

01

SETT
2020

SCUOLA ITALIANA MODERNA

Rivista
per la scuola
primaria

SETTEMBRE

WELCOME BACK TO SCHOOL!

- RICORDI DALLE VACANZE
- PRONTI AL RIPASSO!
- LAVORIAMO CON CANVA, GOOGLE PRESENTAZIONI E HYPERSAY
- MATEMAGICA: ACCOGLIENZA CON LA MAGIA DEI NUMERI





Percorsi su scacchiera

Antonio Faccioli

Formatore e volontario
CoderDojo

Riprendiamo il viaggio nel mondo del coding attraverso altri nove tutorial "step by step".

Questa volta, però, proviamo ad **ampliare la nostra proposta** utilizzando **sia attività digitali sia attività unplugged**, ovvero senza l'uso del computer.

Questo ci aiuterà a estendere a tutte le classi della scuola primaria il lavoro svolto nel corso della precedente annata di Scuola Italiana Moderna. Lo faremo sempre attraverso degli step che quest'anno saranno cinque, idealmente uno per ogni classe del ciclo scolastico.

Si tratta di **nove brevi percorsi multidisciplinari**, piccole tappe di un itinerario più ampio, collegate fra loro da un filo conduttore, ovvero il **calendario scolastico**:

- Percorsi su scacchiera (attività unplugged)
- PixelCoding (attività mista unplugged/digitale)
- Le stagioni (Tinkering/Scratch)
- Merry Xmas (Tinkering/Scratch)
- Fiocchi di neve (Tinkering/Scratch)
- Carnevale (Tinkering/Scratch)
- Circuiti elettrici (Tinkering/Scratch)
- Costruiamo una calcolatrice (Tinkering/Scratch)
- Strumenti musicali (Tinkering/Scratch)

Iniziamo quindi il nostro percorso con un **primo tutorial totalmente unplugged** attraverso la mia personale rivisitazione di un'attività che utilizzo spesso quando introduco il coding: i **percorsi su scacchiera**.

Gli strumenti di cui abbiamo bisogno sono una **scacchiera** e alcune **istruzioni stampate**.

MATERIALI

Nei miei laboratori solitamente uso 4 pezzi di moquette larghi 1 metro e lunghi 2 metri e li combino in base alle esigenze:

- due tappeti separati 2 m x 2 m, per lavorare in parallelo con più alunni;
- un'unica scacchiera 4 m x 2 m, che in genere utilizzo con i bambini delle classi quarta o quinta per piccoli tornei e sfide.

Ogni pezzo di moquette deve essere diviso in quadrati di dimensione uguale; in questo caso ho assemblato 2 pezzi di moquette composti da 3 quadrati in larghezza e 6 in lunghezza.



ISTRUZIONI

On line sono disponibili tre file da utilizzare come modello o da stampare direttamente:

1. **Istruzioni** (avanti, destra, sinistra);
2. **Oggetti** (da utilizzare per indicare dove partire, dove arrivare, errori);
3. **Cicli** (ripetizioni, ripeti fino a quando, se).

In realtà i tre file contengono più informazioni di quelle che utilizzeremo in questo tutorial, ma di ciò parleremo più approfonditamente nelle conclusioni.

CONSIGLI

- Stampiamo due istruzioni/oggetti per pagina e poi tagliamo. Otterremo così delle carte-istruzioni in formato A5 molto comode da tenere in mano.
- Plastifichiamo le pagine; in questo modo dureranno di più e gli alunni potranno tranquillamente calpestarle durante l'attività di gioco.



STEP 1 • LE PRIME ISTRUZIONI

In questo primo step utilizzeremo:

- le istruzioni **avanti**, **destra**, **sinistra**;
- la **bandiera verde** come segnaposto per indicare da dove partiamo;
- la **casa** e la **scuola** come punti di partenza e di arrivo del nostro percorso.

Dividiamo gli alunni in gruppi da due:

- un/una bambino/a, che d'ora in poi chiameremo **programmatore**, dovrà posizionare le istruzioni sul tappeto;
- l'altro/a bambino/a, invece, dovrà eseguire i comandi che verranno indicati dal/dalla compagno/a per raggiungere l'obiettivo.

Disponiamo sulla scacchiera la casa e la scuola e decidiamo quale delle due sarà la partenza e quale l'arrivo. A questo punto, chiediamo al programmatore di utilizzare le istruzioni, di disporle sulla scacchiera e di leggerle ad alta voce per indicare al/alla compagno/a il percorso da seguire. Attenzione: chi esegue il percorso deve muoversi solo dopo che avrà ricevuto l'ordine dal/dalla compagno/a.

L'istruzione **avanti** permette di avanzare di una sola casella per volta; le istruzioni **destra** o **sinistra** consentono di girare verso destra o verso sinistra, rimanendo sul posto (**Figure 1-2-3**).

Lo scopo di questa attività rispetto al coding è quello di **iniziare a elaborare delle sequenze di istruzioni** e **a risolvere piccoli problemi**. Natural-

mente vi sono altri aspetti interessanti, come lo **sviluppo di precise funzioni: spazialità, lateralità, orientamento**.

Alcune piccole accortezze raccolte negli anni, direttamente o indirettamente, possono essere utili in tutti gli step:

- mettiamo ai polsi dei bambini dei braccialetti (per esempio, di carta) dello stesso colore per aiutarli a riconoscere le istruzioni **destra** e **sinistra**; le prime volte questo aiuterà chi è ancora in difficoltà con la lateralità;
- non utilizziamo subito le istruzioni sul tappeto; meglio se invitiamo il programmatore a fare prima il percorso e successivamente a disporre le carte sul tappeto;
- iniziamo con piccoli percorsi di poche caselle, dritti, in modo che i bambini prendano dimestichezza con le istruzioni;
- per usare le frecce, mettiamo sul tappeto alcuni oggetti (per esempio, astucci, cartelle ecc.), come ostacoli che costringano a svoltare;
- ricordiamo sempre che **destra** e **sinistra** fanno girare sul posto;
- chiediamo al programmatore, oppure a un compagno o alla classe intera, di leggere le istruzioni ad alta voce. Ciò servirà a comprendere che **il coding si serve di linguaggi per codificare e decodificare informazioni**.

Figura 1

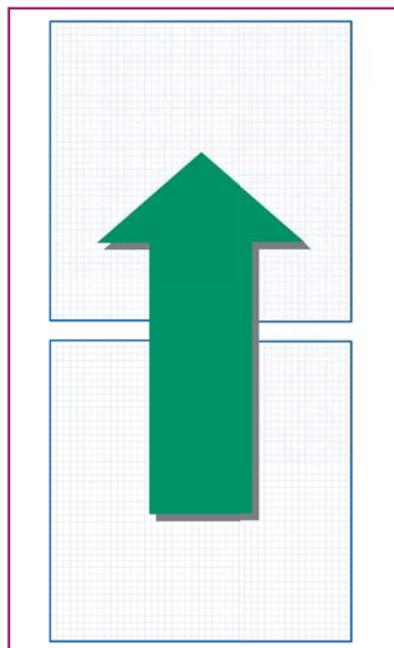


Figura 2

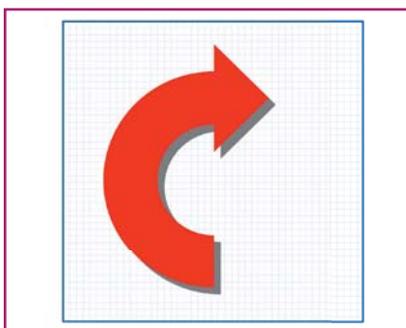
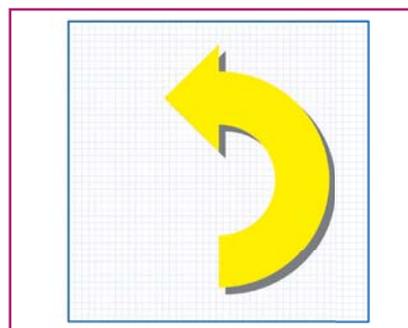


Figura 3



STEP 2 • OSTACOLI

Il passo successivo è quello di inserire diversi ostacoli nel percorso in modo che sulla scacchiera si possano identificare più soluzioni. Possiamo utilizzare oggetti che abbiamo a disposizione oppure i muri che troviamo nel file [Oggetti](#).

Le regole sono le medesime dello step precedente, con l'aggiunta che possiamo chiedere al programmatore di cercare di codificare il percorso più breve, o il più sicuro, per raggiungere la meta. Partiamo da percorsi semplici e poi via via sempre più complessi.

Suggerimento: potremmo mettere alla prova l'orientamento iniziale dell'alunno che deve eseguire il percorso. Possiamo chiedere di partire con le spalle al tappeto, in questo modo il programmatore dovrà aggiungere delle istruzioni ulteriori per farlo arrivare alla posizione più opportuna di partenza.

STEP 3 • TAPPE INTERMEDIE

Complichiamo leggermente i percorsi inserendo delle tappe intermedie. Per esempio: parti da casa, raggiungi l'autobus e poi vai a scuola, o viceversa. Come per lo step precedente, possiamo dare la consegna di identificare il percorso più breve o quello più sicuro (per evitare il maggior numero di ostacoli).

Suggerimenti:

- usiamo la stessa strategia dello step 2 per quanto riguarda la partenza. Ulteriore difficoltà potrebbe essere quella di chiedere di ritornare allo stesso punto di partenza e nella medesima posizione;
- facciamo attenzione a quanti punti intermedi inseriremo nel percorso; per esempio, potrebbe servire un maggior numero di istruzioni **avanti**.

STEP 4 • ALTRE ISTRUZIONI

Aggiungiamo un po' di istruzioni per arricchire il nostro gioco e magari lavorare anche con altre discipline. Nel file [Oggetti](#) è disponibile una carta con una mano che rappresenta l'istruzione **raccolgi/lascia**: la possiamo utilizzare in abbinata con gli oggetti **mela**, **torsolo** e **bidoncino dell'umido** (Figure 4-5-6).

Sul percorso, per esempio, possiamo mettere la mela e/o il torsolo che dovranno essere raccolti con l'istruzione rappresentata dalla mano e, quindi, di nuovo il programmatore dovrà mostrare la stessa carta per indicare di lasciare nel bidoncino quanto raccolto: l'esecutore dei comandi dovrà fisicamente raccogliere la mela o il torsolo e raggiungere il bidoncino.

Suggerimento: proviamo a creare altre carte gioco/istruzioni collegandole con altre materie, inserendo, per esempio, nomi di città, fatti storici, numeri ecc. **Il coding**, come l'informatica in generale, è uno strumento multidisciplinare.

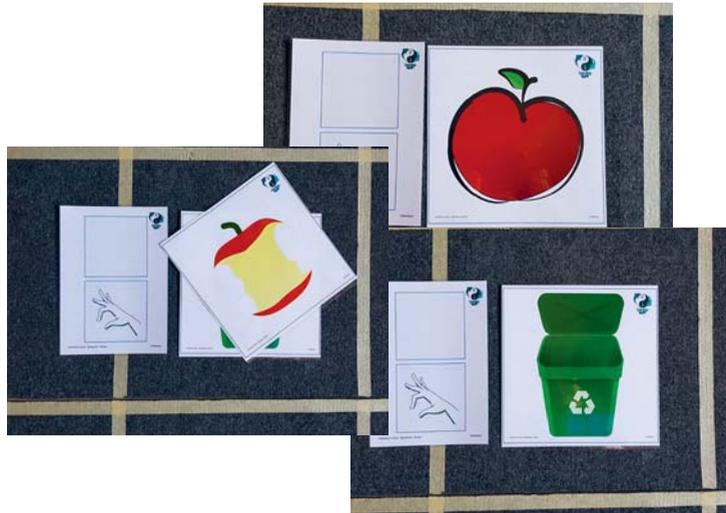


Figura 4



Figura 5

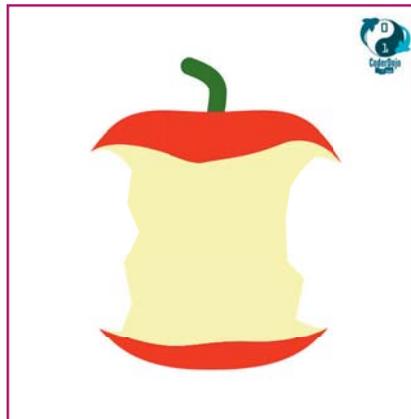


Figura 6



STEP 5 • ERRORI E CORREZIONI

Finora non abbiamo mai parlato di che cosa fare con gli errori.

In quest'ultimo step introduciamo l'uso della **carta bug**, la **coccinella**, e facciamo entrare in gioco altri compagni.

Creiamo alcuni **gruppi da 3 bambini**, ognuno dei quali dovrà avere un compito ben specifico:

- il **programmatore**, lo conosciamo già, è colui che codifica e costruisce le sequenze;
- il **debugger** deve controllare il **codice**, ovvero le **sequenze di istruzioni rappresentate con le carte**, prima di mandarlo in esecuzione e segnalare eventuali errori;
- il **robot** (se preferite cambiate il nome in base al contesto) dovrà "solo" eseguire le sequenze di istruzioni.

Una volta che avrà strutturato la sequenza, il programmatore chiederà al debugger di controllare il codice. Questi segnalerà con la carta della coccinella l'errore riscontrato.

Quando entrambi si sentiranno pronti, potranno chiedere al robot di eseguire il codice.

Suggerimenti:

- arrivati a questo punto, proponiamo delle sfide sempre più complesse, posizionando le tappe intermedie in mezzo ad alcuni ostacoli, per complicare un po' i percorsi;
- se riscontriamo una certa dimestichezza tra gli alunni, e questo vale anche per gli step precedenti, possiamo chiedere loro di non posizionare più le istruzioni lungo il percorso, ma di tenerle in mano, come un mazzo di carte.

CONCLUSIONI

Come anticipato nella nostra introduzione, questa tipologia di attività è utile per concentrarci:

- sulla **semplicità e non ambiguità delle istruzioni**;
- sull'**importanza delle sequenze e dell'ordine dei comandi**.

Sono **concetti alla base di tutti gli strumenti digitali** che possiamo utilizzare per fare coding e robotica.

A conclusione del nostro tutorial introduttivo per la nuova annata, condivido un **ebook** che raccoglie le mie esperienze sull'utilizzo dei percorsi su scacchiera.

In particolare, troverete il gioco **CoderGame**: una sfida da fare a squadre che consente di realizzare dei veri piccoli tornei utilizzando molte delle proposte che abbiamo appena descritto.

