento-2-STAMPA.pdf/4c926cff-afaa-8d3f-7176-09b3ec508d64?t=1703239848691. [Accessed luglio 2024].
- [18] A. J. Nalls and G. Wickerd, "The Jigsaw Method: Reviving a Powerful Positive Intervention.," *Journal of Applied School Psychology*, vol. 39, no. 3, pp. 201-217, 2022.
- [19] M. Ranieri, D. Luzzi, S. Cuomo and I. Bruni, "If and how do 360 videos fit into education settings? Results from a scoping review of empirical research.," *Journal of Computer Assisted Learning*, vol. 38, no. 5, pp. 1199-1219, 2022.
- [20] G. Lampropoulos and Kinshuk, "Virtual reality and gamification in education: a systematic review," *Educational technology research and development*, vol. 72, p. 1691-1785, 2024.

Il Coding in classe

Giovanna Caracciolo¹, Giacomo Cassalia²

¹IC Laureana Galatro Feroletto
gcaracciolo026@gmail.com

²IC Laureana Galatro Feroletto
giacomo.cassalia@gmail.com

Abstract

Il presente studio ha l'obiettivo di analizzare l'utilizzo del coding nella pratica didattica e la sua efficacia nello sviluppo del pensiero computazionale e nel potenziamento delle competenze di *problem posing* e *problem solving*. L'attuazione di una didattica *human centered*, ovvero attenta ai bisogni reali degli studenti e ai loro contributi creativi, non può prescindere dall'utilizzo di approcci innovativi, quali ad esempio il *Design Thinking* che, nella sua articolazione, dalla definizione del problema alla ricerca di una possibile soluzione, passando per le fasi di progettazione per prototipi e *testing*, si avvicina notevolmente alla struttura della programmazione, che sta alla base del coding. Pertanto, in questo studio verranno illustrate le attività di coding svolte nelle classi seconde e terze di una scuola secondaria di I grado, dimostrando quanto esse, in un'ottica trasversale, possano risultare funzionali nel processo di apprendimento e, soprattutto, possano essere utilizzate nell'attuazione di una didattica inclusiva.

1 Introduzione: Pensiero computazionale e Coding

Il pensiero computazionale sta alla base del coding, inteso come “attività ludica o didattica basata sull'uso intuitivo dei principi base della programmazione”.¹ Uno dei software più utilizzati per il coding è Scratch, considerato l'erede di Logo, il linguaggio didattico presentato negli anni '60 da Papert. Scratch presenta molti elementi grafici e permette di usare un codice di programmazione a blocchi che risulta facilmente utilizzabile anche da chi non ha conoscenze informatiche specifiche.²

La filosofia dello *sharing* in Scratch rappresenta un grandissimo potenziale, in quanto permette di osservare i progetti altrui e anche di rimaneggiarli, apportando modifiche o integrazioni attraverso la funzione *remix*. Per utilizzare al meglio Scratch nella pratica didattica si può richiedere un Account Docente e gestire le classi con la creazione degli account per gli studenti che potranno, all'interno della classe, condividere i loro progetti e remixare quelli degli altri.

1.1 Debugging e Warm coding

Uno degli aspetti principali del coding è il *debugging*, ovvero la correzione degli errori, che vengono individuati nel momento in cui si prova il programma. Lunghi dall'essere considerato

¹ A. Bogliolo, *Pensiero computazionale e Coding a scuola-1*. Lezione presentata al Master in "DSA (Disturbi Specifici di Apprendimento), BES (Bisogni Educativi Speciali) e Disturbi dello Sviluppo. Psicopedagogia, Metodologie Didattiche, Pensiero Computazionale (coding) e Didattica dell'Inclusione", Anno Accademico 2022-2023, Università degli Studi di Urbino Carlo Bo, Urbino.

² A. Ferraresso, E. Colombini, *Technologica-Coding e Robotica*, Novara, DeA Scuola, 2017, pp. 6-9.

un'esperienza negativa, quasi come un sinonimo di fallimento, l'errore nel coding, proprio perché avviene in una dimensione non giudicante, indica piuttosto una condizione che permette di apportare delle modifiche affinché il programma possa funzionare. L'errore, quindi, non suscita emozioni negative, ma diventa un modo per potenziare il pensiero creativo e computazionale, in quanto spinge a cercare nuove soluzioni, in una dimensione di continuo miglioramento. Il legame tra apprendimento ed emozioni sta alla base della *Warm Cognition*, elaborata dalla Prof.ssa Daniela Lucangeli, che si riferisce alla centralità delle emozioni nel processo di apprendimento e al legame indissolubile che hanno con le cognizioni. Infatti, dato che nel momento in cui si insegna qualcosa si danno informazioni sia di tipo cognitivo che emotivo, lo studente che sperimenta la paura mentre apprende ricorderà il senso di incapacità e inadeguatezza provato in quel momento e tale sentimento si attiverà in situazioni analoghe, portandolo all'evitamento. Al contrario è più facile apprendere se si sperimentano emozioni positive, che riattivano i circuiti cerebrali contribuendo a rafforzare i procedimenti cognitivi.

La *Warm Cognition* ha rappresentato il filo conduttore dell'evento *#Warm Coding*, un webinar in cui il Professore Bogliolo e la Professoressa Lucangeli hanno analizzato il coding e i meccanismi emotivi e cognitivi che stanno alla base del processo di apprendimento, mettendo in luce il grande valore aggiunto che il coding può apportare alla pratica didattica e allo sviluppo dell'intelligenza emotiva.³

2 Il Coding in classe

L'utilizzo trasversale del coding nella pratica didattica rappresenta una modalità efficace di coinvolgimento degli studenti, attratti principalmente dagli aspetti ludici e dalla dimensione laboratoriale delle attività. Essi, inoltre, sono stimolati a lavorare in modo cooperativo, soprattutto nelle fasi di ricerca e di programmazione, per la cui realizzazione sono chiamati a cercare soluzioni diverse ad uno stesso problema. Il docente svolge il ruolo di facilitatore e di co-costruttore nel processo di apprendimento, valutando sia i progetti sia le interazioni tra gli studenti.

2.1 Attività di Coding

Le attività di seguito descritte sono state realizzate all'interno di progetti europei quali *eTwinning*, *EUCodeWeek* ed *Esa-Education Challenge*, ai quali hanno partecipato gli studenti del nostro Istituto. Sin da subito essi si sono sentiti coinvolti nel processo di apprendimento e sono stati stimolati a realizzare dei prodotti creativi e originali, acquisendo, al tempo stesso, specifiche abilità e competenze.

L'attività *Virtual Tour of U.S. cities using Scratch* si è focalizzata su un argomento di civiltà inglese, prevedendo la realizzazione di mappe interattive su alcune città degli Stati Uniti. Seguendo l'articolazione del progetto *Scratch4Disability*⁴ e utilizzando la tecnica dello *scaffolding*, in una prima fase il programma è stato scritto da noi docenti e presentato agli studenti, ai quali è stato poi richiesto, nelle fasi successive, di creare il loro programma e di migliorarlo attraverso il *testing* e il *debugging*, cercando la soluzione più adatta ai problemi riscontrati (*Design Thinking*). Gli studenti sono stati suddivisi in gruppi eterogenei (*Cooperative Learning*), a ciascuno dei quali è stata assegnata una città e, dopo aver reperito materiale fotografico e informazioni storico-geografiche in lingua inglese (attività svolta a casa), hanno lavorato in classe sulla piattaforma Scratch, supervisionati da noi docenti che li abbiamo assistiti sia nel lavoro di sintesi del materiale raccolto

³ D. Lucangeli, A. Bogliolo, Webinar *Warm Coding*, 9 gennaio 2018. Disponibile su: <https://codemooc.org/warmcoding/> [Data di accesso: 23/06/2023].

⁴ A. Barbero, E. Pantò, M. Davi, (Associazione DSchola), "Scratch4Disability: quando il coding è per tutti", *Bricks*, Anno 8, Numero 4, settembre 2018, pp. 10,11. Disponibile su: http://www.rivistabricks.it/wp-content/uploads/2018/09/BRICKS_4_2018.pdf [Data di accesso: 05/06/2023].

che nella scrittura del programma, svolgendo così il ruolo di facilitatori dell'apprendimento. Gli studenti hanno realizzato delle mappe turistiche interattive, procedendo ad animare gli *Sprite*, ad inserire le immagini e i testi in lingua inglese e a cambiare i *backdrop*. L'attività ha coinvolto diverse discipline quali Inglese, Storia, Geografia e Tecnologia e ha permesso agli studenti di acquisire competenze disciplinari e trasversali. La valutazione è stata sia formativa che sommativa, attraverso l'uso di osservazioni sistematiche e rubriche di valutazione. Fondamentale per la valutazione in itinere è stata la fase del *testing*, non solo perché ha permesso agli studenti di prendere coscienza dell'errore e di cercare una soluzione attraverso prove, tentativi ed esperimenti, in un processo in continua ridefinizione, ma soprattutto perché ha trasmesso la percezione dell'errore come un input per apportare modifiche e miglioramenti, in una dimensione non giudicante. In tal modo, lo studente non ha sperimentato un'emozione negativa in seguito all'errore commesso, ma è stato stimolato a cercare nuove soluzioni, accrescendo quindi la propria autostima (*Warm Cognition*). Per questo motivo l'attività è risultata molto utile soprattutto per gli studenti con disabilità e DSA.

Un'attività simile ha riguardato un progetto di apertura al territorio e si è configurata come vero e proprio compito di realtà che ha permesso agli studenti di creare qualcosa di utile per la loro comunità. Gli studenti hanno infatti realizzato delle presentazioni dei monumenti del loro territorio creando delle mappe interattive su Scratch, in cui hanno inserito il loro avatar (creato con Pixton), che ha svolto il ruolo di guida turistica. Per ciascuna presentazione, poi, è stato creato un QR Code per rendere immediato l'accesso alle mappe.



Figura 1: *Virtual Tour of U.S. cities using Scratch*



Figura 2: *Guida ai monumenti del territorio*

Nel corso del progetto eTwinning “*Music Meets Scratch and Code*” è stata proposta agli studenti un'attività di codifica di alcuni spartiti musicali su Scratch, attraverso l'utilizzo dell'estensione musica, con l'obiettivo di realizzare dei programmi che riproducessero gli spartiti musicali. Gli

studenti hanno così avuto modo di utilizzare le loro competenze musicali trasferendole nella dimensione digitale.



Figura 3: Music Meets Scratch and Code

In occasione della *EU Code Week 2022* è stato realizzato un progetto che si è configurato come una vera e propria attività STEAM, ovvero la creazione di poesie animate realizzate nella piattaforma *Code.org*. Ciascun gruppo ha scelto una poesia e poi, attraverso il codice, ha inserito effetti sonori, background ed effetti scenici per raffigurare le rappresentazioni evocate dal testo e per veicolare le emozioni suscitate.



Figura 4: Poem Art

Inoltre, gli studenti del nostro Istituto hanno partecipato alla *Mission Zero 2022/2023* lanciata da *Esa-Education*, per la quale hanno codificato un programma, all'interno della piattaforma *Mission Zero Hub*, volto a costruire un'immagine che gli astronauti hanno poi visualizzato su un computer Astro PI a bordo della Stazione Spaziale Internazionale.



Figura 5: Mission Zero “Flora and Fauna” (2022/2023)

Le attività sperimentate in classe hanno confermato che gli studenti apprendono meglio se sono pienamente coinvolti nel processo di apprendimento e hanno evidenziato la forte spinta motivazionale determinata dall'uso della programmazione a blocchi, che permette agli studenti di riflettere, in un'ottica metacognitiva, sul loro processo di apprendimento. Infatti, così come sosteneva Papert, “insegnando all'elaboratore a pensare, i bambini si lanciano in un'esplorazione del loro stesso modo di pensare”⁵.

⁵ Papert, S., *Mindstorms: children, computers and powerful ideas*, New York, Basic Books Inc., 1980, *op. cit.* in M. Gabbari, R. Gagliardi, A. Gaetano, D. Sacchi, (Equipe Formazione Digitale), “Integrare Coding e Pensiero Computazionale”, *Bricks*, Anno 10, Fuori Numero 4, marzo 2020, p.1. Disponibile su: https://www.rivistabricks.it/wp-content/uploads/2020/03/FN_2020_OPPI_Coding.pdf [Data di accesso: 0/06/2023].

Bibliografia

Bogliolo, A., *Pensiero computazionale e Coding a scuola-1*. Lezione presentata al Master in "DSA (Disturbi Specifici di Apprendimento), BES (Bisogni Educativi Speciali) e Disturbi dello Sviluppo. Psicopedagogia, Metodologie Didattiche, Pensiero Computazionale (coding) e Didattica dell'Inclusione", Anno Accademico 2022-2023, Università degli Studi di Urbino Carlo Bo, Urbino.

Caracciolo, G., *Il coding per una didattica inclusiva* (Tesi finale- Master in "DSA (Disturbi Specifici di Apprendimento), BES (Bisogni Educativi Speciali) e Disturbi dello Sviluppo. Psicopedagogia, Metodologie Didattiche, Pensiero Computazionale (coding) e Didattica dell'Inclusione", Anno Accademico 2022-2023, Università degli Studi di Urbino Carlo Bo, Urbino.

Ferraresso, A., Colombini, E., *Technologica-Coding e Robotica*, Novara, DeA Scuola, 2017.

Sitografia

Barbero, A., Pantò, E., Davi, M., (Associazione DSchola), "Scratch4Disability: quando il coding è per tutti", *Bricks*, Anno 8, Numero 4, settembre 2018, pp. 10,11. Disponibile su: http://www.rivistabricks.it/wp-content/uploads/2018/09/BRICKS_4_2018.pdf

Lucangeli, D., Bogliolo, A., Webinar *Warm Coding*, 9 gennaio 2018.
Disponibile su: <https://codemooc.org/warmcoding/>

Papert, S., *Mindstorms: children, computers and powerful ideas*, New York, Basic Books Inc., 1980, *op. cit.* in M. Gabbari, R. Gagliardi, A. Gaetano, D. Sacchi, (Equipe Formazione Digitale), "Integrare Coding e Pensiero Computazionale", *Bricks*, Anno 10, Fuori Numero 4, marzo 2020.
Disponibile su: https://www.rivistabricks.it/wp-content/uploads/2020/03/FN_2020_OPPI_Coding.pdf

Co(de)spaces: a long voyage. Insegnare il Pensiero Computazionale con la metodologia CLIL

Chiara Storace¹

¹Università di Genova

chiara.storace@edu.unige.it

Abstract

Già nel 1980 Papert sosteneva che il computer, dal punto di vista dell'uso che se ne può fare, dovesse essere considerato come una matita nelle mani dei bambini del futuro. Con il costante sviluppo delle tecnologie e la loro onnipresenza nella nostra vita personale e lavorativa, oggi diventa ancora più importante non solo saper leggere il flusso di immagini e parole che questi strumenti ci propongono, ma anche riuscire ad essere attivi nel loro utilizzo: scrivendo e creando in modo fantasioso, critico e collaborativo (Burke, 2012). Matita, lettura, scrittura, codice. I collegamenti tra il mondo digitale e l'ambito linguistico sono molteplici e hanno fornito lo spunto per il progetto di questo studio: in una classe seconda di scuola secondaria di primo grado si è svolto un corso sulla programmazione a blocchi con la metodologia *Content and Language Integrated Learning* (CLIL). L'idea è stata quella di portare avanti in parallelo il codice digitale insieme a quello linguistico. L'attività interdisciplinare task-based svolta a piccoli gruppi e tramite metodologia CLIL ha avuto come obiettivo la creazione di un tour virtuale interattivo fruibile tramite un visore VR. L'esperienza didattica ha ottenuto risultati positivi sia a livello di introduzione alla programmazione sia in un'ottica più ampia di integrazione di competenze trasversali e di collaborazione tra diverse discipline del curriculum.

1 Introduzione

Molti studi (Israel, 2015) (Lee, 2020) (Weintrop, 2016) (Yadav, 2014) mostrano come si cerchi di inserire l'insegnamento della Computer Science e le competenze legate al Computational Thinking nell'istruzione scolastica. La programmazione informatica è uno dei modi principali in cui si realizza l'alfabetizzazione mediatica offrendo ai discenti la possibilità di cimentarsi in un processo di costruzione della conoscenza in cui la tecnologia è volta a migliorare l'apprendimento co-creativo (Romero M. L., 2016). Programmare vuol dire essere in grado di analizzare una situazione, identificare le sue componenti chiave, modellare dati e processi e usare un approccio di design-thinking agile per perfezionare o creare un programma (Romero M. L., 2017). Sviluppando un progetto di programmazione creativa gli studenti si trovano ad aver bisogno e a utilizzare competenze e conoscenze interdisciplinari: misure, geometria e piano cartesiano per localizzare e muovere i loro personaggi, oggetti e scenari (matematica), organizzazione nel tempo e nello spazio, società e territori (scienze sociali), schemi narrativi e codici diversi (lingue). Proprio dalla presenza di linguaggi differenti e di diverse attività di pensiero nasce l'idea di adoperare la metodologia CLIL. Questa pratica educativa altamente dinamica auspica "una duplice attenzione alla lingua e al contenuto" (Mehisto, 2008) (Coyle, 2010) e rende possibile agli apprendenti

l'esercizio immediato delle abilità linguistiche, nonché lo sviluppo del pensiero critico (Papadopoulos, 2020). Nasce, quindi, il bisogno di connettere le pedagogie delle varie materie con quella dell'insegnamento linguistico. Per approfondire e migliorare l'integrazione tra discipline dobbiamo cercare le convergenze tra gli obiettivi curricolari e capire come l'insegnamento e l'apprendimento in classe possano lavorare per raggiungere questi obiettivi contemporaneamente (Dalton-Puffer, 2013).

2 Progetto

In una classe seconda di scuola secondaria di primo grado si è svolta un'osservazione partecipata di 54 ore lavorando con una docente di lingua straniera, un docente di tecnologia e un docente di geografia. Il gruppo classe formato da 22 studenti (10 femmine e 12 maschi) ha lavorato diviso in piccoli gruppi alla creazione di un tour interattivo sulla piattaforma Cospaces, usando la programmazione a blocchi in modo creativo. Per valutare l'impatto del corso si stanno analizzando sia i tour virtuali prodotti dai ragazzi che un pre e post test sulla loro conoscenza delle espressioni linguistiche legate al coding. L'osservazione partecipata è avvenuta collaborando con i docenti nella fase di ideazione, progettazione ed erogazione delle attività. Partendo dalle linee guida del *Computing at school* anglosassone (Csizmadia, et al., 2015) e volendo mettere in pratica il "*computational doing*", si sono identificati gli obiettivi formativi specifici riferiti a singoli concetti computazionali:

- Pensiero algoritmico
 - Formulare istruzioni per ottenere un effetto desiderato seguendo un ordine particolare (sequenza)
 - Scrivere set di istruzioni che possano essere eseguite contemporaneamente da agenti diversi (*concurrency*)
 - Progettare soluzioni algoritmiche che tengono in considerazione le abilità, limitazioni, desideri delle persone che le useranno
- Scomposizione
 - Dividere un problema in (più) problemi più semplici che possano essere risolte allo stesso modo (*recursive and divide and conquer strategies*)
- Generalizzazione
 - Identificare pattern e somiglianze tra artefatti
 - Adottare soluzioni/parti di soluzioni ad un'intera classe di problemi simili
 - Trasferire idee e soluzioni da un problema ad un altro
- Valutazione
 - Valutare se un artefatto è adatto ad un certo scopo e se fa la cosa giusta
 - Progettare ed eseguire test ed interpretarne i risultati
 - Valutare se la performance dell'artefatto è sufficientemente buona

Le fasi del lavoro sono state guidate dai docenti: durante le ore di Geografia si sono svolte le ricerche a livello di contenuto, durante quelle di Tecnologia ci si è focalizzati sulle diverse funzioni di Cospaces e sul design del tour virtuale, mentre le ore di Inglese sono servite per approfondimenti linguistici e momenti di esposizione, confronto e riflessione tra i gruppi. Se a livello di programmazione a blocchi si è avanzato gradualmente facendo lavorare gli apprendenti inizialmente ad una sola scena, da subito gli studenti si sono trovati a muoversi tra diversi siti e software. Le prime lezioni, infatti, hanno visto i gruppi produrre una scena a 360° su Genova: l'immagine andava trovata e selezionata su Google Maps, scaricata tramite apposito software, salvata e caricata su Cospaces. I discenti hanno, poi, provato ad inserire

item e a creare le prime animazioni: sia tramite comandi semplificati, sia con la messa in sequenza dei primi blocchi di codice.



Figura 1: Primo progetto Cospaces di un'immagine 360°

Il progetto vero e proprio presenta una forte componente task-based, ovvero il dover progettare e produrre un tour interattivo adatto a un visore VR che trasmetta informazioni di carattere geografico su due nazioni europee, ed è stato impostato come una sorta di scoperta autonoma guidata: dai bisogni dei gruppi sono nati momenti di riflessione e tentativi di applicazione. Gli apprendenti hanno co-costruito la loro conoscenza: chi ha esplorato, ad esempio, l'introduzione dei suoni nella scena, ha condiviso con il resto della classe il proprio know-how diventando, al contempo, un "esperto" a cui gli altri compagni hanno potuto rivolgersi per inserire lo stesso elemento anche nel loro progetto. Questo scambio di buone pratiche tra pari ha fatto gradualmente diminuire la frequenza di richieste di supporto tecnico ai docenti. Al termine dell'anno gli apprendenti nei propri progetti hanno trattato: animazione, personalizzazione e movimento (con comprensione degli assi) di item, movimento della camera, uso della funzionalità "path", interazione tra due o più item tramite "attach" o "gruppi", ricerca ed utilizzo di elementi o intere scene create da altri utenti, inserimento di suoni o link esterni, uso dell'opacità per fini estetici o operativi, creazione di quiz, di sequenze e di funzioni con cicli, condizioni e variabili.

Nel mese di giugno, infine, l'istituto ha partecipato ad un evento in cui gli studenti hanno presentato i loro tour alle famiglie e ai visitatori. Avere un riscontro da parte di un pubblico ha motivato ancora di più i ragazzi a completare i loro progetti, cementando l'unione della classe che ha anche creato una locandina e scelto il nome del progetto: "Codespaces: a long voyage".



Figura 2: foto dell'evento di quartiere



Figura 3: locandina del progetto

Le risposte totali raccolte nel Post-test sono state 229. Anche se l'elemento goliardico è rimasto con la presenza di alcune emoji e con uno studente che ha inserito molte volte un "Chiara that you combine" che avrebbe tradotto come "Chiara, cosa combini?", si può notare come gli inserimenti siano locuzioni ed intere frasi. Alcune espressioni sono anche molto specifiche, come, per esempio, <move forward for 19 meters> o <turn in clockwise of 45°>.

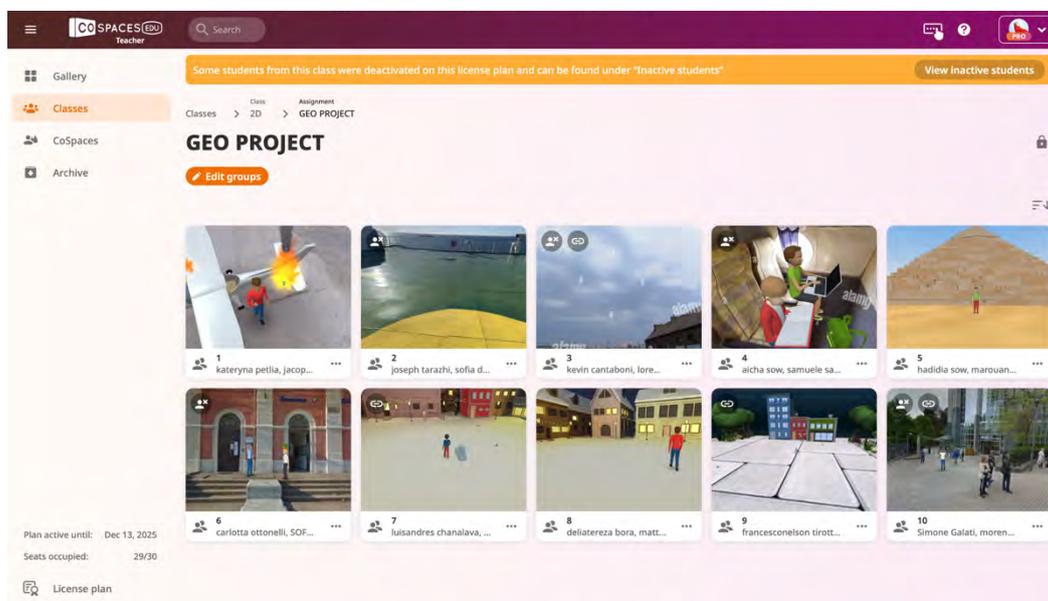


Figura 6: Panoramica progetti Cospaces creati dai gruppi

I tour virtuali interattivi prodotti dagli studenti e il relativo codice sono ancora in fase di analisi; tuttavia, una coppia di apprendenti si è distinta dagli altri portando a termine l'attività con un totale di 26 scene e 530 blocchi utilizzati. Questo team ha creato un videogame completo a livello di contenuto e di "trama" e anche ricco di diversi tipi di attività interattive e di item ed animazioni presenti a schermo. Nella "battaglia finale" la scena ha toccato un picco di 174 blocchi utilizzati. Altri progetti, infine, si sono distinti per originalità, per la presenza di token ed elementi di gamification e per il contenuto linguistico.

4 Conclusioni

L'esperienza didattica portata avanti per questo studio ha ottenuto risultati positivi sia a livello di introduzione alla programmazione sia in un'ottica più ampia di integrazione di competenze trasversali e di collaborazione tra diverse discipline del curriculum. Collegare diverse materie (inglese, geografia e tecnologia) e diverse abilità (linguistiche e di pensiero) permette non solo un interscambio di conoscenze, ma anche la scoperta di prospettive diverse sullo stesso argomento. Il pensiero computazionale affiancato dalla metodologia CLIL sembra poter rappresentare una buona iniziativa per rendere l'informatica più interdisciplinare e integrata nelle materie curriculari a livello di istruzione scolastica.

Bibliografia

- Burke, Q. (2012). The Markings of a New Pencil: Introducing Programming-as-Writing in the Middle School Classroom. *Journal of Media Literacy Education*, 121-135.
- Coyle, D. P. (2010). *CLIL*. Cambridge: Cambridge University Press.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking. A guide for teachers*.
- Dalton-Puffer, C. (2013). A construct of cognitive discourse functions for conceptualising content-language integration in CLIL and multilingual education. *European Journal of Applied Linguistics*, 216-253.
- Israel, M. P. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 263-279.
- Lee, I. G.-S. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, 1-8.
- Mehisto, P. D. (2008). *Uncovering CLIL: Content and Language Integrated Learning in Bilingual and Multilingual Education*. Oxford: Macmillan Education.
- Papadopoulos, S. K. (2020). The Didactic Use of Mantle of the Expert in Teaching CLIL Geography: an Action-Research in Primary Education. *INTERNATIONAL JOURNAL OF HUMANITIES AND EDUCATION (IJHE)*, 17-34.
- Romero, M. L. (2016). The Move is On! From the Passive Multimedia Learner to the Engaged Co-creator. *eLearn*, 1.
- Romero, M. L. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, 1-42.
- Weintrop, D. B. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 127-147.
- Yadav, A. M. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 1-5.

Sara Capecchi è ricercatrice presso il dipartimento di Informatica dell'Università di Torino. È coordinatrice del gruppo di ricerca Didattica dell'Informatica, membro del comitato scientifico del Centro Interdipartimentale per la Ricerca Didattica e l'Aggiornamento degli Insegnanti dell'Università di Torino e del Laboratorio Informatica e Scuola del CINI.

Barbara Catania è professore ordinario presso il DIBRIS, Università di Genova. I suoi interessi di ricerca riguardano la gestione dati e la data education. È coordinatrice del Consiglio dei Corsi di Studio in Informatica presso l'Università di Genova e dell'indirizzo Scienze e Tecnologie per la Società dell'Informazione di IANUA - Scuola Superiore dell'Università di Genova.

Il volume raccoglie gli atti del secondo convegno italiano di didattica dell'informatica (ITADINFO) tenutosi a Genova dal 18 al 20 settembre 2024. I contributi inclusi nel volume includono racconti di esperienze sul campo realizzate da insegnanti, presentazione di risultati di ricerca e di proposte di formazione laboratoriale interattiva.

ISBN: 978-88-3618-287-9