CODING

Possibile percorso da seguire con i bambini delle classi Prima, Seconda e Terza.

Durante le prime lezioni, “istruire” i bambini su come utilizzare in modo consapevole, e il più possibile autonomo, l’aula di informatica (far sperimentare loro che SOLO l’esatto ordine procedurale li porterà ai giochi proposti):

- **Prendere il Pc dal carrello**
- **attaccare il cavo di rete e il mouse**
- **accendere il Pc e inserire la password**
- **accedere a //www.programmailfuturo.it/**
- **nella barra in alto, cliccare su “lezioni tecnologiche” (5 Corsi, suddivisi in una serie di esercizi progressivi)**
- **corso 1**
- **lezione 3**

Di seguito quanto estrapolato, e sperimentato efficacemente, dal sito Programma il futuro.

Il **Corso 1** è progettato per essere utilizzato fin dalla **prima elementare**, ma per la sua semplicità può essere usato con successo come corso introduttivo anche per studenti più grandi. Consente agli studenti di entrare nei meccanismi del **pensiero computazionale** con uno sforzo iniziale molto basso e di procedere in maniera molto graduale, sviluppando progressivamente capacità di risolvere problemi e di perseverare nella ricerca di soluzioni. Alla fine del corso gli studenti creano i loro giochi o le loro storie.

Il corso è il naturale punto di partenza per studenti che stanno iniziando a leggere dal momento che i blocchi usati per la costruzione dei programmi sono corredati da simboli e immagini con uso minimale di testo.

Per renderne la fruizione più agevole anche da parte dei bambini che non riescono a leggere speditamente, è disponibile una funzione che permette di sentire le istruzioni.

I concetti fondamentali coperti dal corso sono quello di **sequenza** di istruzioni e di **ripetizione** di istruzioni.

Corso 1 - Lezione 3: classe prima

Puzzle: Impara come trascinare e rilasciare (drag and drop)

In questa prima lezione interattiva gli studenti apprendono ad usare il mouse e acquisiscono familiarità con l'interfaccia - basata su blocchi - che useranno in tutte le lezioni. Iniziano spostando immagini sullo schermo e progrediscono mettendo in ordine i pezzi di un puzzle.

L'ambiente di apprendimento è semplice da usare, poiché interamente basato sull'approccio denominato "trascina e rilascia" (*drag and drop*). I comandi disponibili sono i blocchi colorati, vengono presi cliccandoci sopra con il mouse e trascinandoli (mentre il mouse rimane premuto) nella zona di destra, dove vengono combinati per costruire le soluzioni ai vari esercizi.

La forma dei blocchi è importante: a seconda di essa e di come vengono combinati, cambia il significato dei programmi che vengono costruiti.

Concetti di base

- L'approccio **trascina e rilascia**, utilizzato per spostare i blocchi e concatenarli tra loro
- Il concetto di **ordine**, con cui sono uniti i blocchi, che cambia il significato del programma nel suo complesso (in questo caso cambia l'immagine dei puzzle che si stanno costruendo, in generale cambia l'ordine delle azioni che i personaggi compiranno)
- I **blocchi a forma di C**, ovvero blocchi che possono contenere al loro interno altri blocchi

Obiettivi della lezione

- Usare il mouse per comunicare informazioni (es. dove muovere un certo blocco) ad un computer
- Concatenare pezzi di un puzzle nel giusto ordine

Corso 1 - Lezione 4: classe prima

Labirinto: Sequenze

Negli esercizi di questa lezione gli studenti sviluppano algoritmi basati su sequenze per spostare un uccellino di Angry Birds da un lato del labirinto verso il maialino dall'altra parte.

Concetti di base

- la **sequenza** (ovvero svolgere un'azione dopo l'altra), realizzata connettendo i blocchi uno sotto l'altro

Obiettivi della lezione

- Esprimere i movimenti tramite una serie di istruzioni
- Impartire le istruzioni di movimento tramite una serie di passi sequenziali
- Contare il numero di volte che un'azione deve essere eseguita e rappresentarla tramite istruzioni di un programma

Corso 1 - Lezione 5: classi prima e seconda

Labirinto: Correzione di Errori

La correzione degli errori è un elemento fondamentale dell'apprendimento della programmazione. In questa lezione gli studenti si trovano di fronte esercizi che sono stati svolti in modo non corretto. Devono quindi esaminare il codice esistente per identificare gli errori, che possono essere costituiti da blocchi mancanti, blocchi superflui o blocchi in ordine sbagliato.

I programmi di reale interesse descrivono procedure molto complesse che vengono affidate ad un calcolatore, cioè un esecutore automatico privo di qualunque intelligenza. Questo esecutore farà esattamente ciò che noi gli diremo di fare, errori compresi!

Anche il più bravo programmatore commette errori e ha bisogno di provare il proprio codice per verificare che funzioni e che faccia esattamente ciò per cui è stato scritto. Se qualcosa non va bisogna saperne individuare la causa e correggerla. Questa fase di verifica e correzione degli errori si chiama in inglese “*debugging* ”. L’origine del nome, che deriva dall’inglese “*bug* ”, risale agli anni ’40 del secolo scorso, quando fu proprio un insetto (“*bug* ”, per l’appunto) a produrre, causando un corto circuito, il malfunzionamento di uno dei primi calcolatori.

Per familiarizzare con la **correzione degli errori**, in questa lezione vengono proposti esercizi già risolti, ma con errori che dovremo individuare e correggere.

Spesso gli studenti hanno la tendenza a non voler eseguire il programma fino a che non è corretto. Spesso, invece, il modo più semplice per capire che cosa non va in programma è proprio quello di "guardarlo sbagliare", dunque dobbiamo far capire agli studenti che non c'è nulla di male ad eseguire un programma prima di averlo sistemato.

Per aiutare gli studenti a correggere gli errori, si dà loro la possibilità di eseguire il programma "passo passo", cioè "un blocco alla volta", per vedere l'effetto di ciascuna istruzione e capire quali possono essere le cause del non corretto funzionamento.

Concetti di base

- la **ricerca e correzione di errori**
- l'**esecuzione passo passo** (eseguita tramite il pulsante **Fai un passo**)

Obiettivi della lezione

- Predire quando un programma commetterà un errore
- Modificare un programma esistente per correggere gli errori che contiene
- Capire che un algoritmo è sbagliato perché i passi non sono nell'ordine giusto
- Riflettere sulle diverse strategie di correzione degli errori

Corso 1 - Lezione 7:classi prima e seconda

Ape: Sequenze

In questa lezione non vengono introdotti nuovi concetti informatici, ma lo studente applica il concetto di sequenza alle azioni, oltre che ai movimenti - come aveva fatto nelle lezioni precedenti.

Concetti di base

- la **quantità**, cioè il numero di unità, in questo caso unità di nettare presenti nel fiore o di miele che un favo più ospitare

Obiettivi della lezione

- Esprimere i movimenti tramite una serie di istruzioni
- Impartire le istruzioni di movimento tramite una serie di passi sequenziali
- Contare il numero di volte che un'azione deve essere eseguita e rappresentarla tramite istruzioni di un programma
- Convertire una quantità espressa con un numero nella corrispondente sequenza di blocchi singoli

Corso 1 - Lezione 13: classe seconda

Labirinto: Cicli

In questa lezione gli studenti usano i **cicli**, cioè i blocchi di ripetizione, per imparare a ripetere un'istruzione o una sequenza di istruzioni in modo più efficiente (senza dover riscrivere le istruzioni, ovvero senza dover inserire copie identiche di blocchetti) e le usano per attraversare il labirinto.

Concetti di base

- la **ripetizione** - o **ciclo** - (ovvero decidere di ripetere un'istruzione o una sequenza di istruzioni), realizzata dal blocco **"ripeti ... volte"**

Obiettivi della lezione

- Identificare i benefici di usare i cicli al posto di una ripetizione manuale
- Creare un programma per un compito dato, in cui si ripete un singolo comando
- Spezzare una lunga sequenza di istruzioni nella sequenza ripetibile più piccola possibile

- Creare un programma per un compito dato, in cui si ripete una sequenza di comandi
- Utilizzare una combinazione di comandi sequenziali e comandi ripetuti tramite un ciclo, allo scopo di raggiungere la fine di un labirinto

Corso 1 - Lezione 14: classe seconda

Ape: Cicli

Nella precedente lezione gli studenti hanno imparato ad usare i cicli per ripetere spostamenti. In questa imparano ad usarli per ripetere azioni, in modo da aiutare l'ape a raccogliere più nettare ed a produrre più miele.

In questa lezione non vengono introdotti nuovi concetti, ma lo studente applica il concetto di ripetizione alle azioni, oltre che ai movimenti, come aveva già fatto nelle lezioni precedenti.

Obiettivi della lezione

- Creare un programma per un compito dato, in cui si ripete un singolo comando
- Identificare le situazioni in cui un ciclo può essere utilizzato per semplificare un'azione ripetitiva
- Utilizzare una combinazione di comandi sequenziali e comandi ripetuti tramite un ciclo per muovere un personaggio e compiere azioni.

Corso 1 - Lezione 16: classi seconda e terza

Laboratorio: Crea una Storia

In questa lezione gli studenti possono applicare tutte le competenze che hanno appreso per creare una storia animata.

In particolare gli studenti creeranno storie aniamte, facendo muovere e parlare personaggi sullo schermo.

In una storia ci sono dei personaggi, e quando accade qualcosa, questi personaggi reagiscono un un certo modo. Per esempio, quando il personaggio Cane incontra il personaggio Gatto, quest'ultimo dice qualcosa.

Il fatto che "*accade qualcosa* " viene chiamato **evento** e i programmi che realizzano giochi interattivi sono *programmi guidati dagli eventi*, cioè programmi in cui si fanno delle azioni in risposta al verificarsi di certi eventi. In questi programmi ci sono quindi dei blocchi del tipo ***quando** accade qualcosa* e a questi blocchi si attaccano i blocchi con le azioni desiderate. Si realizza così un *gestore dell'evento*, cioè un insieme di blocchi che fa reagire il programma a ciò che è accaduto.

Ovviamente si continuano ad usare gli stessi concetti che abbiamo visto in altre lezioni, ad esempio il concetto di *sequenza* (per decidere l'ordine in cui i personaggi parlano o si muovono).

Nei primi esercizi lo studente impara come costruire storie.
Nell'ultimo esercizio, può costruire una storia interattiva usando i blocchetti che ha imparato a conoscere, oltre ad avere la possibilità di scoprirne di nuovi.

Alla fine lo studente può condivididere la sua creazione con gli amici.
La storia che abbiamo realizzato a titolo di esempio si trova qua.

Concetti di base

- l'**evento**, cioè il fatto che accade qualcosa, indicato dai blocchi del tipo ***quando*** *accade qualcosa*;
- il **gestore di evento**, cioè un insieme di blocchi che fa reagire il programma a ciò che è accaduto, realizzato concatenando sotto ai blocchi di evento, le azioni volute

Obiettivi della lezione

- Identificare azioni che sono legate a eventi
- Creare una storia animata e interattiva usando sequenze e gestori di eventi
- Condividere un artefatto creativo da loro realizzato con gli altri studenti

Corso 2

Il Corso 2 è progettato per essere utilizzato da studenti (**da 6 anni in su**) che hanno già imparato a leggere e non hanno precedenti esperienze di programmazione, ma può essere fruito anche da studenti più grandi. Nel corso gli studenti creano programmi per risolvere problemi e sviluppare giochi interattivi o storie da condividere.

Il corso è quindi **raccomandato dalla seconda/terza elementare** (a seconda del livello di maturazione degli studenti) in avanti. I blocchi usati per la costruzione dei programmi contengono testo descrittivo e non più immagini o simboli, come nel Corso 1.

Il nuovo concetto fondamentale introdotto in questo corso è quello di **istruzione condizionale**. Mentre nel precedente Corso 1 tutti i programmi conducevano all'esecuzione di una stessa sequenza lineare di istruzioni, in questo corso gli studenti imparano a scrivere programmi che prendono decisioni e possono quindi eseguire differenti sequenze di istruzioni.

Corso 2 - Lezione 3: classe terza

Labirinto: Sequenze

Negli esercizi di questa lezione gli studenti sviluppano algoritmi basati su **sequenze** per spostare un uccellino di Angry Birds da un lato del labirinto verso il maialino dall'altra

parte. Gli studenti imparano ad attaccare insieme i blocchi che fanno muovere l'uccellino e lo fanno girare.

Concetti di base

- la **sequenza** (ovvero svolgere un'azione dopo l'altra), realizzata connettendo i blocchi uno sotto l'altro.

Obiettivi della lezione

- Esprimere i movimenti tramite una serie di istruzioni
- Impartire le istruzioni di movimento tramite una serie di passi sequenziali
- Contare il numero di volte che un'azione deve essere eseguita e rappresentarla tramite istruzioni di un programma

Corso 2 - Lezione 8:

Ape: Cicli

Nelle precedenti lezioni gli studenti hanno usato i cicli per ripetere spostamenti. In questa imparano a **ripetere azioni**, in modo da aiutare l'ape a raccogliere più nettare ed a produrre più miele.

In questa lezione non vengono introdotti nuovi concetti, ma lo studente applica il concetto di ripetizione alle azioni, oltre che ai movimenti, come aveva già fatto nelle lezioni precedenti.

Obiettivi della lezione

- Creare un programma per un compito dato, in cui si ripete un singolo comando
- Identificare le situazioni in cui un ciclo può essere utilizzato per semplificare un'azione ripetitiva
- Utilizzare una combinazione di comandi sequenziali e comandi ripetuti tramite un ciclo per muovere un personaggio e compiere azioni.

Corso 2 - Lezione 13:

Ape: Istruzioni Condizionali

Fino a questo momento tutti i programmi scritti dagli studenti vengono eseguiti ogni volta esattamente nello stesso modo - questo è molto affidabile ma non è molto flessibile.

In questa lezione si introduce l'istruzione condizionale, cioè un'istruzione che conduce all'esecuzione di differenti azioni in funzione delle condizioni cui si trova di fronte.

I computer infatti devono spesso prendere decisioni che non sono conosciute a priori dal programmatore mentre scrive il codice, perché dipendono da condizioni che si conoscono solo mentre il programma è in esecuzione (in inglese: *a runtime*). Dunque bisogna

prevedere le diverse situazioni che si possono presentare e scrivere un programma che tenga conto delle molteplici possibilità.

Concetti di base

- la **condizione booleana**, cioè una espressione che può essere solo vera o falsa
- l'**istruzione condizionale** (detta anche *selezione*, che permette di decidere di svolgere o meno un'azione solo *se* si verifica una certa *condizione*), realizzata dal blocco **se**

Obiettivi della lezione

- Confrontare valori usando l'operatore di uguaglianza (`=`)
- Tradurre frasi condizionali espresse nel linguaggio naturale in istruzioni di un programma
- Identificare quando è necessario usare un'istruzione condizionale per trattare valori non noti
- Risolvere esercizi usando sia sequenze ripetute sia condizionali