4

LEZIONE:

Programmazione su Carta a Quadretti

Tempo della lezione: 45-60 Minuti. Tempo di preparazione: 10 Minuti. (revisione 2016)

Obiettivo Principale:
Aiutare gli studenti a capire cos'è la programmazione

SOMMARIO:
Lo studente impara a capire cos'è davvero la programmazione dando istruzioni ad un altro studente affinché faccia un disegno. L'obiettivo è far sì che uno studente scriva un programma seguendo il quale un altro studente possa colorare le caselle di un foglio di carta a quadretti in modo da riprodurre un disegno esistente. Se c'è tempo, nella lezione si può affrontare la creazione da parte degli studenti di disegni originali.

OBIETTIVI:
Gli studenti:
- Comprendono la difficoltà di tradurre problemi reali in programmi.
- Imparano che idee che sembrano chiare, possono comunque venir interpretate in modo scorretto da un calcolatore.
- Capiscono la necessità di strutture di programmazione formali come ripetizioni e funzioni.

MATERIALI:
- Kit per Disegni/Algoritmi di esempio (da pag. 8 in avanti).
- Scheda dei Simboli per la Programmazione (a pag. 3).
- Carta a quadretti grandi.
- Pennarelli, penne, o matite (due o tre colori).

PREPARAZIONE:
Stampa un kit per Disegni/Algoritmi per ogni gruppo.

Stampa una Scheda dei Simboli per la Programmazione per ogni gruppo.

Fornisci ad ogni gruppo un po' di fogli per i disegni.

VOCABOLARIO:
Algoritmo – Una serie di passi che descrivono come portare a termine un compito.

Programmazione – Espressione di un algoritmo mediante una sequenza di istruzioni in un formato eseguibile da un calcolatore.

Debug – Attività di correzione degli errori (bug).

Funzione – Una parte di codice alla quale è associato un nome e che può essere chiamata più volte.

Parametro – Informazione aggiuntiva che può essere passata ad una funzione per personalizzarne il comportamento.

RIPASSO:
Questa sezione di ripasso ha lo scopo di far ricordare agli studenti la precedente lezione. Se stai svolgendo le attività in un ordine diverso, sostituiscila con il ripasso degli argomenti svolti.

Le domande che seguono suppongono che questa sia la lezione 4 del percorso completo (interattivo + senza rete).

Domande per la discussione di classe:

- Riuscite a ricordare qualche fase del pensiero computazionale
- Riuscite a ricordare qualcuna delle regolarità che abbiamo trovato tra i mostri della scorsa lezione?

Domande per la discussione tra compagni di banco:

- Che cos'altro potremmo descrivere con gli stessi concetti "astratti" della scorsa lezione? Potremmo descrivere una mucca? Un uccello? Dovremmo cambiare qualcosa per descrivere una teiera?

Lo studente impara a capire cos'è davvero la programmazione dando istruzioni ad un altro studente affinché faccia un disegno.

INTRODUCI:

Inizia chiedendo alla classe se qualcuno ha mai sentito parlare di robotica. Cosa è un robot? Un robot "capisce" veramente quello che le persone dicono? La risposta a quest'ultima domanda è:

"Non allo stesso modo di una persona."

I robot funzionano seguendo "istruzioni", cioè specifiche azioni che sono stati predisposti a compiere. Per riuscire a completare un compito, un robot ha bisogno di avere una precisa sequenza di istruzioni (a volte chiamata "programma") da poter eseguire. Un programma è l'espressione di un "algoritmo" in un formato eseguibile da un calcolatore. Per prendere dimestichezza con i concetti di programma e algoritmo, può essere utile avere un esempio. Per questo esercizio, useremo un linguaggio di programmazione fatto di linee e frecce.

SIMBOLI PER LA PROGRAMMAZIONE

 — Move One Square Forward

 — Move One Square Backward

 — Move One Square Up

 — Move One Square Down

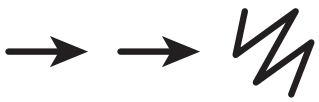
 — Change to Next Color

 — Fill-In Square with Color

In questo caso, i simboli a sinistra sono le istruzioni del "programma" e le parole a destra sono la relativa porzione di "algoritmo". Ciò significa che potremmo scrivere l'algoritmo:

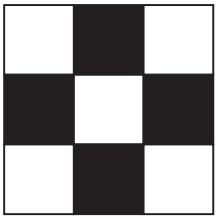
"Vai avanti di una casella, Vai avanti di una casella, Riempi la casella con il colore"

che corrisponde al programma:



È ora di fare un po' di pratica. Inizia la tua lezione nel mondo della programmazione disegnando o proiettando sulla lavagna una figura di riferimento.

Scegli un disegno semplice, come questo, da usare come esempio.

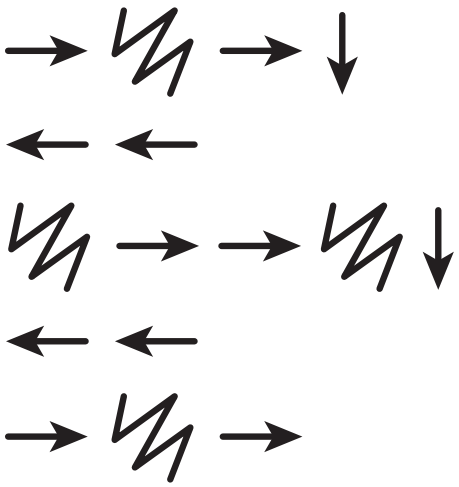


Si tratta di un buon modo per introdurre tutti i simboli della legenda. Per cominciare, puoi far vedere il modo più facile per codificare un'immagine, cioè quello di ritornare all'inizio della riga ogni volta che si passa alla riga successiva.
Completa la figura per la classe, quindi chiedi di aiutarti a descrivere quello che hai appena fatto. Per prima cosa è possibile descrivere un algoritmo a parole, quindi potrete programmare l'algoritmo ottenuto.

Un algoritmo di esempio:

"Avanti, riempi, avanti, riga successiva,
indietro, indietro,
riempi, avanti, avanti, riempi, riga successiva,
indietro, indietro,
avanti, riempi, avanti".

Alcuni studenti potrebbero accorgersi che ci sono dei passi inutili ma falli aspettare fino alla fase di programmazione. Guida la classe nella programmazione dell'immagine:

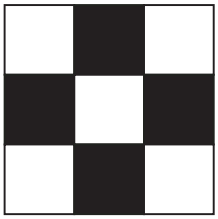


A questo punto, gli studenti potrebbero cercare di offrire suggerimenti. Se gli studenti hanno chiaro lo scopo dell'esercizio, potrebbe essere un buon momento per dare loro un'imbeccata.

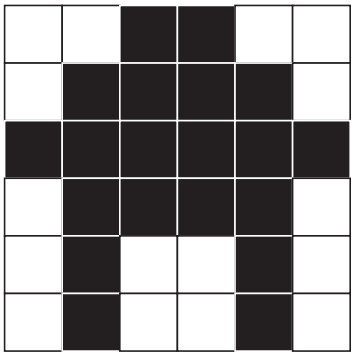
È vero che c'è un po' di ridondanza in quello che abbiamo fatto, ma può essere molto meno chiaro programmare cercando di evitarla. A volte, finché un programmatore non diventa esperto, può essere utile programmare nel modo più comprensibile possibile per assicurarsi che tutto funzioni e rimuovere i passi inutili in un secondo momento.

Ripeti nuovamente l'esempio, prima rimuovendo i passi non necessari dal programma originale e poi programmando da zero senza utilizzare simboli non necessari.

Se la classe sembra persa o confusa prova ad usare un'altra immagine ma salta la fase di "semplificazione". Continua a procedere da sinistra a destra e poi di nuovo indietro.



Se la classe riesce a "spiegare a voce alta" l'algoritmo, e poi a definire i simboli corretti per ogni passo, allora è pronta ad andare avanti. A seconda dei tuoi studenti e della loro età potresti provare a fare un esempio più complicato insieme a loro o farli lavorare subito in gruppi.

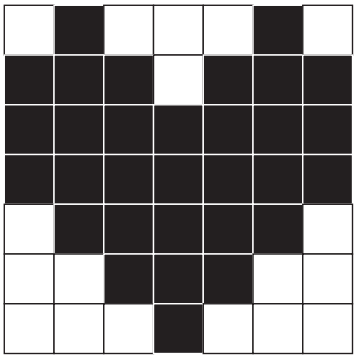


Dai ad ogni gruppo un blocco di immagini e suggerisci loro di scegliere una delle immagini "piccole con un solo colore". Per prima cosa fai scrivere l'algoritmo, quindi fallo convertire in simboli. Dopo che l'hanno fatto per una o due volte, fai scambiare gli algoritmi con un altro gruppo e fai disegnare quello che gli altri hanno programmato.

Se c'è tempo, è possibile spiegare la necessità di funzioni e parametri: ovviamente questo si vede meglio con disegni più grandi e complicati.

Dopo averne disegnato qualcuno di questi più complessi, sarà chiaro che un'immagine di questa dimensione è abbastanza impegnativa. Consideriamo solo due linee di quest'immagine.

Algoritmo:



" Avanti, riempi, avanti, avanti, avanti, avanti, riempi, avanti, riga successiva.
indietro, indietro, indietro, indietro, indietro, indietro.
riempi, avanti, riempi, avanti, riempi, avanti, avanti, riempi, avanti, riempi, avanti, riempi, riga successiva.
indietro, indietro, indietro, indietro, indietro, indietro".

Sfida la classe a cercare qualcosa che si ripete spesso. Accetta dei suggerimenti, ma sicuramente una delle cose che è più utile combinare è "indietro, indietro, indietro, indietro, indietro, indietro". Resta aperto a suggerimenti su come trasformare queste istruzioni in un unico simbolo.

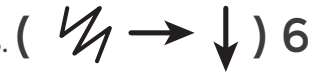
La classe potrebbe arrivare a qualcosa di questo tipo:

Infatti, potrebbero anche aver già fatto qualcosa del genere con l'ultimo gruppo di immagini. Ci sono altre combinazioni che possono essere fatte? Come saltare tre caselle di fila? Colorare tre caselle di fila? Colorare sette caselle di fila?

Probabilmente finirai per avere tanti simboli che contengono vari numeri. Se tutto va bene, a questo punto la classe dovrebbe aver capito cosa sono le funzioni e perché sono utili.

E ora arriva la magia! Puoi rivelare alla classe che hanno appena scoperto le funzioni! Hanno creato una rappresentazione semplice per un raggruppamento complesso di azioni. Ciò è esattamente quello per cui sono fatte le funzioni. Ed il numero? Anche questo ha un nome. Quel numero è chiamato parametro. Nel caso dell'esempio precedente, il parametro permette alla funzione di sapere quante volte muoversi all'indietro.

Osserva queste funzioni. Cosa pensi che facciano?

1. 
2. 
3. 

Risposte:
1. Avanza di sei caselle.
2. Colora sei caselle consecutive.
3. Colora una linea diagonale.

Ci sono altre combinazioni che possono essere utili?

Avendo a disposizione questo nuovo metodo, sfida la classe a scegliere una delle immagini più difficili che possano gestire. Questi simboli aiutano a procedere più velocemente?

Quando i gruppi hanno finito il loro compito, fai scambiare il programma di ogni gruppo (completo di funzioni e parametri) con quello di un altro gruppo, e fai loro provare a disegnare quello che gli altri hanno programmato.

ADATTAMENTI:

Per studenti di prima e seconda elementare: Fai lavorare gli studenti tutti insieme ad un'unica immagine/algoritmo. Usa della carta a quadretti più grandi. Usa solo immagini con un unico colore.

Per studenti di terza, quarta e quinta elementare: Lavora in piccoli gruppi (2-4). Usa griglie più grandi per ridurre il tempo di lavoro.

Per studenti delle scuole medie: Lavora in coppie o singolarmente.

FASI:

1) Scegli un'immagine dall'insieme.
2) Scrivi l'algoritmo per disegnare quell'immagine.
3) Converti l'algoritmo in un programma utilizzando i simboli.
4) Scambia i programmi con un altro gruppo e disegna la loro immagine.
5) Aggiungi delle "funzioni" per rendere più semplici i programmi.
6) Scrivi dei programmi per immagini più complesse.
7) Scambia i tuoi programmi complessi e disegna nuovamente.