

# Pensiero Computazionale: imparare facendo nella Scuola Primaria

Maurizio Vincini<sup>1</sup>, Mariachiara Neri<sup>2</sup> and Alessandra Zoboli<sup>2</sup>

<sup>1</sup> Dipartimento di Ingegneria “E. Ferrari”, Università di Modena e Reggio Emilia, Italia  
maurizio.vincini@unimore.it

<sup>2</sup> Istituto Comprensivo di Nonantola “F.lli Cervi”, Modena, Italia

**Abstract.** Negli ultimi anni il Pensiero Computazionale e il Coding sono divenuti la colonna portante nei progetti innovativi e tecnologici riguardanti la scuola primaria. Da queste proposte è nato un progetto congiunto tra i ricercatori del Dipartimento di Ingegneria “E. Ferrari” di Modena ed un gruppo di insegnanti dell’Istituto Comprensivo F.lli Cervi di Nonantola (Modena) per permettere a ragazzi dell’ultimo anno della scuola primaria di capire e sperimentare le proprie capacità di logica e problem solving usando l’ambiente di Scratch e MIT App Inventor 2, per costruire la propria soluzione a problemi matematici, di logica e di sequenza narrativa e ‘portandola a casa’. L’articolo presenta i progetti realizzati dagli alunni e la loro valutazione secondo il modello di Funke.

**Keywords:** Pensiero Computazionale, Scratch, costruzionismo.

## 1 Introduzione

Negli ultimi anni il Pensiero Computazionale e il Coding sono diventati la colonna portante nei progetti innovativi e tecnologici riguardanti la scuola: numerose ed efficaci iniziative sono state attivate in ambito istituzionale (introdotte con la legge 107/2015 art 1 comma 7) ed altre sono nate in modo spontaneo. Nella prima categoria si pensi al progetto Programma il Futuro, nato dal MIUR in collaborazione con il CINI, nell’altra il movimento CoderDojo (<http://www.coderdojoitalia.org>) che è ormai attivo con i propri club in tantissime città italiane. Queste iniziative fanno riferimento ad altre più ampie ed internazionali quali ad esempio Code.org, che fornisce percorsi dedicati a insegnanti e studenti e con un approccio orientato al problem solving, e Scratch, il famoso e diffusissimo linguaggio a blocchi creato al MIT Media Lab da Mitchel Resnick, sulle idee di Papert, per “promuovere la spirale dell’apprendimento creativo”. Seguendo queste idee, è nato un progetto congiunto tra i ricercatori del Dipartimento di Ingegneria “E. Ferrari” dell’Università di Modena e Reggio Emilia ed un gruppo di insegnanti dell’Istituto Comprensivo F.lli Cervi di Nonantola (Modena), per permettere a ragazzi dell’ultimo anno della scuola primaria, che non si erano mai accostati ad un linguaggio di programmazione informatica, di capire e sperimentare le proprie capacità di logica e problem solving. Le attività hanno riguardato la risoluzione di

problemi matematici, di logica e la creazione di una sequenza narrativa e si sono svolte usando l'ambiente di Scratch e MIT App Inventor 2, per permettere a ciascuno di trovare la propria soluzione, 'portandola a casa'. L'esperienza, durata circa sei mesi, ha permesso agli alunni di acquisire numerose competenze, misurate attraverso l'adozione di un modello di valutazione proposto in ambito di ricerca.

## 2 Il progetto

Il Costruzionismo è una teoria educativa sviluppata da Seymour Papert, basata sulla teoria del Costruttivismo elaborata da Piaget, col quale Papert ha lavorato tra la fine degli anni 50 e i primi anni 60. Partendo da Piaget, Papert fonde il Costruttivismo con le nuove tecnologie e con gli strumenti informatici, dando vita al Costruzionismo: il bambino impara facendo, attraverso gli artefatti. Il computer e i linguaggi di programmazione possono permettere al bambino di creare, su base esperienziale, ambienti e situazioni di proprio interesse e quindi molto più incisivi nel proprio percorso di crescita formativa. Si passa cioè dall'essere "consumatori" di informazioni a "produttori" di conoscenza (i cosiddetti prosumer). Mitchel Resnick, a partire dagli anni 80, all'interno del MIT Lifelong ha realizzato queste idee costruendo Scratch, il diffusissimo linguaggio a blocchi che si è dimostrato uno strumento particolarmente adatto ad esperienze di tipo costruzionista.

Il nostro progetto ha voluto percorrere la strada aperta da Resnick, lavorando per "classi aperte e gruppi di livello", in modo da permettere una didattica individualizzata e personalizzata. Infatti l'organizzazione di gruppi per livelli di competenze può rispondere in modo più adeguato e mirato alle richieste dei diversi stili cognitivi. Inoltre, l'incontro di alunni provenienti da classi diverse ha favorito occasioni di socializzazione ed integrazione tra coetanei, permettendo anche il confronto e l'interazione tra percorsi didattici in parte differenti.

Gli alunni coinvolti non avevano alcuna esperienza in ambito di coding, pertanto il primo obiettivo ha riguardato lo sviluppo dell'idea di pensiero computazionale per la risoluzione di problemi affrontati nelle varie discipline. L'ambito interessato è stato ampio, spaziando dai quiz logico-matematici delle prove INVALSI alla rappresentazione di un testo narrativo elaborato per piccoli gruppi, ideati e realizzati sinergicamente fra insegnanti e ricercatori. Il percorso si è sviluppato in piccoli progetti legati alle attività e ai contenuti che nel periodo in questione gli alunni avevano affrontato nelle materie curriculari.

L'obiettivo è stato perseguito cercando di offrire agli alunni l'astrazione di algoritmo, quale sequenza di azioni, attraverso lezioni "unplugged", concrete, in cui essi hanno costruito un bracciale/portachiavi definendo ed eseguendo brevi comandi, per poi passare all'astrazione delle variabili, espressioni condizionali e iterazioni per giungere al coding.

Il secondo obiettivo ha riguardato l'introduzione del linguaggio di programmazione ad istruzioni quale strumento di creazione dell'algoritmo, portando quindi gli alunni al concetto di coding: il nuovo linguaggio è diventato per loro lo strumento operativo per risolvere problemi concreti, costruire artefatti, comunicare ed esprimersi. Seguendo

le proposte del learn to code, l'apprendimento è stato di tipo incrementale attraverso la risoluzione di problemi via via più complessi, introducendo di volta in volta gli strumenti di programmazione adeguati alla soluzione, senza fornire basi teoriche e formali. Come detto, è stata scelta la piattaforma del MIT Scratch, un linguaggio a blocchi visuale disponibile sotto forma di web-app e pertanto usata in laboratorio, sia su PC che su tablet, sia a casa dagli alunni per completare i problemi o affrontarne spontaneamente dei nuovi.

### 3 Il modello di valutazione

I progetti degli alunni, alcuni realizzati a gruppi di 3-4, altri singolarmente, sono stati analizzati dal punto di vista strutturale e della qualità ottenuta, usando il sistema di categorizzazione presentato da Funke [1], [2]. Il modello è progettato per misurare le abilità del pensiero computazionale attuato attraverso il coding, individuando la presenza di 24 elementi di programmazione, suddivisi in quattro categorie principali:

1. Requisiti: diverse sprite, movimento di sprite, iterazione, dichiarazione condizionale
2. Concetti di programmazione: sequenza, variabili, liste, gestione degli eventi, discussioni, condizione e sincronizzazione, input da tastiera, logica booleana, iterazione
3. Organizzazione del codice: blocchi personalizzati, nomi sprite, nomi di variabili
4. Operatività funzionalità, personalizzazione sprite, personalizzazione dello stage, interattività, usabilità, tipo di progetto

L'uso del modello ha permesso di schematizzare e valutare i vari progetti degli alunni, misurandone i progressi e le competenze acquisite al termine del percorso.

In più, il *level of understanding* di ciascun progetto è stato calcolato usando la SOLO taxonomy [3]: prestructural (L1), unistructural (L2), multistructural (L3), relational (L4), extended abstract (L5).

### 4 Risultati di progetto

Nel percorso logico-matematico gli alunni hanno dapprima affrontato e risolto un problema contenente 4-6 variabili e risolvibile con un'espressione di 5-6 operazioni annidate. Si è poi passati a risolvere un quesito tratto dalle prove INVALSI che genera un albero di decisione binario a tre livelli e, infine, il disegno e il calcolo del perimetro di un qualunque poligono regolare. Per l'area linguistica gli alunni, suddivisi in gruppi, hanno ideato e realizzato una sequenza narrativa con almeno 4 personaggi, che interagiscono tra loro, si muovono, parlano, con una colonna sonora che collega le varie scene. Per ciascuno di questi progetti sono state valutate le quattro categorie inerenti la presenza di elementi di programmazione.

1. Requisiti: tutti i progetti contenevano i quattro elementi minimi sulle sprite indicate dal modello.
2. Concetti di programmazione: tutti i progetti hanno usato sequenze, variabili, eventi (almeno 5 ciascuno), condizioni, input da tastiera. Inoltre due progetti hanno utilizzato sia la logica Booleana che le liste.

3. Organizzazione del codice: in tutti i progetti le sprite e le variabili sono state rinominate usando nomi significativi. Nessun progetto ha usato i blocchi personalizzati, non essendo stati precedentemente presentati agli allievi.

4. Operatività: quasi tutti i progetti, eventualmente conclusi a casa dai singoli alunni, sono completamente funzionanti e contengono input da parte dell'utente. Gli stage e gli sprite sono stati personalizzati per garantire interattività e usabilità.

Valutando il level of understanding in accordo alla SOLO taxonomy, tutti i progetti possono essere collocati almeno a livello L3 (Multistructural). Il 25% dei progetti a livello quattro L4 (Relational), ed il 10% al livello massimo, L5 (Extended Abstract), dimostrando quindi il buon livello generale raggiunto.

Sulla scorta di queste buone valutazioni si è svolto un test finale individuale, con l'obiettivo di verificare il punto 2 del modello (Concetti di programmazione), dal quale è emerso che oltre l'80% degli alunni ha acquisito le competenze riguardo gli strumenti di coding definiti come obiettivi di progetto.

## 5 Conclusioni

L'articolo presenta il progetto congiunto tra i ricercatori del Dipartimento di Ingegneria "E. Ferrari" di Modena ed un gruppo di insegnanti dell'Istituto Comprensivo di Nonantola (Modena), svoltosi da ottobre 2018 a marzo 2019 in due classi quinte della scuola primaria, per permettere di capire e sperimentare le proprie capacità di logica e problem solving. I risultati sono stati valutati positivamente sia secondo la metodologia Funke che la SOLO taxonomy.

Il buon livello di "programmatori" raggiunto dagli alunni partiti 'digiuni' di coding è dimostrato dalla scelta di partecipare al concorso "Programma una storia", bandito il 6 marzo 2019 con una Circolare del MIUR nell'ambito dei Programma il Futuro, nonostante le regole del concorso, imponendo di utilizzare il linguaggio a blocchi della piattaforma Code.org, diverso da Scratch. Il cambio di piattaforma e linguaggio ha permesso di verificare le capacità acquisite dagli alunni in ambito di programmazione, evidenziate dalla semplicità con cui gli alunni hanno si sono adeguati al nuovo ambiente.

## Bibliografia

1. A. Funke, K. Geldreich, and P. Hubwieser, "Analysis of scratch projects of an introductory programming course for primary school students," in *2017 IEEE Global Engineering Education Conference, EDUCON 2017, Athens, Greece, April 25-28, 2017*. IEEE, 2017, pp. 1229–1236.
2. A. Wilson, T. Hainey, and T. M. Connolly, "Using scratch with primary school children: An evaluation of games constructed to gauge understanding of programming concepts," *IJGBL*, vol. 3, no. 1, pp. 93–109, 2013.
3. L. M. Seiter, "Using SOLO to classify the programming responses of primary grade students," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education, Kansas City, MO, USA, March 4-7, 2015*, A. Decker, K. Eiselt, C. Alphonse, and J. Tims, Eds. ACM, 2015, pp. 540–545.