

BRICKS | TEMA

Dal coding unplugged al coding creativo: percorsi H-STEAM nella scuola primaria

a cura di:

Roberto Gigliotti, Elisa Garieri



Coding, motivazione, coinvolgimento, DidatticaAttiva

1. Introduzione: il coding come linguaggio espressivo

Cosa accade quando il coding smette di essere un esercizio tecnico e diventa uno spazio di creazione? Questa domanda ha guidato la progettazione di un percorso didattico che ha coinvolto tutte le classi di una scuola primaria.

L'esperienza è stata condotta cercando di proporre un approccio progressivo e differenziato per fascia d'età. Gli artefatti digitali prodotti dai bambini sono diventati oggetti culturali autentici: animazioni, storie, trasformazioni, cicli naturali tradotti in codice, immagini e suoni. Farfalle che si trasformano, stagioni che si avvicinano, labirinti da esplorare, personaggi che dialogano.

L'orizzonte pedagogico è stato quello dell'approccio H-STEAM (Humanities, Science, Technology, Engineering, Art, Mathematics), in cui i linguaggi umanistici e artistici dialogano con quelli scientifico-tecnologici per promuovere creatività, espressione e pensiero critico (Resnick, 2017; Kafai & Burke, 2015), dove il coding non è stato un fine, ma un mezzo: un mezzo per pensare, per raccontare, per creare.

2. Gli strumenti: CodeJr.org e CodeKidi

La scelta degli ambienti di programmazione non è stata casuale. CodeJr.org e CodeKidi sono stati selezionati per la loro complementarità: il primo offre un contesto più strutturato e guidato, ideale per consolidare i concetti fondamentali del pensiero computazionale; il secondo apre uno spazio di creazione libera, in cui il codice si intreccia ancor di più con immagini, suoni e narrazione.

3. Il percorso classe per classe: progressione e differenziazione

Ogni classe ha lavorato su attività pensate per il proprio livello di sviluppo cognitivo e di esperienza digitale, mantenendo però un filo conduttore comune: la produzione di artefatti digitali espressivi attraverso il coding. Di seguito vengono indicati per ciascuna classe gli strumenti utilizzati, degli esempi rappresentativi delle attività svolte e le principali competenze sviluppate.

Classe Prima

Strumenti

- Reticolo a pavimento (nastro o tappeto a griglia)
- Carte comando fisiche (partenza, arrivo, ostacoli, azioni)
- CodeJr.org

Esempi di attività e artefatti realizzati

- Coding unplugged: costruzione e risoluzione di labirinti fisici sul reticolo
- Torneo a squadre: una squadra progetta il labirinto, l'altra lo risolve
- CodeJr.org: prime sequenze con sprite su percorsi semplici

Competenze sviluppate

- Comprensione corporea di sequenza, istruzione e algoritmo
- Prime strategie di ottimizzazione (percorso minimo)
- Cooperative learning e verbalizzazione delle strategie
- Trasposizione dall'esperienza fisica al codice digitale

Classe Seconda*Strumenti*

- CodeJr.org
- CodeKidi

Esempi di attività e artefatti realizzati

- CodeJr.org: percorsi con ostacoli e labirinti digitali di complessità crescente
- CodeJr.org: riconoscimento di pattern e percorsi a serpentina
- CodeKidi: storie animate con due personaggi e breve dialogo
- CodeKidi: animazioni con cambio di sfondo e sequenza di scene

Competenze sviluppate

- Consolidamento del concetto di sequenza e debugging
- Riconoscimento di schemi ripetitivi
- Prime forme di sincronizzazione tra personaggi
- Autonomia operativa nell'ambiente digitale

Classe Terza*Strumenti*

- CodeJr.org
- CodeKidi

Esempi di attività e artefatti realizzati

- CodeJr.org: stage complessi con ottimizzazione del percorso e debugging autonomo
- CodeJr.org: progettazione di percorsi originali da parte degli studenti
- CodeKidi: storie con più personaggi, dialoghi e sincronizzazione di eventi
- CodeKidi: animazioni con effetti sonori e musica integrata

Competenze sviluppate

- Ottimizzazione e pensiero algoritmico avanzato
- Gestione della sincronizzazione tra più sprite
- Integrazione di codice, suono e narrazione
- Creatività progettuale e agency nell'uso degli strumenti

Classe Quarta*Strumenti*

- CodeJr.org
- CodeKidi

Esempi di attività e artefatti realizzati

- CodeKidi: esempio rappresentativo, il ciclo delle stagioni con navigazione tra scene
- CodeKidi: storie animate con più scenari e transizioni tra scene
- CodeKidi: progetti con interazione tra personaggi e ambienti diversi

Competenze sviluppate

- Struttura multi-scena e gestione delle transizioni
- Connessioni interdisciplinari (scienze, arte, narrazione)
- Progettazione di artefatti complessi e coerenti
- Co-creazione e lavoro collaborativo

Classe Quinta*Strumenti*

- Codejr.org / CodeKidi

Esempi di attività e artefatti realizzati

- CodeKidi: percorsi liberi ideati e progettati autonomamente dagli studenti
- CodeKidi: animazioni interattive con risposta agli input dell'utente
- CodeJr.org: esempio rappresentativo, la metamorfosi del bruco in farfalla in 4 scenari (uovo, bruco, crisalide, farfalla)
- CodeKidi: esempio rappresentativo, il sistema Terra-Luna: animazione del giorno e della notte e delle fasi lunari
- CodeKidi: progetti multimediali con integrazione di risorse autoprodotte (disegni, suoni, voci)

Competenze sviluppate

- Progettazione autonoma e pensiero computazionale maturo
- Interattività e gestione degli input dell'utente
- Integrazione di risorse autoprodotte nell'artefatto digitale
- Riflessione metacognitiva sul processo creativo

4. Gli artefatti come opere: esempi in primo piano

Tra i molti artefatti prodotti durante il percorso, tre esempi emergono con particolare forza espressiva e didattica, ciascuno rappresentativo di un diverso modo in cui il coding si intreccia con arte, scienze e narrazione, di seguito le progettazioni dei tre esempi indicati.

4.1 Il labirinto: dal pavimento allo schermo

Nella sua versione unplugged, realizzata con nastro sul pavimento dell'aula, il labirinto ha permesso ai bambini più piccoli di sperimentare con il corpo i concetti di sequenza e algoritmo: una squadra progettava il percorso con carte fisiche, l'altra doveva risolverlo con le carte azione.

Questa dinamica di gioco, in cui chi progetta e chi risolve si alternano e si sfidano, ha generato una tensione cognitiva produttiva: per costruire un buon labirinto devi capire come pensano gli altri, e per risolverlo devi scomporre il problema in passi elementari. Due processi fondamentali del pensiero computazionale, vissuti come un gioco.

Figura 1 - Coding unplugged in classe prima: costruzione e risoluzione di labirinti fisici sul reticolo



Successivamente, il labirinto è migrato su CodeJr.org: lo stesso problema, trovare un percorso efficiente tra ostacoli, ma affrontato con strumenti simbolici digitali. Lo sprite sullo schermo è diventato il prolungamento virtuale del bambino sul reticolo. Il debugging, correggere gli errori nella sequenza, ha assunto un significato concreto e immediatamente verificabile.

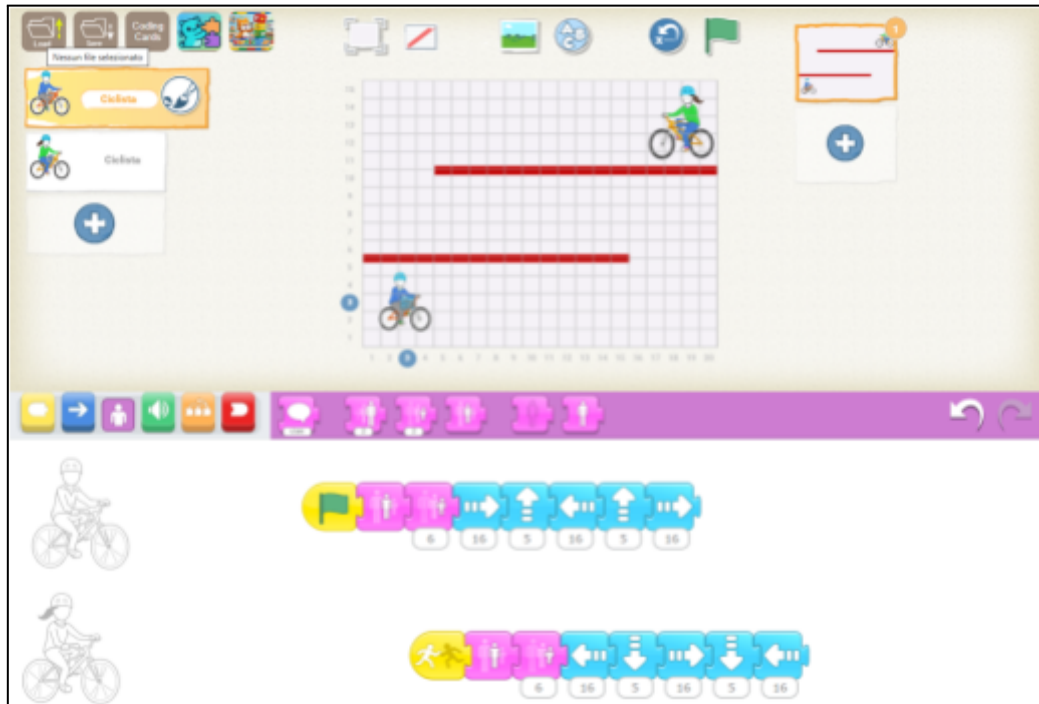


Figura 2 - Il labirinto digitale su CodeJr.org: dallo spazio fisico allo spazio virtuale

4.2 Il ciclo delle stagioni: interattività e tempo naturale

Nelle classi quarte, uno degli artefatti più apprezzati è stato il ciclo delle stagioni interattivo: un'animazione in cui le quattro stagioni si susseguono, ciascuna con le proprie caratteristiche visive, sonore e narrative, e in cui l'utente può navigare liberamente da una stagione all'altra.

Rispetto alla metamorfosi, questo progetto ha aggiunto una dimensione nuova: l'interattività. L'utente agente che sceglie il proprio percorso attraverso le scene.

Anche qui le connessioni H-STEAM si sono attivate naturalmente: la geografia e le scienze hanno fornito il contenuto scientifico, l'arte e la musica hanno guidato la progettazione degli sfondi e dei suoni stagionali



Figura 3 - Attività di storytelling sul ciclo delle stagioni

4.3 La metamorfosi: quattro scenari per una trasformazione

Tra i progetti realizzati nelle classi quinte, la metamorfosi del bruco in farfalla ha suscitato nei bambini particolare interesse. La struttura narrativa, uovo, bruco, crisalide, farfalla si è tradotta naturalmente in quattro scenari distinti, ognuno con il proprio sfondo, i propri personaggi disegnati digitalmente da ciascun bambino e la propria sequenza di eventi animati.

Dal punto di vista computazionale, questo progetto ha introdotto il concetto di struttura multi-scena: come si passa da uno scenario all'altro? Come si gestisce la continuità narrativa attraverso cambi di ambientazione e di personaggio?

Le connessioni interdisciplinari in questo progetto sono state molteplici e spontanee: le scienze naturali hanno fornito il contenuto (il ciclo biologico della farfalla), l'arte ha guidato la creazione digitale degli sprite e la progettazione visiva di ogni scena, la narrazione ha dato struttura all'intera sequenza. Il coding ha tenuto insieme tutto questo, come un linguaggio che sa parlare con tutti gli altri.

4.4 Il sistema Terra-Luna: due fenomeni, un'unica animazione

Un altro artefatto significativo realizzato nelle classi quinte ha avuto come protagonista il sistema Terra-Luna, con l'obiettivo di rappresentare visivamente due fenomeni astronomici distinti ma correlati: l'alternarsi del giorno e della notte, causato dalla rotazione della Terra su se stessa, e le fasi lunari, causate dalla rivoluzione della Luna intorno alla Terra.

La distinzione tra i due fenomeni è stata il punto di partenza scientifico del progetto. Il lavoro di progettazione dell'animazione, dover decidere cosa mostrare, in quale scena, con quale movimento, ha costretto a chiarire concettualmente la differenza. Il coding, in questo caso, ha funzionato come strumento di comprensione scientifica oltre che di espressione creativa.

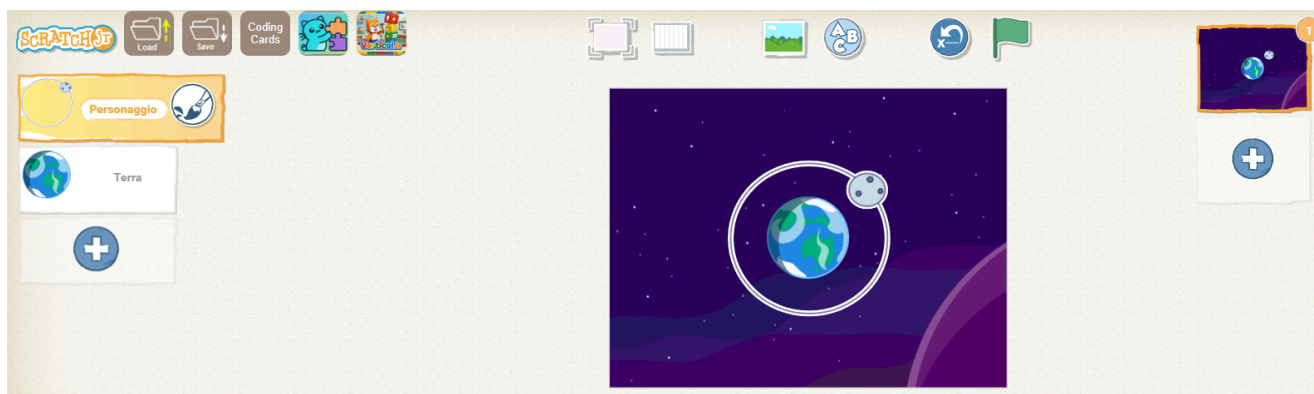


Figura 4 - Fase della progettazione della rotazione della luna intorno alla terra

Quest'attività ha richiesto una gestione precisa del movimento ciclico e della sincronizzazione tra più oggetti animati: la Terra che ruota, la Luna che orbita, il Sole che rimane fisso come riferimento. Dal punto di vista H-STEAM, le connessioni con le scienze (astronomia, fisica della luce e delle ombre) e con l'arte (la rappresentazione visiva del cosmo, la scelta dei colori e delle proporzioni) sono state particolarmente ricche.

5. L'approccio H-STEAM: quando le discipline incontrano il codice

Uno degli esiti più significativi del percorso è stato l'emergere spontaneo di connessioni interdisciplinari in ogni classe e in ogni fase. Il coding non è mai stato vissuto come un'attività isolata, avulsa dal resto dell'apprendimento. Al contrario, ha funzionato come un catalizzatore: uno spazio in cui i saperi di discipline diverse si incontravano, si intrecciavano e si arricchivano reciprocamente.

La tabella seguente sintetizza, a titolo esemplificativo, le principali connessioni disciplinari attivate nelle diverse fasi del percorso.

Attività / Progetto	Scienze	Arte	Italiano / Narrazione	Musica	Informatica
Labirinto (unplugged + digitale)	Geometria, spazio	Mappa e composizione	Strategie verbali	—	Sequenze, algoritmi, debugging
Storie animate (più personaggi)	Ambienti naturali	Illustrazione digitale	Dialogo, trama, personaggio		Sincronizzazione eventi
Metamorfosi (4 scenari)	Ciclo vita farfalla	Scenari visivi per fase	Racconto trasformazione	—	Struttura multi-scena, eventi
Ciclo delle stagioni	Fenomeni stagionali	Colori e sfondi stagionali	Poesia, simbolismo	Suoni stagionali	Interattività, input utente
Sistema Terra-Luna	Astronomia, luce e ombre	Rappresentazione del cosmo	—	—	Movimento ciclico, sincronizzazione

Tabella n. 1 - Connessioni disciplinari H-STEAM in alcuni progetti rappresentativi del percorso

6. Risultati e riflessioni

Al termine del percorso, emergono risultati significativi su più piani: quello delle competenze computazionali, quello delle competenze creative ed espressive, e quello della motivazione e del senso di sé come autori. Per quanto riguarda le Competenze computazionali, si può affermare che tutti gli studenti, indipendentemente dalla classe, hanno acquisito una comprensione operativa dei concetti fondamentali del pensiero computazionale: sequenze, algoritmi, debugging, pattern, eventi. La progressione tra classi ha garantito che questi concetti fossero affrontati con crescente profondità e astrazione, evitando sia la ripetizione sterile sia il salto di livello troppo brusco.

In merito alle competenze creative ed espressive, gli artefatti prodotti dagli studenti, testimoniano un livello di investimento creativo notevole: cura per i dettagli visivi, coerenza narrativa, scelte espressive consapevoli. Infine in quanto a motivazione, il risultato più importante, può considerarsi il cambiamento nel modo in cui gli studenti si sono percepiti: non come fruitori di tecnologia ma come suoi creatori.

Il percorso dimostra in breve che il coding creativo, quando è pensato in modo progressivo, differenziato e integrato con altri linguaggi espressivi, può diventare un'esperienza di apprendimento autentica e significativa.

La chiave non sta nello strumento, CodeJr.org e/o CodeKidi, ma nell'approccio: partire dall'esperienza concreta, costruire la complessità gradualmente, lasciare spazio alla creatività e alla co-costruzione, integrare il coding con i saperi disciplinari invece di isolarlo in un compartimento separato.

Riferimenti bibliografici

- Bell, T., Witten, I.H. & Fellows, M. (2009). *Computer Science Unplugged: Off-line activities and games for all ages*. University of Canterbury.
- Brackmann, C.P., Román-González, M., Robles, G., Barone, D., Moreno-León, J. & Casali, A. (2017). Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*.
- Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*.
- Grover, S. & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Kafai, Y.B. & Burke, Q. (2015). *Connected code: Why children need to learn programming*. MIT Press.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Resnick, M. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT Press.
- Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.



Elisa Garieri

elisa.garieri@univr.it

Università degli studi di Verona

Laureata in Economia e Commercio e in Scienze dell'Educazione. Ricercatrice presso l'Università di Verona. Docente scuola Secondaria di Secondo Grado in provincia di Padova. Docente a contratto presso l'Università di Padova per l'insegnamento di Competenze Informatiche e presso l'Università di Verona per la formazione dei Docenti di sostegno in ambito Tecnologico. Formatrice esperta: PNSD, PNRR, animatori digitali, team per l'innovazione, docenti. Esperta in metodologie e didattiche innovative con le nuove tecnologie per studenti con BES. Componente dell'Equipe Formativa territoriale Veneto 2021-2024.



Roberto Gigliotti

gigliotti.roberto69@gmail.com

IIS "E.U. Ruzza" - Università eCampus - Università degli Studi di Verona - Università degli Studi di Sassari - Università Magna Graecia di Catanzaro - Università degli Studi della Basilicata. Docente di matematica presso l'Istituto E. Uselli Ruzza di Padova attualmente distaccato presso l'ateneo eCampus. Laureato in Informatica e appassionato di Nuove Tecnologie. Direttore della Fondazione ITS Academy COSMO dal 2010 al 2025. Co-Autore del testo: "L'atelier digitale, dalle forbici al mouse". Formatore in percorsi di aggiornamento per studenti, docenti e per dipendenti di aziende pubbliche e private in ambito digitale. Docente a contratto presso le Università di Verona, Sassari, Basilicata e Magna Grecia di Catanzaro.