

Coding con Scratch nella scuola primaria per introdurre il concetto di angolo

Paolo Teani^{1,2*} Marco Lazzari² Chiara Giberti^{2,3}

¹ Istituto Comprensivo “Giosuè Carducci”, Dalmine (BG)

² Università di Bergamo

³ Università di Modena e Reggio Emilia

Abstract

In questo lavoro affrontiamo il concetto di pensiero computazionale, tratteggiandone le basi teoriche e le implicazioni didattiche. L'obiettivo principale è evidenziare come la programmazione e il coding possano trasformare la didattica tradizionale in un'esperienza attiva e laboratoriale, che favorisce nei bambini lo sviluppo di competenze disciplinari, trasversali e cognitive. Riportiamo dunque un'esperienza didattica svolta con una classe quinta della scuola primaria, intesa a collocare l'insegnamento della geometria in un contesto più motivante e accessibile per gli studenti, per favorire un apprendimento attivo e interattivo, e sviluppare al contempo il pensiero computazionale. Tramite l'uso del linguaggio Scratch si è mirato a lavorare con gli alunni sulla costruzione del significato degli oggetti geometrici, con un focus particolare sul concetto di angolo

1 Il pensiero computazionale

Il pensiero computazionale è emerso come area di interesse accademico nei primi anni 2000, grazie al contributo di Jeannette Wing (2006), che lo ha definito come un insieme di abilità, applicabili in tutti i campi del sapere, che tutti gli studenti dovrebbero sviluppare nel percorso educativo. Precursore di questo approccio è stato Seymour Papert che, sulla scorta della sua collaborazione con Jean Piaget, ha sviluppato la teoria del costruzionismo (Papert, 1986): secondo Papert, l'apprendimento è più efficace quando lo studente è coinvolto nella costruzione di artefatti concreti, che favoriscono l'interiorizzazione delle conoscenze (Papert, 1980).

Papert ha reso operativo questo principio con la creazione del linguaggio di programmazione LOGO, destinato a insegnare ai bambini la geometria tramite la programmazione dei movimenti nel piano di una piccola tartaruga che si sposta sul pavimento o, nella versione grafica, sullo schermo di un computer. L'approccio di Papert trova eco in progetti e strumenti più recenti come il linguaggio Scratch, volto a rendere la programmazione accessibile, intuitiva e coinvolgente per studenti di ogni livello,

* L'esperienza didattica qui riportata è stata sviluppata per un progetto di tesi di laurea in Scienze della formazione primaria da PT, con MT come relatore e CG come correlatrice. La redazione del contributo è stata condivisa dai tre autori.

ingaggiati nello sviluppo di progetti personalmente significativi come storie animate e giochi (Maloney, 2010, p. 1). Mitchel Resnick e il suo gruppo di ricerca hanno ulteriormente sviluppato il quadro teorico, introducendo l'idea di "creative thinking spiral", un modello di apprendimento iterativo che prevede fasi di immaginazione, creazione, sperimentazione, condivisione e riflessione (Resnick, 2008). Questo processo stimola il pensiero creativo e critico, un aspetto chiave nell'era digitale (Thornhill-Miller et al., 2023). L'interfaccia grafica di Scratch consente agli utenti di creare programmi mediante l'assemblaggio di blocchi colorati, che rappresentano comandi e funzioni logiche, senza doversi cimentare con una sintassi complessa. Si tratta di un ambiente dichiaratamente concepito per permettere agli studenti di "imparare a programmare" e "programmare per imparare" (Resnick, 2013): si possono creare storie interattive e giochi, e condividere i propri progetti con gli altri utenti, favorendo la collaborazione e il feedback, promuovendo un approccio attivo alla conoscenza, sviluppando competenze computazionali, disciplinari e trasversali, riducendo l'ansia da apprendimento (Wu, Chang & He, 2010).

A partire da queste esperienze, negli ultimi anni si è sviluppato un dibattito sull'opportunità di introdurre il pensiero computazionale nelle scuole, intendendolo come un'abilità fondamentale per tutti, alla stregua di lettura, scrittura e matematica. Il pensiero computazionale implica la capacità di applicare strategie di astrazione, scomporre problemi complessi in sottoproblemi più gestibili, e identificare soluzioni algoritmiche, e viene stimolato tramite attività per le quali i discenti possono trovare un senso e una rilevanza pratica (Gutiérrez, 2014, p. 20) in termini di generalizzabilità e di trasferibilità nelle esperienze del quotidiano (Lazzari, 2017, p. 30).

Il sistema scolastico italiano ha riconosciuto l'importanza del pensiero computazionale già a partire dalle "Indicazioni Nazionali" del 2012, che enfatizzano il ruolo delle competenze digitali nel percorso educativo. La legge 107/2015 ne ha ribadito la necessità, includendo lo sviluppo del pensiero computazionale tra gli obiettivi formativi prioritari. Le "Indicazioni nazionali e nuovi scenari" del 2018 hanno ancor più rafforzato questo proposito, sottolineando il valore del coding come strumento educativo trasversale. In questa temperie, l'uso di Scratch e di altre piattaforme di programmazione assume un ruolo chiave nel favorire una didattica innovativa, basata su creatività e problem-solving.

2 Un percorso analogico-digitale per l'apprendimento del concetto di angolo

L'integrazione del pensiero computazionale nella didattica della geometria è un'opportunità per sviluppare abilità logiche e di problem-solving. Questo studio presenta un percorso sperimentale condotto in una classe quinta primaria, composta da 17 alunni (11 femmine e 6 maschi), nato dall'esigenza di offrire agli studenti un'esperienza di apprendimento di concetti geometrici fondamentali che andasse oltre i tradizionali approcci astratti alla matematica e che favorisse, invece, un percorso di costruzione attiva della conoscenza.

Il nostro lavoro si è fondato su tre obiettivi principali. In primo luogo, abbiamo mirato a sviluppare un'attività che puntasse alla zona di sviluppo prossimale degli studenti (Vygotskij, 1990), ovvero un contesto in cui l'insegnamento accompagnasse e stimolasse la crescita e il processo di devoluzione degli studenti (Brousseau, 1998), piuttosto che limitarsi alla trasmissione di conoscenze. In secondo luogo, abbiamo voluto promuovere il pensiero computazionale come strumento per il problem solving, evidenziando l'importanza dell'errore come parte integrante del processo di apprendimento e valorizzandolo come forza generativa (Castelli Gattinara, 2017), piuttosto che come ostacolo alla buona riuscita del progetto. Infine, abbiamo utilizzato Scratch per favorire un coinvolgimento attivo degli studenti e osservare il suo impatto sull'acquisizione delle conoscenze geometriche.

La scelta di integrare il coding nella didattica della geometria non è casuale. La geometria, sebbene sia uno dei quattro ambiti delle Indicazioni Nazionali, nella pratica didattica nel primo ciclo di istruzione

è spesso relegata ai margini, insegnata in modo astratto e distante dall’esperienza concreta degli alunni. Scratch, con la sua interfaccia visiva e il concetto di “geometria della tartaruga” (Solomon & Papert, 1976), permette di riportare la geometria al centro del percorso di apprendimento, facendo leva sulla motivazione e sulla sperimentazione diretta.

Nello specifico, il nostro progetto ha coinvolto gli studenti in diverse attività di geometria, dall’orientamento spaziale alla rappresentazione di figure geometriche, passando per il concetto di angolo e il suo utilizzo nella programmazione. I bambini hanno avuto l’opportunità di costruire giochi in cui gli *sprite* di Scratch si muovevano seguendo istruzioni basate su principi geometrici, nonché di progettare algoritmi che consentissero agli *sprite* di disegnare poligoni regolari.

Un aspetto particolarmente rilevante del progetto è l’attenzione dedicata al concetto di angolo, che sovente chi pratica il coding dà per scontato, in quanto la rotazione sul posto è una primitiva grafica del linguaggio, così com’era per LOGO. Scratch si è rivelato un valido strumento per favorire una visione dinamica del concetto di angolo, considerato come rotazione di una semiretta, che permette di superare importanti difficoltà legate a questo concetto (Sbaragli & Santi, 2012). La classe a cui è stato proposto il progetto aveva affrontato il concetto di angolo nell’anno scolastico precedente e, attraverso un test preliminare, è emerso che la definizione in possesso degli alunni è riconducibile al cosiddetto “modello statico”, secondo cui l’angolo è una parte di piano compresa tra due semirette che hanno la stessa origine. Questo approccio formale produce l’acquisizione di un’idea di angolo piuttosto oscura e distorta, spesso causa di una serie di misconcezioni (Malara, 1996). Per tentare di superare il problema abbiamo fornito una varietà di mezzi semiotici di oggettivizzazione (Sbaragli & Santi, 2012); in particolare, ispirandoci agli studi di Mitchelmore e White (1998, 2000) sull’apprendimento progressivo del concetto di angolo, abbiamo sviluppato un percorso parallelo che ha incluso esperienze fisiche, come l’uso di una pedana rotante. Attraverso questa esperienza concreta, centrata sull’osservazione di corpi in movimento rotatorio e sulla loro rappresentazione grafica, i bambini hanno potuto esplorare la misura dell’angolo come rotazione, per poi trasferire in maniera consapevole nell’ambiente di programmazione visuale la conoscenza così acquisita.

Un elemento chiave del nostro approccio è la valorizzazione della collaborazione e del confronto tra pari, elementi che non solo facilitano l’apprendimento, ma lo rendono più significativo (Johnson & Johnson, 1991): seguendo il principio di Papert “chiedi a tre, poi a me”, gli alunni sono stati incoraggiati a risolvere i problemi discutendo con i compagni prima di chiedere aiuto all’insegnante. Questo approccio ha stimolato più autonomia nel problem solving e ha favorito anche le positive ricadute del lavoro cooperativo in termini di interdipendenza positiva, interazione promozionale faccia a faccia e sviluppo di abilità sociali.

L’integrazione tra geometria e programmazione ha portato a risultati interessanti: sulla base di osservazioni e test svolti in classe e del confronto con l’insegnante, è emerso che i bambini hanno mostrato sincero interesse per la materia, acquisito una comprensione profonda dei concetti geometrici e sviluppato capacità di astrazione e modellizzazione.

Inoltre, l’uso della programmazione ha permesso loro di verificare le proprie ipotesi in tempo reale, passo dopo passo, e di correggere eventuali errori, montessorianamente intesi come guida; in questo modo si è incentivato un apprendimento basato sulla sperimentazione, nel quale l’esperienza concreta sostiene la formulazione di concetti astratti (Kolb, 1984; Uyen, Tong & Lien, 2022).

3 Conclusioni e futuri sviluppi

Il nostro progetto evidenzia come un approccio che integri tecnologia e didattica disciplinare possa costituire uno strumento efficace per migliorare l’apprendimento della geometria. L’uso di Scratch ha reso possibile un’esperienza coinvolgente, concreta e accessibile, consentendo agli studenti di esplorare concetti matematici attraverso un processo attivo e partecipativo.

Questa esperienza ci suggerisce che l’insegnamento della matematica può trarre grande beneficio dall’uso di strumenti digitali e che il coding, oltre a favorire lo sviluppo di competenze computazionali, può rappresentare un ponte per una comprensione più profonda della geometria.

Nell’immediato futuro prevediamo di ripetere l’esperienza con un campione più ampio di allievi e di individuare inoltre nuovi temi di geometria da affrontare con il medesimo approccio.

Bibliografia

- Brousseau, G. (1998). *Didattiche disciplinari: Aspetti teorici e metodologici*. Pitagora Editrice.
- Castelli Gattinara, E. (2017). Epistemologia dell’errore in Federigo Enriques e Gaston Bachelard. *Matematica, Cultura e Società*, 2(2), 195-208.
- Gutiérrez, K. D. (2014). Relevance to practice as a criterion for rigor. *Educational researcher*, 19-23.
- Johnson, D., & Johnson, R. (1991). *Learning mathematics and cooperative learning : lesson plans for teachers*. Interaction Book Company.
- Kolb, D. (1984). *Experiential learning: Experience as the source of learning and development*. Prentice Hall.
- Lazzari, M. (2017). *Istituzioni di tecnologia didattica*. Studium.
- Malara, N. (1996). L’insegnamento della geometria nella scuola media. Questioni teoriche e didattico-metodologiche. In *L’insegnamento della geometria: seminario di formazione per docenti della istruzione secondaria di primo grado* (p. 13-76). Ministero della Pubblica Istruzione.
- Maloney, J. R. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 1-15.
- Mitchelmore, M., & White, P. (1998). Development of angle concepts: A framework for research. *Mathematics Education Research Journal*, 10(3), 4-27.
- Mitchelmore, M., & White, P. (2000). Development of angle concepts by progressive abstraction and generalisation. *Educational Studies in Mathematics*, 41, 209-238.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Papert, S. (1986). *Constructionism: A new opportunity for elementary science education*. Massachusetts Institute of Technology.
- Resnick, M. (2008). Sowing the seeds for a more creative society. *Learning & Leading with Technology*, 35(4), 18-22.
- Resnick, M. (2013). Learn to code, code to learn. *EdSurge*, 8, 54.
- Sbaragli, S., & Santi, G. (2012). Le scelte dell’insegnante relative al concetto di angolo. *Bollettino dei docenti di matematica*, 65, 35-55.
- Solomon, C., & Papert, S. (1976). A case study of a young child doing Turtle Graphics in LOGO. *Proceedings of the AFIPS 1976 National Computer Conference* (p. 1049-1056). American Federation of Information Processing Societies.
- Thornhill-Miller, B., Camarda, A., Mercier, M., Burkhardt, J., Morisseau, T., Bourgeois-Bougrine, S., & et al. (2023). Creativity, critical thinking, communication, and collaboration: assessment, certification, and promotion of 21st century skills for the future of work and education. *Journal of Intelligence*, 11(3), 54:1-54:32.
- Uyen, B., Tong, D., & Lien, N. (2022). The effectiveness of experiential learning in teaching arithmetic and geometry in sixth grade. *Frontiers in Education*, 7(858631), 1-13.
- Vygotskij, L. (1990). *Pensiero e linguaggio*. Laterza.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wu, W. Y., Chang, C. K., & He, Y. Y. (2010). Using Scratch as game-based learning tool to reduce learning anxiety in programming course. *Global Conference on Learning and Technology* (p. 1845-1852). Association for the Advancement of Computing in Education.