



CoderDojo Firenze

Quanto durano 10 secondi

Gioco molto semplice in cui bisogna premere il tasto spazio esattamente (+/- 1 s) dopo 10 secondi dall'avvio del gioco.

Si presta a miglioramenti estetici e funzionali per esempio restringendo la tolleranza.





CoderDojo Firenze

Scratch: Acchiappa la palla

Una palla cade dall'alto in posizione casuale e il giocatore deve intercettarla con un mirino mosso dal mouse, facendo un punto ogni volta che la palla viene intercettata.

Palla che cade dall'alto



Sprite per mirare la palla col mouse



Complicazioni possibili:

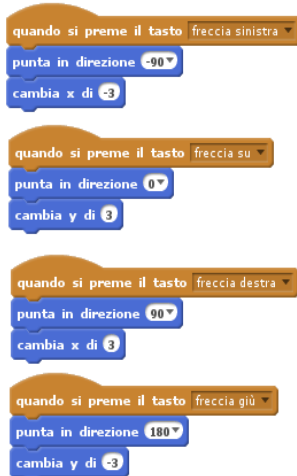
- Inserire il click sull'intercettazione
- usare una seconda palla come proiettile



Scratch: giochi di movimento

Movimenti usando i tasti

Per spostare uno sprite nelle 4 direzioni usando le frecce o altri tasti (es. “a s d w”) e ruotandolo nella direzione del moto, ci si aggancia all'evento “Quando si preme il tasto.....”



Va gestito ogni tasto di movimento in maniera separata. In questo caso il comando “punta in direzione...” serve a direzionare lo sprite nella direzione del moto. Il numero di pixel di spostamento (3 nell'esempio) può essere variato per muovere più o meno velocemente lo sprite. Per ottenere un moto continuo è necessario tenere premuto il tasto della direzione del moto. Questo tipo di movimento si usa per muovere sprite lungo percorsi o labirinti.

Se si vuole un moto continuo, comandando soltanto la direzione di moto attraverso i tasti, è necessario utilizzare una variabile che conservi il tasto che di volta in volta viene premuto. Si può dunque produrre uno script come il seguente:

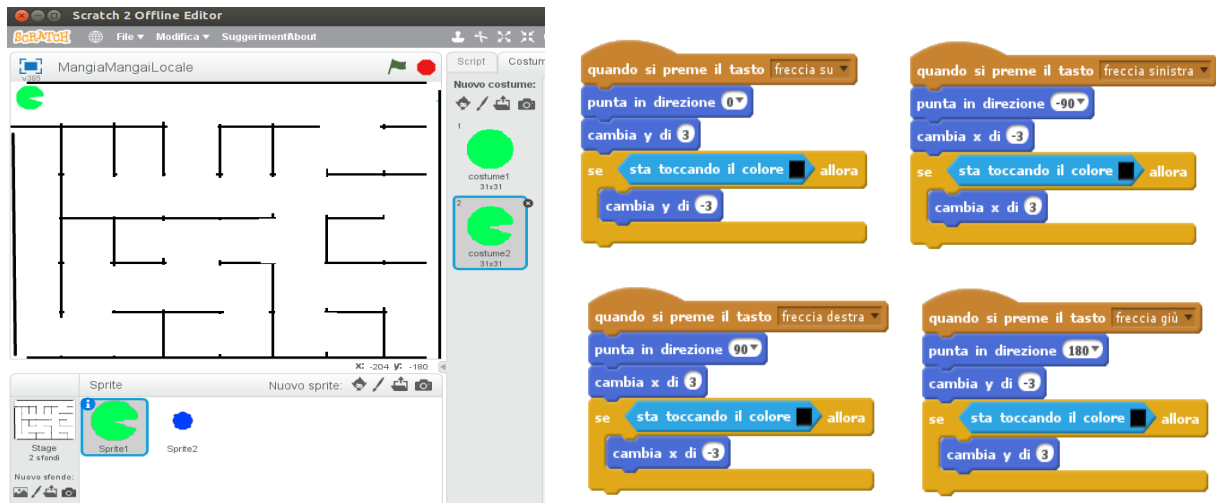


In questo caso si fa uso della variabile “tasto” che viene impostata ogni volta che si preme un tasto, di un ciclo infinito (“per sempre”) in cui facciamo spostare lo sprite nella direzione definita dal tasto.

Questo tipo di movimento si usa nei giochi stile “TRON”, ovvero in cui si deve muovere lo sprite in concorrenza con un altro giocatore e non ci si può fermare.

Movimento in un labirinto

Per questo esercizio è necessario disporre di uno sprite e di uno sfondo che rappresenta un labirinto, simili a quelli mostrati nella figura seguente. Le pareti del labirinto devono essere di un colore specifico in modo da poter controllare l'evento di collisione tra lo sprite e le pareti e far “rimbalzare” lo sprite se va contro la parete.



Il sensore “sta toccando il colore ...” identifica la condizione in cui lo sprite tocca la parete del labirinto, per cui viene fatto rimbalzare generando il movimento opposto a quello che lo ha portato contro la parete.

Gioco TRON

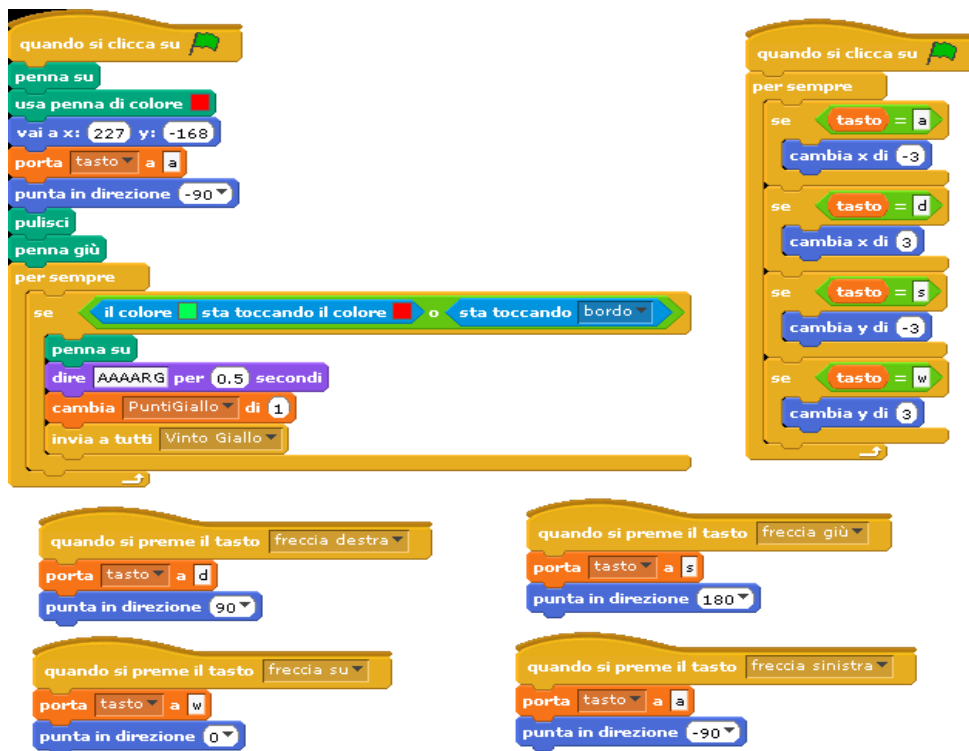
Tron è un vecchio videogioco che prende spunto da un film di fantascienza del 1982, in cui delle motociclette si muovono lasciando scie che non devono incrociarsi. Per riprodurre un gioco simile, è necessario usare due sprite (uno per giocatore) che lascino una scia colorata, e gestire le collisioni degli sprite con le scie.

Il movimento viene gestito solo per quanto riguarda la direzione per cui la pressione di un tasto di movimento provoca la rotazione dello sprite, ma la velocità dello sprite è sempre la stessa.

La scia viene tracciata utilizzando il comando “penna giù” che disegna una linea di un colore predefinito lungo il percorso dello sprite. Nell'esempio le scie vengono disegnate con il colore rosso.

La gestione delle collisioni con le scie viene fatta tramite il sensore “il colore ... sta toccando il colore”.

Gli sprite vanno disegnati con una forma affusolata e con una punta colorata che serve appunto a determinare la collisione con le scie. Gli sprite inoltre non possono toccare i bordi del campo di gioco. Nell'esempio gli sprite sono realizzati con forme triangolari nera e gialla ambedue con la punta di colore verde.



Lo script riportato è per lo sprite di colore nero, va duplicato per quello di colore giallo modificandolo opportunamente.

La gestione della fine gioco è demandata allo sfondo con il seguente script

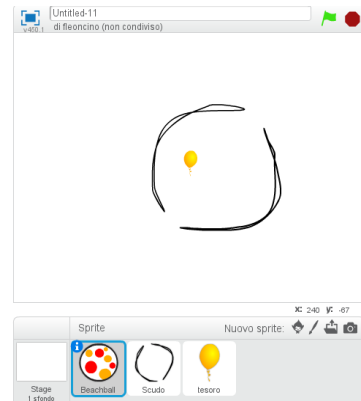


In pratica quando uno sprite va in collisione con una scia o con il bordo invia un messaggio che viene intercettato dallo sfondo che mostrerà un messaggio del tipo “Ha vinto il giallo” o “Ha vinto il nero” oltre al punteggio accumulato dai due giocatori.

GIRA LO SCUDO

Proiettile

```
quando si clicca su [bandierina]
nascondi
porta dimensione al 15 %
per sempre
  attendi numero a caso tra 1 e 3 secondi
  crea clone di me stesso
quando ricevo bloccato
nascondi
quando vengo clonato
mostra
se numero a caso tra 1 e 2 = 1 allora
  vai a x: 240 y: numero a caso tra -240 e 240
altrimenti
  vai a x: -240 y: numero a caso tra -240 e 240
scivola in 1 secondi a x: 0 y: 0
nascondi
```



tesoro

```
quando si clicca su [bandierina]
porta dimensione al 50 %
vai a x: 0 y: 0
per sempre
  se sta toccando Beachball allora
    cambia dimensione di -1
    passa al costume seguente
```

Scudo

```
quando si clicca su [bandierina]
vai a x: 0 y: 0
per sempre
  se sta toccando Beachball allora
    invia a tutti bloccato
quando si preme il tasto freccia destra
ruota di 5 gradi
quando si preme il tasto freccia sinistra
ruota di -5 gradi
```



CoderDojo Firenze

Griglia

Gioco semplice in cui in una griglia realizzata clonando uno sprite bisogna cliccare sugli sprite che si illuminano per un breve tempo. Se si fa in tempo lo sprite sparisce. Viene dato un numero di accensioni fisso, terminato il quale il gioco termina e mostra il punteggio ottenuto.

Si fa uso di uno sprite (un cerchio rosso) e di uno stage con due costumi.

Si presta a modifiche e stravolgimenti, sia nell'aspetto che nella logica del gioco.

Script dello Stage:

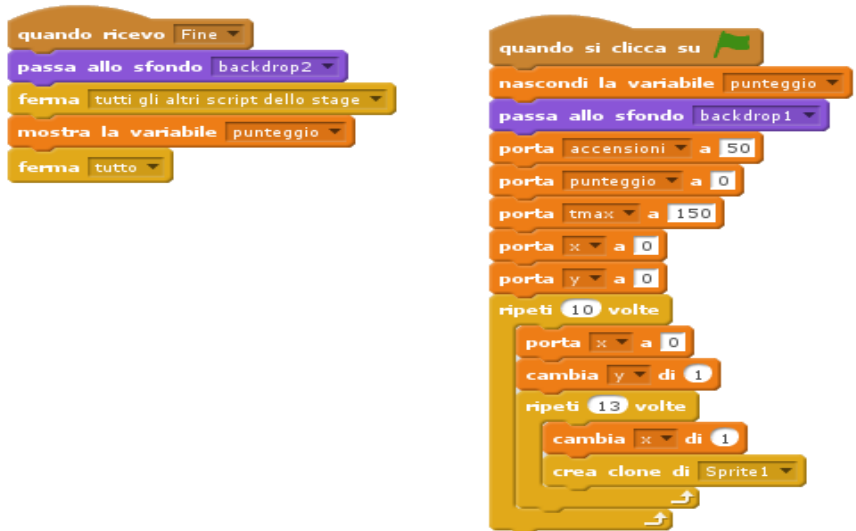
si fa uso delle seguenti variabili "globali":

accensioni: numero di sprite che risulteranno cliccabili

punteggio: per contare i

tmax: serve per diminuire il tempo massimo di attesa per l'accensione col crescere del punteggio (concetti di distribuzione di probabilità ed ergodicità?), anche se non è una scelta felicissima.

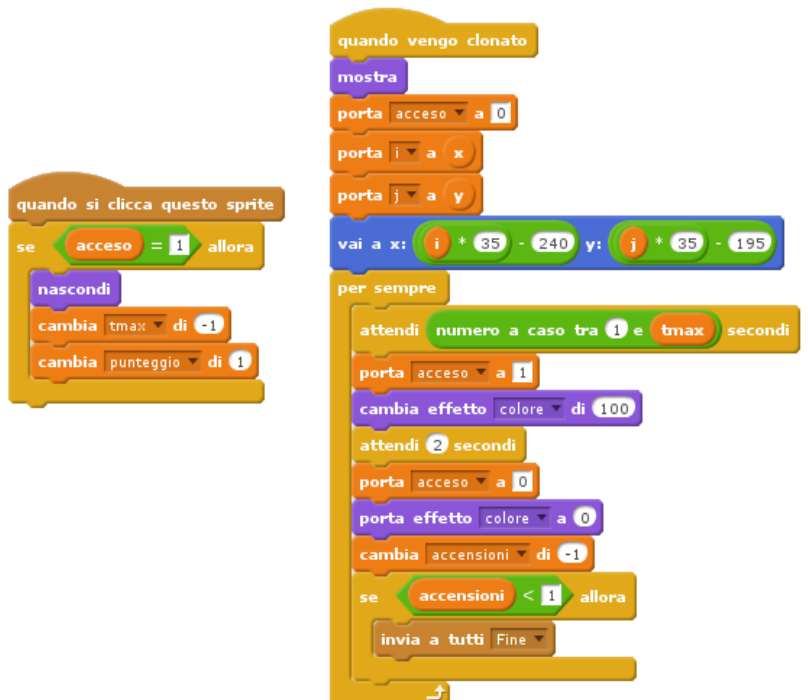
x,y: posizione dello sprite



Script dello sprite

le variabili *i* e *j* (private dello sprite) non sono utilizzate, ma possono risultare utili per modificare la logica del gioco.

La variabile **acceso** (privata) serve per sapere se il clone è cliccabile





CoderDojo Firenze

Scratch: Mostro mangia Palla (Festival dei Bambini) BASE

Script mostro

```
quando si clicca su [bandierina]
mostra
per sempre
  se < distanza da puntatore del mouse > > 5
    punta verso puntatore del mouse
    fai 5 passi
quando ricevo STOOOOP
nascondi
```

Script Stage (ha due sfondi)

```
quando si clicca su [bandierina]
passa allo sfondo sfondo1
porta punti a 0
azzerà cronometro
per sempre
  se < cronometro > > 30
    invia a tutti STOOOOP
    attendi 0.1 secondi
    passa allo sfondo fine
    azzerà cronometro
    ferma tutto
Lo Stage ha due sfondi: iniziale bianco e sfondo Fine
```

Script di ciascuna palla (una volta scritto per una, è sufficiente duplicare lo sprite)

```
quando si clicca su [bandierina]
nascondi
per sempre
  attendi numero a caso tra 0.5 e 3 secondi
  vai a x: numero a caso tra -220 e 220 y: numero a caso tra -160 e 160
  mostra
quando si clicca su [bandierina]
per sempre
  se < sta toccando mostro >
    produci suono Duck
    nascondi
    cambia punti di 1
quando ricevo STOOOOP
nascondi
La palla gestisce la variabile punti che però è globale
la palla ha anche un suono associato: Duck
```


Sequenza di creazione guidata

- 1) Sprite Mostro: si muove inseguendo il mouse (lo sprite Mostro deve essere libero di ruotare)
- 2) Sprite Palla:
 - apparizione casuale dopo un tempo casuale
 - gestione dello 'sta toccando Mostro': 'produci suono', nascondi
 - inserimento variabile globale 'punti' e aumento gestito nello 'sta toccando Mostro'
- 3) Stage: creazione sfondo 'FINE' e gestione della sua apparizione e del cronometro (che fa finire il gioco dopo 30 secondi). Invio messaggio STOOOOP.
- 4) Sprite Palla e Mostro: gestione della scomparsa su ricezione di STOOOOP

AVANZATO

Script Mostro

```

quando si clicca su
mostra variabile punti
mostra variabile tempo
azzerà cronometro
porta punti a 0
porta dimensione al 40 %
mostra
per sempre
se cronometro > 30
  dire AAAARG per 2 secondi
  nascondi
  invia a tutti FINE
  porta tempo a 30 - cronometro

```

```

quando si clicca su
per sempre
  se distanza da puntatore del mouse > 5
    punta verso puntatore del mouse
    fai 5 passi

```

```

quando ricevo Ostacolo
ripeti 5 volte
  cambia effetto colore di 25
  cambia effetto colore di 50
cambia punti di -1
porta effetto colore a 0

```

```

quando ricevo Ostacolo
fai -25 passi
dire Ahiiii per 0,5 secondi

```

```

quando ricevo preso
dire Gnam per 0,2 secondi
cambia punti di 1

```

Script Palla1

```

quando si clicca su
per sempre
  se sta toccando mostro
    invia a tutti preso
    nascondi

```

```

quando ricevo FINE
nascondi

```

```

quando si clicca su
nascondi
porta dimensione al 35 %
per sempre
  attendi numero a caso tra 1 e 3 secondi
  vai a x: numero a caso tra -220 e 220 y: numero a caso tra -160 e 160
  mostra

```

Script Mostro

```

quando si clicca su
per sempre
  se sta toccando mostro
    invia a tutti Ostacolo

```

```

quando ricevo FINE
nascondi

```

Script Ostacolo

```

quando si clicca su
porta dimensione al 35 %
mostra
per sempre
  attendi numero a caso tra 1 e 3 secondi
  vai a x: numero a caso tra -220 e 220 y: numero a caso tra -160 e 160

```

Script Stage

```

quando si clicca su
passa allo sfondo sfondogioco

```

```

quando ricevo FINE
passa allo sfondo game over
mostra variabile punti
nascondi variabile tempo
ferma tutto

```

Si fanno modifiche e aggiunte al primo programma:

- Si Introduce un ostacolo
- Si introduce una variabile tempo per mostrare quanto manca alla fine del gioco
- Si fa maggior uso dei messaggi

CON GLI ALGORITMI È POSSIBILE COSTRUIRE GIOCHI

Per esempio proviamo a descrivere un gioco in cui un pappagallo deve cercare di acchiappare delle mele che scorrono in un fiume evitando un leone...

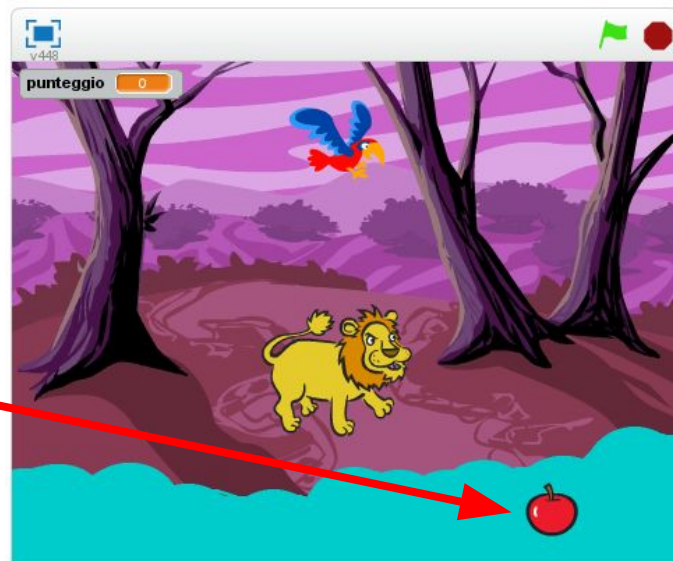


OGNI OGGETTO DEL GIOCO HA UN PROGRAMMA CHE LO ANIMA...

La mela si sposta da sinistra verso destra scorrendo nel fiume...

Script della mela:

- 1) Posizionati sul lato sinistro dello schermo
- 2) Ripeti i seguenti passi continuamente:
 - ❖ Muoviti di pochi passi a destra
 - ❖ Se raggiungi il lato destro dello schermo:
 - Torna al lato sinistro
 - ❖ Se tocchi il pappagallo:
 - Aggiungi +1 al punteggio del pappagallo
 - Ricomincia dal lato sinistro



OGNI OGGETTO DEL GIOCO HA UN PROGRAMMA CHE LO ANIMA...

Il pappagallo, controllato dal giocatore cerca di acchiappare le mele lanciandosi dall'alto...

Script del pappagallo:

- 1) Posizionati sull'angolo in alto a sinistra dello schermo
- 2) Ripeti i seguenti passi continuamente:
 - ❖ Se il giocatore preme la freccia destra:
 - Spostati un poco a destra (se possibile)
 - ❖ Se il giocatore preme la freccia sinistra:
 - Spostati un poco a sinistra (se possibile)
 - ❖ Se il giocatore preme il tasto spazio:
 - Voli in picchiata verso il fiume
 - Torna rapidamente in cima



OGNI OGGETTO DEL GIOCO HA UN PROGRAMMA CHE LO ANIMA...

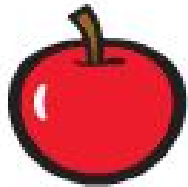
Il leone, l'antagonista del giocatore, cerca di catturare e mangiare il pappagallo...

Script del leone:

- 1) Posizionati nel mezzo dello schermo
- 2) Ripeti i seguenti passi continuamente:
 - ❖ Se il pappagallo è alla mia destra:
 - Spostati un poco a destra
 - ❖ Se il pappagallo è alla mia sinistra:
 - Spostati un poco a sinistra
 - ❖ Se tocco il pappagallo:
 - Termina il gioco



SCRIPT DELLA MELA



```
quando si clicca su   
vai a x: -228 y: -141  
punta in direzione 90  
per sempre  
  fai 3 passi  
  se  posizione x > 240 allora  
    vai a x: -228 y: -141  
  se  sta toccando Parrot allora  
    cambia punteggio di 1  
    vai a x: -228 y: -141
```


SCRIPT DEL PAPPAGALLO










```
quando si clicca su [bandierina]
punta in direzione 90
porta stile rotazione a sinistra-destra
vai a x: -207 y: 110
per sempre
  se [tasto freccia destra premuto] allora
    se [posizione x < 235] allora
      punta in direzione 90
      fai 5 passi
  se [tasto freccia sinistra premuto] allora
    se [posizione x > -235] allora
      punta in direzione -90
      fai 5 passi
  se [tasto spazio premuto] allora
    scivola in 3 secondi a x: [posizione x] y: -160
    scivola in 1 secondi a x: [posizione x] y: 110
per sempre
```

```
quando si clicca su [bandierina]
per sempre
  attendi 0.15 secondi
  passa al costume seguente
```

SCRIPT DEL LEONE



```
quando si clicca su 
porta stile rotazione a sinistra-destra
vai a x: 0 y: -41
per sempre
  se  posizione x di Lion < posizione x di Parrot  allora
    punta in direzione 90
    fai 1 passi
  se  posizione x di Lion > posizione x di Parrot  allora
    punta in direzione -90
    fai 1 passi
  se  sta toccando Parrot  allora
    ferma tutto
```



Script Auto

```
quando si clicca su [bandierina verde]
  porta punteggio a 0
  punta in direzione 90
  vai a x: 0 y: 72

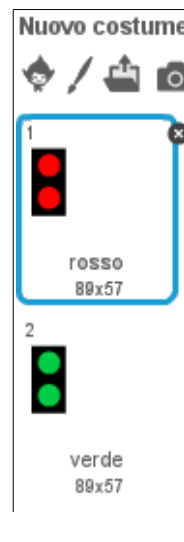
quando ricevo VIA!!!!
  per sempre
    punta verso puntatore del mouse
    se distanza da puntatore del mouse > 10 allora
      fai 5 passi

quando si clicca su [bandierina verde]
  per sempre
    se il colore [giallo] sta toccando il colore [bianco] allora
      cambia punteggio di -1
      fai -5 passi

quando si clicca su [bandierina verde]
  per sempre
    se il colore [giallo] sta toccando il colore [rosso] allora
      cambia punteggio di 2
```

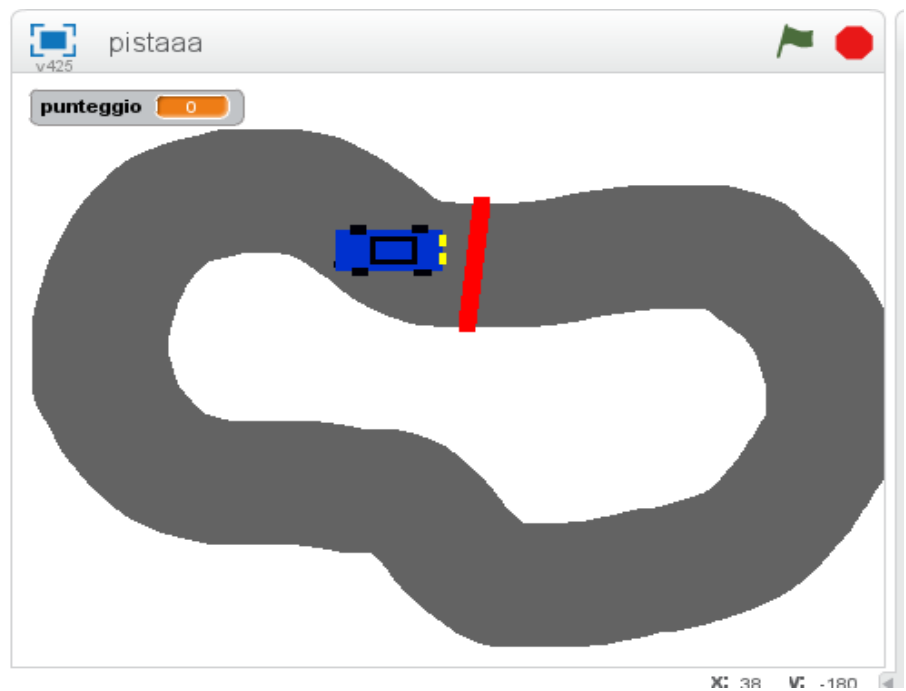
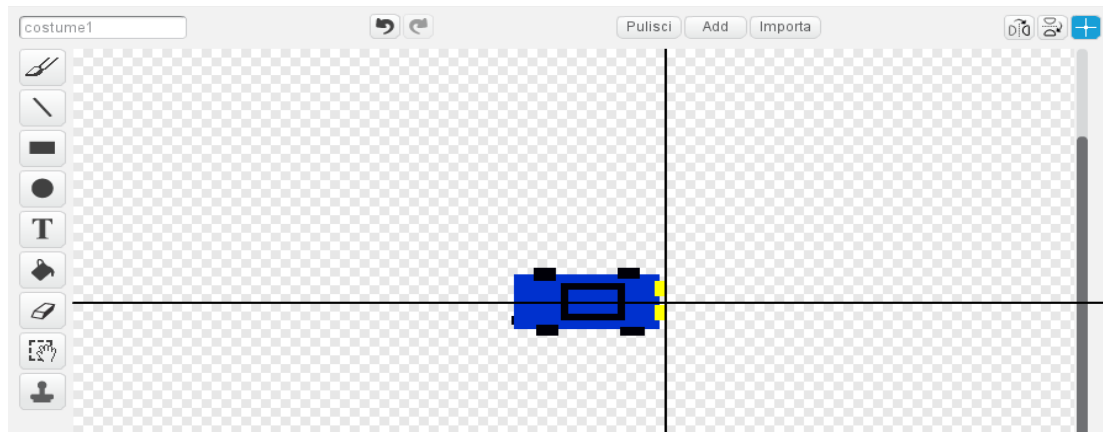
Script Semaforo

```
quando si clicca su [bandierina verde]
  passa al costume rosso
  attendi 2 secondi
  passa al costume verde
  invia a tutti VIA!!!! e attendi
```



Script Stage

```
quando ricevo VIA!!!!
  porta tempo a 30
  azzera cronometro
  per sempre
    porta tempo a arrotonda(30 - cronometro)
    se tempo < 0 allora
      porta tempo a 0
      arresta tutto
```



```

quando si clicca su [bandierina verde]
  porta punteggi a 0
  punta in direzione 90
  vai a x: -31 y: 72
  attendi 1 secondi
  per sempre
    punta verso puntatore del mouse
    se distanza da puntatore del mouse > 10 allora
      fai 5 passi
  
```

```

quando si clicca su [bandierina verde]
  per sempre
    se il colore [giallo] sta toccando il colore [bianco] allora
      cambia punteggi di -1
  
```

```

quando si clicca su [bandierina verde]
  per sempre
    se il colore [giallo] sta toccando il colore [rosso] allora
      cambia punteggi di 2
  
```



CoderDojo Firenze

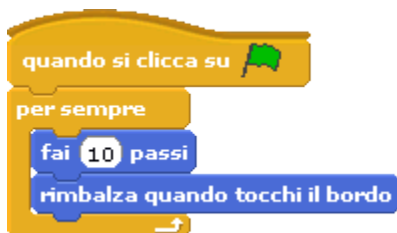
Scratch Primi passi e gioco dello squalo mangia pesce.

Primi passi:

Aperto scratch l'ambiente si presenta con uno sfondo (stage) vuoto e uno sprite a forma di gatto con due costumi.

Va fatta una brevissima introduzione del concetto di sprite e di stage e script.

Un primo approccio è quello di far muovere il gatto, utilizzando il comando “fai 10 passi”, e facendolo rimbalzare sul bordo con il comando “rimbalza quando tocchi il bordo”



In questo modo mostriamo già anche gli elementi di controllo “quando si clicca su Start” e “per sempre”.

Si può dire ai ragazzi di provare a cambiare il numero di passi e vedere cosa succede, anche usando numeri negativi e spiegare l'associazione fra passo e pixel.

Un'altra sperimentazione può essere fatta cambiando la proprietà delle sprite che lo blocca o meno a movimenti ortogonali



Il passo immediatamente successivo è quello di simulare una camminata attraverso il cambio di costume e direzionare lo sprite attraverso il mouse, in questo caso va tolto il rimbalzo al bordo perché va in contrasto con l'inseguimento del mouse.



Il gioco dello squalo che mangia il pesce

Il concetto è quello di realizzare un semplice gioco in cui uno squalo comandato dal giocatore tramite il mouse, insegue e mangia un pesce più piccolo.

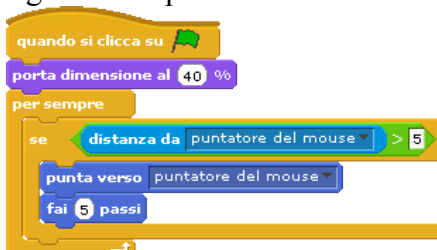
Pertanto le istruzioni che dobbiamo impartire sono per lo squalo di inseguire il mouse e mangiare il pesce quando lo raggiunge, e per il pesce di andare a zozzo per il mare.

In realtà non possiamo impartire allo squalo l'istruzione di mangiare il pesce, si tratta di un effetto "speciale", ovvero quando lo squalo raggiunge il pesce, impartiamo al pesce l'istruzione di sparire.

Si riparte dal codice precedente in cui lo sprite insegue il mouse e:

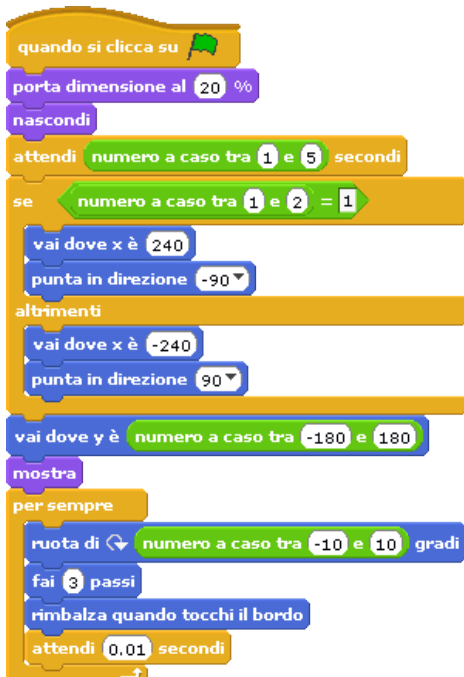
- Si caricano come costumi dello sprite tramite il bottone importa le immagini "shark1-a" e "shark1-b", ovvero l'immagine di uno squalo a bocca chiusa (1-a) e uno a bocca aperta (1-b), che serviranno per l'animazione del boccone.
- Si carica come sfondo l'immagine "underwater" che rappresenta il fondo marino.

Il primo step è di portare lo squalo a dimensioni più piccole e di fargli inseguire il mouse con il seguente script:



Rispetto allo script fatto in precedenza è stato aggiunto un blocco "se" con la condizione "distanza da puntatore del mouse" > 5. Questo blocco fa sì che se il mouse è fermo quando lo squalo lo raggiunge non ci sia un effetto "schizofrenico" e anche lo squalo rimanga fermo.

A questo punto si introduce la preda, quindi si carica un altro sprite utilizzando il file "fish1", si crea uno script che lo fa apparire in una posizione casuale dello schermo e che lo fa muovere in modo casuale.



A questo punto abbiamo un pesce che si muove in modo “casuale” e uno squalo che viene “guidato” dal mouse. Dobbiamo far sì che quando lo squalo raggiunge il pesce si produca una animazione che apre e chiude la bocca dello squalo (passando da costume shark1-a a shark1-b) e il pesciolino sparisca.

Lo script dello squalo diventa:

```

quando si clicca su [bandierina]
  porta dimensione al 40 %
  passa al costume shark1-a
  per sempre
    se < distanza da puntatore del mouse > 5
      punta verso puntatore del mouse
      fai 5 passi
    se sta toccando preda
      passa al costume shark1-b
      cambia effetto colore di 50
    altrimenti
      passa al costume shark1-a
      porta effetto colore a 0
  
```

mentre quello della preda diventa:

```

quando si clicca su [bandierina]
  porta dimensione al 20 %
  nascondi
  attendi numero a caso tra 1 e 5 secondi
  se numero a caso tra 1 e 2 = 1
    vai dove x è 240
    punta in direzione -90
  altrimenti
    vai dove x è -240
    punta in direzione 90
  vai dove y è numero a caso tra -180 e 180
  mostra
  per sempre
    ruota di numero a caso tra -10 e 10 gradi
    fai 3 passi
    rimbalza quando tocchi il bordo
    attendi 0,01 secondi
    se sta toccando Sprite1
      attendi 0,05 secondi
      produci suono Pop
      nascondi
      attendi numero a caso tra 1 e 5 secondi
      se numero a caso tra 1 e 2 = 1
        vai dove x è 240
        punta in direzione -90
      altrimenti
        vai dove x è -240
        punta in direzione 90
      vai dove y è numero a caso tra -180 e 180
      mostra
    
```

```
when I start as a clone
switch costume to costume2
change livello by 1
if livello > 10 then
broadcast message1
stop this script

go to x: pick random -80 to 100 y: -94 + 30 * livello

forever
change x by shift
if touching edge? then
set shift to shift * -1
if mouse down? then
glide 1 secs to x: x position y: -125 + 30 * livello
if abs of 16 - x position < sens then
change precisione by abs of 16 - x position
switch costume to costume3
change punti by 1
change sens by -5
if sens < 10 then
set sens to 10
create clone of myself
stop this script
else
create clone of myself
stop this script
```

```
when I receive asdffdas
broadcast vai
delete this clone
```

```
when clicked
set livello to 0
set precisione to 0
set punti to 0
set shift to 8
set sens to 75
hide variable precisione
broadcast vai
```

```
when I receive vai
go to x: 16 y: -125
set livello to 0
change shift by 2
hide variable precisione
wait 1 secs
switch costume to costume3
create clone of myself
```



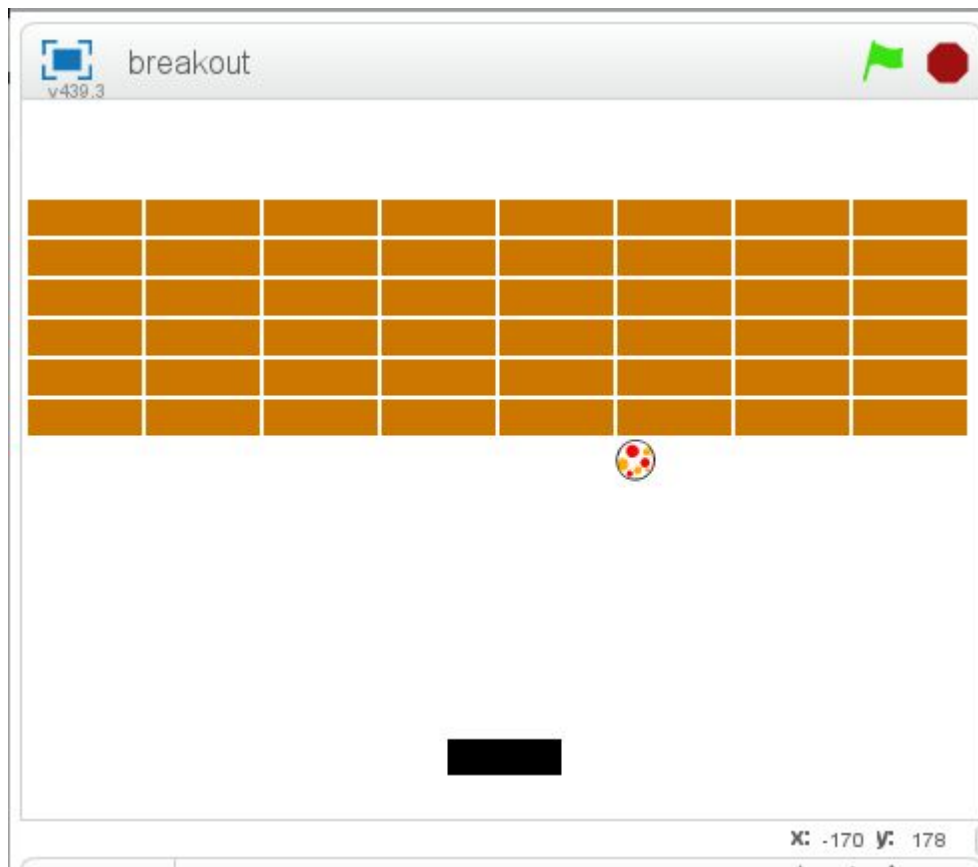
```
when I start as a clone
  switch costume to costume2
  change livello by 1
  if livello > 10 then
    broadcast message1
    stop this script

  go to x: pick random -80 to 100 y: -94 + 30 * livello

  forever
    change x by shift
    if touching edge? then
      set shift to shift * -1
    if mouse down? then
      glide 1 secs to x: x position y: -124 + 30 * livello
      if abs of 16 - x position < sens then
        switch costume to costume3
        change livello by -1
        change punti by 1
        change sens by -1
        if sens < 10 then
          set sens to 10
        create clone of myself
        delete this clone
      else
        change precisione by abs of 16 - x position
        create clone of myself
        stop this script
```

```
when clicked
  go to x: 16 y: -125
  set livello to 0
  set precisione to 0
  set punti to 0
  set shift to 10
  set sens to 75
  hide variable precisione
  wait 1 secs
  switch costume to costume3
  create clone of myself
```

Coderdojo Firenze -- Tutorial Breakout



Scopo del gioco è demolire il muro di mattoni nella parte alta dello schermo utilizzando la pallina rimbalzante. Tutte le volte che la pallina tocca un mattone questo viene demolito e scompare. Quando tutti i mattoni sono scomparsi hai vinto.

Se la pallina tocca la parte bassa dello schermo questa viene persa. Ci sono più palline, quanto tutte vengono perse, il gioco termina.

Per evitare che la pallina arrivi in fondo allo schermo, essa deve essere rinviata verso l'alto utilizzando una racchetta. La racchetta viene controllata dal giocatore utilizzando, per esempio, i tasti freccia destra e sinistra, oppure il puntatore del mouse.

1: La pallina: movimento nell'area di gioco



Iniziare un nuovo progetto ed eliminare lo sprite del gatto Scratch che per questo gioco non serve.

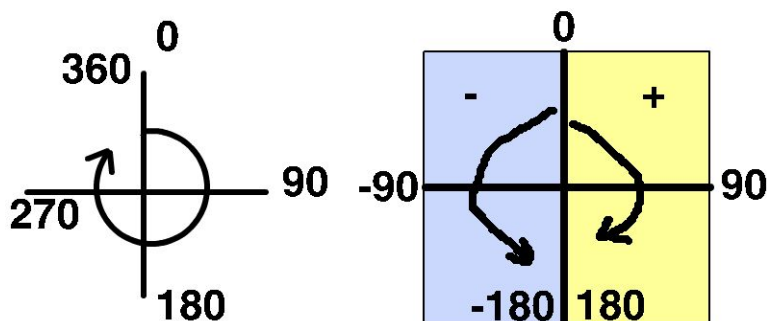
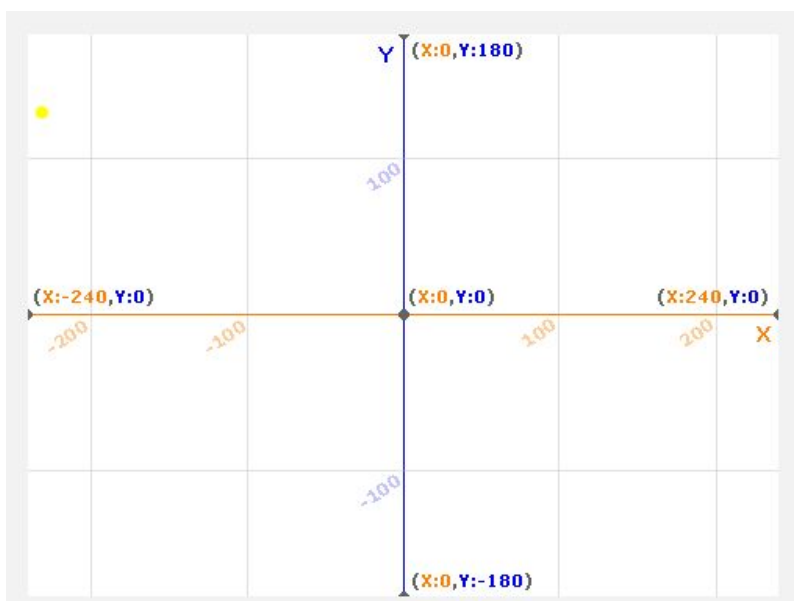
Creare un nuovo sprite, scegliendo dalla libreria una pallina, per esempio la "Beachball". Cambiargli il nome in "Pallina".

Scrivere uno script che fa muovere la pallina a velocità costante all'interno del campo di gioco. Fare in modo che tutte le volte che si riavvia il programma, la pallina parta da una posizione all'interno dello schermo che abbia come x un valore a caso tra tutti quelli possibili e come y il valore 0. Con questa scelta tutte le volte che il gioco inizia la pallina si troverà in un punto a caso della linea orizzontale che attraversa il centro dell'area di gioco (vedi figura a lato).

Fare in modo che la pallina si muova in una direzione a caso verso il basso. Per impostare la direzione, ricordarsi di come si misurano gli angoli nello stage di gioco. Fare riferimento alla figura sotto; in particolare notare come gli angoli si possano misurare in due modi: da 0 a 360 gradi girando in senso orario e da -180 a 180 passando per lo zero (che indica sempre la direzione verso l'alto).

Per evitare che la pallina sia troppo grossa, ridurre inoltre la dimensione utilizzando il

blocco "porta dimensione al NN%". Nel caso della BeachBall portare la sua dimensione al 20%. Ultima cosa: la pallina deve rimbalzare quando tocca il bordo.



Il programma risultante è riportato sotto.



2. La pallina: come farla scomparire e far terminare il programma se tocca la parte bassa dello schermo?

Per realizzare questo esistono vari modi, oggi proveremo quello della “**zona Mangia Palle**”. Definiamo un nuovo sprite, di nome “**ZonaMangiaPalle**” costituito da una semplice linea orizzontale di colore giallo che attraversa tutto lo schermo e collochiamolo nella parte bassa dello schermo.

Tip 1: per fare in modo che la linea venga perfettamente orizzontale, mentre si disegna tenere premuto il tasto “Maiuscolo” (o Shift).

Tip 2: ricordarsi centrare lo sprite in modo preciso.

Fare in modo che all’avvio che questo sprite si collochi nella parte bassa del campo di gioco tramite lo script mostrato a lato.



A questo punto aggiungere allo Sprite **Pallina** lo script a lato.

Dopo questa modifica il gioco risulterà estremamente corto! Infatti all’avvio del gioco Pallina partirà diretta verso il basso e come dopo pochi secondi toccherà **ZonaMangiaPalle** scomparirà dando termine al gioco. **Dobbiamo trovare il modo di controllare la pallina in modo da reinviarla verso l’alto!**



3. La racchetta: rispedire la pallina verso l'alto

Per far durare il gioco di più costruiamo un nuovo Sprite chiamato “**Racchetta**” che altro non è che un rettangolo nero che si può spostare solo in orizzontale nella zona immediatamente sopra la zona mangia palle.

Tip 3: definire lo stile di rotazione dello Sprite di tipo “solo orizzontale”.

Per far muovere la racchetta si possono utilizzare diversi metodi, tra cui:

- utilizzare i tasti freccia sinistra e destra
- seguire la posizione del mouse

Qui sotto viene riportato il programma da realizzare per ciascuno dei due casi

The image shows two Scratch scripts side-by-side. The left script starts with a 'quando si clicca su' block, followed by 'vai a x: 0 y: -160' and 'porta stile rotazione a sinistra-destra'. It then enters a 'per sempre' loop with two 'se' blocks. The first 'se' block checks for the 'tasto freccia sinistra premuto' event, and if true, it checks 'se posizione x > -200 allora', which leads to 'punta in direzione -90' and 'fai 15 passi'. The second 'se' block checks for the 'tasto freccia destra premuto' event, and if true, it checks 'se posizione x < 200 allora', which leads to 'punta in direzione 90' and 'fai 15 passi'. The right script starts with 'quando si clicca su', followed by 'vai a x: 0 y: -160' and 'porta stile rotazione a sinistra-destra'. It then enters a 'per sempre' loop with a single 'vai a x: x del mouse y: -160' block. Below the scripts is a yellow note with the text: 'ATTENZIONE! QUESTI DUE SCRIPT SONO ALTERNATIVI: SCEGLIERE SE SI VUOLE CONTROLLARE LA RACCHETTA CON IL MOUSE O CON I TASTI E COSTRUIRE IL RELATIVO SCRIPT'.

A questo punto rimane da aggiungere a **Pallina** il codice per farla rimbalzare quando tocca **Racchetta**.

ATTENZIONE: questa è la vera parte complicata di questo tutorial. L’algoritmo che implementa il rimbalzo è particolarmente complicato. Se siete dei veri **Ninja** provate a farlo da solo. Se non ci riuscite trovate la risposta nella prossima pagina.

Vista la sua complessità l'algoritmo di "rimbalzo" merita di essere contenuto all'interno di un blocco dedicato, realizzato tramite il sistema degli "Altri Blocchi". Per fare questo andare nella sezione "Altri blocchi" e cliccare sul tasto "Crea un blocco".

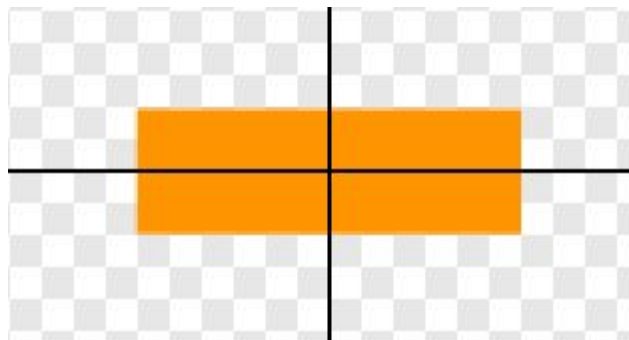
Dare al blocco il nome "RIMBALZA", quindi scrivere il codice da eseguire sotto il "cappello viola" con nome "definisci RIMBALZA" (vedi sotto a destra). Adesso tutte le volte che vorremo far rimbalzare la pallina basterà utilizzare il nuovo blocco "RIMBALZA" (vedi sotto a sinistra).



A questo punto è possibile divertirsi a far rimbalzare la pallina verso l'alto in modo da non farla cadere nella "Zona Mangia Palle".

4. Il muro da abbattere

Adesso passiamo alla costruzione del muro da abbattere con la pallina. Come sappiamo, il muro è formato da mattoni. Costruiamo quindi un nuovo sprite chiamato “**Mattono**”, che altro non è che un rettangolo di colore più o meno marrone. Aiutarsi con la “grid” e fare più o meno un rettangolo delle dimensioni in figura.



Per costruire il muro occorrono molti mattoni.

Per evitare di dover definire molti sprite utilizzeremo la funzione di “**Clonazione**”.

Ricordiamoci inoltre che il muro deve occupare solo la parte superiore dello Stage ($y > 0$).

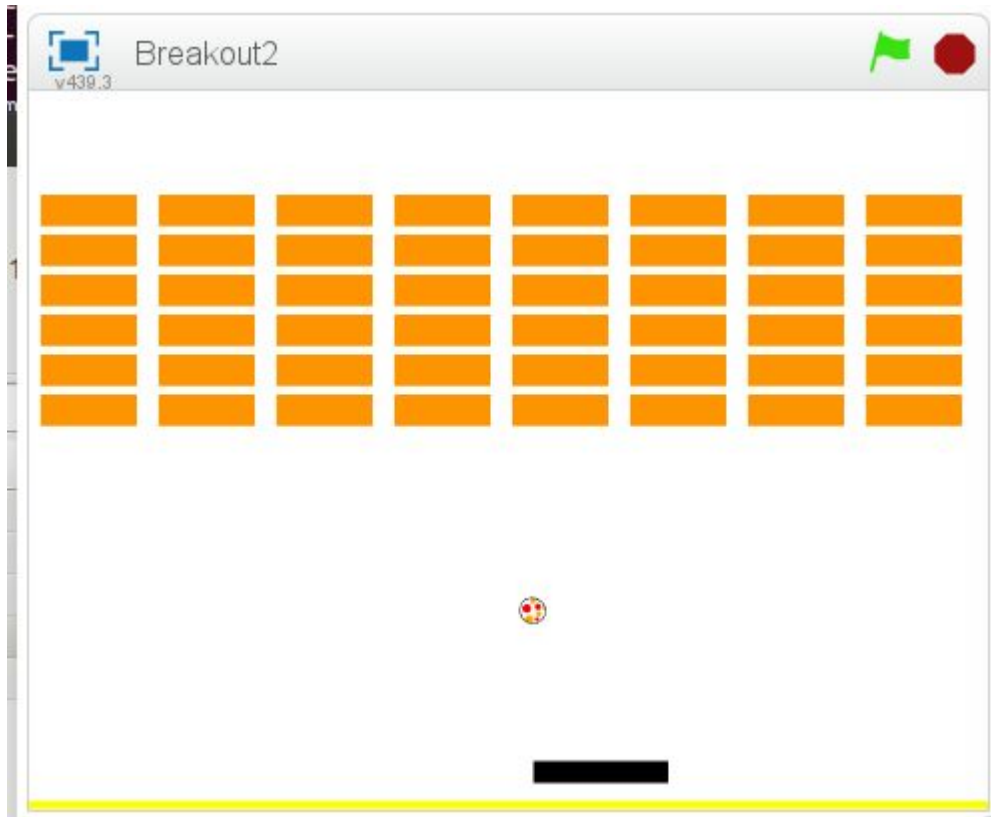
Ecco quindi un esempio di script per generare il muro. E' abbastanza complesso, vanno definite tre variabili (**numero_mattoni**, **indice** e **riga**) e i numeri presenti nell'algoritmo potrebbero dover essere adattati in base alla dimensione del mattone, in modo da generare il muro più regolare e compatto possibile. A voi, come esercizio, capire come funziona...



N.B. A seconda della dimensione del mattone, può darsi che i numeri contenuti nell'istruzione “**vai a**” debbano essere modificati un poco. Cercate rendere il muro il più possibile regolare in modo che non rimanga troppo spazio tra i mattoni.

Da notare poi l'istruzione **invia a tutti il segnale “Inizio”** che appare nello script principale alla fine delle operazioni di clonazione. Capiremo più tardi a cosa serve questa istruzione.

Nella figura che segue si vede come dovrebbe apparire a questo punto il campo di gioco



PROBLEMA. Si noti come la procedura di creazione del muro con tutte le operazioni di clonazione duri diversi secondi e come durante questa fase la pallina sia già in moto. Questo risulta piuttosto fastidioso. Sarebbe meglio che durante la fase di costruzione del muro il gioco sia ancora fermo e che la pallina cominci a muoversi solo nel momento che il muro è terminato.

Come fare? Prima di proseguire con la lettura, pensateci e provate a risolvere il problema.

SOLUZIONE: utilizzare il segnale **"Inizio"** inviato dal mattone una volta terminate le operazioni di clonazione. Nello script che governa il movimento della pallina (quello visto a pagina 4)

sostituire il blocco  con il blocco 

Allo Sprite **Pallina** va poi aggiunto un miniblocco che all'avvio lo nasconda:



5. Distruggere il muro!

A questo punto occorre realizzare l'interazione tra muro e pallina.

Cosa succede a una palla quando colpisce il muro? Rimbalza! Al programma di **Pallina** è quindi necessario aggiungere il seguente Script:



N.B. Notare come aver definito l'algoritmo di rimbalzo tramite un nuovo blocco, ci permetta adesso di utilizzarlo in modo semplice

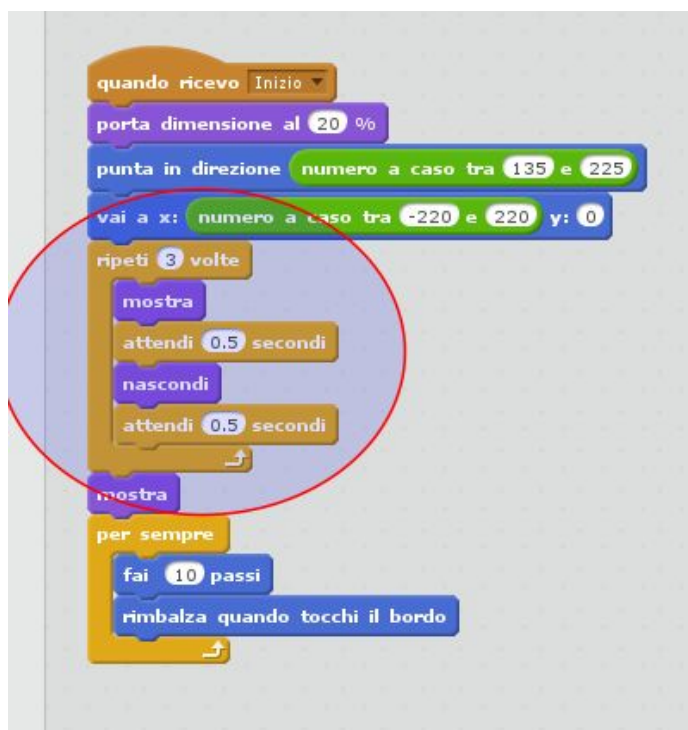
Nel nostro gioco succede poi un'altra cosa: quando la pallina colpisce un mattone, il mattone viene distrutto. Questo si ottiene tramite il seguente script da aggiungere allo Sprite **Mattone**.



6. Migliorie

6.1. Ritardare la partenza della pallina

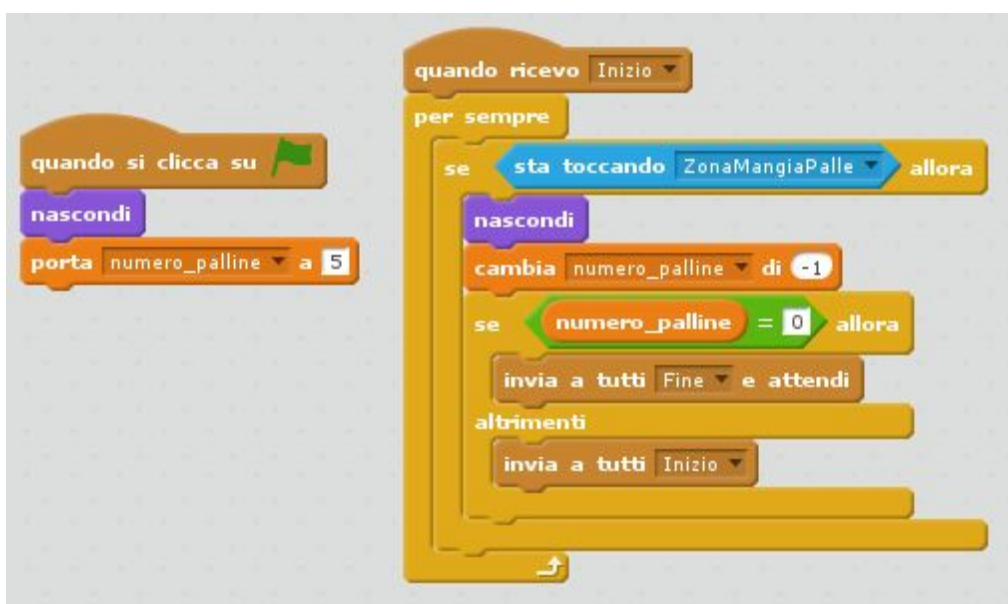
Appena finito di costruire il muro **Pallina** compare immediatamente in una posizione causale e parte verso una direzione causale. Per dare tempo al giocatore di posizionare **Racchetta** sotto la pallina e prepararsi a prenderla, aggiungere il seguente pezzo di codice allo Script di avvio della pallina:



6.3. Aumentare il numero di palline a disposizione del giocatore

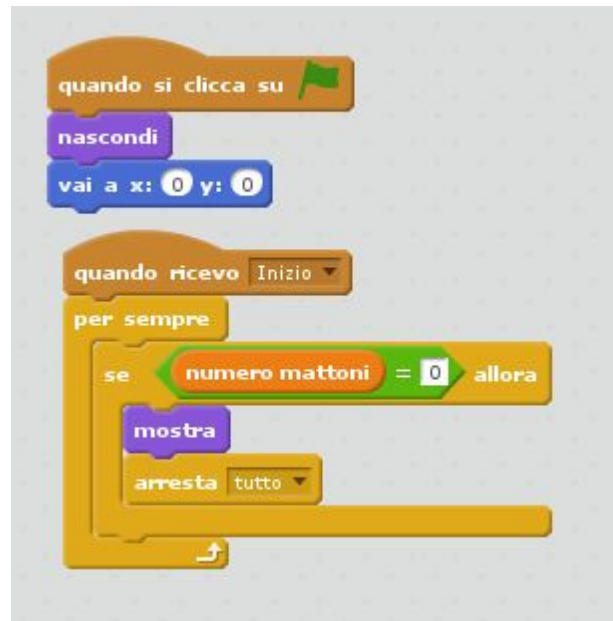
Fare questo comporta una modifica agli script di **Pallina**.

In particolare introdurre una nuova variabile di nome “**numero_palline**” e modificare gli script visti a pagina 3 nel seguente modo:



6.2. Inserire le schermate di fine gioco

Definire uno Sprite con una scritta di “**Livello Terminato**” che appaia alla fine della partita in caso di successo, cioè quando tutti i mattoni sono stati distrutti (la variabile **numero_mattoni** arriva a zero).



Definire uno Sprite di “**Game Over**” che appare quando la missione fallisce, cioè le palline finiscono e sono presenti ancora mattoni.



Appendice (per i pigri): tutti gli script

PALLINA

The script for the BALL object consists of several event-driven blocks:

- when clicked:** hides the object and sets the ball count to 5.
- when clicked:** enters a loop that triggers a bounce when touching the 'Mattoni' (bricks) object.
- when clicked:** enters a loop that triggers a bounce when touching the 'Racchetta' (paddle) object.
- when received 'Inizio':** sets the ball size to 20%, points in a random direction between 135 and 225 degrees, and moves to a random x-coordinate between -220 and 220 at y=0. It then repeats a sequence of showing, waiting 0.5 seconds, and hiding the ball three times.
- when clicked:** shows the ball and enters a loop that moves the ball 10 steps and triggers a bounce when touching the edge.
- when clicked:** enters a loop that triggers a change in ball count when touching the 'ZonaMangiaPalle' (ball-eating zone) object. If the count reaches 0, it sends a 'Fine' message; otherwise, it sends an 'Inizio' message.

A custom block named 'RIMBALZA' (bounce) is defined with the following logic:

- if direction > 0:** turn 2 * direction - 90 degrees.
- otherwise:** turn 2 * direction + 90 degrees.

ZONA MANGIA PALLE

The script for the ZONA MANGIA PALLE object is simple:

- when clicked:** moves the object to x=0 and y=-178.

RACCHETTA

quando si clicca su

vai a x: 0 y: -160

porta stile rotazione a sinistra-destra

mostra

per sempre

se **tasto freccia sinistra premuto** allora

se **posizione x > -200** allora

punta in direzione -90

fai 15 passi

se **tasto freccia destra premuto** allora

se **posizione x < 200** allora

punta in direzione 90

fai 15 passi

quando si clicca su

vai a x: 0 y: -160

porta stile rotazione a sinistra-destra

mostra

per sempre

vai a x: x del mouse y: -160

SE SI VUOLE CONTROLLARE CON I TASTI FRECCIA

SE SI VUOLE CONTROLLARE CON IL MOUSE

ATTENZIONE!

QUESTI DUE SCRIPT SONO ALTERNATIVI: SCEGLIERE SE SI VUOLE CONTROLLARE LA RACCHETTA CON IL MOUSE O CON I TASTI E COSTRUIRE IL RELATIVO SCRIPT

MATTONI

quando si clicca su

porta indice a 0

porta riga a 0

porta numero mattoni a 0

nascondi

ripeti 48 volte

crea clone di me stesso

invia a tutti Inizio

quando vengo clonato

cambia numero mattoni di 1

vai a x: -210 + indice * 59 y: 20 + riga * 20

mostra

cambia indice di 1

se **indice = 8** allora

cambia riga di 1

porta indice a 0

quando vengo clonato

per sempre

se **sta toccando Pallina** allora

attendi 0.05 secondi

cambia numero mattoni di -1

elimina questo clone

LIVELLO TERMINATO



GAME OVER



Coderdojo Firenze -- Tutorial Fantasma Acchiappa Dolcetti

Scopo del gioco è far entrare nel sacchetto del fantasma i dolcetti che cascano dall'alto verso il basso in modo da fare più punti possibile, contemporaneamente cercando di evitare pipistrelli e Jack Skellington che fanno invece perdere punti e svanire il fantasma. Il fantasma può muoversi orizzontalmente nella parte bassa dello schermo e può essere controllato con le frecce destra e sinistra.



1: Il primo dolcetto cadente

Iniziare un nuovo progetto ed eliminare lo sprite del gatto Scratch che per questo gioco non serve.

Creare un nuovo sprite che rappresenta un dolcetto, una caramella o un cioccolatino. Potete disegnarlo a piacere oppure utilizzare uno degli sprite presenti in libreria, per esempio **“Donut”**.

Scrivere uno script che fa muovere la pallina a velocità costante all'interno del campo di gioco dall'alto verso il basso.

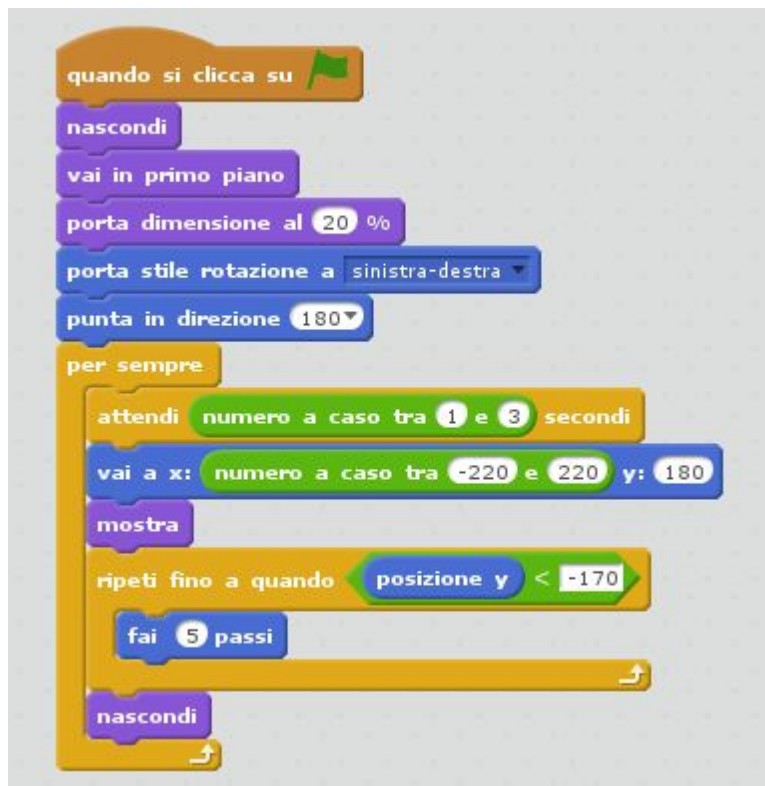
Fare in modo che tutte le volte che si riavvia il programma, il dolcetto parta dalla parte più alta dello stage (Y = 180) e da una posizione X a caso, tra i valori massimi -220 e 220.

Fare in modo che la pallina si muova verso il basso (Direzione = 180)

Per evitare che il dolcetto sia troppo grosso, ridurre inoltre la dimensione utilizzando il blocco “porta dimensione al 20%”.

Ultima cosa: la pallina deve sparire quando arriva a toccare il fondo dello schermo e quindi dopo pochi secondi ricomparire in cima.

Il programma risultante è riportato a lato.



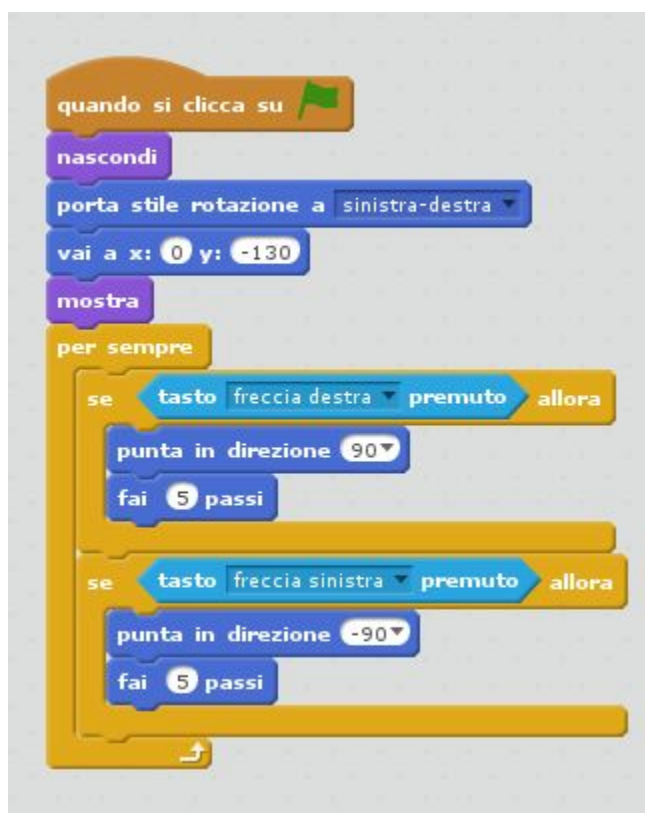
2. Il fantasma acchiappa dolcetti



Il nostro fantasma acchiappa dolcetti ha un sacchetto dove dobbiamo far entrare le caramelle. Si posiziona sulla parte bassa dello schermo e si muove solo in orizzontale. Per farlo spostare si utilizzano le frecce destra e sinistra.

Potete provare a disegnare il fantasma da soli oppure fatevi dare il fantasma a lato dai mentor. Se lo disegnate da soli, l'unica cosa importante **è che la parte superiore del sacchetto sia colorata uniformemente di un colore riconoscibile e non utilizzato in nessun'altra parte del gioco.** Per esempio nel fantasma a lato si vede che la parte superiore del sacchetto è colorata di verde.

Scrivere quindi lo script che permette di spostare il fantasma:



Adesso sullo schermo appare un fantasma che potete spostare con le frecce e dei dolci che cadono. Non particolarmente divertente, ma comunque un inizio!

3. Acchiappare il dolcetto!

A questo punto provate ad acchiappare il dolcetto mettendovi sotto di esso con il sacchetto mentre sta cadendo. Cosa vi aspettate? il dolcetto dovrebbe finire dentro il sacchetto. Invece continua a cadere. Del resto nel codice che abbiamo scritto finora non c'è niente che fa sì che il dolcetto venga acchiappato. Ma con una semplice modifica al programma del dolcetto si ottiene il risultato voluto. In particolare basta modificare il codice nel punto cerchiato di rosso qui sotto:



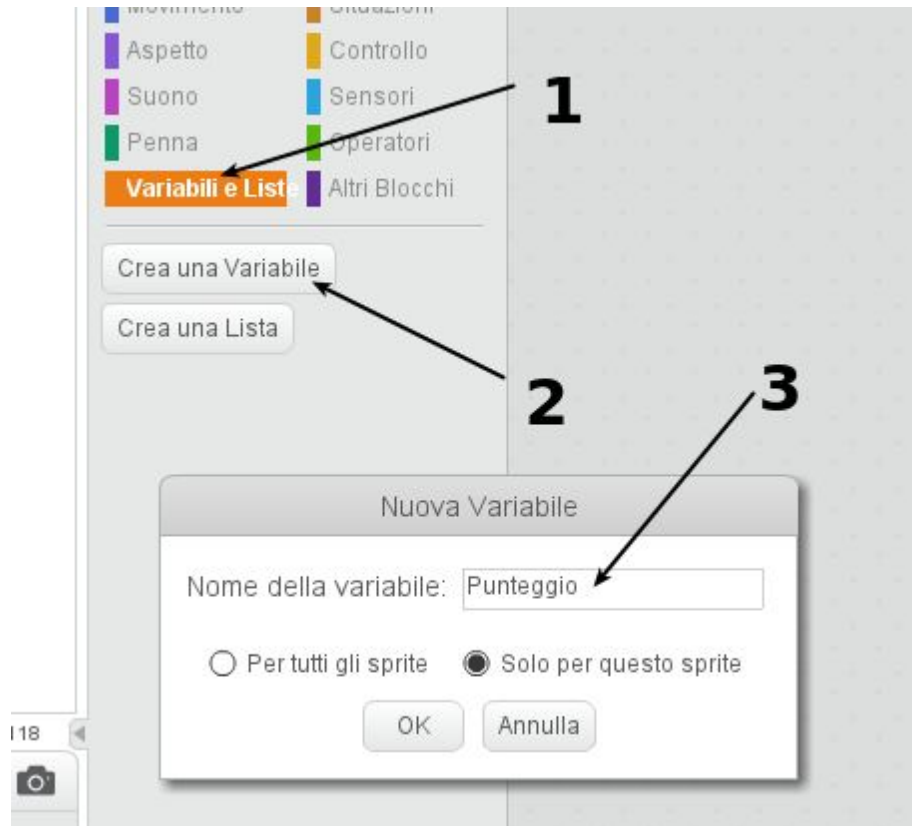
Praticamente sfruttiamo il colore superiore del sacchetto per far “capire” al dolcetto che è stato preso. Se dopo questa modifica si prova a eseguire nuovamente il programma, adesso quando mettiamo il sacchetto sotto il dolcetto, questo sparisce dando proprio l'impressione di finire dentro il sacchetto!

4. Collezionare punti

Per dare interesse al gioco e fare gare con gli amici e le amiche aggiungiamo il punteggio! Ogni volta che un dolcetto viene catturato questo ci “regala” dei punti (o delle calorie?!?!). Associamo quindi al fantasma una variabile “**Punteggio**” che inizializziamo a zero all'avvio.

Per fare questo, effettuare le seguenti operazioni:

- selezionare lo sprite del fantasma e andare nell'area "Script"
- **1** selezionare il blocco "Variabili e liste"
- **2** cliccare su "Crea una variabile".
- **3** dare alla variabile il nome "Punteggio".
- Indicare che la variabile è "Solo per questo Sprite"



Aggiungere quindi nella parte iniziale dello script del fantasma il blocco:



Come si fa ad aumentare il punteggio del fantasma? Deve accorgersi di aver preso una caramella. Si può fare in molti modi, ma il più semplice consiste nell'utilizzare i **SEGNALI**. Quando il dolcetto "Donut" viene preso (tocca il colore verde) invia al fantasma un segnale con il suo nome. Quando il fantasma riceve il segnale "Donut" aumenta la sua variabile del valore di questo dolcetto. Supponiamo per esempio che valga 5. Ecco il nuovo codice del dolcetto (con evidenziata la parte da aggiungere);



E il nuovo script da aggiungere al fantasma:



Come vedete abbiamo aggiunto anche un suono che viene generato tutte le volte che “catturiamo” un dolcetto. Per poter aggiungere l’istruzione con il suono occorre selezionare il tab “Suoni” e quindi aggiungere un nuovo suono cliccando sull’icona a forma di altoparlante (Scegli un suono dalla libreria). Scegliete il suono che più vi aggrada. Noi vi proponiamo “fairydust”. Provate a rigiocare! Adesso vedrete che tutte le volte che acchiapperete il dolcetto il vostro punteggio aumenterà

5. Aumentare i dolcetti!

Per aumentare i dolcetti basta utilizzare il meccanismo della “clonazione”. Periodicamente generiamo un nuovo clone del nostro dolcetto. Per rendere più interessante la cosa utilizziamo più costumi in modo da generare dolcetti sempre diversi. Disegnate caramelle, cioccolatini o i vostri dolci preferiti, oppure scegliete cose da mangiare tra gli sprite presenti nella libreria di Scratch. Noi che siamo pigri li abbiamo presi tutti dalla libreria! (Vedi a lato).

Per rendere più variabile il gioco inseriamo una variabile per definire la velocità di caduta e diamogli un valore a caso.

Alla fine il codice del dolcetto risulterà quello riportato nella prossima pagina.

ESERCIZIO: come si può fare ad assegnare un punteggio diverso ad ogni tipo di dolcetto?

6. Esseri dispettosi

Definiamo adesso un nuovo Sprite che disturba invece il nostro fantasma muovendosi liberamente nello schermo e mangiandosi i dolcetti quando li incontra. Inoltre quando il fantasma viene toccato da questi “disturbatori”, vengono persi punti e il fantasma comincia a svanire.



Inseriamo un nuovo sprite utilizzando per esempio qualche elemento dalla libreria (per esempio il pipistrello), oppure utilizzando il bellissimo Jack Skellington di “Nightmare before Christmas” messo a disposizione dai mentor. Facciamo muovere liberamente nello stage il nostro Jack per esempio con il codice sempre nella prossima pagina




```

quando si clicca su
per sempre
  crea clone di me stesso
  attendi 10 secondi
quando vengo clonato
  nascondi
  vai in primo piano
  porta dimensione al 20 %
  porta stile rotazione a sinistra-destra
  punta in direzione 180
per sempre
  passa al costume numero a caso tra 1 e 5
  porta Velocità a numero a caso tra 2 e 7
  attendi numero a caso tra 1 e 3 secondi
  vai a x: numero a caso tra -220 e 220 y: 180
  mostra
  ripeti fino a quando posizione y < -170 o sta toccando il colore
  fai Velocità passi
  se sta toccando il colore allora
    invia a tutti Donut
  nascondi

```

Annotations in the image:

- Clonazione (points to the cloning loop)
- Clonazione (points to the 'when cloned' event)
- Scelta casuale del costume (points to the costume selection block)
- Velocità (points to the velocity selection and movement blocks)

NUOVO CODICE DEI DOLCETTI

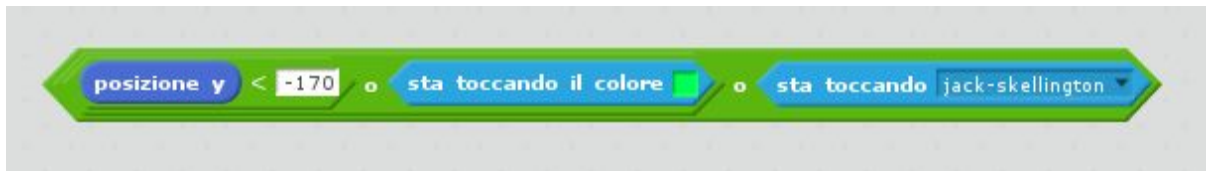
```

quando si clicca su
nascondi
vai a x: numero a caso tra -200 e 200 y: numero a caso tra 150 e 0
mostra
per sempre
  fai numero a caso tra 2 e 7 passi
  ruota di numero a caso tra 1 e 2 gradi
  rimbalza quando tocchi il bordo

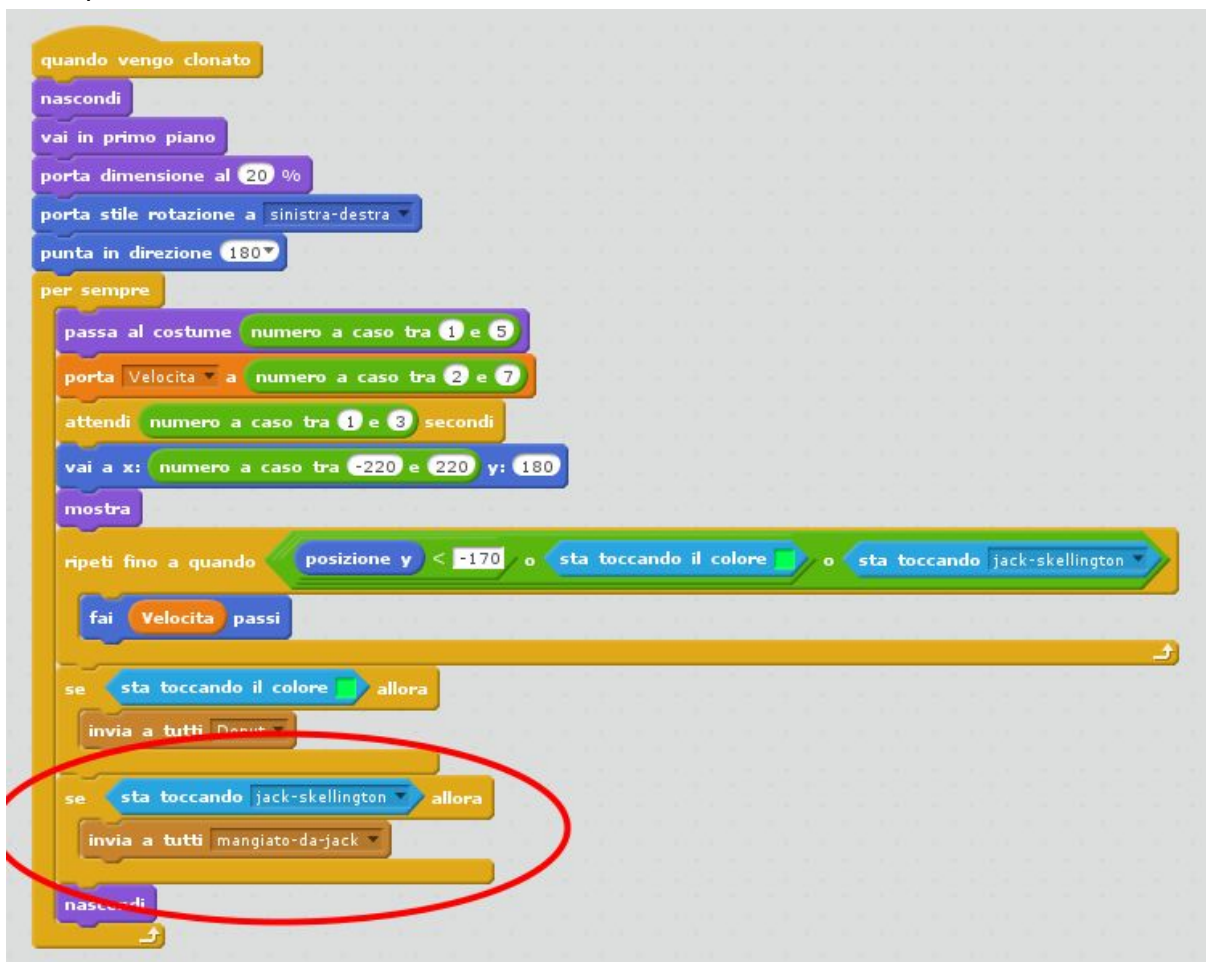
```

CODICE DI JACK SKELLINGTON

Modifichiamo adesso ulteriormente il codice dei dolcetti: il dolcetto deve sparire **anche** quando viene toccato da Jack. La condizione che determina fino a quando il dolcetto deve continuare a cadere diventa la seguente:



E complessivamente il codice del dolcetto diventa:

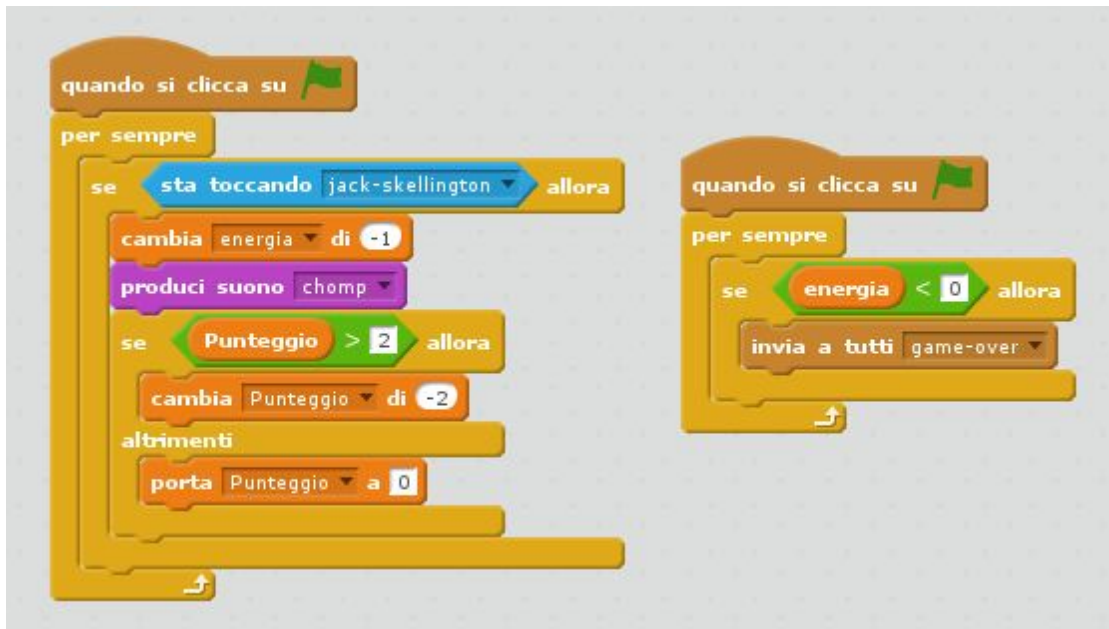


Infine occorre aggiungere allo Sprite Jack la gestione del messaggio “mangiato-da-jack” per fargli fare quello che vogliamo al momento che mangia un dolcetto (per esempio emettere un suono).



Adesso facciamo in modo che quando il fantasma viene toccato da Jack perde forza e quando la sua energia termina il gioco termina.

Inserire una variabile "energia" per lo sprite Fantasma e aggiungere i seguenti script:



NUOVI SCRIPT PER IL FANTASMA

Infine definiamo un nuovo sfondo per lo Stage per rappresentare la condizione di "Game Over" (creare un nuovo sfondo dove scriveremo "Il gioco è finito" o "Game Over" se vi piace l'inglese).

Aggiungiamo infine il seguente codice allo stage:

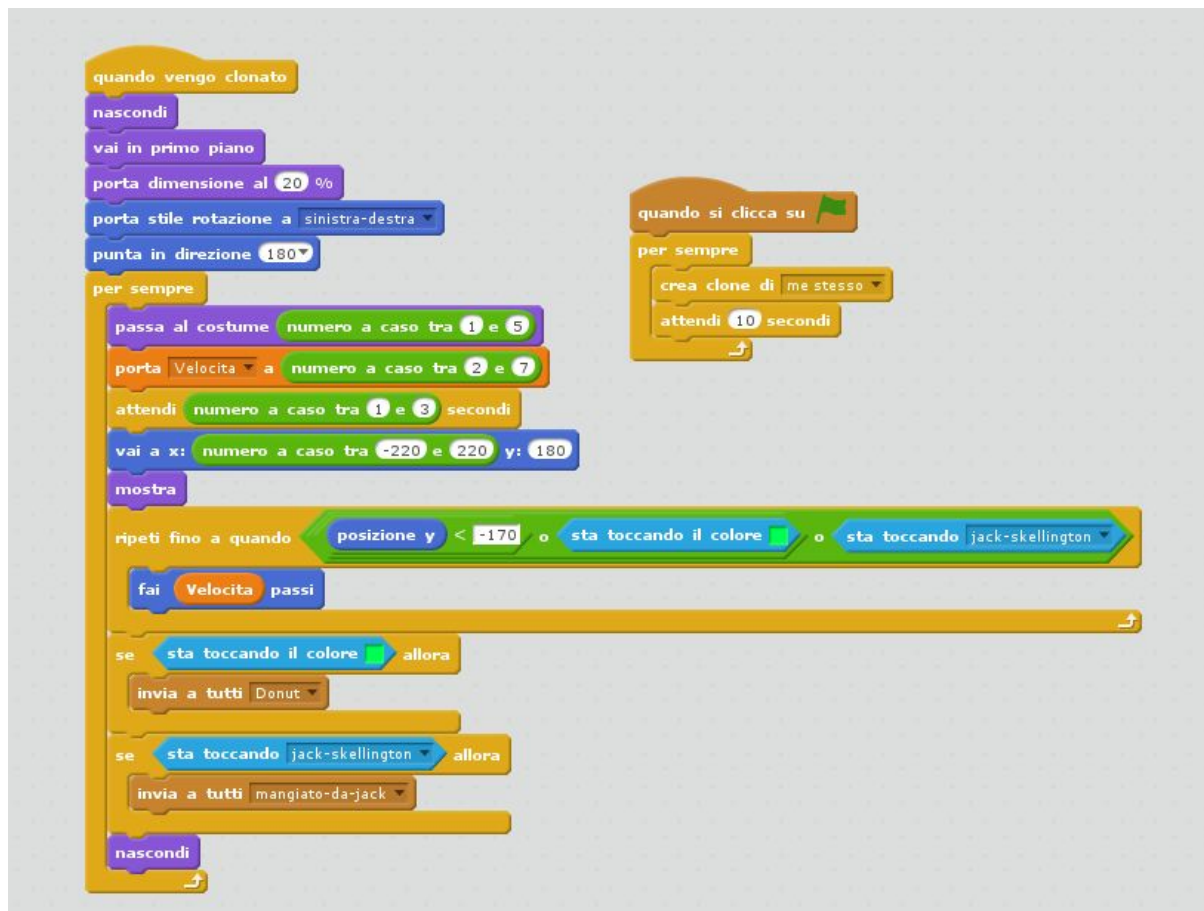


Appendice (per i pigri): tutti gli script

STAGE



DONUT



FANTASMA

```
quando si clicca su [bandierina]
nascondi
porta Punteggio a 0
porta energia a 100
porta stile rotazione a sinistra-destra
vai a x: 0 y: -130
mostra
per sempre
  se [tasto freccia destra premuto] allora
    punta in direzione 90
    fai 5 passi
  se [tasto freccia sinistra premuto] allora
    punta in direzione -90
    fai 5 passi

quando ricevo Donut
produci suono fairydust
cambia Punteggio di 5

quando si clicca su [bandierina]
per sempre
  se [sta toccando jack-skellington] allora
    cambia energia di -1
    produci suono chomp
  se [Punteggio > 2] allora
    cambia Punteggio di -2
  altrimenti
    porta Punteggio a 0

quando si clicca su [bandierina]
per sempre
  se [energia < 0] allora
    invia a tutti game-over
```

JACK SKELLINGTON

```
quando si clicca su [bandierina]
nascondi
vai a x: numero a caso tra -200 e 200 y: numero a caso tra 150 e 0
mostra
per sempre
  fai numero a caso tra 2 e 7 passi
  ruota di numero a caso tra 1 e 2 gradi
  rimbalza quando tocchi il bordo

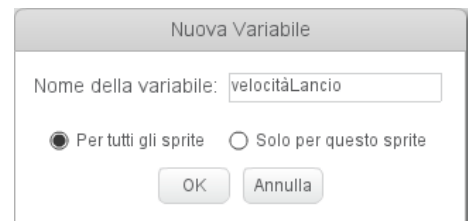
quando ricevo mangiato-da-jack
produci suono chee chee
```

CODERDOJO FIRENZE – TUTORIAL “JUMPY MONKEY” (tratto da “Coding games in Scratch” di Jon Woodcock)

Scopo del gioco: lanciare la scimmia per prendere le banane evitando l’albero.

1. Il lanciatore

Eliminare lo sprite gatto. Dalla libreria selezionare uno sprite “Arrow”. Creare una variabile “velocitàLancio” e assicurarsi di associarla a “tutti gli sprite”.



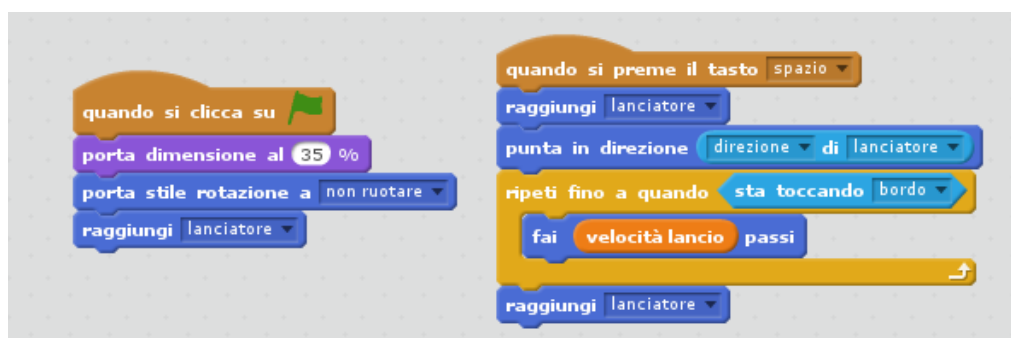
Realizzare i seguenti script per lo sprite:



Il lanciatore si posiziona nella parte bassa a sinistra dello stage di gioco. Tramite le frecce destra e sinistra si regola la direzione di lancio, tramite le frecce su e giù si regola la velocità di lancio. **[NOTA MENTOR: Spiegare il concetto di “Evento” o “Situazione” (blocchi marroni): pressione tasti, mouse click, movimento, suoni...]**

2. La scimmia

Selezionare uno degli sprite che rappresenta una scimmietta (o comunque un qualunque protagonista a piacere) e realizzare il seguente codice:



[NOTA MENTOR: Spiegare il blocco "ripeti fino a quando", più flessibile del "per sempre" e "ripeti"]

3. Le banane

Selezionare un nuovo sprite da libreria: il casco di banane. Useremo la clonazione per creare più banane, posizionandole in posizioni a caso sullo schermo. Definiamo anche una variabile "numBanane" per decidere quanti cloni fare. Ecco il codice:



Bananas

```
quando si clicca su [bandiera]
nascondi
porta numbananas a 5
ripeti numbananas volte
  crea clone di me stesso
quando vengo clonato
  vai a x: numero a caso tra 0 e 200 y: numero a caso tra -140 e 140
  porta dimensione al numero a caso tra 50 e 100 %
  porta effetto colore a numero a caso tra -10 e 20
  mostra
  attendi fino a quando sta toccando Monkey1
  cambia numbananas di -1
  se numbananas = 0 allora
    invia a tutti game over
  elimina questo clone
```

Adesso si può provare a giocare: cercare di prendere tutti i caschi di banana lanciando la scimmia.

4. La palma (l'ostacolo...)

Selezionare l'ennesimo sprite da libreria: la palma.

Posizionarlo in basso leggermente spostato a sinistra rispetto al centro.

Modificare il codice della scimmia nel seguente modo:



Palmtree

```
ripeti fino a quando sta toccando bordo
  fai velocità lancio passi
ripeti fino a quando (sta toccando bordo) o (sta toccando Palmtree)
  fai velocità lancio passi
```

ERA

DIVENTA

[NOTA MENTOR: Spiegare gli operatori logici "o", "e" e "non"]

A questo punto il gioco sarà diventato più difficile e probabilmente ci saranno delle banane impossibili da raggiungere (nascoste dietro la palma...).

4. La gravità (what goes up must come down...)

Aggiungiamo due nuove variabili: **gravità** e **velocità Caduta**, e modifichiamo gli script della scimmia nel seguente modo (notare le parti aggiunte):



Provare a rieseguire il gioco: adesso la gravità fa cadere la scimmia, permettendogli di raggiungere le banane nascoste dietro l'albero.

Notare come l'uso della gravità permette di rendere il gioco più realistico. Provare a giocare con il parametro "gravità" per vedere cosa succede. Cosa succede se gli viene assegnato un valore positivo?

5. Creiamo uno sfondo

Divertiamoci ad aggiungere o ancor meglio disegnare uno sfondo. Per esempio:



5. Gestione del fine gioco (Game Over)

Abbiamo già previsto un "segnale" per comunicare la condizione di fine gioco (vedere lo script della banana, che genera il segnale "gameOver").

Creare un nuovo sprite, questa volta disegnandolo. Creare un'ellisse con la scritta dentro come a lato. Notare lo spazio tra le parole "effettuato" e "lanci".

Definiamo una variabile di nome "lanci" e scriviamo il seguente codice per lo sprite di fine gioco:



Alla fine della partita apparirà la variabile con il numero dei lanci: la prima volta che appare posizionarla tra le parole "effettuato" e "lanci". Le volte successive si ricorderà la sua posizione.

5. HACKING & TWEAKING

a) Facciamo un po' di rumore

Aggiungiamo effetti sonori ai momenti chiave del gioco: per esempio al momento del lancio (*boing*), o quando la scimmia prende una banana (*chomp*) o quando sbatte contro l'albero.

b) Cambiamo il modo di visualizzare le variabili.

Usare per esempio un cursore (o slide) per impostare la gravità (impostare anche il massimo e minimo correttamente).

Cosa succede se la gravità è positiva? Come occorre modificare il gioco per fare sì che sia comunque interessante?

[NOTA MENTOR: la "fisica del gioco". Nella realizzazione di molti tipi di videogiochi è importante realizzare delle simulazioni degli effetti fisici del mondo reale come per esempio la caduta, il rimbalzo, ecc. Ma nei videogiochi non siamo limitati dalle leggi fisiche del mondo reale, possiamo inventarci giochi che hanno una "fisica" tutta loro]

c) Aggiungiamo più tipi di frutta

Aggiungiamo anche il punteggio e associato a frutti diversi punteggi diversi...

d) Lotta contro il tempo

Aggiungere un cronometro che scorre e fa terminare il gioco quando scade.

e) Controllare i parametri di lancio con il mouse

...invece che con la tastiera. Modificare il lanciatore usando i blocchi a lato.



f) Baco o funzionalità??

Avete notato che è possibile cambiare la velocità della scimmia mentre è in volo usando i tasti freccia su e giù? Per risolvere il problema basta introdurre un'ulteriore variabile "**velocitàScimmia**" e assegnargli il valore di **velocitàLancio** al momento del lancio. Usare poi **velocitàScimmia** nell'algoritmo del movimento della scimmia. Oppure non è più divertente lasciare le cose come stanno? Baco o funzionalità?

g) Ulteriori pericoli

Aggiungere un ulteriore ostacolo più pericoloso della palma che se toccato fa terminare il gioco (per esempio un serpente gigante...)

h) Banane mobili

Rendere le banane più difficili da prendere facendole muovere a spasso per lo schermo



Gioco dell'elicottero:

Si comanda il volo di un elicottero con le frecce (esclusa sinistra) che viene continuamente spinto verso sinistra. La posizione dell'elicottero è preferenzialmente sul bordo sinistro dello schermo. L'elicottero può sparare un razzo premendo spazio, e può essere colpito da proiettili che arrivano dal lato destro dello schermo.

I proiettili sono di due tipo, uno sparato “a caso”, l'altro sparato verso l'elicottero.

L'elicottero può scansare i proiettili o colpirli con il razzo.

Se l'elicottero viene colpito da un proiettile precipita e esplode.

Varianti:

Gestione cronometro e punteggio

Gestire salute elicottero

Inserire un bersaglio che l'elicottero deve proteggere

Aumentare via via la difficoltà

Se l'elicottero tocca i bordi precipita

Script Elicottero

Da fare successivamente (dopo i proiettili)

```
quando si clicca su [bandierina verde]
  passa al costume [helicopter]
  vai a x: -152 y: 87

quando si clicca su [bandierina verde]
  per sempre
    se [tasto freccia su premuto] allora
      cambia y di 5
    se [tasto freccia giù premuto] allora
      cambia y di -5
    se [tasto freccia destra premuto] allora
      cambia x di 5
    se [tasto spazio premuto] allora
      invia a tutti [sparo]

quando ricevo [colpito]
  passa al costume [helicopter2]
  scivola in 1 secondi a x: [posizione x] y: -180
  passa al costume [bang]
  attendi 1 secondi
  invia a tutti [Fine]

quando si clicca su [bandierina verde]
  per sempre
    cambia x di -3
    attendi 0.3 secondi
```


Script razzo (freccia)

quando si comincia viene nascosto, quando riceve il messaggio “sparo” andiamo all'elicottero

```
quando si clicca su [bandierina verde]
nasconditi

quando ricevo [sparo]
raggiungi [Helicopter]
mostrati
scivola in 1 secondi a x: 234 y: [posizione y di Helicopter]
nasconditi
```

Proiettile 1

Parte dal bordo destro e va verso il bordo sinistro

```
quando si clicca su [bandierina verde]
per sempre
  nasconditi
  attendi [numero a caso tra 1 e 2] secondi
  vai a x: 234 y: [numero a caso tra -180 e 180]
  mostrati
  scivola in 2 secondi a x: -234 y: [numero a caso tra -180 e 180]

quando si clicca su [bandierina verde]
per sempre
  se [sta toccando Arrow1] allora
    nasconditi

quando si clicca su [bandierina verde]
per sempre
  se [sta toccando Helicopter] allora
    invia a tutti colpito
```

Il proiettile 2 è uguale, salvo che va verso l'elicottero per cui per muoverlo si cambia l'istruzione scivola:

```
scivola in 2 secondi a x: [posizione x di Helicopter] y: [posizione y di Helicopter]
```