

2^ Scuola estiva di Formazione per i Docenti del Primo Ciclo di istruzione
«Insegnare Matematica: inclusione e cooperazione»
Pizzoferrato (CH) – 22, 23, 24 e 25 luglio 2018

Logica e pensiero computazionale



A cura del Dott. Fabio Manuppella (www.fabiomanuppella.it)

Mi presento



Laureato in Informatica Applicata presso l'Università degli Studi «Carlo Bo» di Urbino, da oltre 10 anni lavoro come consulente informatico nel settore delle tecnologie dell'informazione.

Esperto nella gestione di piattaforme e-learning e iscritto all'Albo Regionale dei Formatori PNSD dell'USR Abruzzo, progetto e realizzo Learning Objects destinati alla pubblicazione sul Web e alla realizzazione di lezioni interattive compatibili con qualsiasi modello di LIM.

Realizzo e gestisco Siti Web per Privati e PA, conformi alle direttive sull'accessibilità e sull'usabilità del Web.

Mi occupo anche di Editoria 2.0; collaboro con Enti di Formazione certificati dal MIUR (ai sensi della Direttiva 170/2016) e sono responsabile dell'impaginazione delle riviste scientifiche «HoPuE», «Ratio Mathematica» e «Science & Philosophy» e della realizzazione dei quaderni operativi dell'APAV pubblicati sul sito www.eiris.it.

Sono, infine, proprietario del sito web www.fabiomanuppella.it attraverso il quale promuovo la mia attività.



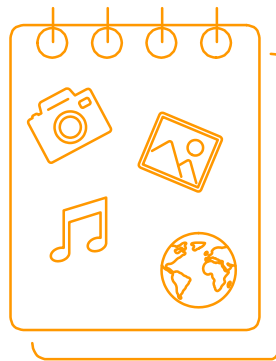
Di cosa parleremo

■ Logica, pensiero computazionale e Coding

- ▷ Le conoscenze preliminari (algoritmi e loro rappresentazione)
- ▷ L'approccio ludico alla programmazione
- ▷ Perché è così importante fare Coding a Scuola

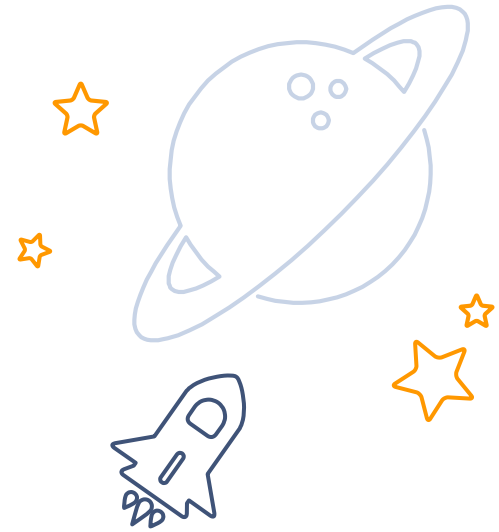
■ Strumenti visuali per il Coding: Scratch e Scratch JR

- ▷ Cosa è Scratch e come funziona
- ▷ Coding a Scuola: l'esperienza di Palmoli
- ▷ Laboratorio pratico



Non può esserci uso consapevole della tecnologia senza pensiero computazionale, una capacità da coltivare e applicare in modo interdisciplinare.

Per questo il Coding diventa fondamentale ed è un'attività in cui l'Italia primeggia, grazie ai moltissimi insegnanti che si sono messi in gioco.





Logica e pensiero computazionale



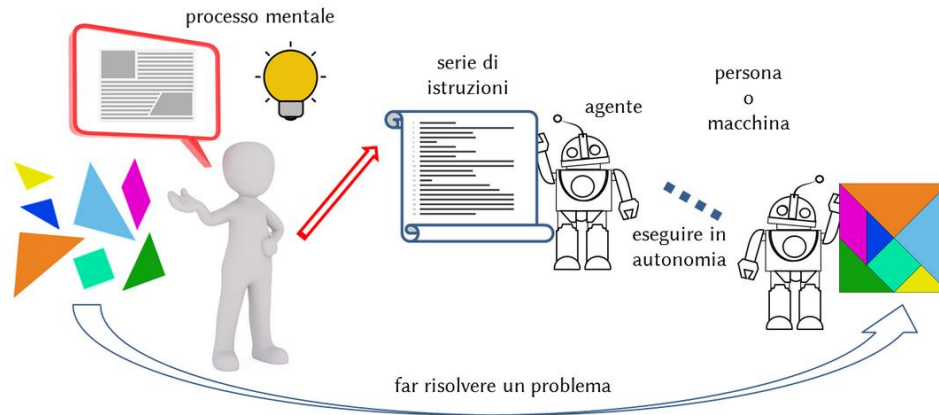
La vita di tutti i giorni ci pone di fronte a una serie di scelte, spesso difficili. Siamo chiamati a risolvere problemi di varia natura e ad affrontare gli inevitabili imprevisti.

Ogni volta che iniziamo a riflettere su cosa fare per risolvere un dato problema mettiamo in atto, in modo del tutto automatico, il **pensiero computazionale**, ovvero il processo mentale per la risoluzione di problemi, costituito dalla combinazione di metodi caratteristici e di strumenti intellettuali, che hanno tutti valore generale.



Logica e pensiero computazionale

Pensare in modo computazionale significa **suddividere il processo decisionale in singoli step**, ragionare passo-passo sul modo migliore per raggiungere un obiettivo. Un comportamento che in realtà – quasi senza accorgercene – mettiamo in atto tutti i giorni, per esempio quando stabiliamo il percorso più breve per raggiungere una destinazione oppure, più semplicemente, quando giochiamo ai videogiochi e dobbiamo elaborare un piano per superare un livello.





Logica e pensiero computazionale

Il pensiero computazionale serve a trasformare una intuizione in un procedimento costruttivo che ci porti alla soluzione di un problema.

Non è un concetto esclusivamente informatico, come si potrebbe pensare, ma molto più ampio: è un'abilità (come scrivere, leggere e fare calcoli) che permette di acquisire elasticità mentale e capacità nel risolvere problemi.

Quando dal pensiero si passa all'azione, ovvero all'esecuzione vera e propria delle azioni cui si era pensato, si parla di **Coding**, oggi spesso indicato come la **palestra del pensiero computazionale**.



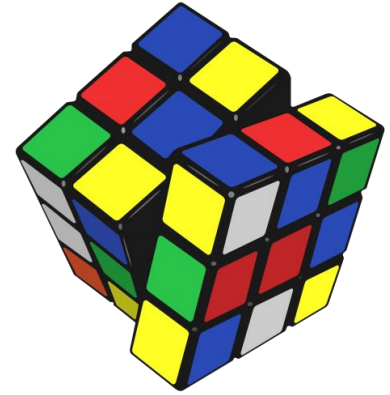


Logica e pensiero computazionale

I benefici del “pensiero computazionale” si estendono a tutte le professioni. Medici, avvocati, dirigenti di azienda, architetti, funzionari di amministrazioni – solo per citarne alcune – ogni giorno devono affrontare problemi complessi; ipotizzare soluzioni che prevedono più fasi e la collaborazione con altri colleghi o collaboratori; immaginare una descrizione chiara di cosa fare e quando farlo.

Il segreto sta tutto nel metodo: **poca teoria e tanta pratica.**

L'obiettivo non è formare una generazione di futuri programmatori, ma educare i più piccoli al pensiero computazionale, che è la capacità di risolvere problemi – anche complessi – applicando la logica, ragionando passo-passo sulla strategia migliore per arrivare alla soluzione.



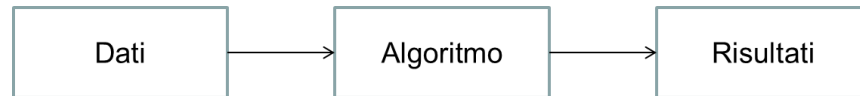


Le conoscenze preliminari

Gli algoritmi e la loro rappresentazione

Nelle scuole primarie e secondarie di primo grado tutti i docenti possono affrontare nelle loro classi l'attività di Coding. Vi è, però, la necessità di alcune conoscenze preliminari per affrontare in maniera corretta l'argomento e non trovarsi, successivamente, in difficoltà.

La teoria degli algoritmi, la pseudo-codifica e i diagrammi a blocchi sono un ottimo punto di partenza. Riferendoci ad un computer, **un algoritmo non è altro che una sequenza di istruzioni**. Esso comunica con l'esterno acquisendo dati e restituendo i risultati della computazione.





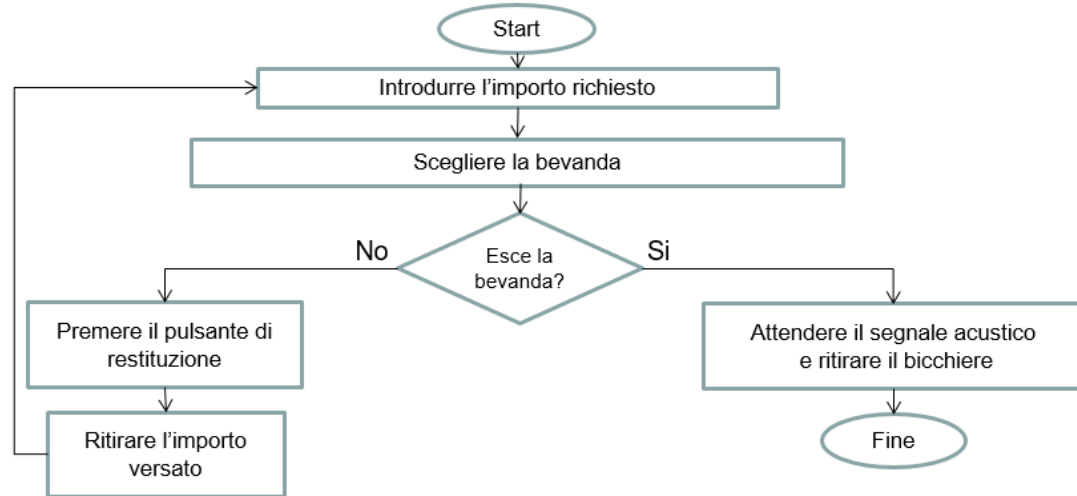
La rappresentazione degli algoritmi

Un esempio pratico: la macchinetta del caffè

Pseudo-codifica

1. Introdurre l'importo richiesto
2. Premere il pulsante di scelta della bevanda
3. Se esce la bevanda, attendere il segnale acustico e ritirare il bicchiere
4. Altrimenti, premere il pulsante per la restituzione e ritirare l'importo versato.

Diagramma a blocchi





Pseudo-codifica VS Diagramma a blocchi

Come si evince dall'esempio, **la rappresentazione a blocchi fornisce una descrizione molto più accurata** dell'algoritmo. Infatti, osservando anche solo per un attimo lo schema, è possibile capire che è presente un blocco di inizio (blocco ovale), cinque istruzioni di azione (blocchi rettangolari), una condizione (blocco a forma di rombo) e un blocco di fine (blocco ovale).

La pseudo-codifica, invece, si avvicina di più al linguaggio parlato, ma va letta (tutta!) per potersi fare un'idea almeno di massima su cosa fa l'algoritmo in questione.

Dopo aver rappresentato l'algoritmo da realizzare, si passa alla fase di coding e, quindi, ad impartire le varie istruzioni al calcolatore.



Il Coding a Scuola NON è una novità

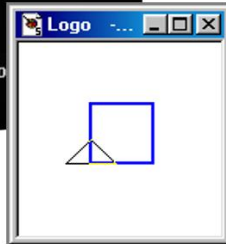
Facendo un veloce excursus storico, dobbiamo dire che nelle “scuole elementari” e “medie” del tempo (ovviamente le più avanzate) si lavorava soprattutto con due linguaggi “base”; il **Logo per la scuola elementare** ed il **Basic per la scuola media**, mentre si usava il **Pascal nelle scuole superiori**. Ma, tranne qualche raro caso di successo, l'introduzione della programmazione si è rivelata un fallimento, soprattutto perché si tendeva ad insegnare “la programmazione”, mentre non deve e non può essere questo lo scopo dell'attività di Coding.

```
C:\Temp>test.bat
I have 4 books you can borrow.
Pick one:

0> Avatar
1> The Green Mile
2> The Count of Monte Cristo
3> Babo

2

You picked The Count of Mo
C:\Temp>
```

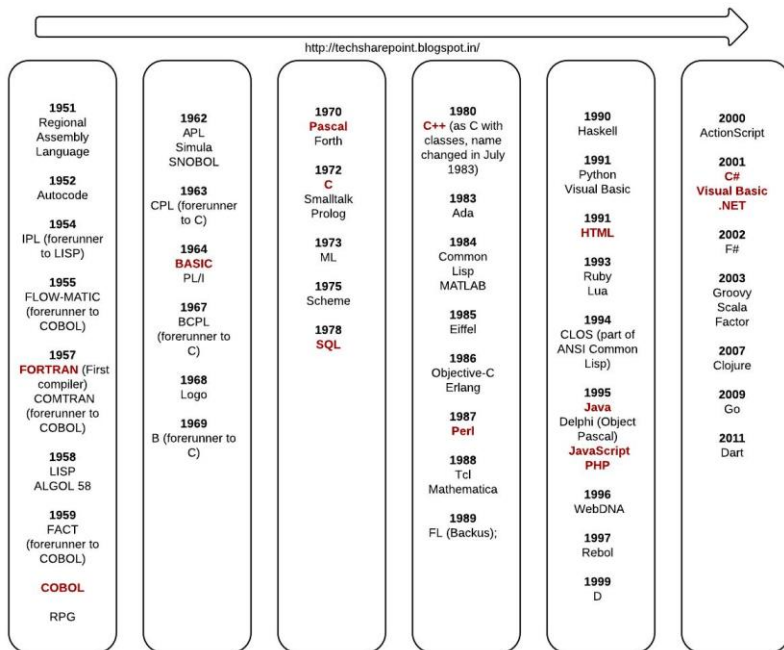


Dopo ore ed ore di impegno dei docenti e degli studenti, si riusciva a realizzare il solito **banale programmino** che prevedeva l'immissione di dati e una loro elaborazione con stampa a video dei risultati su enigmatiche ed “interessantissime” schermate a sfondo nero o verde con caratteri bianchi (i monitor di una volta non erano a colori).



Il Coding a Scuola NON è una novità

A queste considerazioni va aggiunto che i linguaggi di programmazione hanno, in genere, una **rapidissima evoluzione** e ciò che oggi è stato imparato con tanta fatica e tanto dispendio di energie, si rivela essere domani di nessuna utilità perché obsoleto.



**È necessario, quindi,
un cambio di approccio.**



L'approccio ludico alla programmazione

La scelta di un approccio ludico è strategica poiché consente di **attirare l'attenzione dei più piccoli** che, pensando di giocare, cercano di risolvere un problema più o meno complesso per la realizzazione di un videogioco o di una fiaba illustrata e sonorizzata, scrivendo una serie di istruzioni che la macchina interpreta ed esegue.

Usando gli attuali software a disposizione, **in brevissimo tempo (anche poco più di un'ora) si può creare un piccolo videogioco.**

E, contrariamente a quanto accadeva negli anni scorsi, **il risultato di questo lavoro è**, in genere, per gli alunni, **pieno di soddisfazioni**: ricordiamo che essi non si accontenteranno di un semplice triangolino bianco su fondo verde che ha bisogno di un gran numero di istruzioni per disegnare a video un semplice quadrato (Logo).

Il videogioco realizzato deve essere "bello", possibilmente deve "suonare e parlare" e deve essere "impegnativo" per l'utente finale, altrimenti il giocatore "si stufa subito". Questa è la situazione reale in cui ci troviamo quando affrontiamo la tematica nelle nostre classi.



L'approccio ludico alla programmazione

I ragazzi via via maturano anche una presa di coscienza, iniziano, quindi, a vedere le cose da una prospettiva diversa. Nella realizzazione del prodotto finito verrà profuso un impegno sempre più stringente nel passare dalle prime versioni di prova a versioni sempre più avanzate e complesse. E che dire, poi, del **lavoro cooperativo** che si innesca nei gruppi che il docente avrà avuto cura di creare? Quale gruppo avrà realizzato il videogioco più interessante? Ma non ci limitiamo solo alla realizzazione di videogiochi; si può realizzare al computer anche una lezione interattiva sulla spiegazione della struttura della cellula o la dimostrazione del Teorema di Pitagora o qualsiasi altra cosa possa venire in mente!

Per fare tutto ciò servono gli strumenti adatti

Esistono ormai diversi software che permettono la realizzazione di programmi più o meno complessi. Tra questi, quello che ha avuto più successo nelle scuole del primo ciclo di istruzione è **Scratch**.

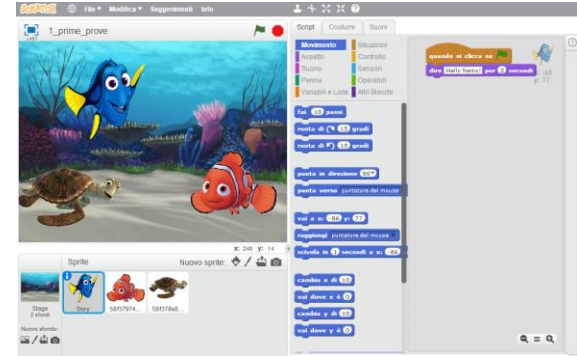
Strumenti visuali per il Coding: Scratch e Scratch JR





Cosa è Scratch e come funziona

Scratch è uno strumento di programmazione visuale ideato al MIT (Massachusetts Institute of Technology), che permette di creare giochi e animazioni senza dover scrivere una sola riga di codice. Rende semplice e divertente la creazione di storie interattive, giochi e animazioni. È un software completamente gratuito e disponibile sia nella versione installabile sul PC, sia nella versione on-line.



Ne esiste persino una versione "junior" per chi ancora non sa leggere. Scratch Junior è una app per il coding gratuita che si può scaricare e installare su tablet Android e su iPad. Nel primo caso è sufficiente avviare Google Play, cercare l'app e scaricarla. È necessario avere un tablet con display di almeno 7 pollici e Android 4.2 (Jelly Bean) o successivi. Nel secondo caso la stessa procedura va fatta dopo essersi connessi a iTunes.



Cosa è Scratch e come funziona

L'idea alla base del software è quella di mettere a disposizione dell'utente una serie di mattoncini colorati suddivisi in base alla loro funzionalità. Per realizzare una vera e propria animazione basta sfogliare il catalogo dei mattoncini, scegliere uno ad uno quelli utili al progetto e collegarli tra loro, incastrandoli attraverso gli appositi intagli, come si fa con i Lego o con un semplice puzzle.

A ciascun mattoncino corrisponde un'azione, una linea di codice che non ha bisogno quindi di essere digitato. Basta muovere o assemblare i mattoncini tra di loro – nell'ordine giusto – e il gioco è fatto.

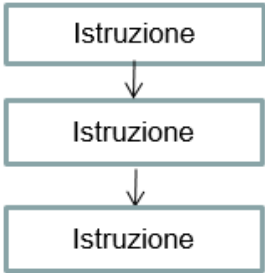
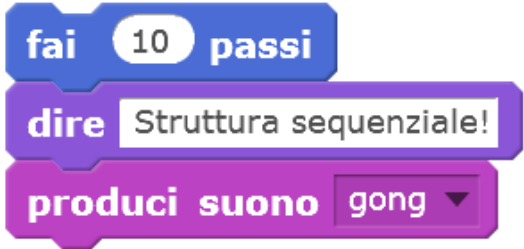
Facile a dirsi ... ma vedremo durante il laboratorio che non è sempre semplice ottenere il risultato voluto!





Tornando alla rappresentazione degli algoritmi...

La logica dei mattoncini di Scratch ricorda tanto quella dei Diagrammi a blocchi

<i>Struttura sequenziale</i>	
<i>Diagramma a blocchi</i>	<i>Scratch</i>
 <pre>graph TD; A[Istruzione] --> B[Istruzione]; B --> C[Istruzione];</pre>	



Tornando alla rappresentazione degli algoritmi...

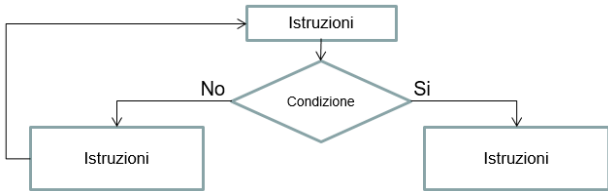
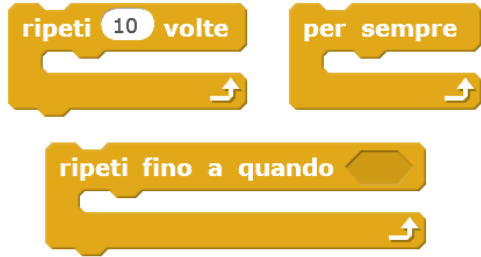
La logica dei mattoncini di Scratch ricorda tanto quella dei Diagrammi a blocchi

<i>Struttura di selezione</i>	
<i>Diagramma a blocchi</i>	<i>Scratch</i>
	



Tornando alla rappresentazione degli algoritmi...

La logica dei mattoncini di Scratch ricorda tanto quella dei Diagrammi a blocchi

<i>Struttura di iterazione</i>	
<i>Diagramma a blocchi</i>	<i>Scratch</i>
	



Coding a Scuola: l'esperienza di Palmoli

Il diario operativo del corso

Nell'ambito del progetto PON di inclusione e di lotta al disagio dal titolo "Gioco per contare di più" ho lavorato con i ragazzi delle classi seconda e terza media dell'Istituto Comprensivo Statale di Palmoli. Il diario operativo del corso è stato il seguente:

5/4/2018: presentazione di "Scratch" e primo approccio degli alunni con il software.

Dopo aver visitato il Sito Web ufficiale del progetto sono stati mostrati numerosi esempi di giochi e di animazioni; i ragazzi, animati da un forte interesse per questa nuova esperienza, hanno iniziato, guidati, a sperimentare e a realizzare animazioni semplici. È da notare il fatto che, al termine della lezione, alcuni studenti hanno pensato di documentare il loro lavoro con brevi riprese video da mostrare successivamente a genitori ed amici.



Coding a Scuola: l'esperienza di Palmoli

Il diario operativo del corso

12/4/2018: illustrazione delle funzionalità di base di Scratch e realizzazione di animazioni interattive e non.

In particolare, i ragazzi hanno imparato come si anima un personaggio, come cambiare lo sfondo dello stage in base a determinate condizioni logiche e come risparmiare tempo duplicando le istruzioni già impartite ad uno o più personaggi. Gli studenti hanno mostrato un crescente interesse verso il Coding ed hanno iniziato ad ottimizzare le animazioni realizzate, curandone i singoli dettagli.

19/4/2018: realizzazione di due animazioni complesse.

Sono stati introdotti diversi concetti importanti, come quello delle variabili, delle condizioni di arresto dei livelli. I ragazzi hanno iniziato a ragionare in modo autonomo, a porsi problemi e a proporre valide soluzioni.

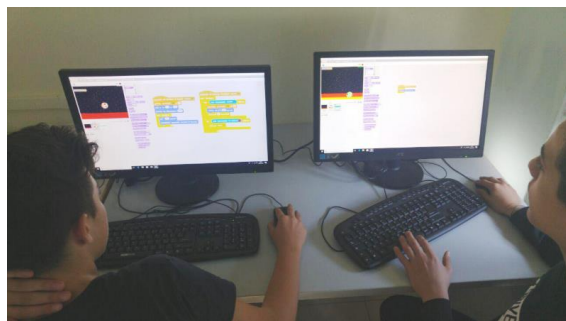
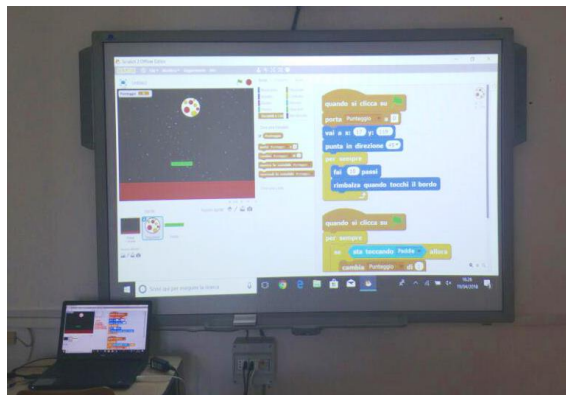
8/5/2018: realizzazione di due giochi interattivi.

La complessità degli stessi ha consentito il ripasso di tutte le conoscenze apprese durante il corso. I lavori realizzati dai ragazzi sono stati raccolti per la dimostrazione finale, che si è tenuta alla presenza dei genitori il 9/6/2018.



Coding a Scuola: l'esperienza di Palmoli

Progetti realizzati dai ragazzi sotto la guida del Docente



1. Controllo dei personaggi tramite la tastiera del computer
2. Galleria di immagini con effetti di transizione
3. Animazione delle lettere del proprio nome
4. Gioco della palla impazzita (rimbalzo sui bordi dello schermo)
5. Gioco della palla avvelenata (evitare che la palla tocchi terra)
6. Gioco «Acchiappami ... se ci riesci» (colpire i personaggi per accumulare punti)
7. Gioco «Attacca le astronavi nemiche»

Il corso realizzato ha avuto una durata di **10 ore** divise in 4 incontri di due ore e trenta minuti ciascuno.



Coding a Scuola: l'esperienza di Palmoli

Considerazioni al termine del corso

Osservando le date in cui si è svolto il corso, si evidenziano alcuni aspetti degni di nota:

- Durante la prima lezione i ragazzi hanno iniziato a sperimentare le funzionalità del programma, lavorando soprattutto **"di improvvisazione"**. Si sono, però, ben presto accorti che non basta collegare "a caso" i mattoncini per realizzare l'animazione desiderata. Volutamente il corso è stato realizzato come attività "ludica", senza inutili appesantimenti concettuali, irrealizzabili anche a causa della brevità del corso.
- Durante le prime tre lezioni, peraltro molto ravvicinate nel tempo, i ragazzi hanno realizzato uno o più progetti via via sempre più complessi, arrivando, durante l'ultima lezione a realizzare due giochi in cui è presente una discreta **interattività**.
- Fin da subito si è potuto notare un **atteggiamento positivo e propositivo** e un grande impegno nell'ottimizzazione e nella cura di ogni singolo dettaglio delle animazioni e dei giochi realizzati.



Adesso tocca a voi!

Ecco qualche idea:

Animazioni non interattive:

- ▷ Realizzare una presentazione di immagini (cambio dello sfondo ogni tot secondi)
- ▷ Realizzare un fumetto animato (due o più personaggi che dialogano tra loro)
- ▷ Realizzare un'animazione che disegni una figura geometrica (utilizzare i blocchi della categoria «penna»)

Animazioni interattive:

- ▷ Realizzare un'animazione che permetta il controllo del movimento di un personaggio tramite la tastiera
- ▷ Realizzare un'animazione in cui sia possibile modificare l'aspetto di un oggetto tramite tastiera (ad esempio, tasto A = ingrandimento, tasto S = rimpicciolimento, tasto D = cambiamento del costume)
- ▷ Realizzare l'animazione proposta nel tutorial "Catch game"

Potete condividere i lavori che fate o vedere quelli condivisi dagli altri corsisti: <https://scratch.mit.edu/studios/5241788/projects/>

Per testare la nuova versione di Scratch, attualmente in sviluppo:

<https://preview.scratch.mit.edu/>



Codice QR per Smartphone e Tablet



Sapevate che...

Proprio in questi giorni **Scratched** (<http://scratched.gse.harvard.edu/>), la community degli educatori dell'Università di Harvard ha lanciato il progetto «Getting Unstuck», ovvero «Sbloccati», dedicato agli insegnanti che desiderano cimentarsi nel mondo della programmazione visuale.

L'idea è semplice ma molto interessante: una volta iscritti all'iniziativa si riceve, quotidianamente una email che propone una piccola **sfida di programmazione creativa**, progettata per aiutare l'insegnante ad esplorare le funzionalità di Scratch e a sperimentare le sue innumerevoli potenzialità. È possibile condividere i progetti realizzati e confrontarli con quelli prodotti dagli altri partecipanti. Al progetto hanno preso parte più di 1500 persone e il numero delle iscrizioni è in crescita.

Il progetto è partito il 5 luglio 2018 e terminerà il 25 dello stesso mese. Chi volesse prenderne parte può iscriversi, gratuitamente, in qualsiasi momento e avere accesso non solo alla sfide future ma anche a quelle trascorse.

Per maggiori informazioni vi invito a visitare il sito web del progetto: <https://gettingunstuck.gse.harvard.edu/>
Al link seguente è pubblicato l'archivio delle sfide di programmazione creativa, accessibile anche ai non iscritti:
<https://us18.campaign-archive.com/home/?u=5dd4d8b32f719a30cbde507c6&id=cc0b423621>



Grazie per l'attenzione

Restiamo in contatto!



0854210779 - 3389948384



fabio.manuppella@gmail.com



www.fabio.manuppella.it



fabio.manuppella

Accessibilità



Editoria 2.0



Formazione (anche con Carta del Docente)



Multimedia



Social Media Marketing



Sviluppo Siti Web per PA e Privati

