



Progetto PON CODING E ROBOTICA 10.2.7.A2-FSEPON-INDIRE-2017-1, CUP  
B59B17000000006

## Il coding e la robotica educativa

*di Beatrice Miotti*

Il termine coding nella accezione di metodologia usata in ambito scolastico, è mutuata dall'ambiente informatico dove significa "programmazione". Ad un certo punto la parola coding è stata introdotta in ambito educativo per indicare quelle attività che facendo o non facendo uso di uno strumento tecnologico permettevano di sviluppare la competenza del problem solving e del pensiero computazionale.

Sono nate quindi attività di due tipologie: il coding unplugged che non utilizza strumenti tecnologici e come si trova nella Indicazioni Nazionali Nuovi Scenari (2018) "nella didattica, si possono proficuamente mettere a punto attività legate al pensiero computazionale anche senza le macchine. Ogni situazione che presupponga una procedura da costruire, un problema da risolvere attraverso una sequenza di operazioni, una rete di connessioni da stabilire (es. un ipertesto), si collocano in tale ambito, a patto che le procedure e gli algoritmi siano accompagnati da riflessione, ricostruzione metacognitiva, esplicitazione e giustificazione delle scelte operate". Un esempio riguarda la acquisizione di competenze legate al campo di esperienza "la conoscenza del mondo" tali da raggiungere il seguente traguardo: "Individua le posizioni di oggetti e persone nello spazio, usando termini come avanti/dietro, sopra/sotto, destra/sinistra, ecc; segue correttamente un percorso sulla base di indicazioni verbali" in questo caso i bambini sperimentano, anche usando il proprio corpo, le modalità di impartire istruzioni al fine di risolvere un problema o realizzare un percorso. Questa semplice attività mette in moto una serie di riflessioni che non sono banali riguardo al concetto di istruzione, codice e linguaggio.

L'altra tipologia di attività comprende l'uso di artefatti tecnologici come per la robotica educativa o applicativi software come Scratch o Blockly per la programmazione visuale.

Il coding, quindi, non è solo programmazione anzi, la scrittura di codice (seppure tramite approcci visuali) è proprio l'ultimo passo di un percorso in cui è la presa di coscienza del processo e la consapevolezza della logica sottesa sono le competenze importanti da acquisire. Nel 2015 Resnick spiega così il coding: «Noi vediamo il coding come un nuovo modo, per le persone, di organizzarsi, esprimersi e condividere le proprie idee».



Anche in Italia negli ultimi anni, in particolare all'interno del programma "La buona scuola", sono stati introdotte attività orientate allo sviluppo di competenze di programmazione con il progetto "Programma il futuro", mentre nel 2018 con le Indicazioni Nazionali Nuovi Scenari il pensiero computazionale è stato proprio inserito come competenza da acquisire ed è così descritto "Per pensiero computazionale si intende un processo mentale che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici pianificando una strategia."

Per ampliare il quadro dei riferimenti di letteratura, rispetto all'uso del coding ma anche di artefatti tecnologici che devono essere comunque istruiti tramite un programma al fine di compiere azioni (robotica educativa), riporto qua brevemente alcuni cenni storici.

Saymur Papert già negli anni '70 aveva sviluppato un artefatto cognitivo per lo studio della geometria collegato al linguaggio LOGO [Papert, 1998] che aveva inventato. Successivamente questo artefatto da reale (una specie di robot con la forma di una tartaruga) divenne virtuale (un piccolo triangolo) che poteva muoversi sullo schermo e tracciare figure geometriche secondo istruzioni tramite linguaggio LOGO da parte degli studenti.

Papert di formazione costruttivista introdusse la definizione di costruzionismo [Harel, Papert 1991][Papert, 1986], riprendendo gli stessi principi del costruttivismo (studente al centro del proprio processo di apprendimento, docente con ruolo di facilitatore, creazione di artefatti) ma di fatto rendendo gli artefatti da mentali a reali ossia oggetti fisici da manipolare, costruire, provare e sbagliare.

Appariva già allora (si parla degli anni '70) evidente che alcune caratteristiche intrinseche dei bambini fossero la chiave di volta per un processo di apprendimento efficace, in particolar modo la loro capacità di costruire cose, di lavorare su oggetti concreti, la loro curiosità nello scoprire e manipolare che ha portato alla necessità di dare stimolo e seguito alla loro fantasia lasciandoli liberi di sperimentare. In questo senso Papert si avvale del concetto di Pensiero concreto [Papert, 1980] indicando come l'apprendimento sia più efficace e padroneggiato quando non è solo mentale, ma è supportato da una costruzione reale, da una attività come la costruzione di un progetto significativo fisico o virtuale che sia.

In accordo alla definizione di costruzionismo è quindi evidente come alcune metodologie didattiche per la conduzione delle attività in classe siano ovviamente da preferire quando ci avviciniamo al coding o alla robotica educativa, strategie che oggi sono applicate quotidianamente dalla maggior parte degli insegnanti anche nello studio delle singole discipline. Nella metodologia "apprendimento per scoperta" [Bruner, 1961], così come negli approcci di



tipo learning by doing [Dewey,1938] o problem based learning cambia la modalità tradizionale di insegnamento: l'alunno è artefice del proprio processo di apprendimento, che porta avanti in collaborazione con i compagni ed in parte con studio autonomo, sotto la supervisione del docente che ha in questo caso la funzione di facilitatore. Le problematiche che sono proposte agli studenti sono reali e implicano da parte dei ragazzi fasi di osservazione, riflessione, analisi e modellazione dei problemi per trovare poi una soluzione che non necessariamente è univoca. È il percorso che lo studente affronta per raggiungere l'obiettivo che conferisce valore educativo all'attività.

Inoltre, le attività di coding e robotica educativa permettono anche una rivalutazione degli errori: infatti nel momento in cui il programma scritto o il robot programmato non agisce nella modalità aspettata, lo studente è chiamato a riflettere sui motivi del comportamento anomalo e quindi a rivalutare le proprie osservazioni ed implementazioni così da giungere ad una corretta soluzione.

Appare quindi evidente come il percorso per l'acquisizione di competenze per il problem solving, competenze che porteranno poi alla capacità di sviluppare un programma informatico, non sia immediato ma derivato da una serie di conoscenze che i bambini devono apprendere in modo graduale.

Uno dei vantaggi dei percorsi di coding e robotica educativa è legato proprio anche alla sfera motivazionale: quando lo studente costruisce o modella il proprio artefatto fisico o virtuale instaura con esso un senso di affiliazione che lo incentiva a superare anche parziali fallimenti. Si pensi ad esempio all'utilizzo di robot in ambito ospedaliero dove i bambini identificano nell'oggetto un loro eguale.

Il coding e la robotica educative non sono discipline vere e proprie, nonostante ci siano alcune scuole che hanno organizzato l'orario in modo da ritagliare un tempo settimanalmente per proporre la studio di informatica o dei robot, tipicamente sono utilizzate come metodologie a supporto delle altre discipline.

Ma sono estremamente utili perché permettono di fare fronte al problema della contestualizzazione delle conoscenze poiché i saperi devono essere incorporati in campi di esperienza significativi per acquistare senso nella mente di chi apprende e solo dopo questo processo lo studente può rielaborare la conoscenza acquisita applicandola ad altri contesti [Merlo,2017]. Inoltre, favoriscono l'inclusività perché si basano sostanzialmente su lavori di gruppo e favoriscono le intelligenze multiple che emergono nel momento in cui si propongono attività diversificate per specie e competenze impegnate.



Il coding e la robotica educativa si posizionano sostanzialmente su due linee nella didattica e nella organizzazione scolastica: secondo un approccio in orizzontale, sia per una singola disciplina e sia fungendo da raccordo fra più discipline in un'ottica interdisciplinare, oppure in verticale in una prospettiva di curricolo.

Nel primo caso, oltre a supportare come strumento una singola disciplina, permettono di affrontare argomenti multidisciplinari agendo come strumenti e collante fra più progettazioni. Proprio perché versatili e adattabili a diversi contesti in cui lo studente è elemento attivo nel proprio processo di apprendimento, il coding e la robotica educativa possono spaziare da diversi ambiti: la matematica, la fisica e più in generale le TEAM sono forse gli ambiti di più immediato utilizzo, ma sono stati realizzati anche applicazioni che coinvolgono le lingue, le materie umanistiche (ad esempio il teatro, o lo storytelling).

Per costruire un robot e per programmarlo con delle istruzioni a raggiungere un obiettivo, così come per progettare ed implementare un algoritmo che risolve una problematica, occorrono tante competenze diverse, ed è già questa una attività tipicamente interdisciplinare. La parte disciplinare sarà ovviamente sempre presente ma non necessariamente l'alunno deve esserne immediatamente consapevole, sarà il docente che evidenzierà al momento opportuno la specificità di una disciplina nascosta dentro un aspetto particolare dell'attività che stanno svolgendo gli alunni così da far prendere loro coscienza del sapere in gioco. Sarà poi sempre compito del docente fare in modo che quello stesso sapere sia interiorizzato dagli studenti e esteso anche ad altri contesti.

Appare quindi evidente come l'utilizzo della robotica e del coding avulsi dalla programmazione didattica e dallo specifico curricolo sia in realtà fine a sé stesso e poco incisivo sugli apprendimenti degli studenti. È compito del docente interagire con gli studenti proponendo inferenze e sollecitando riflessioni e approfondimenti.

L'altro piano su cui possiamo agire utilizzando è quello verticale, proponendo quindi un curricolo che tenga conto della attività svolta dagli studenti in tutti gli ordini scolastici e supportando quindi la continuità educativa, metodologica e didattica. Il peer tutoring che già emerge anche fra studenti della stessa età può rivelarsi una strategia vincente proprio nel momento in cui si lavora con studenti di età diverse. Organizzare questa tipologia di approccio è sicuramente più impegnativo rispetto alla interdisciplinarietà perché al contrario di questa che agisce all'interno dello stesso consiglio di classe o con i docenti della stessa classe nella scuola primaria, in questo caso è necessario andare a progettare il curricolo su ordini di scuola diversi e questo è possibile solo se c'è una forte indicazione e motivazione anche da parte della dirigenza poiché significa andare a scardinare una struttura scolastica ormai consolidata. Per fortuna negli ultimi anni, grazie proprio al processo di innovazione



scolastica in atto, questo processo di contaminazione e peer tutoring tra ordini scolastici diversi è sempre più frequente e fa ben sperare per la trasformazione in prassi di una didattica attiva ed incentrata sullo studente.

### **Bibliografia di riferimento**

- Bruner, J. S. (1961). The act of discovery. *Harvard Educational Review*, 31, 21–32.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books. Trad. It. *Mindstorms. Bambini computer e creatività*. Milano, Emme, 1984.
- Harel, I., Papert, S. (eds) (1991). *Constructionism*. Norwood. New Jersey: Ablex.
- Papert, S. (1986). *Constructionism: A new opportunity for elementary science education*. NSF Grant Application.
- Papert, S. (1998). *Strumenti di didattica multimediale*. MicroMondi. Il logo multimediale per imparare nel 2000. Roma: Garamond.
- Dewey, John. (1938). *Experience and education*. New York: Collier.
- Howard S. Barrows, MD, Robyn M. Tamblyn (1980), *Problem-Based Learning: An Approach to Medical Education*, Springer
- Merlo, D. *La robotica educativa nella scuola primaria*, Ed. Teli ISBN- 9788826465432

## UNITA DIDATTICA

### “Dalla pixel art a scratch: un percorso nel mondo del coding”

Di Beatrice Miotti

L'attività descritta in questo video è nata durante il primo periodo di lockdown desiderando sperimentare con i bambini della seconda un breve percorso di coding unplugged in modalità collaborativa. Questa attività oltre a sviluppare semplici nozioni di pensiero computazionale è stata pensata per far interagire tra loro i bambini non solo durante l'ora di lezione ma anche in orario extrascolastico al fine di portare a termine il compito assegnato.

I contenuti che sono stati affrontati in questo percorso didattico sono:

- La pixel art
- Il codice (linguaggio condiviso, non ambiguo)
- Come funziona un computer e come si programma
- Gli algoritmi
- Scratch

Partendo quindi da esempi di pixel art siamo arrivati a proporre ai bambini semplici esempi di algoritmi con scratch.

#### Fase1 (consegna)

Richiesta: Ai bambini è stato richiesto di ricopiare il disegno di un pulcino fornito dagli insegnanti e realizzato su un foglio a quadretti da 1cm, con cornice di lettere e numeri.

Note: Non è stata data alcuna indicazione sulle modalità con cui realizzarlo (per righe, colonne, riferendosi a numeri o lettere ecc.)

Restituzione:

- sottoposto un semplice questionario per capire il grado di interesse per l'attività e le difficoltà riscontrate.
- Discussione in plenaria sulle modalità di realizzazione

*Nota per l'insegnante:* -

#### Fase 2 (consegna)

Richiesta: Ai bambini è stato richiesto:

- di ricopiare un disegno di un albero fiorito più complesso del precedente e realizzato su foglio a quadretti da 0.5cm e senza indicazioni di numeri e lettere sulla cornice
- Di scrivere un breve testo in cui spiegavano come avevano realizzato il disegno (per righe colonne, scrivendo numeri e lettere ecc.)

Note: Non è stata data alcuna indicazione sulle modalità con cui realizzarlo (per righe, colonne, possibilità di inserire numeri o lettere ecc.)

### Restituzione:

- Discussione in plenaria sulle modalità di realizzazione
- Lettura del loro testo e brevi interviste
  - o Era più facile o più difficile del pulcino?
  - o Ti è piaciuto?
  - o Vorresti rifarlo?

*Nota per l'insegnante:* -

### **Fase 3 (consegna)**

Richiesta: Ai bambini è stato richiesto di realizzare il disegno a partire dalla scheda composta da:

- una legenda: Carattere – Colore (ad esempio B – Bianco; R – Rosso; RS- Rosa ecc)
- un elenco di righe e per ogni riga il colore di ogni quadretto identificato da Lettere/Numero e la sigla del colore.

Note: Non è stata data alcuna indicazione sulle modalità con cui realizzarlo (per righe, colonne, possibilità di inserire numeri o lettere ecc.)

### Restituzione:

- Discussione in plenaria sulle modalità di realizzazione
  - o Hai avuto difficoltà
  - o Ti pare simile a qualche lavoro che già hai svolto?

*Nota per l'insegnante:* Questa modalità con incroci (colonna/riga) era stata già affrontata dai bambini l'anno precedente nelle lezioni sulla griglia topologica, quindi, sapevano muoversi e localizzare già i quadretti come su una matrice.

### **Fase 4 (lezione)**

Attività: presentazione delle slide 1 sui Codici

Durata: 30 minuti di presentazione delle slide + 10 minuti di discussione plenaria

*Nota per l'insegnante:* Le slide 1 descrivono il concetto di Codice e sollecitano gli studenti ad un confronto sulla parola stessa. Poi tramite alcuni esempi di prassi condivise (i colori del semaforo, i cartelli stradali, il cartellino rosso o giallo dell'arbitro) si cerca di far apprendere il concetto di linguaggio condiviso come necessità per essere tutti concordi e consapevoli delle azioni. Ad esempio, nel nostro linguaggio e quindi nel codice del semaforo il rosso significa stop, mentre il verde avanti e se uno conosce questo modo di interpretare il colore saprà sempre che cosa significa.

### **Fase 5 (2 ore)**

Richiesta: Ad ogni gruppo di bambini è stato richiesto di realizzare il disegno a partire dal materiale dato:

- una legenda: Carattere – Colore (ad esempio B – Bianco; R – Rosso; RS- Rosa ecc)
- un elenco di righe e per ogni riga il colore di ogni quadretto identificato da Lettere/Numero e la sigla del colore.

Successivamente dovevano osservare il disegno e provare ad inventare con i compagni di gruppo un codice alternativo (a quello presentato nella consegna), ossia un modo diverso per far disegnare ai compagni il proprio lavoro.

Note: i bambini sono stati suddivisi in gruppi ed hanno lavorato all'interno di breakout rooms sotto la sorveglianza dei docenti che passavano da stanza a stanza. Ognuno lavorava sul proprio quaderno per la parte di realizzazione del disegno. Per quanto riguarda la parte di invenzione del codice hanno proseguito il lavoro ritrovandosi nel pomeriggio in stanze create dai docenti per ogni gruppo e trasmesse tramite invito [Opzionale]. Hanno poi completato i lavori nuovamente durante l'orario scolastico nelle breakout rooms. A gruppi hanno poi condiviso (1 studente del gruppo) il proprio schermo e mostrato ai compagni il codice creato.

Restituzione:

- o Presentazione del proprio risultato da parte di ogni gruppo ai compagni.
- o Intervista da parte dei docenti ad ogni gruppo per capire come hanno lavorato e come hanno ottenuto la loro soluzione

*Nota per l'insegnante:* L'idea principale è stata proprio quella di far trovare ai bambini, che non avevamo mai avuto alcun tipo di approccio al coding, alla codifica di informazioni, e alla definizione di linguaggio o algoritmi, una modalità di codifica che rispondesse alle esigenze di replicabilità e condivisione. Per fare questo passaggio i bambini hanno lavorato a gruppi inventando un proprio codice che permettesse di disegnare una immagine e lo hanno poi condiviso con i compagni. La discussione che ne è derivata ha permesso agli insegnanti di orientare poi i ragazzi a trovare un codice comune che oltre ad essere univoco per tutti, semplificasse anche il passaggio delle informazioni (codice tipico della pixel art).

Con questa riflessione e facendo riferimento a quella sulla parola codice e sulle caratteristiche che sono necessarie per poter dare istruzioni agli altri della lezione precedente, è stata introdotto il codice della pixel art della lezione seguente.

## **Fase 6 (lezione)**

Attività: presentazione delle slide 2 Pixel Art

Durata: 10 minuti

*Nota per l'insegnante:* A completamento della lezione precedente

**PRESENTAZIONE DELLE SLIDE LEZIONE 2 DA RICERCATORI INDIRE**

## **Fase 7 (consegna)**



Richiesta: Ai bambini è stato richiesto di seguire la seguente consegna articolata in punti:

“Cari bambini,

in questa attività vi chiediamo di lavorare in coppia con un vostro compagno che è lo stesso che trovate tra i destinatari di questo messaggio.

Abbiamo diviso l'attività in fasi diverse che abbiamo numerato per voi. Seguite l'ordine per procedere con il lavoro

---

### **FASE 1 - trova immagine**

Cerca nella posta il messaggio che ti ha inviato il maestro, il messaggio ha come titolo *IMMAGINE per Creare il codice*.

Nel messaggio troverai l'immagine che è stata assegnata a te. Disegna questa immagine sul tuo quaderno.

---

### **FASE 2 - disegna e crea il codice**

Una volta disegnata l'immagine controlla che sia identica all'originale ricevuta dal maestro.

Adesso crea il codice usando il "nuovo linguaggio della Pixel Art" scoperto a lezione, che ti permette di raggruppare sequenze di quadrati dello stesso colore in un unico codice. Ad esempio: A-10G significa nella Riga A colorare di Giallo 10 quadrati in sequenza, eccetera);

---

### **FASE 3 - invia il tuo codice (*entro indicare data*)**

Passa il codice che hai creato al tuo compagno. Dopo aver scritto il codice sul tuo quaderno, fagli una foto e invialo tramite messaggio e-mail. Fai attenzione che il codice nella foto sia leggibile bene. Prova tu stesso a leggere il codice dell'immagine che hai fatto prima di mandarlo.

Come fare ad inviare la foto del codice:

- vai su gmail, dove trovi tutti i tuoi messaggi di scuola e cerca proprio questo messaggio che stai leggendo adesso e che ti ha inviato la maestra;
- una volta aperto questo messaggio, clicca su “rispondi a tutti” (se non lo trovi clicca sui 3 puntini in alto a destra, comparirà la parolina “altro”, clicca e poi scegli “rispondi a tutti”);
- adesso allega la foto del tuo codice;
- premi 'invia' e manda così il codice al compagno.  
Fai attenzione che tra i destinatari di questo messaggio ci siano, oltre al tuo compagno, anche i maestri.

---

### **FASE 4 - disegna dal codice ricevuto (*da indicare azione*)**

Controlla nella tua posta di aver ricevuto un messaggio dal tuo compagno con i codici del suo disegno. Scrivi sul quaderno il codice ricevuto e poi ricrea il disegno del compagno.

Se *entro indicare data* non ricevi il codice dal tuo compagno contattalo, sempre per messaggio.

---

### **FASE 5 - invia il disegno (*entro indicare data*)**

Invia al tuo compagno la foto del disegno che hai realizzato usando il suo codice. Usa sempre gmail, seguendo le indicazioni della FASE 3

---

### **FASE 6 - controlla il disegno del compagno** (*entro indicare data*)

Controlla nella tua posta di aver ricevuto un messaggio dal tuo compagno. Allegato al messaggio troverai il disegno che lui ha realizzato usando il tuo codice.

Verifica che il suo disegno sia uguale al tuo originale (quello che hai usato per creare il codice che gli hai inviato).

Se il tuo disegno e quello del tuo compagno non sono uguali, se ci sono delle differenze, controlla il codice che hai creato per questa immagine e che hai passato al tuo compagno. Verifica se la differenza tra i due disegni è dovuta ad un errore nelle tue istruzioni o se è il tuo compagno ad aver interpretato male il tuo codice.

Vai sul tuo quaderno e trova il codice che hai creato nella FASE 2. Cerchia con una matita il punto del codice dove hai trovato la o le differenze.

Rifletti: Dove sta l'errore? È nel codice che hai creato oppure nel disegno del tuo compagno?

---

### **FASE 7 - comunica il risultato** (*per indicare data*)

Scrivi al tuo compagno (sempre come indicato nella FASE 3) e, se il tuo compagno non ha fatto errori, fagli i complimenti. Se invece hai trovato degli errori, allega una foto del codice dove hai segnato gli errori (FASE 6) e cerca con le tue parole di spiegare l'errore.

Se secondo te l'errore sta nel tuo codice, magari il tuo compagno lo ha trovato: parlatene insieme, anche trovandovi su Meet come per l'esercizio scorso. “

Note: I docenti hanno suddiviso i bambini a coppie per rendere più semplice lo scambio di mail e messaggi, ma se fatta direttamente online in breakout room, potrebbe essere condotta anche in gruppi più numerosi. I docenti avevano previsto circa una settimana fra l'invio di questo messaggio e la restituzione finale. I docenti hanno inviato ad ogni coppia di bambini due disegni tramite mail: ogni bambino ovviamente non conosceva il disegno del suo compagno. I docenti hanno supervisionato gli scambi di mail restando sempre in copia a tutti i messaggi.

#### Restituzione:

- Discussione in plenaria sulle modalità di realizzazione e sugli errori effettuati.

*Nota per l'insegnante:* questa attività oltre a permettere ai bambini di familiarizzare coi codici della pixel art ha introdotto un importante concetto del coding che è quello del “debug”. Infatti, ogni studente per effettuare la conclusione della consegna ha dovuto “debuggare” il disegno del compagno alla eventuale ricerca di un errore, e anche a ritroso trovare eventuali errori nel proprio codice fornito che hanno portato all'errore nel disegno del compagno. In ogni caso l'errore è da considerarsi costruttivo.

## Fase 8 (lezione)

Attività: presentazione delle slide 3 sul linguaggio delle cose

Durata: 20 minuti di presentazione delle slide + 10 minuti di discussione plenaria

*Nota per l'insegnante:* Le slide 3 cercano di descrivere in modo generico ma calandosi nel mondo della informatica che cosa significa dare una istruzione al computer. Cosa accade quando clicchiamo sulla icona di un bel videogioco? Che cosa sta dietro a tutto questo? Viene introdotto MicroPi il cervello del computer (processore) che ragiona in modo da risolvere problemi proprio come gli studenti fanno con i problemi di matematica: cercano i dati, pensano alle operazioni da eseguire, eseguono le operazioni, e scrivono una risposta. La stessa cosa accade nel MicroP quando clicchiamo sulla icona del gioco: vengono cercati i dati (le skin dei personaggi, i mondi di Mirko ecc.) e poi in base alle istruzioni che qualcuno ha scritto vengono fatte delle azioni. Ma chi scrive le istruzioni? E che linguaggio parla MicroP? Le istruzioni vengono scritte dai programmatori, che sono coloro che insegnano a MicroP quali azioni eseguire. Il nostro linguaggio, quello umano, però è complicato per MicroP, perché è ambiguo e le parole hanno significati diversi a secondo del contesto, quindi MicroP usa un linguaggio fatto solo da 0 e 1 che disposti in sequenze rappresentano le istruzioni. I programmatori non usano gli 0 e gli 1 ma un vocabolario di parole come potrebbero essere ad esempio le frecce per programmare il percorso di un robot per andare da un punto ad un altro.

## Fase 9 (consegna)

Consegna: i bambini devono determinare la sequenza giusta di istruzioni affinché il robot si sposti dal punto di inizio (la moltiplicazione) verso il giusto risultato.

*Nota per l'insegnante:* -possono essere creati percorsi e storie basandosi sul concetto di mappa e movimento del robot, e programmarlo usando frecce o comandi condivisi dai bambini. Per fare delle lezioni interattive si possono usare ad esempio gli scacchi o la dama che si spostano sulla scacchiera ognuno secondo regole prestabilite condividendo lo schermo e lasciando il controllo via via a ogni bambino perché possa sperimentare i percorsi.

## Fase 10 (lezione)

Attività: presentazione delle slide 4 "facciamo ordine con il coding"

Durata: 20 minuti di presentazione delle slide + 10 minuti di discussione plenaria e/o gruppi di lavoro per la realizzazione di algoritmi

*Nota per l'insegnante:* Le slide 4 riassumono brevemente quanto detto nella precedente lezione, puntualizzando e innescando una riflessione su come quanto detto la volta scorsa non fosse solo "informatica" anzi, ma fossero intervenute più discipline: la matematica per la risoluzione di problemi e le tabelline, la geografia con le indicazioni topologiche, l'italiano con l'ambiguità del linguaggio. Appare evidente e necessario far notare che il coding è interdisciplinare. Dopo un breve brainstorming sulla modalità con cui raccontare a qualcuno che non sa una qualche azione (ad esempio come attraversare la strada, come risolvere un problema di matematica; come fare la limonata) ed aver definito una sequenza di istruzioni per eseguirle, si introduce il concetto di

algoritmo che deve essere non ambiguo e chiaro e che permette di mettere in ordine le istruzioni. Si procede quindi con esempi mettendo in evidenza anche i costrutti di Se-Allora e il ciclo.

## **Fase 11 (consegna)**

Consegna: i bambini devono scrivere degli algoritmi su azioni semplici e quotidiane, o su problemi di matematica.

*Nota per l'insegnante*: attività da svolgersi a gruppi o singolarmente

## **Fase 12 (lezione)**

Attività: presentazione di Scratch

Durata: 20 minuti di presentazione delle slide + 10 minuti di discussione plenaria e/o gruppi di lavoro per la realizzazione di semplici esempi

*Nota per l'insegnante*: In questa lezione si presentano le basi di Scratch ai bambini partendo da semplici azioni dello sprite (sequenza di movimenti, voci, costrutti di selezione, costrutti di ciclo), presentazione degli sfondi, modifica degli sprite e semplici esempi di algoritmi.

I ragazzi poi lavoreranno a gruppi su consegne da parte dei docenti sperimentando algoritmi legati alle sfere disciplinari: disegna una figura geometrica; narra una storia ecc.