

# IMPARIAMO A PROGRAMMARE CON SCRATCH

**Lezione 2: Cos'è un Algoritmo e il secondo tutorial**

# PROGRAMMARE

IL computer ha la caratteristica di seguire ciecamente ed esattamente le istruzioni, che gli vengono fornite, e di essere molto veloce.

Non è però in grado di fare ragionamenti o di pensare da solo.

**Programmare significa riuscire a dare istruzioni dettagliate, precise e ben organizzate.**

Dobbiamo anche imparare a parlare la sua lingua.

# COS'È UN ALGORITMO?

L'algoritmo è un insieme ordinato di operazioni non ambigue ed effettivamente computabili che, quando eseguito, produce un risultato e si arresta in tempo finito.



# ALGORITMO

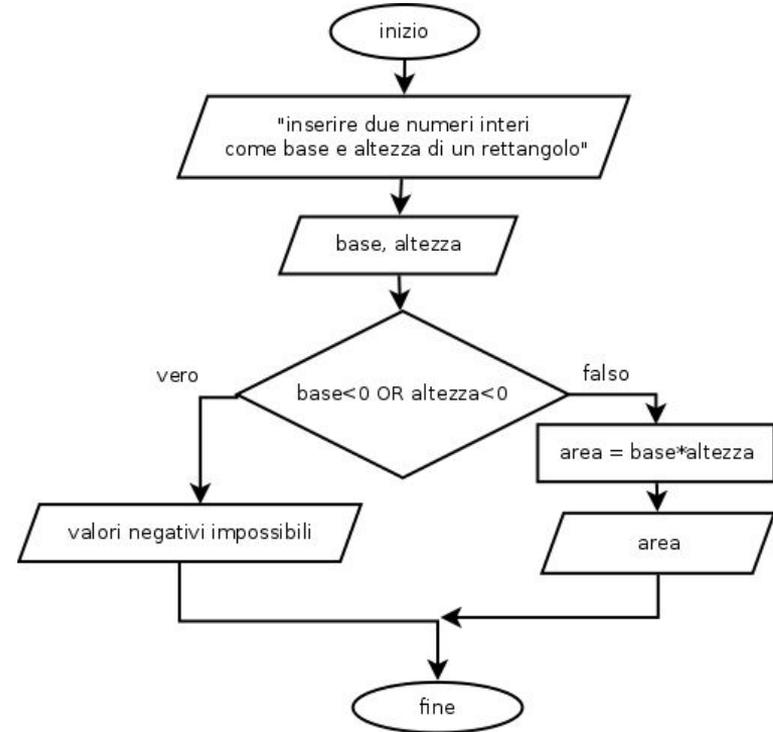
Seguire una ricetta o dare indicazioni stradali per raggiungere un luogo o spiegare a qualcuno come allacciarsi le stringhe delle scarpe significa aver già sperimentato cosa vuol dire dare istruzioni dettagliate e precise. Proviamo a dare delle istruzioni per preparare un toast.

Prepariamo un toast.

- 1° Prendere due fette di pane per i toast
- 2° Mettere una fetta di prosciutto cotto sopra una delle due fette di pane
- 3° Mettere una fetta di formaggio sopra la fetta di prosciutto che è sopra la fetta di pane
- 4° Mettere la seconda fetta di pane sopra il formaggio che è sopra il prosciutto che è sopra la prima fetta di pane

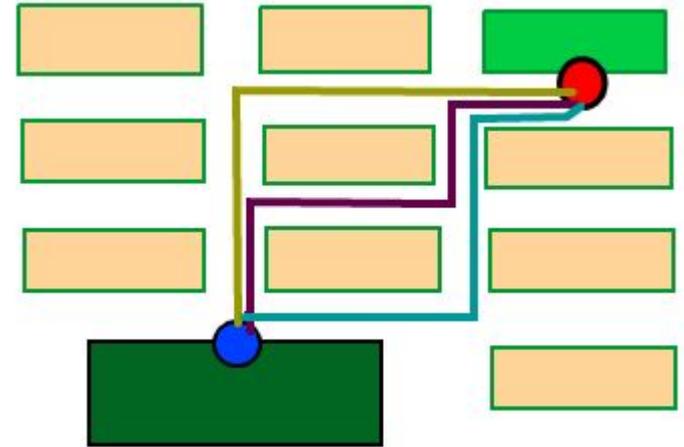
# ALGORITMO

In questo modo abbiamo scritto un algoritmo, cioè una sequenza finita di passi necessari per risolvere un problema.



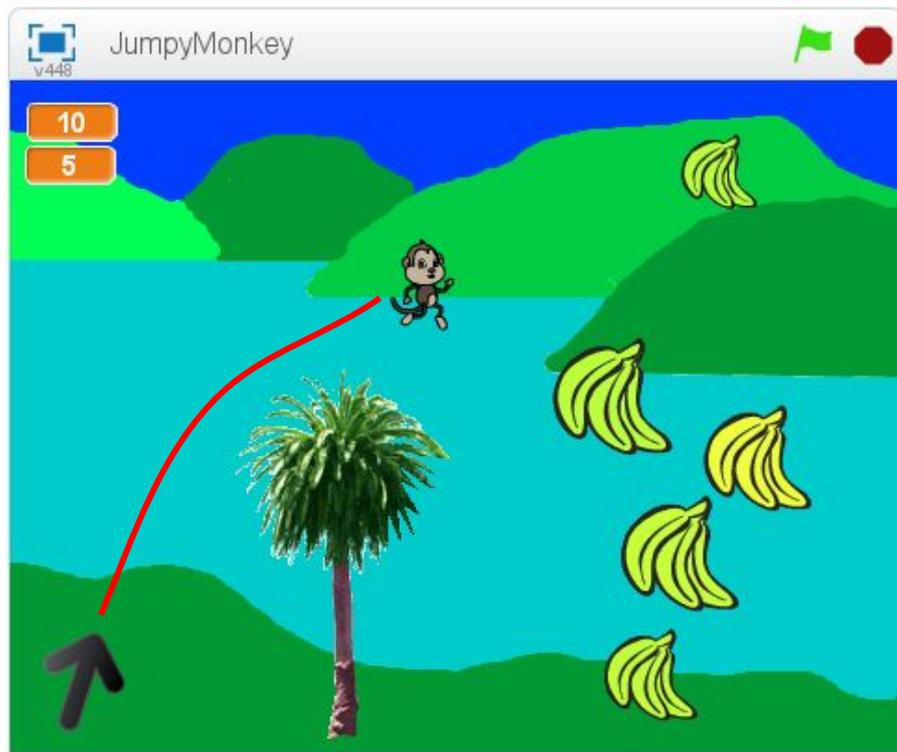
# DIVERSI ALGORITMI PER LO STESSO PROGRAMMA

Ci possono essere diversi algoritmi per risolvere lo stesso problema. Esempio: dare indicazioni su una mappa per raggiungere un hotel dalla stazione.



# IL SECONDO TUTORIAL: JUMPY MONKEY

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



# COMINCIAMO...

Dovremo creare 4 sprite:

- Un lanciatore
- Una scimmia
- Caschi di banane
- Una palma



# 1. IL LANCIATORE



Nuova Variabile

Nome della variabile:

Per tutti gli sprite  Solo per questo sprite

Eliminare lo sprite gatto.

Dalla libreria selezionare uno degli sprite "Arrow" (per esempio **Arrow2**).

Creare una variabile di nome "**velocitàLancio**" e assicurarsi di associarla a "tutti gli sprite".

Realizzare gli script al lato per lo sprite.

A collection of Scratch scripts for the Arrow2 sprite. The scripts are as follows:

- Script 1:** "quando si clicca su" (when green flag clicked) → "porta velocità lancio a 10" (set velocity launch to 10) → "vai a x: -200 y: -140" (go to x: -200 y: -140) → "vai in primo piano" (go to front).
- Script 2:** "quando si preme il tasto freccia su" (when space key pressed) → "se velocità lancio < 20 allora" (if velocity launch < 20 then) → "cambia velocità lancio di 0.1" (change velocity launch by 0.1).
- Script 3:** "quando si preme il tasto freccia sinistra" (when left arrow key pressed) → "ruota di 1 gradi" (rotate 1 degrees).
- Script 4:** "quando si preme il tasto freccia destra" (when right arrow key pressed) → "ruota di 1 gradi" (rotate 1 degrees).
- Script 5:** "quando si preme il tasto freccia giù" (when down arrow key pressed) → "se velocità lancio > 1 allora" (if velocity launch > 1 then) → "cambia velocità lancio di -0.1" (change velocity launch by -0.1).

# COSA STIAMO FACENDO?

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.



The image shows a Scratch workspace titled 'Untitled' with version 'v448'. A 'quando si preme il tasto' (when key pressed) event block is selected, with 'freccia destra' (right arrow) chosen from the dropdown menu. To the right, a flowchart is visible, featuring Scratch characters like the orange cat, a green dragon, and a yellow bird. The flowchart starts with a 'START' block, followed by 'Move 10 steps', 'if on edge, bounce', and a decision diamond 'Is it touching the other sprite?'. The 'Yes' path leads to 'Change colour', and the 'No' path leads to 'when clicked', 'ask What your name?', 'answer for secs', and 'wait for secs'.

## Eventi e Situazioni

Il computer è in grado di interagire con il mondo esterno tramite “eventi”. Esempi di eventi sono la pressione di tasti e i click del mouse ma anche livelli sonori o movimenti rilevati dalla webcam

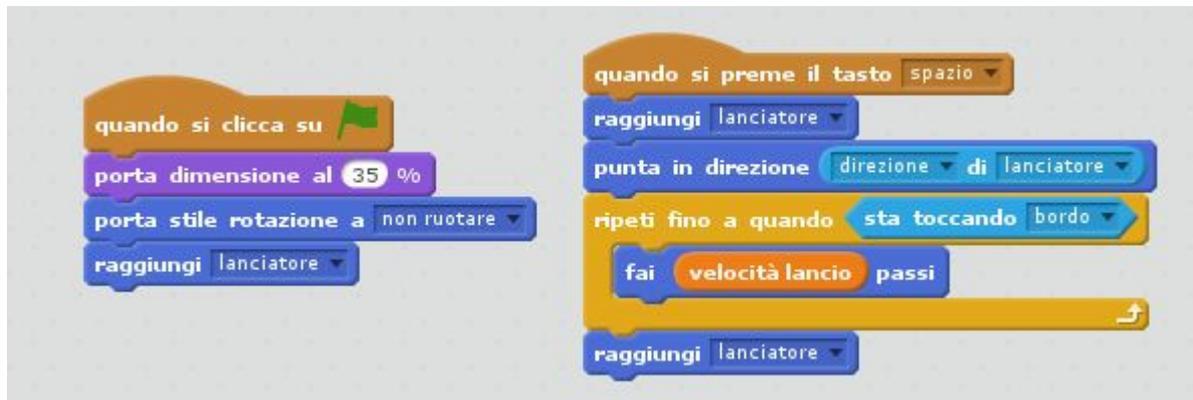


## 2. LA SCIMMIA

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



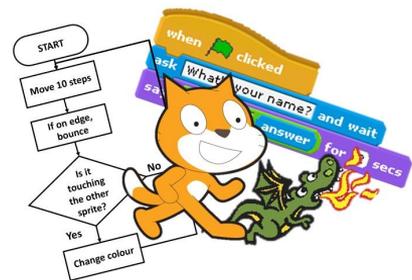
Monkey1



# L'ANGOLO DELL'ESPERTO! << RIPETI FINO A QUANDO >>

Più versatile dei blocchi “per sempre” e “ripeti” permette di eseguire delle azioni fino a che non si verifica una certa condizione

Condizione di uscita



### 3. LE 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

Selezionare un nuovo sprite da libreria: il casco di banane



Bananas

```
quando si clicca su [bandiera verde]
  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
```

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.



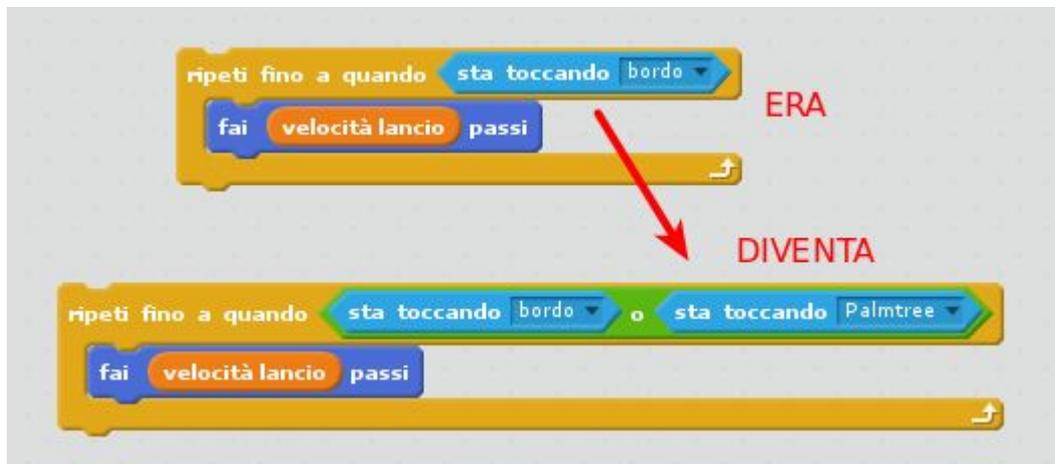
# LA PALMA DEVE COSTITUIRE UN OSTACOLO PER LA SCIMMIA

Se la scimmia colpisce l'albero il lancio fallisce.

Modificare quindi il codice della scimmia come indicato a lato



Monkey1

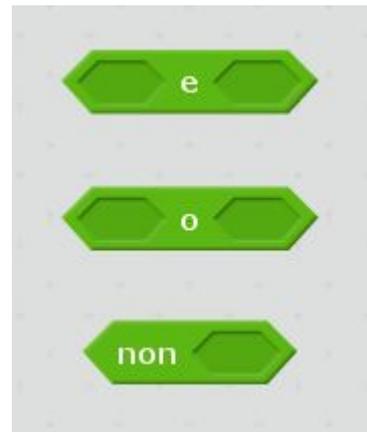
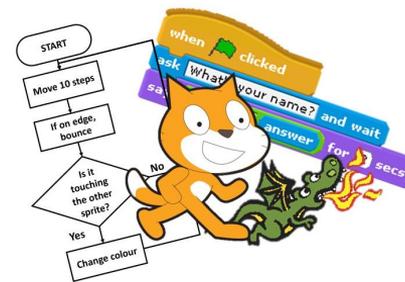


# L'ANGOLO DELL'ESPERTO <<OPERATORI LOGICI>>

Finora con i blocchi **“Se ... Allora”** abbiamo controllato solo condizioni singole del tipo: “Se sta toccando... allora”.

Se vogliamo controllare due cose contemporaneamente possiamo utilizzare gli operatori logici **“e”** ed **“o”**. Il primo operatore ritorna **“Vero”** solo se entrambe le condizioni controllate sono vere, il secondo se almeno una delle due lo è.

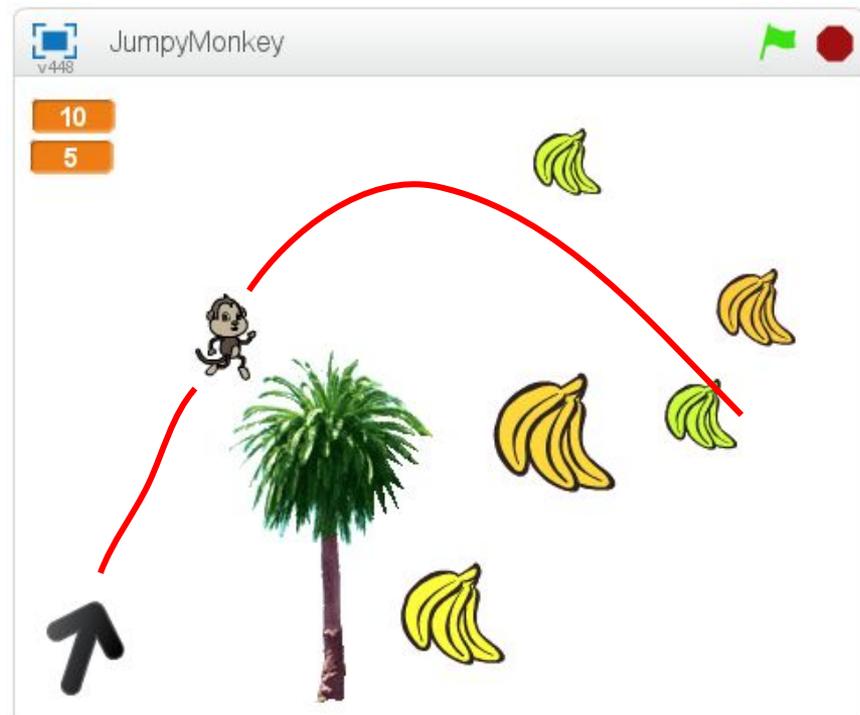
L'operatore logico **“non”** invece ritorna **“Vero”** se la condizione è falsa



# COME RAGGIUNGERE LE BANANE DIETRO LA PALMA??

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

Cosa possiamo fare?



# QUELLO CHE SALE, DEVE SCENDERE... SIMULIAMO LA GRAVITÀ

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



# UN TOCCO DI REALISMO RENDE PIÙ INTERESSANTE IL GIOCO

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

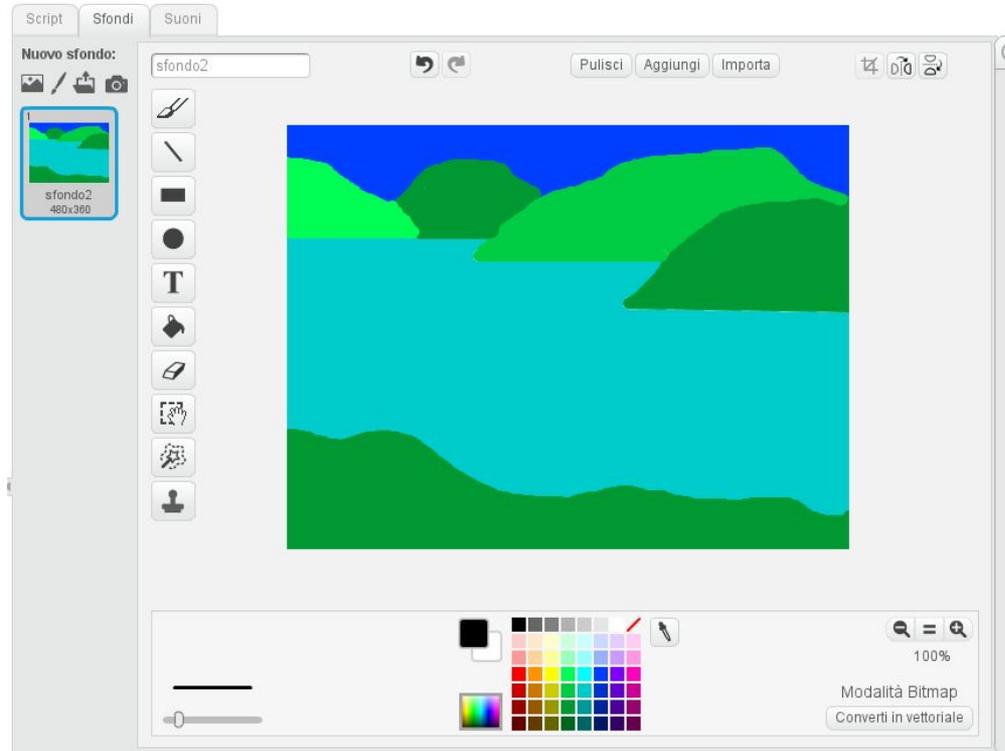
Notare come l'uso della gravità permette di rendere il gioco più realistico. Provare a giocare con il parametro "gravità" per vedere cosa succede.

Cosa succede se gli viene assegnato un valore positivo?

Nei giochi oltre a simulare la realtà possiamo anche creare dei mondi impossibili...

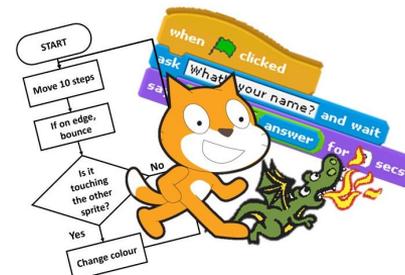
# A PROPOSITO DI REALISMO: CREIAMO UNO SFONDO

Dopo tanto  
programmare un  
po' di relax:  
proviamo a  
disegnare uno  
sfondo



# GESTIONE DELLA FINE DEL GIOCO: GAME OVER

Nello script delle banane abbiamo visto l'uso di una misteriosa istruzione:



Nella terminologia di Scratch **game over** è un **Segnale**.

I segnali sono un modo per inviare un messaggio a tutti gli sprite che popolano il nostro gioco in modo che possano essere effettuate specifiche operazioni in modo sincrono all'accadere di determinati eventi.

# USARE UN SEGNALE PER TERMINARE IL GIOCO

Nel nostro caso sfrutteremo il segnale per terminare il gioco facendo nel contempo apparire uno sprite speciale che riepiloga i punti fatti

Disegnare un nuovo sprite come indicato a lato (ricordarsi di lasciare un spazio tra le parole “effettuato” e “lanci”)



# GLI SCRIPT PER LO SPRITE DI FINE GIOCO

BEN FATTO!

hai effettuato lanci

Definiamo una variabile di nome “**lanci**” e scriviamo il codice a lato per lo sprite di fine gioco



# POSIZIONARE CORRETTAMENTE LA VARIABILE “NUMERO LANCI”

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.







# HACKS & TWEAKS (3)

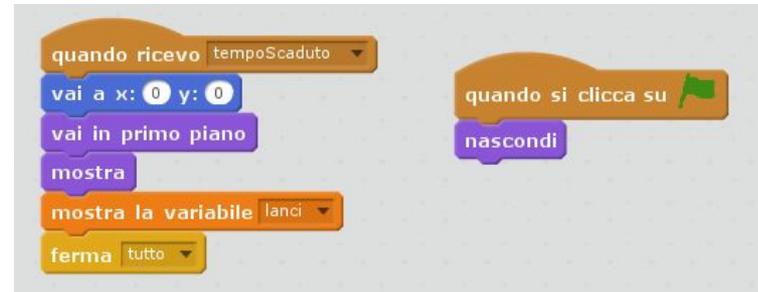


d) **Lotta contro il tempo:** Aggiungere un cronometro che scorre e fa terminare il gioco quando scade

Aggiungere questo script allo sprite scimmia



Aggiungere un nuovo sprite simile a quello di game over per indicare “Tempo Scaduto”





# HACKS & TWEAKS (5)



**g) Ulteriori pericoli:** Aggiungere un ulteriore ostacolo piú pericoloso della palma che se toccato fa terminare il gioco (per esempio un serpente gigante...)

**h) Banane mobili:** Rendere le banane piú difficili da prendere facendole muovere a spasso per lo schermo

**i) Usare Cursori per regolare il valore dei parametri di gioco:** Provare a cliccare con il tasto destro del mouse su una delle variabili e scegliete la modalit  di visualizzazione "Cursore". Sar  piú facile regolarle e sperimentare con il gioco

