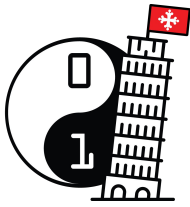


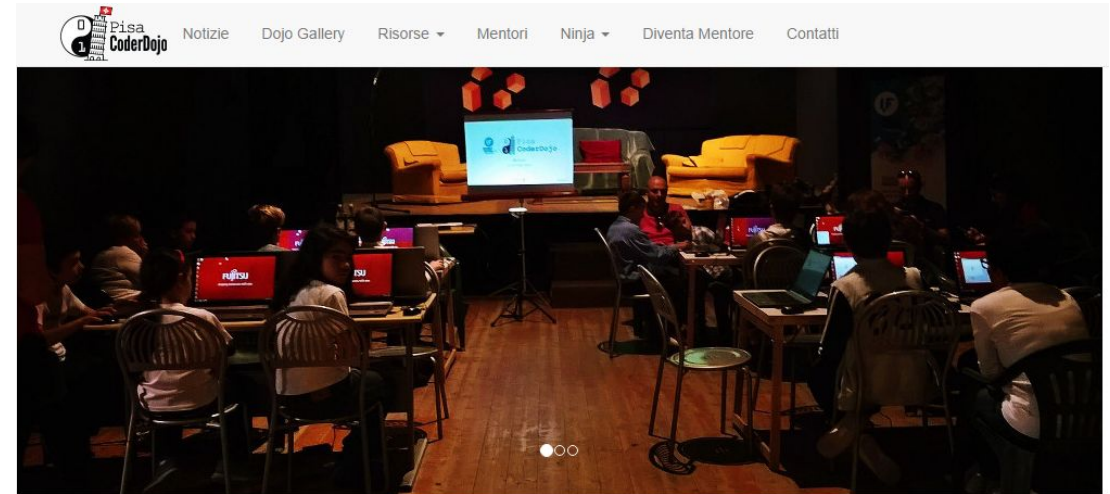
Be Cool!

**#iorestoacasa**



# Chi siamo?

- Movimento internazionale
- Club volontario per insegnare/imparare la programmazione
- 40+ incontri con Python, Scratch, AppInventor, micro:bit, HTML...
- Studenti di Informatica e non solo
- [pisa.coderdojo.it](https://pisa.coderdojo.it), Facebook e Twitter!



## Partecipa!

Pisa CoderDojo si riunisce una volta al mese a [SMS Biblio](#), controlla il nostro calendario e acquista il biglietto gratuito su Eventbrite.

attualmente non ci sono eventi in programma.

## Dojo@Scuola!

Sei un insegnante o un preside di scuole elementari nel Comune di Pisa e vuoi organizzare un Dojo, coinvolgendo una o più classi? Allora [clicca qui](#): ti faremo sapere quando possiamo organizzare **gratuitamente** l'evento nei laboratori di Informatica della tua scuola. Sempre in gamba!

## Che cos'è?

Un Dojo è un'organizzazione volontaria di persone che costituisce, attiva e mantiene un club basandosi sul regolamento etico di CoderDojo al fine di facilitare l'apprendimento gratuito della programmazione informatica per i giovani fra i 7 e i 17 anni.

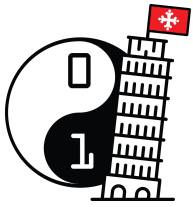
Coder significa Programmatore e Dojo significa Tempio dell'Apprendimento.

Se vuoi imparare il Karate vai in un KarateDojo,

## Perché un Dojo?

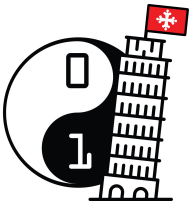
Se ci guardiamo intorno, vediamo PC dappertutto. Il mondo intero viene mandato avanti dai computer. Ma che cosa manda avanti un computer? Il codice. Scritto da programmatori e da gente comune. A mano. Ad oggi mancano programmatori. Sempre più ci appoggiamo ai computer anche per scopi di sopravvivenza e d'altra parte i corsi universitari di Informatica sperimentano un abbandono del 50%.

E' come se ci fosse un picco di richiesta di programmi e... potrebbe



# Una notte tempestosa...

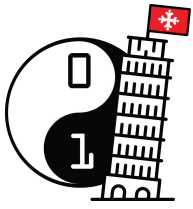




...Steven, un coraggioso  
pipistrello noncurante del  
pericolo, si tuffa nella tempesta,  
ma non sa che potrebbe finire  
male!...

...Sta a noi aiutarlo, ma come?





# Steven the Bat

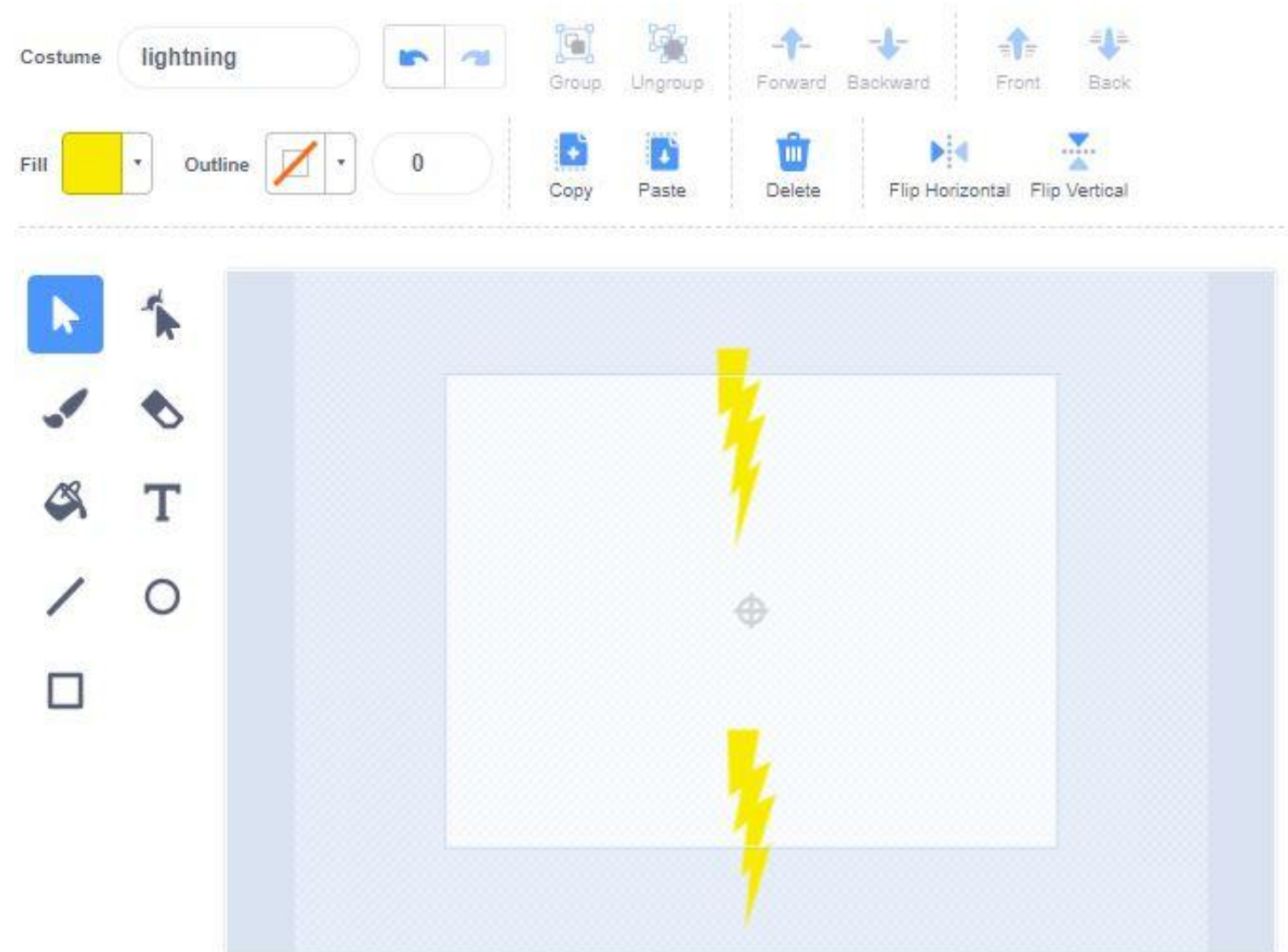


- L'obiettivo del gioco è quello di aiutare Steven ad evitare più fulmini possibili: più fulmini evitiamo, più il nostro punteggio aumenterà!
- Lista degli «ingredienti»:
  - Il nostro Steven ed il suo movimento di volo;
  - Variabili punteggio e gravità (data la forza della tempesta, sarà difficile volare dritti!);
  - I fulmini da evitare;
  - Lo sfondo di fine partita;
  - Lo sfondo del gioco;

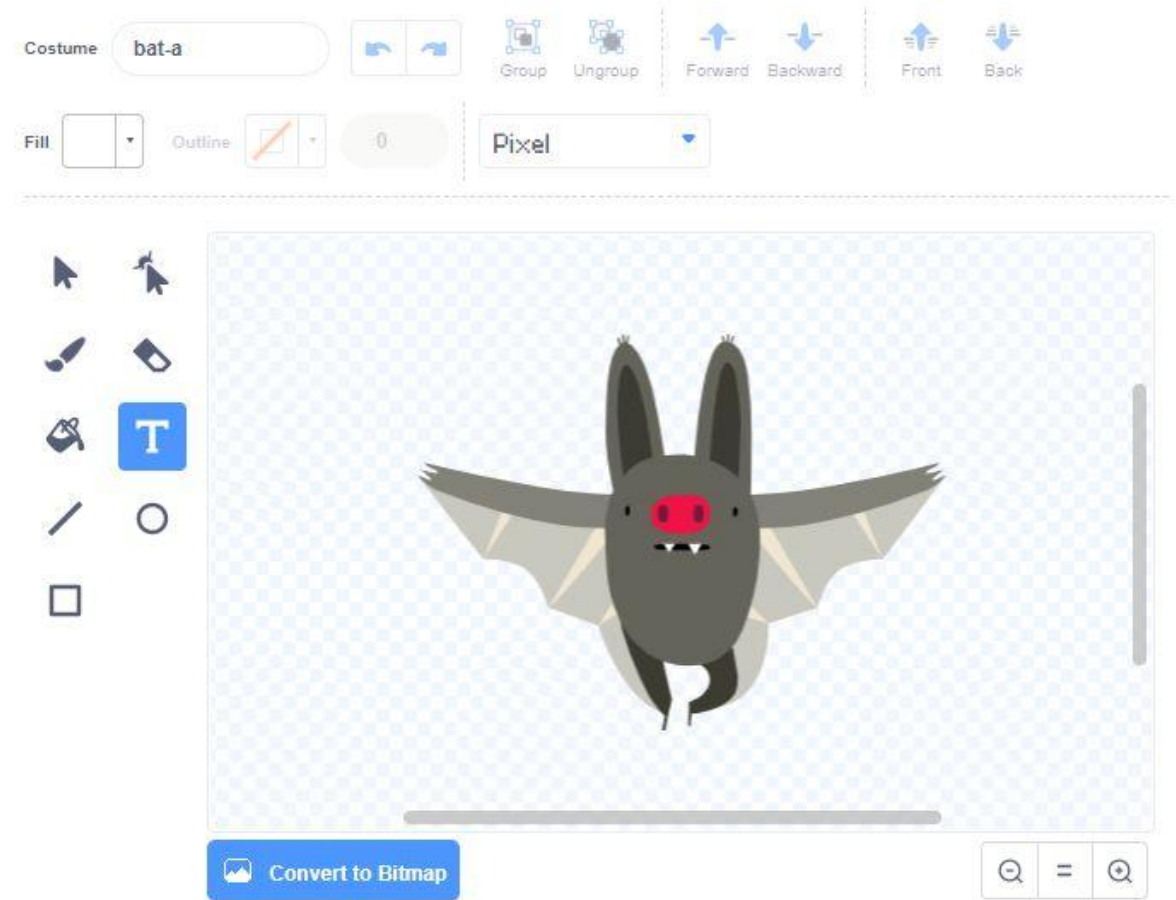
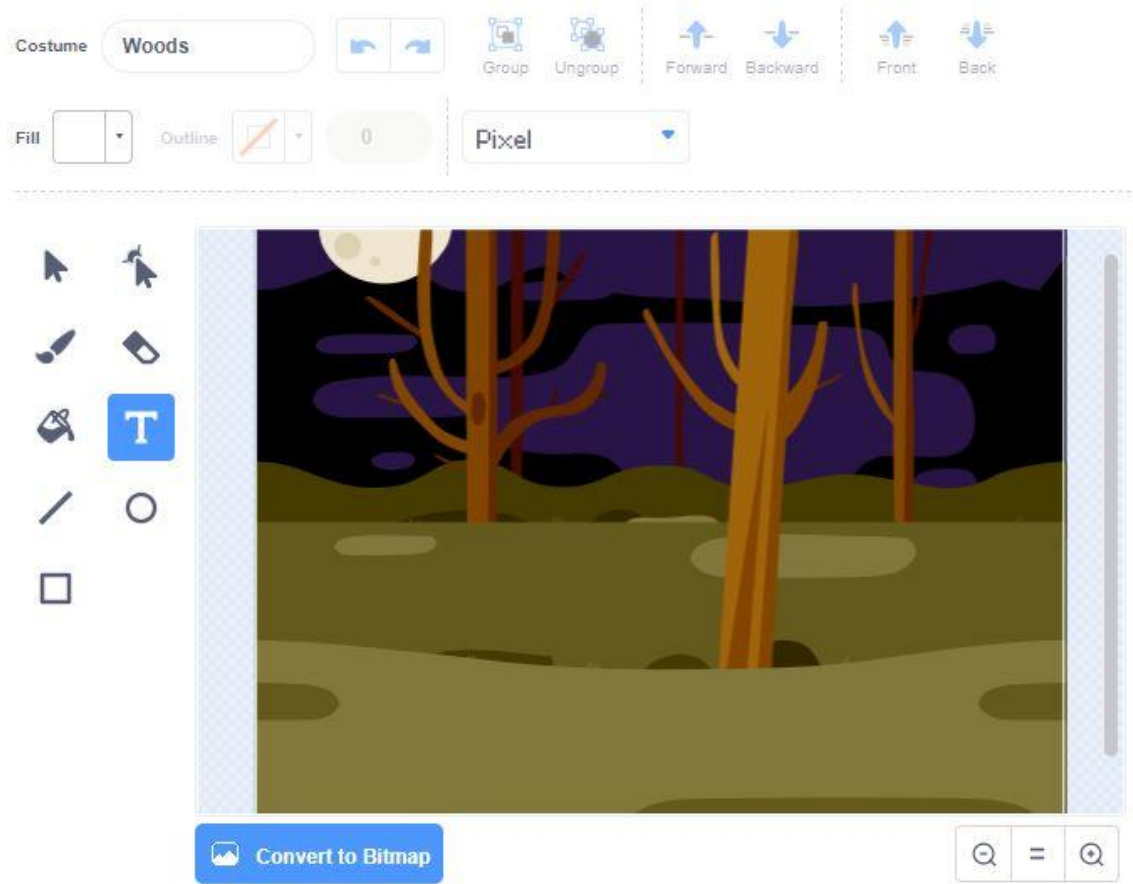
# Selezionare sprite e sfondi

- Per i fulmini, poiché vorremo averne due in colonna, dato che Steven ci passerà attraverso, possiamo fare in questo modo:

- 1) Selezioniamo dagli sprite (usando il tasto «Scegli uno Sprite» e cercando nella sezione «Fantasy») il nostro fulmine;
- 2) Duplichiamo il fulmine , e sovrapponiamo come nella figura affianco un fulmine con l'altro;
- 3) Sistemiamo i fulmini (anche facendoli uscire un po' dal riquadro, come fatto a destra) in modo che nella schermata di gioco possiamo vederli bene;

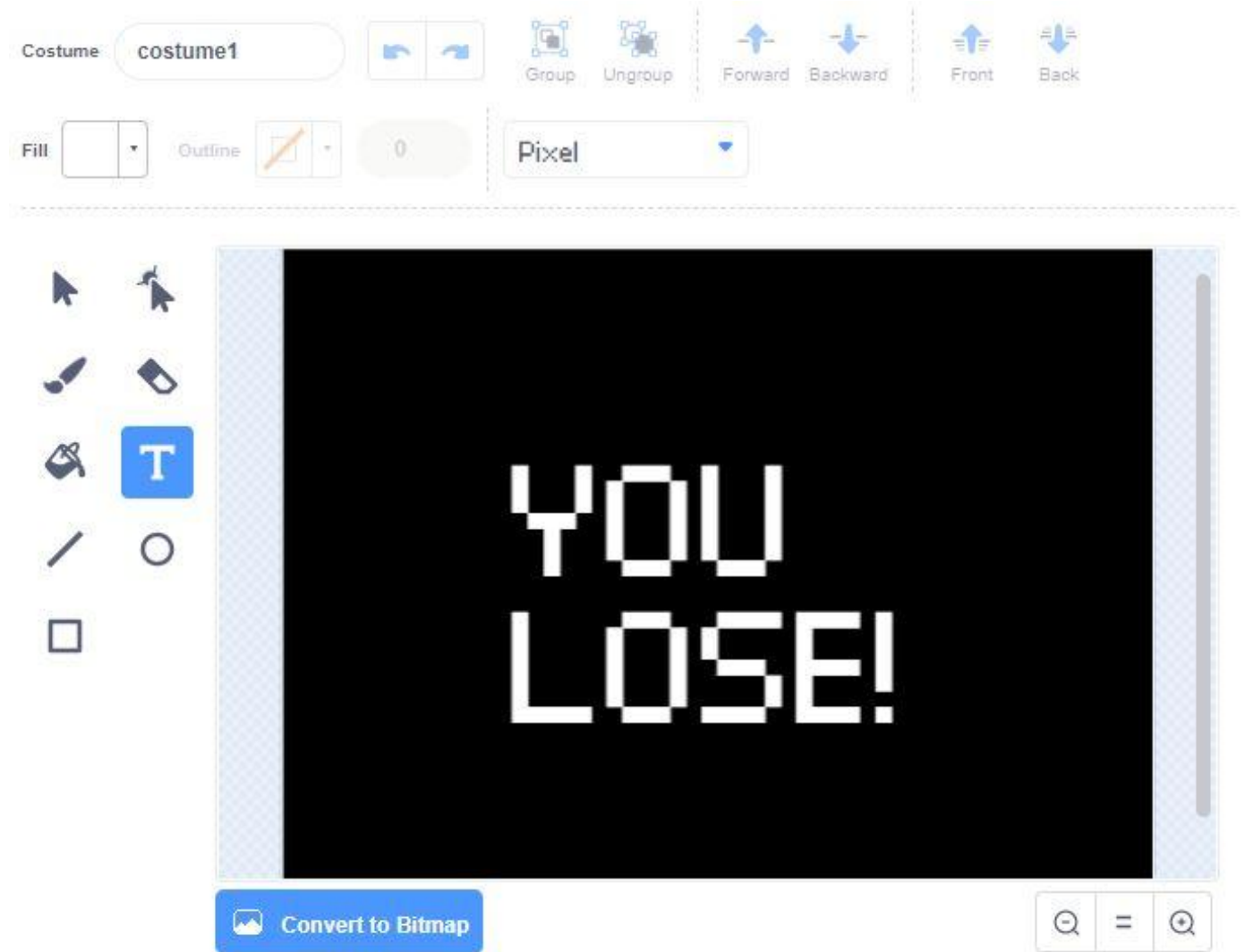




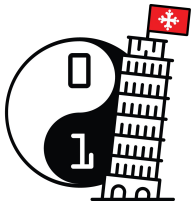


- Si può selezionare lo sfondo cliccando in basso a destra sulla schermata, sul tasto «Scegli uno sfondo» , mentre Steven si può trovare cliccando in basso su «Scegli Sprite» e cercandolo nella sezione «Fantasy» degli sprite;

- Per fare la schermata di fine partita, possiamo selezionare uno spazio bianco per gli sprite, dipingerlo di nero (cliccando sull'icona con su scritto «dipingi») e poi scriverci sopra la scritta di fine partita (ad esempio «You Lose!») con il tipo di scrittura che vogliamo (ad esempio il tipo «Pixel»)
- Scriviamo adesso i codici che ci permetteranno di giocare!



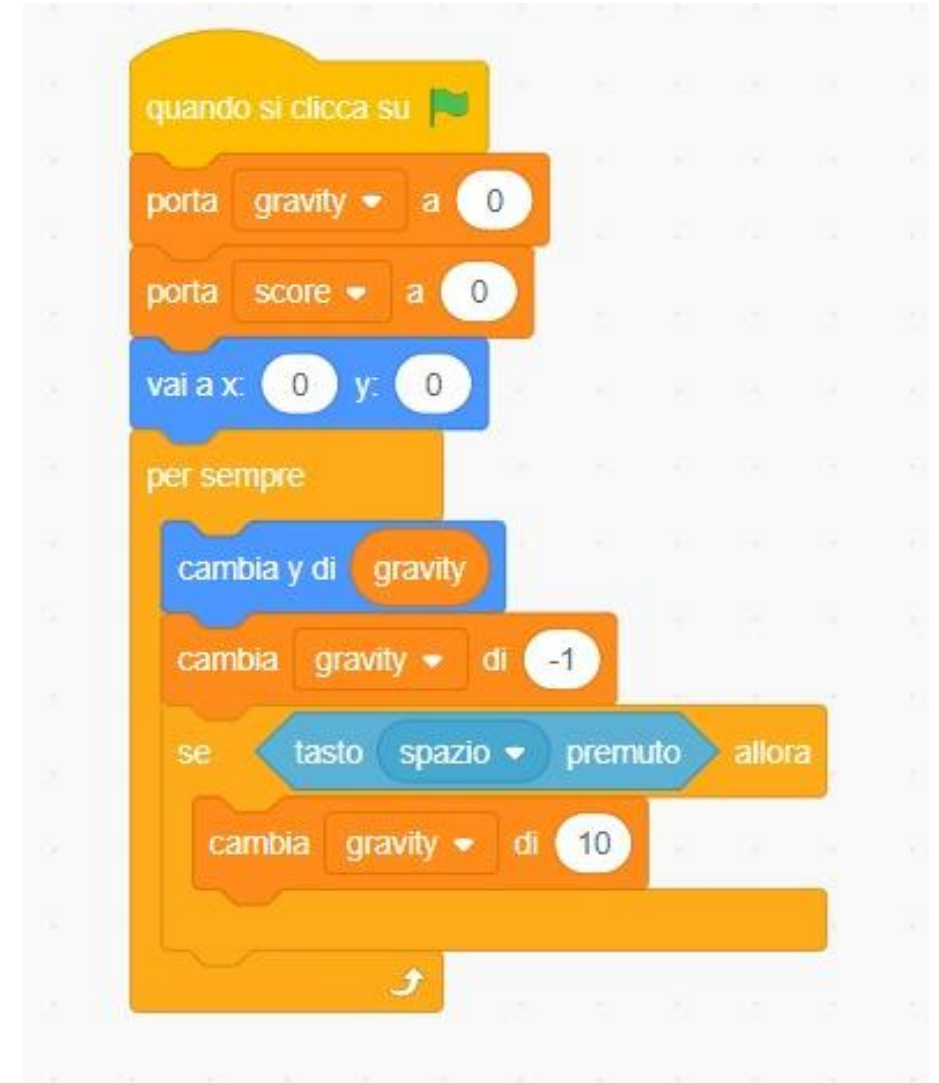


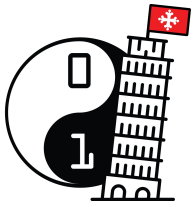


# Il pipistrello Steven [1]

Steven può essere implementato in 3 blocchi (uno è presente in questa slide, a destra, mentre gli altri nelle slides successive):

- Nel primo si creano due variabili, gravity (gravità) e score (punteggio), e si pongono a 0, insieme alle nostre coordinate x e y;
- Dopodichè dobbiamo settare la forza di gravità (es.  $gravity = -1$ ) e quanto Steven è forte nel contrastarla quando premo spazio ( $gravity = 10$ );

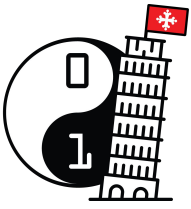




## Il pipistrello Steven [2]

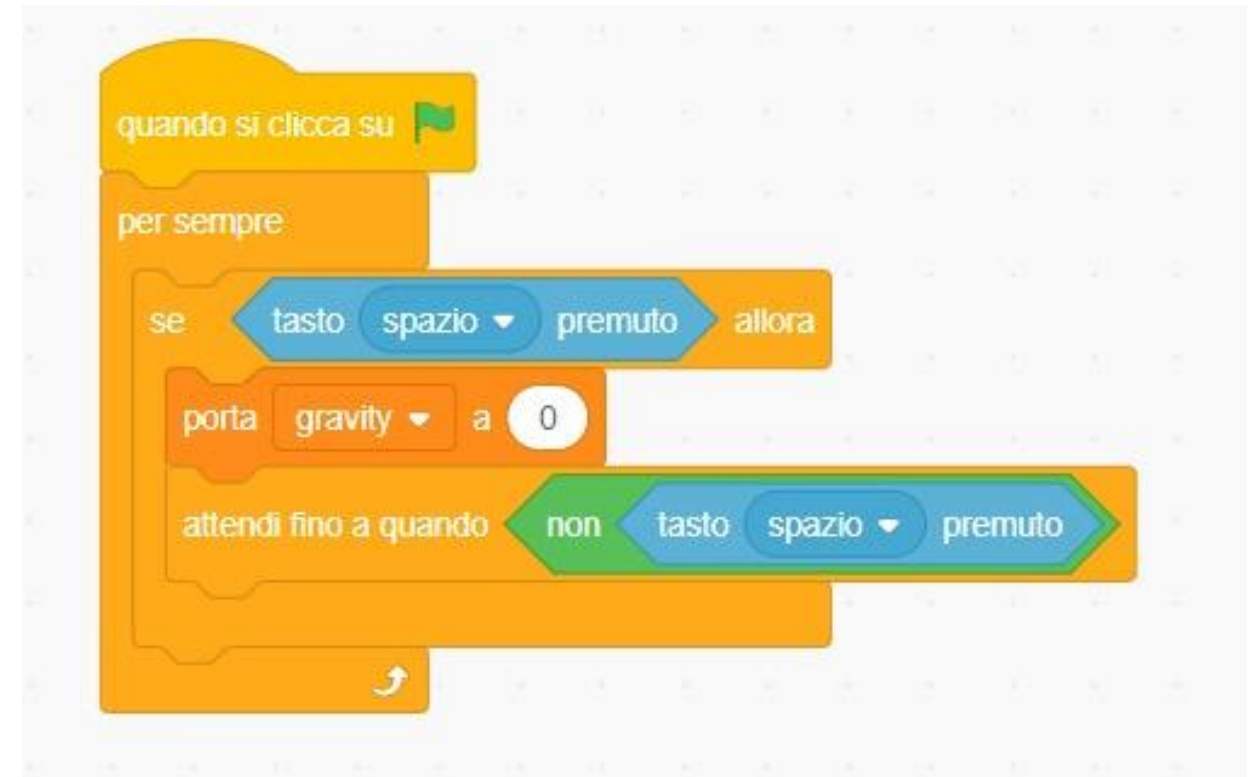
- Nel secondo, bisogna programmare il modo in cui possiamo perdere;
- Noi perdiamo se tocchiamo i fulmini (Lightning) pertanto dovremo porre un'istruzione che varrà per sempre, che ci dica quando perdiamo (mandando il messaggio «You lose» (hai perso!) che programmeremo tra poco...)

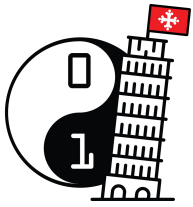




# Il pipistrello Steven [3]

- Nel terzo, dobbiamo invece programmare il movimento contro la gravità vero e proprio;
- Quando premiamo lo spazio, Steven salta con una certa forza, ma se non sbatte le ali (che noi gli facciamo sbattere premendo spazio) continua a cadere per sempre...
- Passiamo adesso ai fulmini!

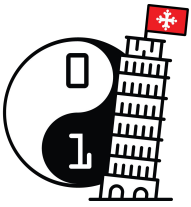




# I fulmini [1]

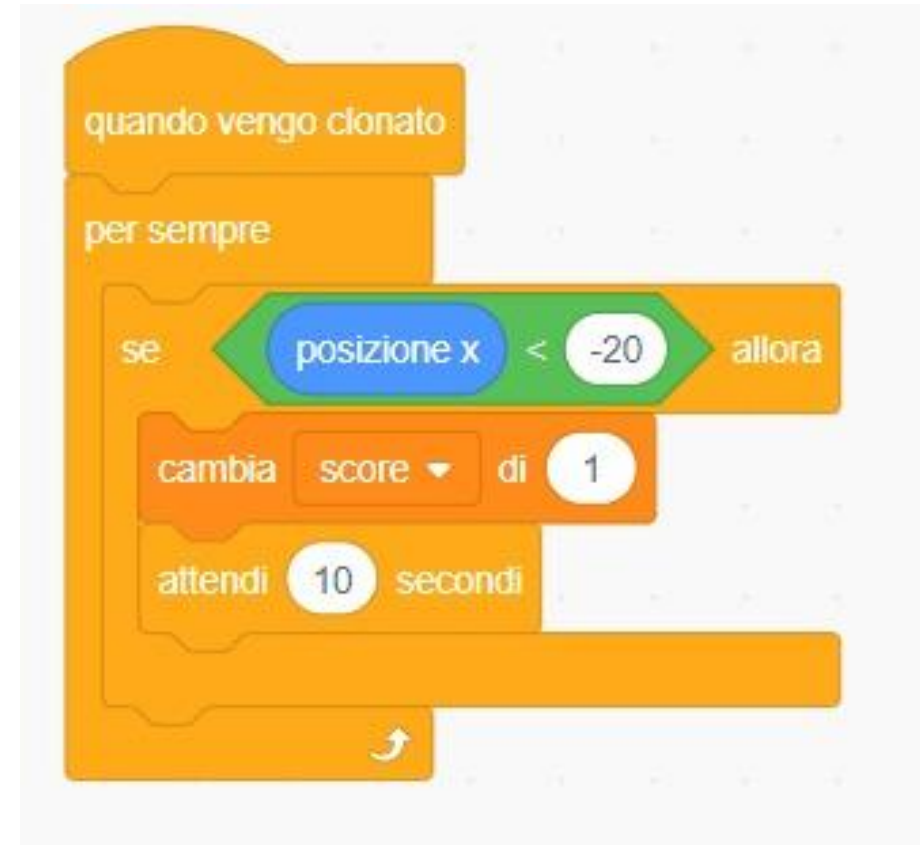
- Anche i fulmini possono essere programmati in 3 miniblocchi :
- Nel primo, si creano i vari fulmini: dobbiamo fare tante copie dei fulmini e nasconderle inizialmente (si mostreranno nel corso del gioco...)

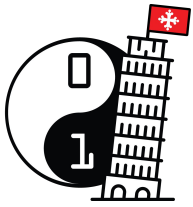




## I fulmini [2]

- Nel secondo, dobbiamo stabilire come vogliamo far scorrere il punteggio: ogni volta che schiviamo un fulmine, lasciandocelo alle spalle, aumentiamo il nostro punteggio di 1;



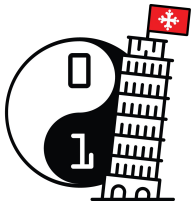


## I fulmini [3]

- Nel terzo, si dovrà stabilire come compariranno nello sfondo i nostri fulmini, una volta che andiamo avanti; per rendere il gioco più divertente, facciamo variare l'altezza dei nostri fulmini in maniera casuale;
- Inoltre, una volta che i fulmini ce li lasciamo alle spalle, «ricompariranno» più avanti, in una posizione diversa da quella di prima



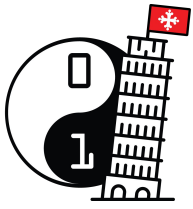




# Lo sfondo di fine partita

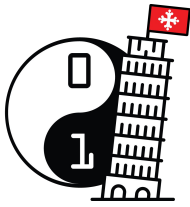
- Anche lo sfondo dovrà comportarsi in un certo modo: vogliamo che quando tocchiamo un fulmine la partita finisca;
- Mentre stiamo giocando, lo sfondo di fine partita non si vede, ma appena tocco il fulmine, si mostra subito, ed interrompe tutto il gioco!





# Sfide per voi!

- 1) Aumentare la velocità una volta raggiunto un determinato punteggio (una volta ogni 20 punti, ad esempio) può essere un buon modo per rendere più divertente il gioco!
- 2) Si può aggiungere un suono «elettrico» nel momento in cui Steven sbatte contro un fulmine;
- 3) Aggiungere dei bonus che Steven può raccogliere per aumentare il punteggio (una caramella, una moneta, ecc.)



# Io resto a casa e programmo!



## #IORESTOACASA

ESCI SOLO PER ESIGENZE ESSENZIALI

