

CoderDojo @Campoformido

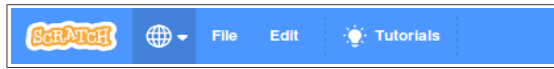
Sala polifunzionale A. Geatti

8 dicembre 2019



Prati stabili

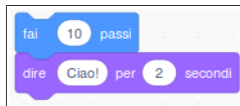
- avviare Scratch e impostare la lingua italiana selezionandola dal menu delle lingue:



- trascinare il blocco `fai (10) passi` nell'area di lavoro – non succede niente:



- click sul blocco: il gattino si sposta verso a destra;
- selezionare il gruppo *Aspetto*;
- trascinare il blocco `dire (Ciao!) per (2) secondi` e agganciarlo sotto al blocco `fai (10) passi` – notare l'area grigia che appare quando i due blocchi sono vicini: evidenzia il punto di aggancio:



- click su uno qualunque dei blocchi: il gattino si muove e saluta;
- sganciare il blocco `dire (Ciao!) per (2) secondi` trascinandolo lontano dal blocco `fai (10) passi`;



- agganciarlo sopra al blocco `fai (10) passi`;
- click su uno qualunque dei blocchi, il gatto saluta e si muove: **l'ordine dei comandi conta!**
- sganciare i due blocchi spostando il blocco `fai (10) passi` (si sgancia sempre da sotto, se si trascina il blocco in alto si ottiene lo spostamento dell'intera catena di blocchi);
- eliminare il blocco `dire (Ciao!) per (2) secondi` trascinandolo nell'area dei blocchi – non importa la posizione precisa;
- facciamo fare una passeggiata al gattino: selezioniamo il gruppo *Controllo* e trasciniamo il blocco `per sempre` nell'area di lavoro; avvicinandolo al blocco `fai (10) passi` si incastra circondandolo; non succede nulla:



- click sul blocco e il gattino si sposta fino al bordo destro dello *Stage*;
- lo *Script* è in esecuzione, lo si riconosce dall'alone giallo che contorna il codice; se trasciniamo il gattino al centro dello *Stage* con il mouse, riprende a spostarsi verso destra;
- per fermare un blocco `per sempre` bisogna premere il pulsante rosso in alto; notare che l'alone giallo sparisce;
- vicino al pulsante di arresto c'è una bandierina verde: serve ad avviare i giochi;

- selezioniamo il gruppo *Situazione* e agganciamo il blocco `quando si clicca su` sopra al blocco `per sempre`:



- riportiamo il gattino al centro dello *Stage*, e verificiamo che se ne sta fermo dove l'abbiamo lasciato; premiamo la bandierina ed esso riprende a muoversi;
- premere il pulsante d'arresto e riportare il gattino al centro della scena;
- vogliamo guidare il gattino con i tasti cursore, come fosse un'automobilina; per fare questo ricorriamo ai blocchi `ruota ~ di (15) gradi` e `ruota ^ di (15) gradi`;
- le rotazioni devono avvenire solo se sono premuti rispettivamente i tasti "freccia destra" e "freccia sinistra"; accedere al gruppo *Controllo* e portare un blocco `se <> allora` nell'area di lavoro, quindi inserire un blocco `ruota ~ di (15) gradi` al suo interno:



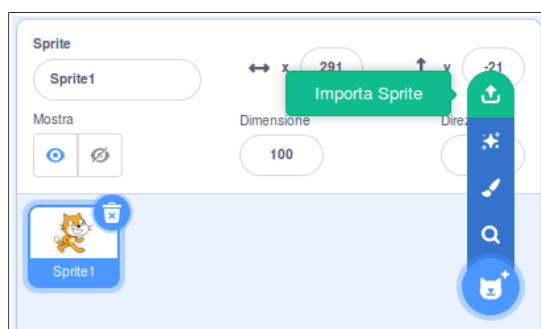
- notare lo spazio vuoto esagonale tra il *se* e l'*allora*: lì va indicata la condizione per la quale devono essere eseguiti i blocchi interni; le varie condizioni disponibili si trovano nel gruppo *Sensori*;
- selezionare il gruppo *Sensori* e trascinare il blocco `<tasto [spazio] premuto>` nello spazio, verificando che l'agganciamento sia avvenuto nel modo corretto (assicurarsi che il perimetro dell'esagono si evidenzi di bianco prima di rilasciare il blocco); aprire il menu a tendina e selezionare il tasto *freccia destra*;
- inserire il blocco `se <> allora` nel programma principale, prima del blocco `fai (10) passi`:



- premere la bandierina verde; il gattino comincia a muoversi verso destra; premere il tasto cursore a destra, ed esso curva verso destra! fare alcune prove, al termine premere il pulsante d'arresto;
- in modo analogo si prepari un blocco `se <> allora` per la rotazione sinistra:



- verificare che ora il controllo sul gattino è completo; provare a modificare i parametri dei comandi: che effetto hanno sulla mobilità del personaggio?
- poiché il tema del nostro gioco è un'ape, carichiamo lo sprite **bee.sprite3**: click sul pulsante *Scegli uno Sprite* in basso a destra, quindi *Importa Sprite*:



- il nuovo personaggio appare al centro della scena, e l'icona che lo rappresenta si aggiunge a quella di Scratch nell'area degli sprite:



- il programma che avevamo scritto è sparito! Niente paura, ogni sprite ha il suo programma: selezionare *Sprite1* ed ecco che il programma ricompare; trasciniamolo sopra l'icona dell'ape, in modo da copiarlo (attendere che l'icona si evidenzi di blu prima di rilasciare il codice); selezionare lo sprite dell'ape e verificare che il programma è stato copiato; a questo punto si può cancellare lo sprite di Scratch selezionandolo nell'area degli sprite e cliccando l'icona a forma di cestino;
- adattiamo il programma alle dimensioni del nuovo personaggio:



- l'ape ha due costumi: alternandoli, si ha l'impressione che essa svolazzi; applichiamo l'effetto al programma aggiungendo il blocco `passa al costume seguente` dopo il blocco `fai (3) passi`:

```

quando si clicca su
per sempre
se [tasto freccia destra] premuto allora
ruota di 5 gradi
se [tasto freccia sinistra] premuto allora
ruota di 5 gradi
fai 3 passi
passa al costume seguente
    
```

- all'inizio del gioco vogliamo che l'ape occupi una posizione centrale; notare che quando sposto lo sprite con il mouse le coordinate del blocco `vai a x: () y: ()` cambiano; posizionare l'ape al centro, e inserire questo blocco prima del `per sempre`:

```

quando si clicca su
vai a x: 0 y: 0
per sempre
se [tasto freccia destra] premuto allora
ruota di 5 gradi
se [tasto freccia sinistra] premuto allora
ruota di 5 gradi
fai 3 passi
passa al costume seguente
    
```

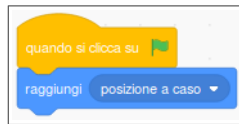
- è ora di salvare il gioco!

L'obiettivo

- scopo del gioco è raccogliere quanto più polline possibile prima che arrivi l'inverno; carichiamo allora lo sprite `sunflower.sprite3` che rappresenta l'obiettivo che l'ape deve raggiungere:



- all'inizio del gioco il girasole deve apparire in una posizione a caso:



- se avvio il gioco e guido l'ape sul girasole non succede nulla; facciamo in modo che il fiore sparisca quando viene toccato dall'ape; il sensore da usare è `sta toccando (puntatore del mouse)`, ricordandosi di sostituire *puntatore del mouse* con *bee*:



- abbiamo ottenuto l'effetto voluto! se però adesso riavvio il gioco, il fiore non riappare; per questo dobbiamo aggiungere un blocco `mostra` all'inizio, dopo il blocco `raggiungi (posizione a caso)`:



- per far riapparire il fiore devo cliccare sulla bandierina verde; perché non riappare automaticamente, magari dopo un breve lasso di tempo, da un'altra parte?



- aggiungiamo uno sfondo per dare un'ambientazione consona al gioco; premere il pulsante *Scegli uno Sfondo* in basso a destra e scegliere l'immagine preferita (es. **Forest**):



- si può giocare a schermo intero premendo il pulsante *Modalità presentazione* in alto a destra; premere il riquadro nella stessa posizione per tornare alla modalità di lavoro;
- è ora di aggiungere il punteggio: contiamo quanti sono i girasoli sui quali l'ape è riuscita a posarsi... selezionare *Variabili*, quindi *Crea una Variabile*, chiamarla "Punti" e confermare premendo *OK*; cosa succede?
- abbiamo detto che facciamo un punto ogni volta che l'ape si posa su un girasole; selezioniamo lo sprite del girasole ed aggiungiamo il blocco `cambia [my variable] di (1)` all'interno del corpo del blocco `se <sta toccando (bee)> allora`, avendo l'accortezza di sostituire la variabile "my variable" con la variabile "Punti":



- provare il gioco e verificare che il punteggio aumenta quando l'ape raggiunge il girasole;
- notate un problema? quando il gioco riparte, il punteggio non si azzer! la soluzione è aggiungere il blocco `porta [Punti] a [0]` all'inizio del programma:



L'antagonista

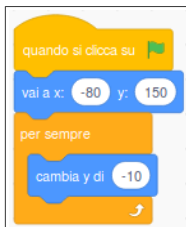
- rendiamo il gioco più difficile: poiché siamo in autunno, l'ape deve evitare le foglie che cadono dagli alberi; se tocca una foglia, perde un punto;
- carichiamo l'ultimo sprite, **leaf.sprite3**, e posizioniamolo in alto sullo *Stage*:



- come facciamo a far cadere la foglia? beh, sappiamo che si comincia con qualcosa di simile a:



- [brevi cenni al sistema di coordinate cartesiane...]
- per far spostare uno sprite verso il basso bisogna far diminuire la sua coordinata y; agganciamo perciò un blocco `cambia y di (-10)`; attenzione al meno, se ve lo scordate la foglia salirà in cielo!



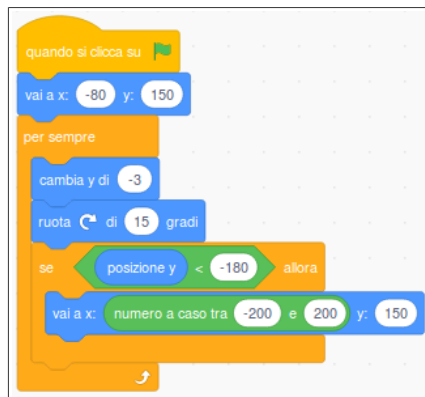
- se ritenete che la foglia scenda troppo velocemente, potete abbassare il valore a -3;
- per rendere la caduta più realistica possiamo aggiungere una rotazione:



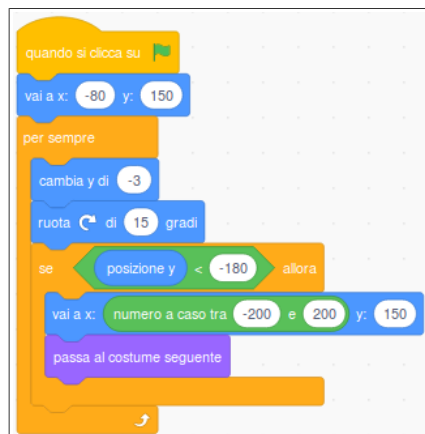
- quando la foglia raggiunge il fondo dello *Stage* deve sparire; un modo per capire se la foglia è giunta a terra è verificare il valore della sua coordinata y: se è inferiore a -180 allora la si può riportare nella sua posizione iniziale:



- per rendere il gioco più vario, possiamo fare in modo che la posizione orizzontale venga scelta a caso dal programma:



- poiché lo sprite *leaf* è dotato di molti costumi possiamo cambiare l'aspetto ad ogni nuova discesa:



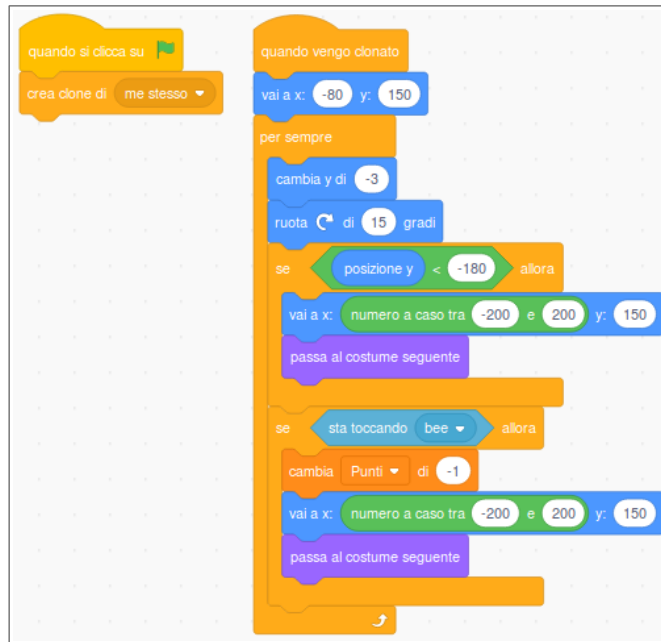
- se la foglia tocca l'ape allora si perde un punto:



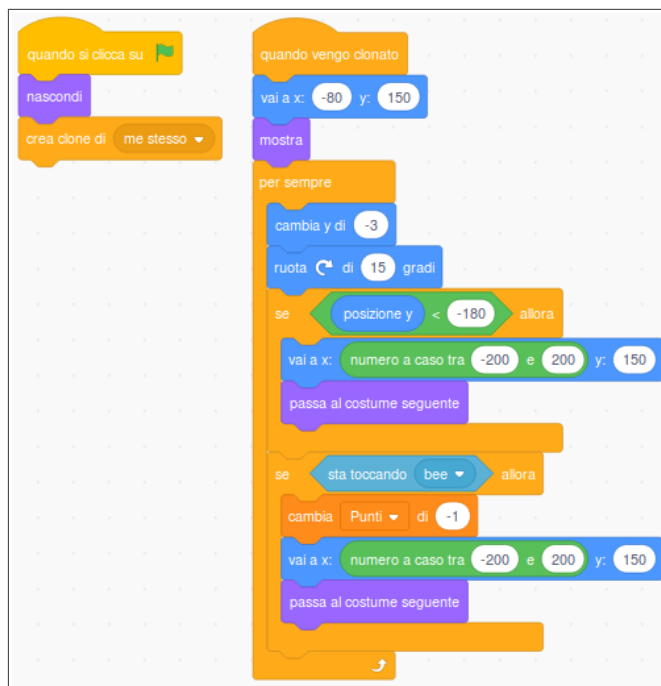
- provare il gioco andando a finire di proposito su una foglia... cosa accade? perché? se la cosa non piace, come si potrebbe risolvere? beh, per esempio riportando la foglia in alto, esattamente come accade quando tocca terra:



- una foglia sola non complica abbastanza il gioco? se ne può aggiungere un'altra! una possibilità è duplicare lo sprite (magari ritardando l'apparizione delle copie per aumentare gradualmente il livello di difficoltà del gioco), l'altra è usare i *Cloni*;
- cloni: lo sprite principale ha il solo scopo di generare copie di sè stesso, e sono le copie (i cloni) che eseguono il programma vero e proprio:



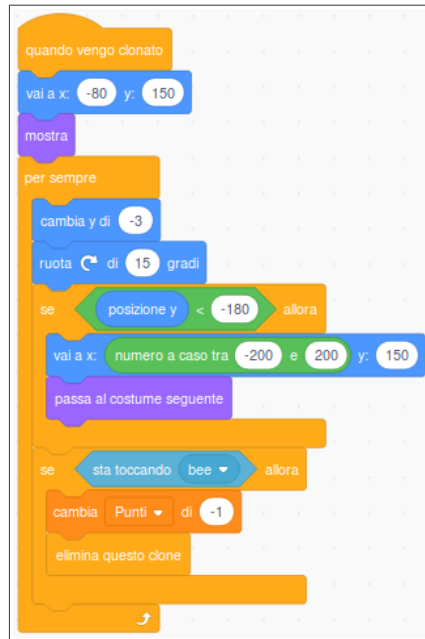
- conviene nascondere il clone principale della foglia: ferma lì in mezzo allo schermo dà solo fastidio – ricordiamoci di inserire un blocco `mostra` nel programma principale, altrimenti anche i cloni rimarranno invisibili:



- il vantaggio dei cloni è che è semplice definire quando e quanti cloni devono essere in gioco, ad esempio possiamo fare apparire una foglia aggiuntiva ogni 5 secondi:



- quando un clone tocca l'ape è possibile metterlo "fuori gioco" anziché nasconderlo, è più corretto:



Partita a tempo

- facciamo in modo che il gioco abbia una durata predefinita... possiamo usare una nuova variabile, la impostiamo ad un valore iniziale che viene decrementato ogni secondo e, quando questa raggiunge lo zero, interrompiamo il gioco con il blocco `ferma [tutto]` (oppure la si incrementa a partire da zero e si interrompe il gioco quando raggiunge un valore predefinito);
- questo script può essere assegnato a uno sprite qualunque, anche allo sfondo!
- creare una nuova variabile "Tempo", quindi:



Altri spunti

- aggiungere gli effetti sonori;
- sapreste simulare l'effetto del vento sulle foglie? e un vento di intensità variabile?
- è accettabile un punteggio negativo? come si potrebbe fare per evitare questa condizione?
- aggiungere una schermata finale al gioco;
- aggiungere una schermata iniziale di presentazione del gioco con le istruzioni;
- il girasole sparisce se l'ape non lo raggiunge entro 5 secondi;
- per chi ha scelto lo sfondo **Forest**, fare in modo che il fiore appaia solo tra l'erba (soluzione: sostituire i due blocchi `raggiungi (posizione a caso)` con `vai a x: (numero a caso tra (-200) e (200)) y: (numero a caso tra (-140) e (-30))`);
- quando una foglia colpisce l'ape, questa resta ferma per un secondo (utilizzare i messaggi);
- aggiungere un altro sprite (es. **boots.sprite3**) che insegue l'ape e che se la raggiunge causa la fine del gioco:



- far terminare il gioco all'esaurimento delle tre vite dell'ape anziché allo scadere dei 30 secondi;
- aggiungere un altro sprite (es. **mushroom.sprite3**) che appare di tanto in tanto e che "intontisce" temporaneamente l'ape (invertire i comandi destra/sinistra per 3/5 secondi):



- ...