

Advanced Scratch: Gatto Raccogli Stelle

ATTENZIONE: Per questo gioco non si parte a scrivere il codice da zero. Farsi dare dai mentor il codice di partenza (file: *CoderDojoAdvancedScratch_Starter.sb2*)

Questo gioco contiene un "motore" che simula la fisica del mondo reale. Nel codice di partenza il codice di questo motore è già scritto.

- **Proviamo intanto a giocare per capire come funziona**

Dopo di chè usiamo scratch per completare il gioco.

Gestione fine partita

Per prima cosa notiamo che il gioco non termina mai.

Aggiungiamo la gestione del fine gioco completando lo script "termina-gioco" del personaggio principale:



Notare come la generazione del segnale "game over" verrà recepito dagli sprite "Piattaforme" e "Limiti" per scomparire. Allo stesso modo è presente il codice che fa riapparire questi sprite al lancio di una nuova partita:



Aggiungiamo lo stesso tipo di blocchi anche allo sprite dei "Collezionabili":



Aggiungiamo nuovi collezionabili

Aggiungiamo un nuovo tipo di collezionabili. Creiamo un nuovo costume. Facciamo in modo che abbia più o meno la stessa dimensione della stella.



Abbiamo già una funzione **"scegli-tipo"** che ci permette di selezionare un costume in base a un parametro. E abbiamo un punto dove il costume viene scelto tramite una chiamata a questa funzione. Inoltre usiamo una variabile **"collez-tipo"** per scegliere un tipo di costume. Dobbiamo fare in modo che **collez-tipo** assuma valori casuali, e vogliamo che il nuovo costume abbia una probabilità di 1 a 50 di apparire. Per questo aggiungiamo il seguente pezzo di codice al programma principale, e modifichiamo la funzione **scegli-tipo** in questo modo:



Adesso dobbiamo decidere cosa fare quando il giocatore riesce ad acchiappare il costume "speciale". Per fare questo possiamo utilizzare la funzione **"toccato"**. Regaliamo una nuova vita e lanciamo una **"pioggia di collezionabili"**.

Aggiungiamo il seguente blocco che attiva la pioggia:



Analizzate il codice... Perché la generazione di collezionabili aumenta?

Adesso che abbiamo capito il metodo possiamo aggiungere tutti i "collezionabili speciali" che ci pare: quelli che valgono molti punti, quelli che regalano molte vite, e così via...

Attenti al cane!

Aggiungiamo un avversario, nel nostro caso un cane. Facciamo in modo che abbia più o meno la stessa dimensione del gatto.

Scriviamo il seguente codice:



The image displays two sections of Scratch code for a dog character. The left section shows the initial setup and movement logic, while the right section shows the dog's behavior during a game, including a collision detection with a cat.

Left Section (Dog Setup and Movement):

- quando si clicca su
 - mostra
 - porta numero-passi-cane a 5
 - porta stile rotazione a sinistra-destra
 - vai a x: -60 y: 6
 - punta in direzione 90
 - per sempre
 - fai numero-passi-cane passi
 - se non sta toccando Piattaforme allora
 - punta in direzione direzione * -1

- quando ricevo game over
- nascondi

Right Section (Dog Behavior):

- quando si clicca su
 - reset-gioco
 - per sempre
 - fisica-principale
 - se posizione y < -179 allora
 - nascondi
 - reset-giocatore
 - cambia vite di -1
 - attendi 0.5 secondi
 - mostra
 - se sta toccando Dog1 allora
 - nascondi
 - vai a x: -187 y: 42
 - cambia vite di -1
 - attendi 0.5 secondi
 - mostra
 - se vite < 1 allora
 - termine-gioco

Adesso dobbiamo aggiungere al gatto il codice che gli fa perdere una vita quando entra in contatto con il cane. Dobbiamo fare in modo che il contatto duri pochissimo per evitare che il gatto perda più vite.

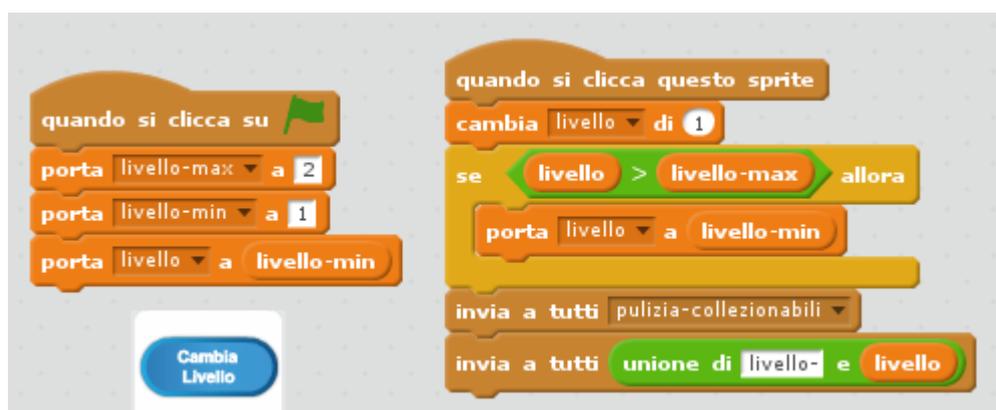
Aggiungiamo il codice a lato al programma principale del gatto:

Nuovi livelli

Aggiungiamo un nuovo livello di gioco. Inseriamo uno sprite "bottonone" che l'utente può utilizzare per cambiare il livello.



Scriviamo del codice "intelligente" per fare in modo che ogni volta che si preme il bottone si cambia di livello indipendentemente da quanti livelli sono disponibili.



E aggiungiamo quindi questo mini pezzo di codice per far sparire tutti i **collezionabili** quando si cambia di livello:



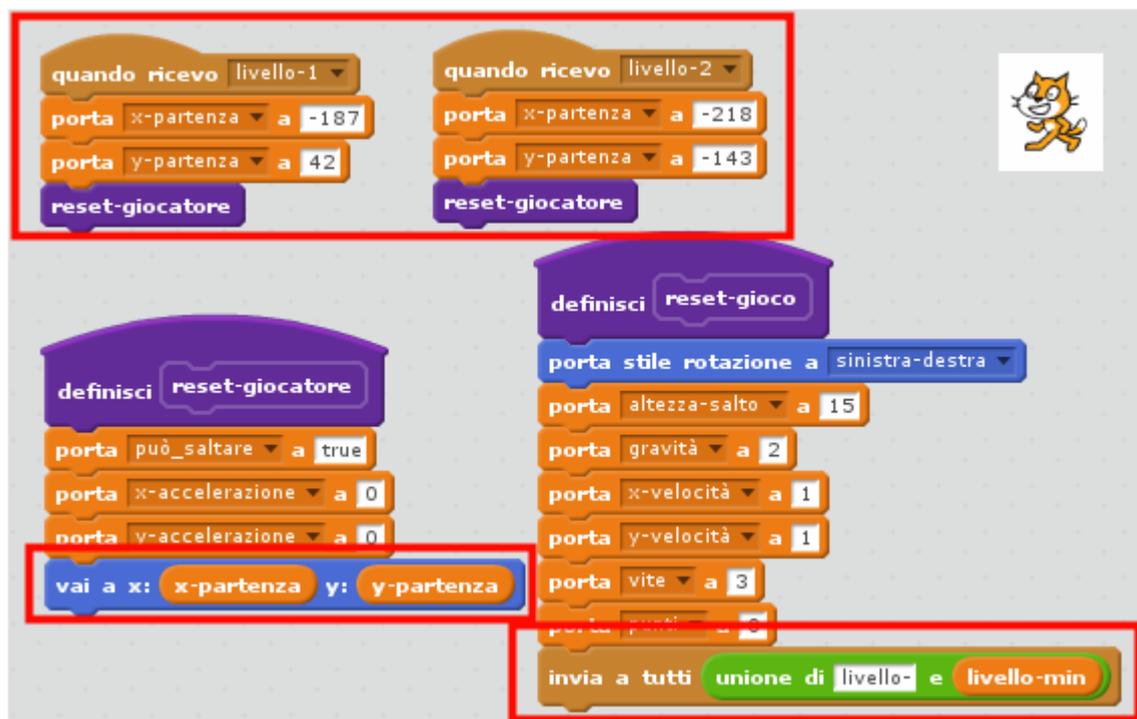
Adesso rimane solo da cambiare le piattaforme. Aggiungiamo il seguente codice per passare a diversi costumi dello sprite **piattaforme**:



Aggiungiamo il codice a lato allo sprite del canino avversario per farlo sparire quando si passa al livello 2 (in alternativa si potrebbe riposizionare su un'altra piattaforma):



Ecco infine come modificare il codice del giocatore:



Piattaforme Mobili

Notiamo che nel secondo costume delle piattaforme c'è una zona centrale vuota che si presta bene per inserire delle piattaforme mobili che, partendo dal basso, ci permettono di far salire il nostro personaggio come se usasse un ascensore.

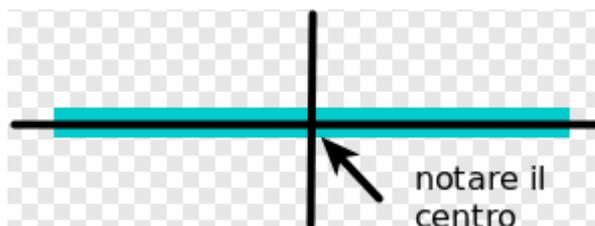
Creiamo quindi un nuovo sprite chiamato **"Piattaforma mobile"** che

questa volta disegneremo da soli. Lo sprite sarà costituito semplicemente da una piattaforma dello stesso colore di quelle del secondo livello.

Inizialmente la posizioneremo in basso e quindi scriveremo il codice per farla salire. Inoltre poiché non vogliamo una sola piattaforma usiamo la funzione clone per crearne più di una.



Lo sprite è composto da un semplice rettangolo dello spessore delle piattaforme già esistenti e della larghezza più o meno della zona centrale del secondo costume dello sprite "piattaforme". Ricordarsi - come sempre - di centrare lo sprite.



Creare una nuova variabile

"crea-piattaforme" e scrivere il seguente codice:

The code consists of several event-driven blocks:

- quando si clicca su** (when clicked):
 - nascondi (hide)
 - per sempre (forever) loop:
 - vai a x: 20 y: -160 (go to x: 20 y: -160)
 - attendi 8 secondi (wait 8 seconds)
 - se crea-piattaforme = true allora (if crea-piattaforme = true then):
 - crea clone di me stesso (create clone of myself)
- quando vengo clonato** (when cloned):
 - mostra (show)
 - per sempre (forever) loop:
 - se posizione y < 180 allora (if y position < 180 then):
 - cambia y di 1 (change y by 1)
 - attendi 0.02 secondi (wait 0.02 seconds)
 - altrimenti (otherwise):
 - elimina questo clone (delete this clone)
- quando ricevo livello-1** (when receive level-1):
 - porta crea-piattaforme a false (set crea-piattaforme to false)
 - nascondi (hide)
- quando ricevo livello-2** (when receive level-2):
 - porta crea-piattaforme a true (set crea-piattaforme to true)
- quando ricevo game over** (when receive game over):
 - nascondi (hide)
 - porta crea-piattaforme a false (set crea-piattaforme to false)

Non abbiamo però ancora finito! Come infatti si potrà notare il nostro giocatore cadrà attraverso le piattaforme! Perché? Semplice: il nostro giocatore ancora ignora l'esistenza di queste nuove piattaforme. Dobbiamo modificare il codice del giocatore in modo che dove compaiono i blocchi di controllo per decidere se sta toccando o non sta toccando le piattaforme si tenga conto anche delle piattaforme mobili:

The code shows two modifications to collision detection logic:

- sta toccando Piattaforme** (is touching Platforms) is replaced by **sta toccando Piattaforme o sta toccando Piattaforme Mobili** (is touching Platforms or is touching Mobile Platforms).
- non sta toccando Piattaforme** (is not touching Platforms) is replaced by **non sta toccando Piattaforme e non sta toccando Piattaforme Mobili** (is not touching Platforms and is not touching Mobile Platforms).

E per continuare a divertirsi...

- **Tabella degli "High Scores"**: Tenere traccia del nome e dei punteggi dei giocatori più bravi. Alla fine del gioco, se il punteggio è abbastanza alto da entrare in classifica, utilizzare il blocco "chiedi" dei sensori per avere il nome del giocatore e mostrare la tabella aggiornata dei punteggi più alti
- **Nuovi Collezionabili**: aggiungere nuovi tipi di collezionabili che portano dei premi speciali:
 - Immunità ai nemici
 - Più vite
 - Giocatore più grande o più piccolo
- **Livelli che cambiano da soli**: fare in modo che i livelli cambino automaticamente quando il giocatore raggiunge un bordo destro o sinistro
- **Livelli da completare**: con il gioco che abbiamo visto i livelli non finiscono mai. Come si può fare per fare in modo che i livelli cambino automaticamente quando si raggiunge un certo punteggio
- **Giochiamo con la Fisica**: provare a cambiare alcuni dei valori nel motore della fisica del gioco cambiando i parametri **gravità**, **altezza-salto**, **x-velocità**, **y-velocità**...
- **Aggiungiamo altri livelli**: realizzare ancora altri livelli, divertirsi a fare gli artisti modificando gli sfondi utilizzando lo **stage**.
- **Effetti sonori**: aggiungere effetti sonori e musica di sottofondo per rendere il gioco più accattivante
- **Non solo gatti**: fare in modo che il giocatore possa scegliere il personaggio con cui giocare. Dare a personaggi diversi caratteristiche diverse: la grandezza, forza del salto, vite, e così via...

Credits

Questo gioco è la versione italiana del tutorial "Advanced Scratch" disponibile sul sito della Fondazione Coderdojo e realizzato da Philip Harney. La versione originale è disponibile al seguente link:

http://kata.coderdojo.com/wiki/Advanced_Scratch