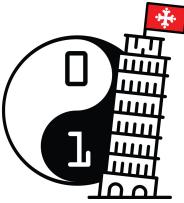


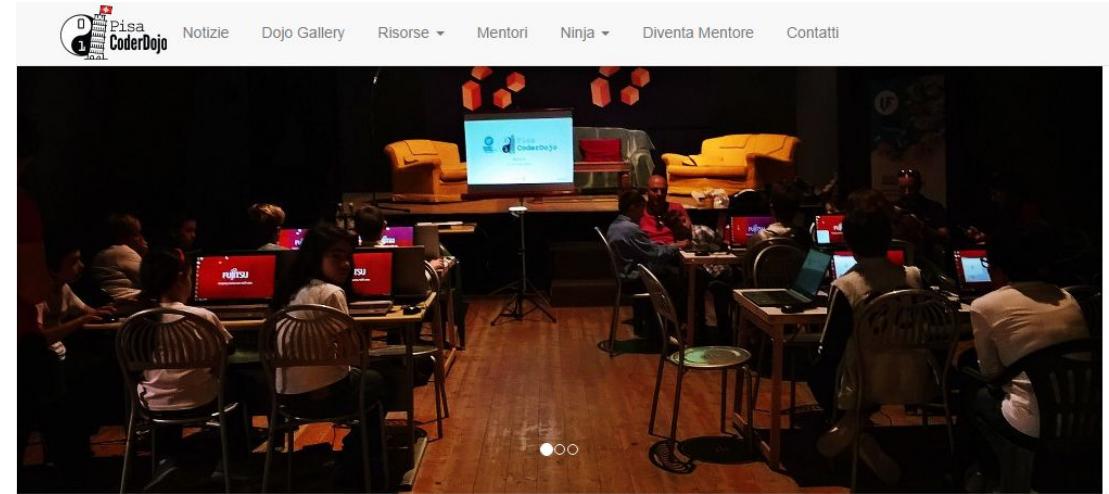
Be Cool!

#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](http://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.

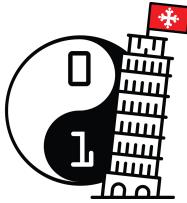
Coder significa Programmatore e Dojo significa Tempio dell'Apprendimento.

Se vuoi imparare il Karate vai in un KarateDojo,

## Perchè un Dojo?

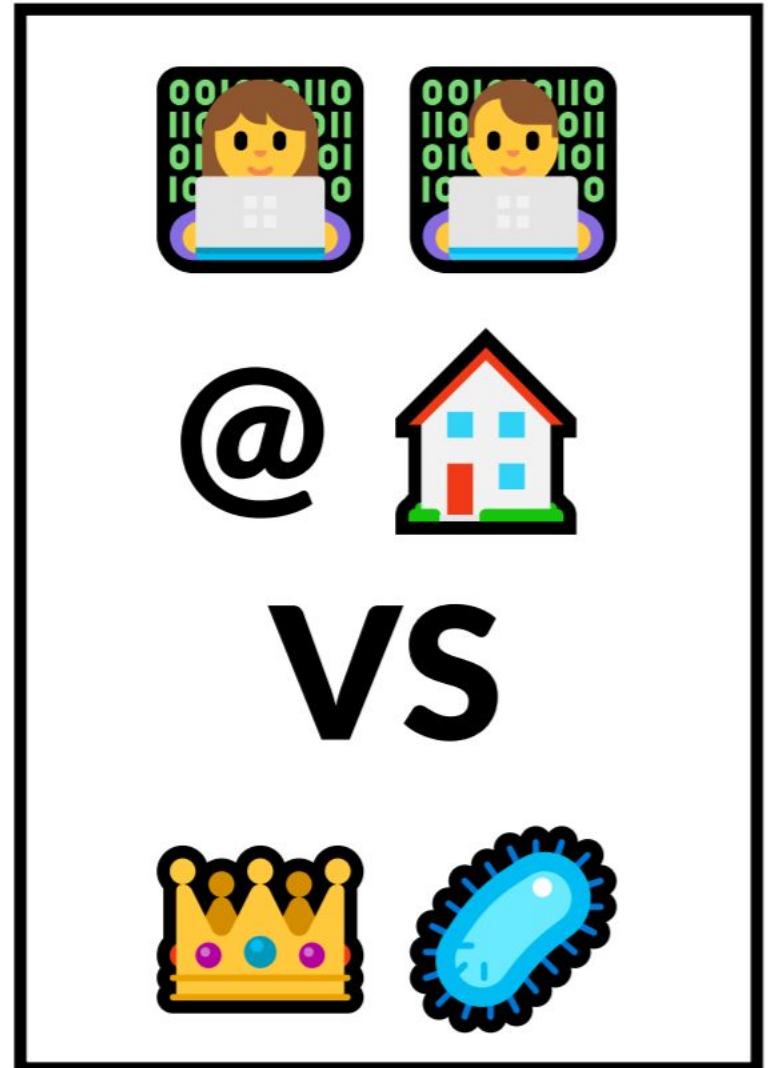
Se ci guardiamo intorno, vediamo PC dappertutto. Il mondo intero viene mandato avanti dai computer. Ma che cosa manda avanti un computer? Il codice. Scritto da programmatore e da gente comune. A mano. Ad oggi mancano programmatore. Sempre più ci appoggiamo ai computer anche per scopi di sopravvivenza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

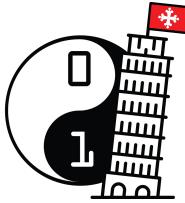
E' come se ci fosse un picco di richiesta di programmi e... potrebbe



# #iorestoacasa e programmo!

- Una **nuova iniziativa** del Pisa Coderdojo per tutti i Ninja e per i loro genitori!
- Una serie di **video-tutorial** per **mantenere vivo lo spirito di CoderDojo** in questi giorni così difficili per tutti.
- Facciamo la nostra parte per sconfiggere il coronavirus: **rimaniamo a casa e programmiamo insieme.**





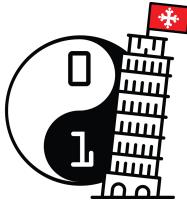
# Idle Games



- “Idle” significa pigro, inattivo.
- Cliccando si accumulano risorse
- Spendendo risorse in upgrade si aumenta la velocità di accumulo

Esempi:

- Cookie Clicker
- Clicker Heroes



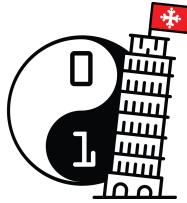
# Monster Clicker

In questo tutorial realizzeremo *Monster Clicker*, un idle game dove il nostro obiettivo sarà quello di accumulare denaro sconfiggendo mostri sempre più forti.

Sarà possibile acquistare upgrade per aumentare i nostri danni da click o aumentare il danno automatico inflitto per secondo.

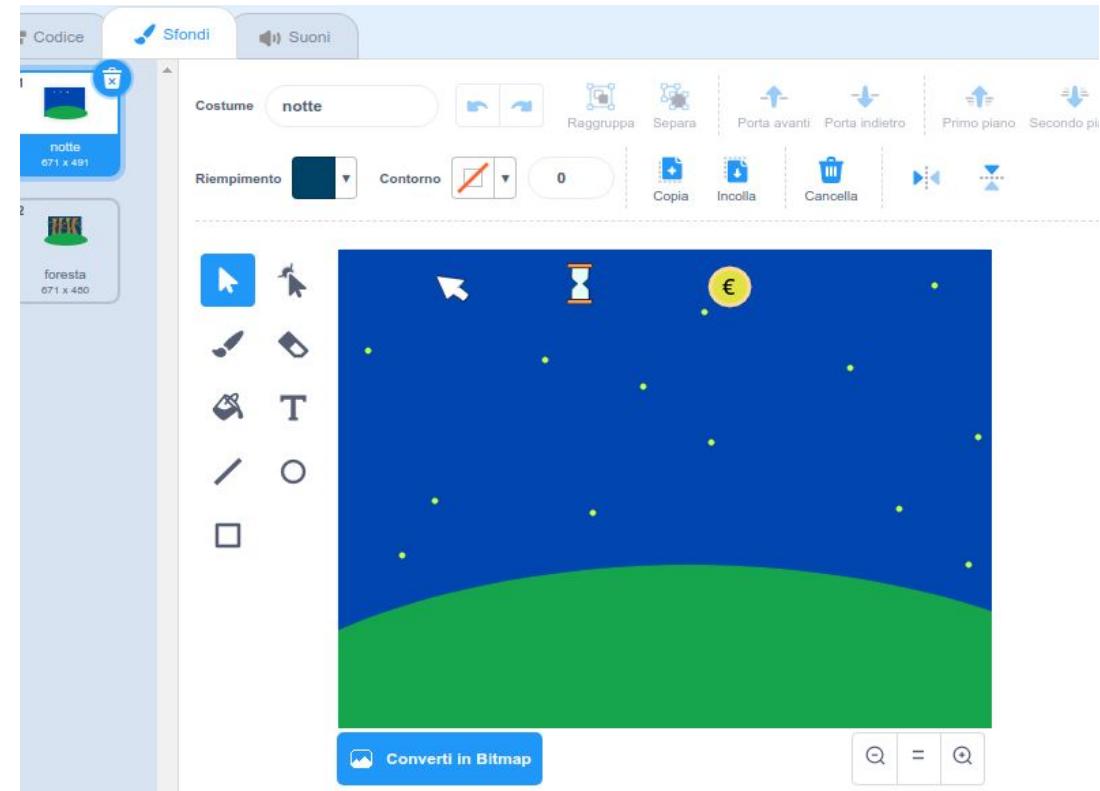
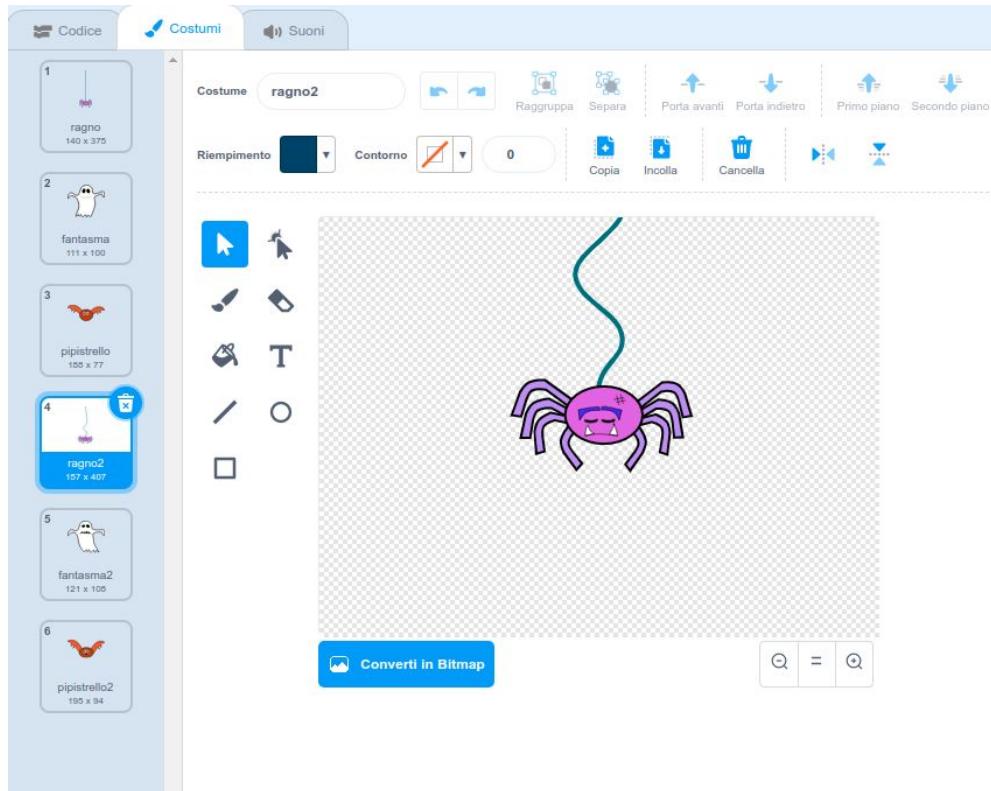


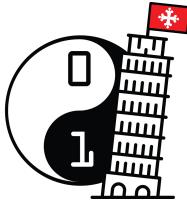
Link Remix Base <https://scratch.mit.edu/projects/390896417>  
Completo: <https://scratch.mit.edu/projects/378267923>



# Lo Stage

Iniziamo creando uno **Sprite** chiamato *Mostro* dove i primi 3 costumi saranno i nemici base, e i successivi 3 la loro animazione da mostrare quando vengono colpiti. Poi passiamo allo **Stage** e disegniamo il nostro *sfondo*.



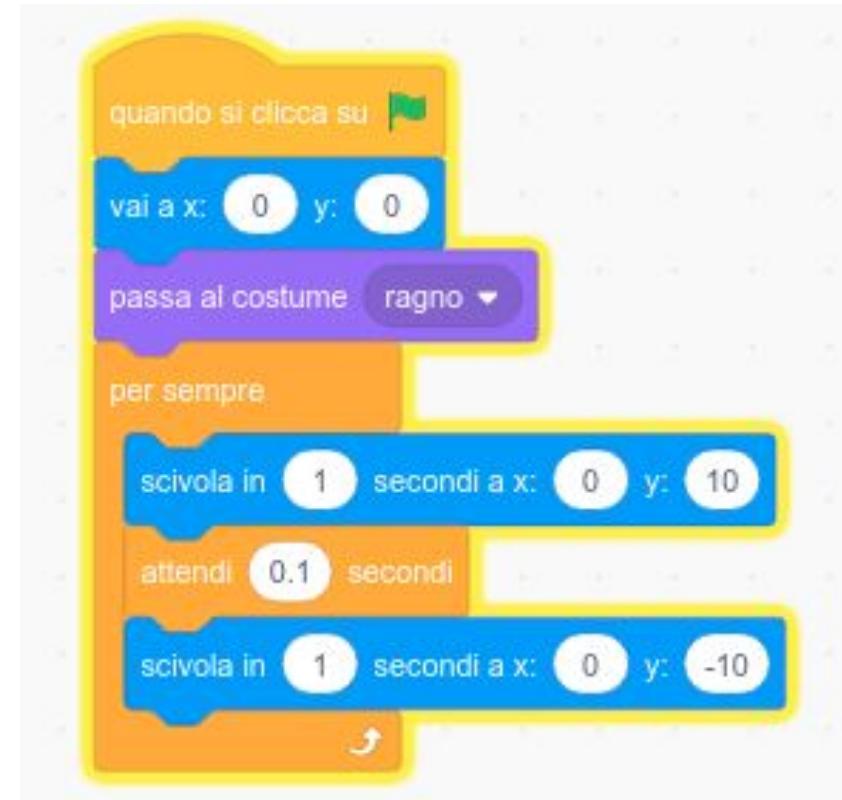


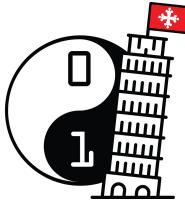
# Movimento!

Selezioniamo la **Sprite Mostro**.

Quando clicchiamo start **questo sprite** verrà posizionato al centro dello schermo, e selezioniamo il costume iniziale.

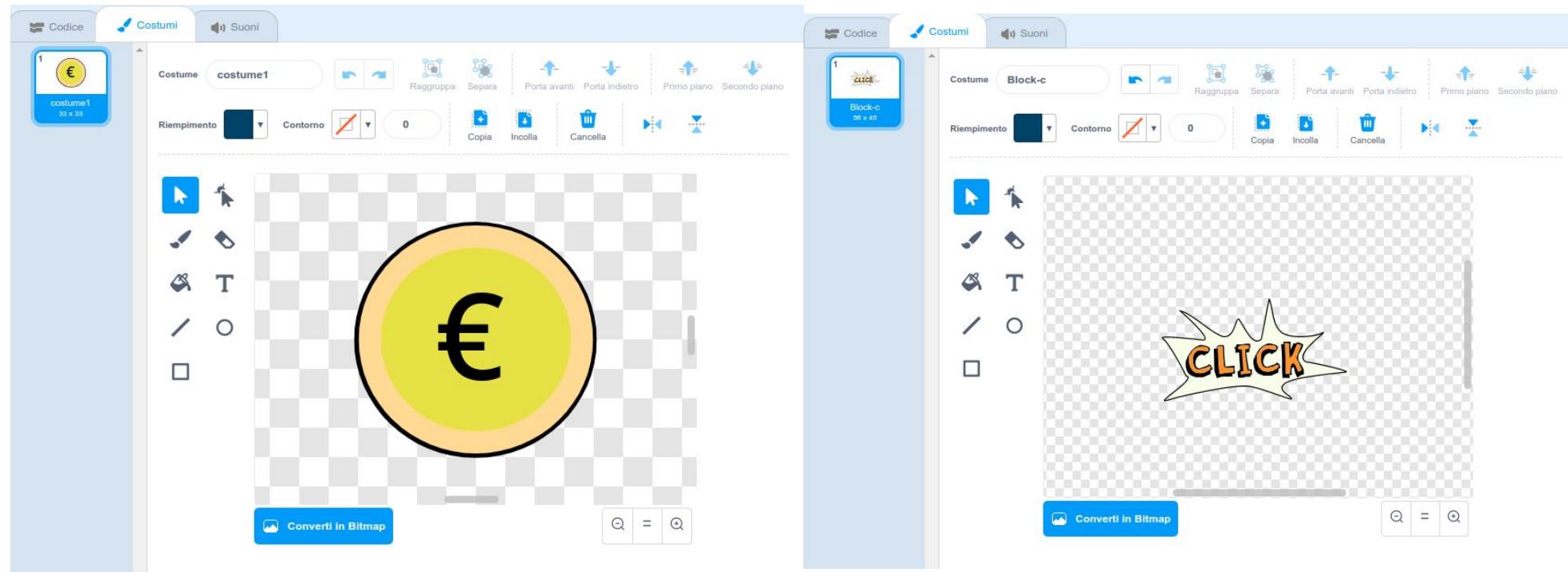
A questo punto per sempre animiamo il nemico in modo che *scivoli sull'asse y*.

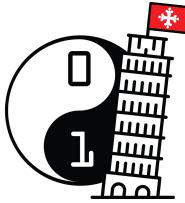




# Altri Sprite

Creiamo altri due **Sprite** chiamati *Euro* ed *Effetto* dove il primo rappresenta le monete che otteniamo sconfiggendo i nemici, l'altro darà un feedback al colpo inferto dal click.





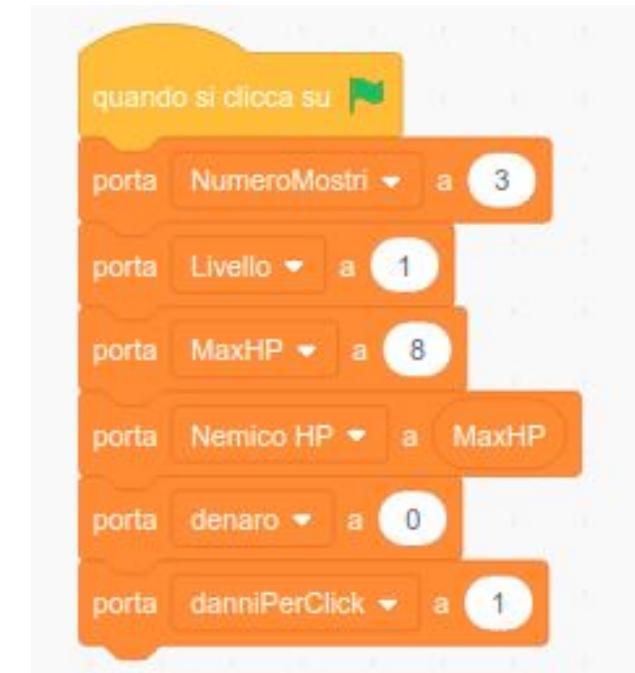
# Variabili

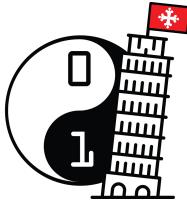
Abbiamo bisogno di tener traccia della vita iniziale dei nemici, la vita corrente, il danno inflitto dal click ed il denaro accumulato.

Creiamo le variabili:

- NumeroMostri
- *Livello*
- *MaxHP*
- *NemicoHP*
- *Denaro*
- *DanniPerClick*

Selezioniamo il nostro sfondo ed iniziamo a dichiarare le nostre variabili ed inizializzarle.





# All'attacco!

Selezioniamo la **Sprite Mostro**.

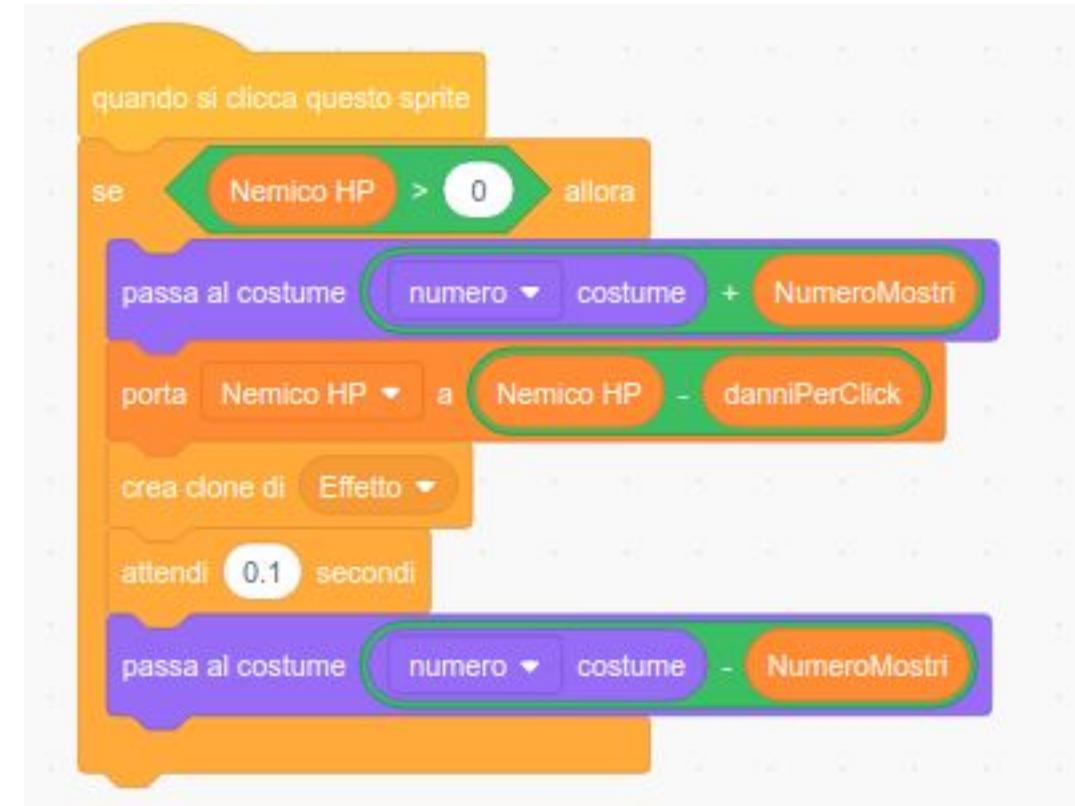
Quando si clicca **questo sprite** vogliamo controllare che abbia punti vita,

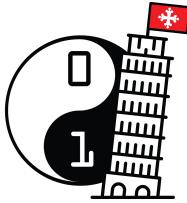
A questo punto mostrare il costume da ferito .

Step importante, applichiamo il danno:

NemicoHP = NemicoHP - DanniPerClick.

**Cloniamo Effetto** per dare un indizio visivo del colpo.

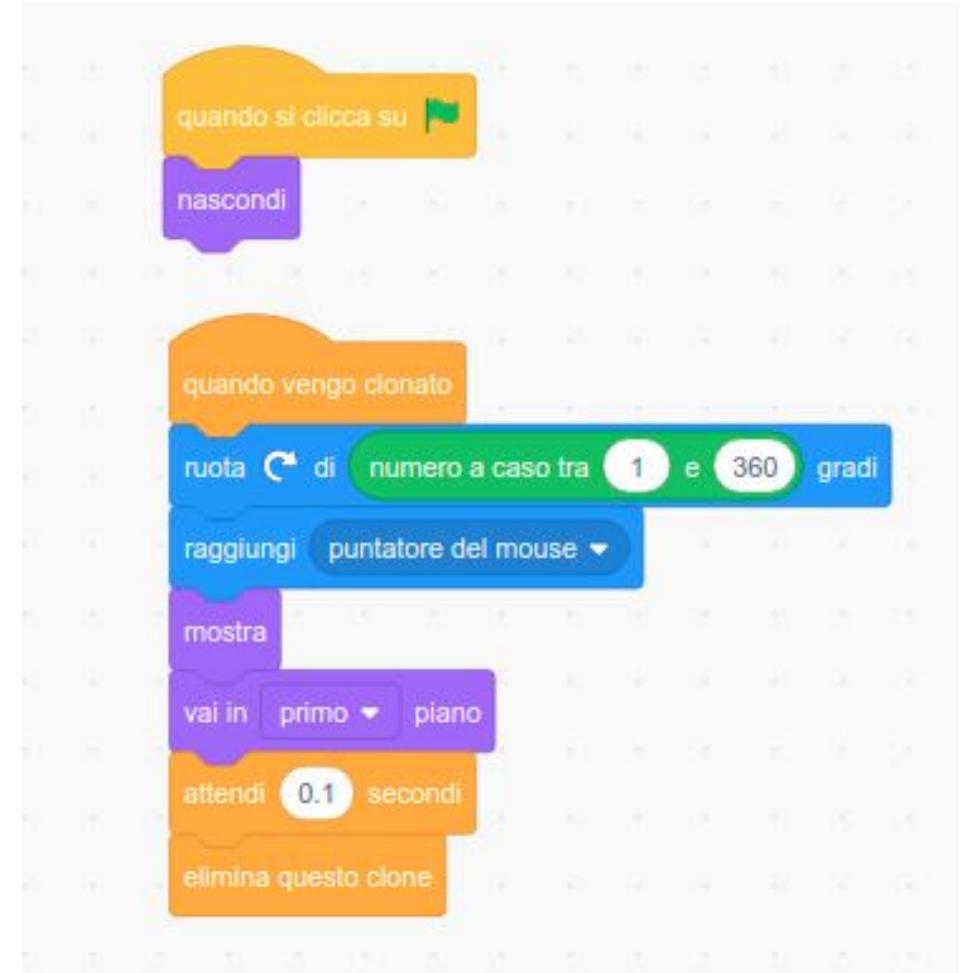




# Effetto del click

Selezioniamo lo **Sprite** chiamato *Effetto*. Per dare al giocatore la sensazione di aver colpito il nemico inizialmente sarà nascosto e verrà mostrato quando verrà clonato.

Aggiungiamo un pò di variabilità ruotandolo casualmente.



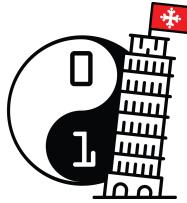


# Sconfitto!

Ora i punti vita del nemico diminuiscono con i nostri attacchi, dobbiamo aggiungere un controllo per sapere se è stato sconfitto, ottenere le nostre monete e passare allo sfidante successivo!

Aggiungiamo anche *Livello* volte **Clone** *Euro* per visualizzare le monetine che acquisiamo.

```
when green flag clicked
forever
  if [Nemico HP < 1] then
    play sound [ghost v1] for [50] ticks
    set [Nemico HP v] to [0]
    repeat
      set [Livello v] to [0]
      create clone of [Euro v]
      change [denaro v] by [Livello]
      wait [0.5] seconds
      play sound [ghost v1] for [50] ticks
      set [Euro v] to [0]
      switch costume to [random 1..NumeroMostri v]
      change [Uccisioni v] by [1]
    end
  end
end
```

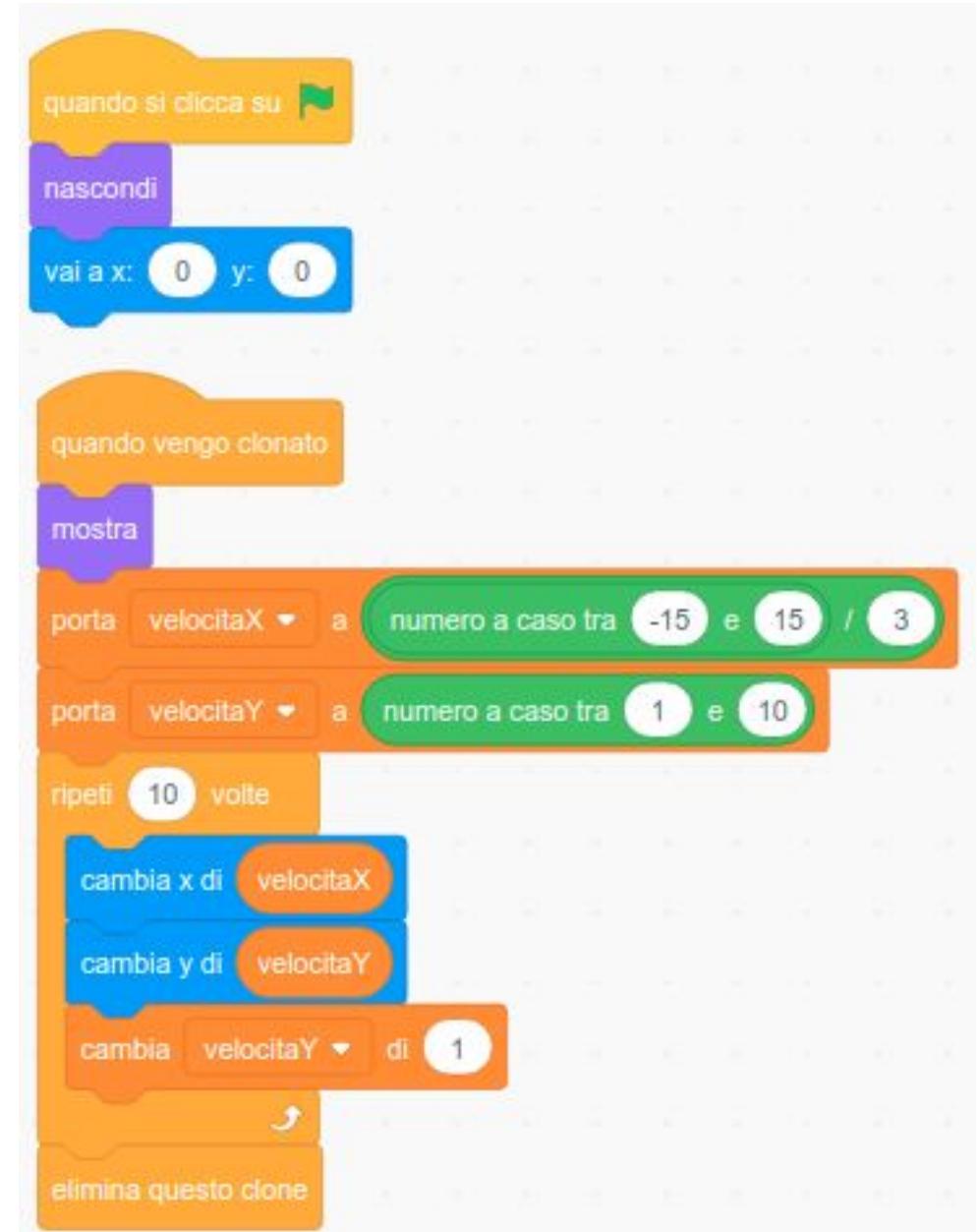


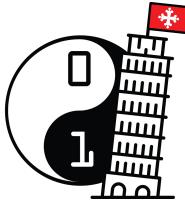
# Ricompense!

Selezioniamo lo **Sprite Euro**.

Le monete saranno mostrate solo quando un nemico è stato sconfitto.

Quando queste vengono clonate animiamole facendole salire in alto e scomparire.





# Altre Variabili

Dobbiamo aggiungere altre variabili per gestire gli “Upgrade” (significa potenziamento).

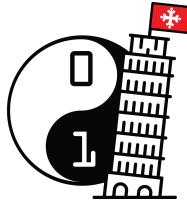
Il costo del potenziamento aumenterà dopo ogni singolo acquisto.

Creiamo le variabili:

- *CostoUpgradeClick*
- *FattoreDiScalaClick*

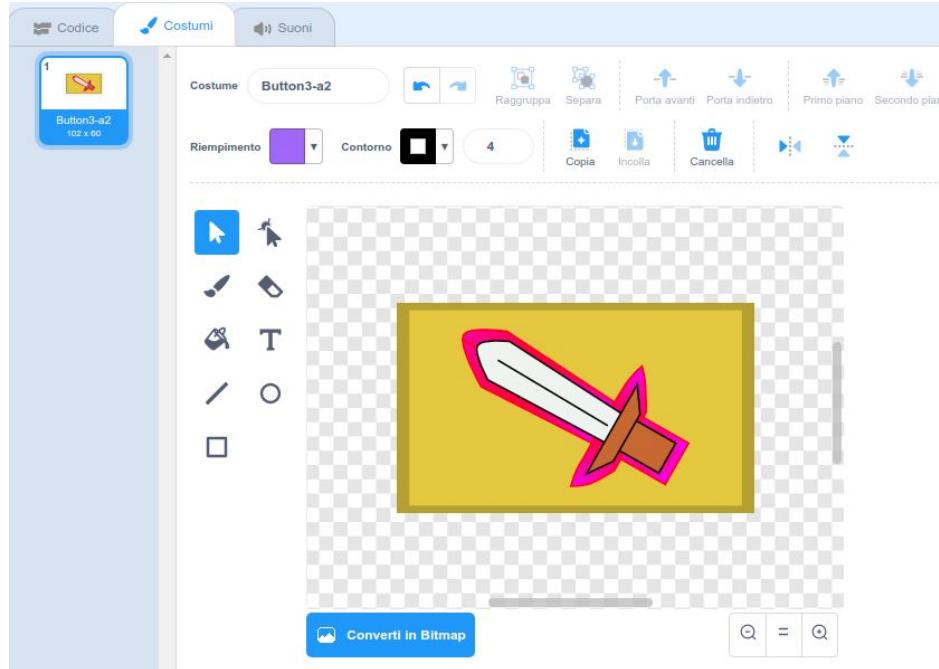
Selezioniamo il nostro sfondo ed dichiariamo ed inizializziamo le altre variabili.



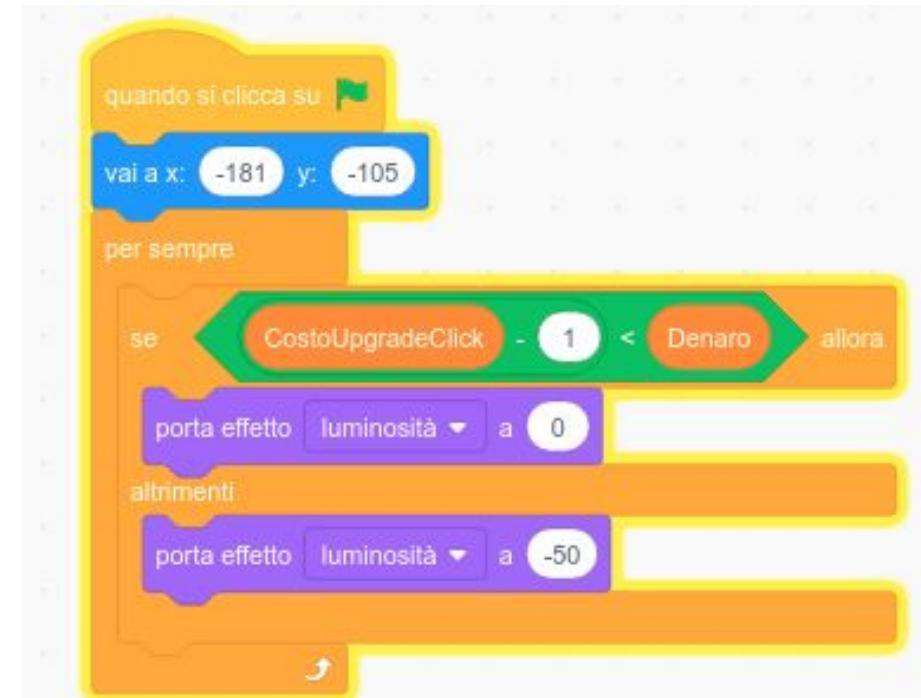


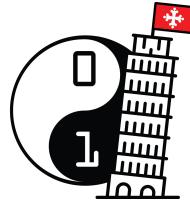
# Upgrade del Click

Aggiungiamo un nuovo **Sprite** chiamato *UpgradeClick*.

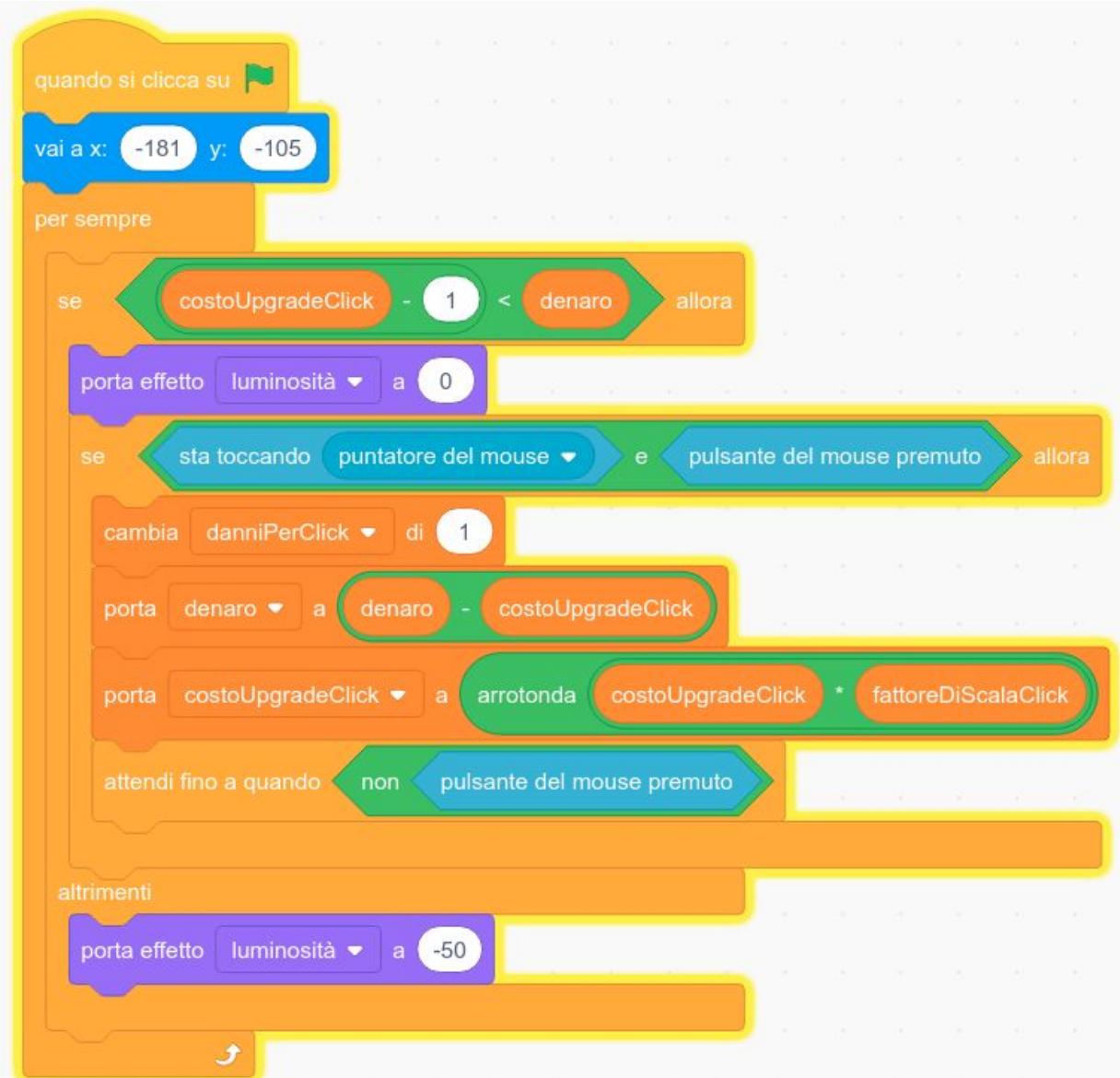


Cliccando su questo bottone aumentiamo i danni inflitti ai nemici per il singolo click.  
Aggiungiamo un effetto di luminosità per segnalare quando è pronto per l'acquisto.





# Upgrade del Click

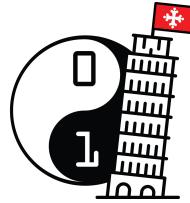


Quando questo **Sprite** viene premuto aumenteremo i *DanniPerClick* di 1

Sottraiamo al *Denaro* il costo del potenziamento.

Aumentiamo il costo del prossimo acquisto moltiplicandolo per il *fattore di scala*, che regola quanto velocemente vogliamo farlo crescere.

Arrotondiamo per non avere numeri dopo la virgola.



# Progressione

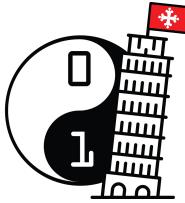
Selezioniamo lo **Sprite Mostro**.

Aggiungiamo la progressione dei livelli tenendo traccia di quanti nemici abbiamo sconfitto.

Ogni 10 uccisioni aumentiamo il livello.

Aumentiamo la difficoltà facendo salire la vita massima dei mostri, moltiplicando la *MaxHP* per 1.5.

```
when green flag clicked
  set [Uccisioni v] to [0]
  set [Livello v] to [1]
  set [MaxHP v] to [100]
  set [Nemico HP v] to [100]
  forever
    if [Nemico HP < 1] then
      apply effect ghostly 50
      set [Nemico HP v] to [0]
      repeat [Livello] times
        create clone of [Euro v]
        change [denaro v] by [1]
        wait [0.5] seconds
      end
      apply effect ghostly 0
      pass costume [numero a caso tra 1 e NumeroMostri v]
      change [Uccisioni] by [1]
      if [Uccisioni = 10] then
        reset [Uccisioni v]
        set [MaxHP v] to [round (MaxHP * 1.5)]
        set [Livello v] to [Livello + 1]
        set [Nemico HP v] to [MaxHP]
      end
    end
  end
```



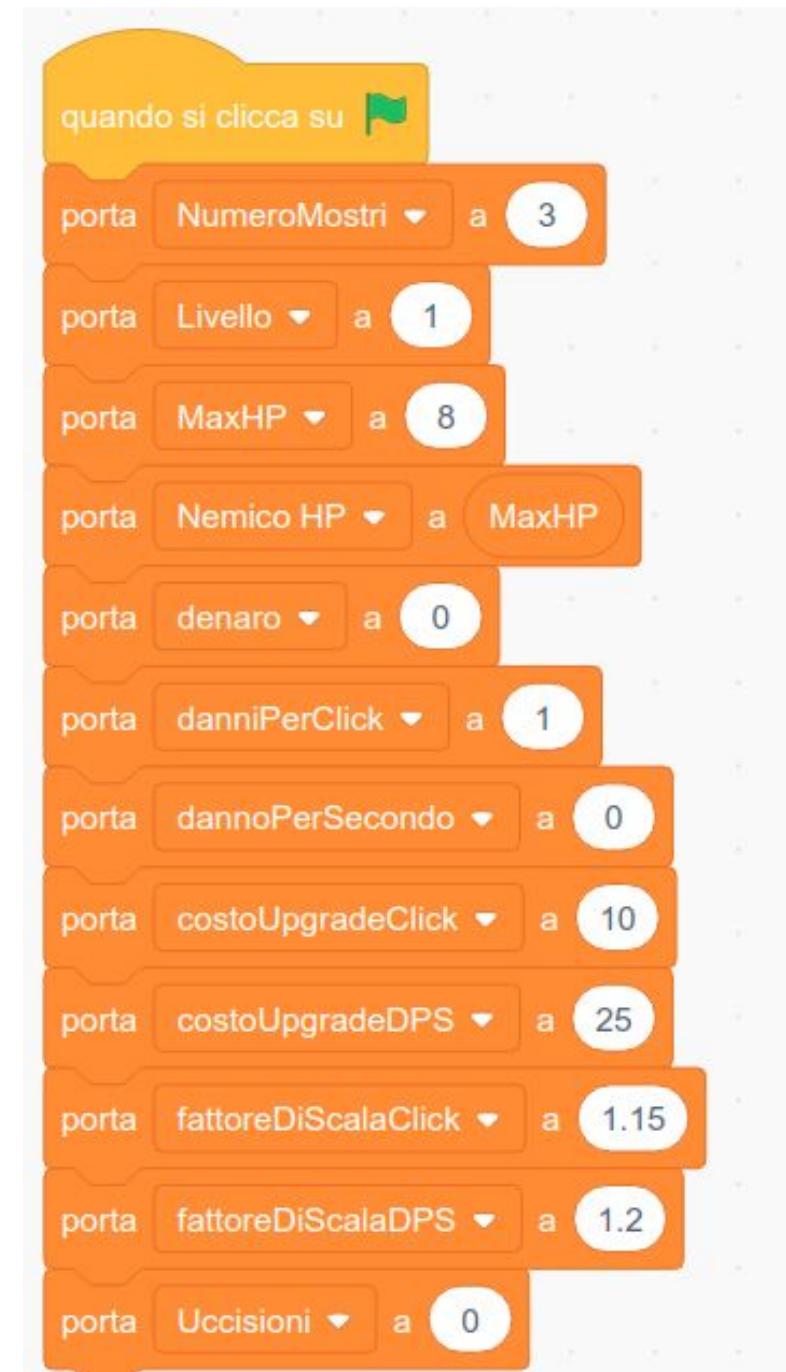
# Ancora altre Variabili

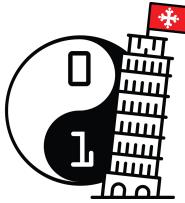
Nuovamente dobbiamo aggiungere altre variabili per gestire il nuovo “Upgrade” DPS.

Sempre selezioniamo il nostro sfondo ed dichiariamo ed inizializziamo le nuove variabili.

Creiamo le variabili:

- *DanniPerSecondo*
- *CostoUpgradeDPS*
- *FattoreDiScalaDPS*

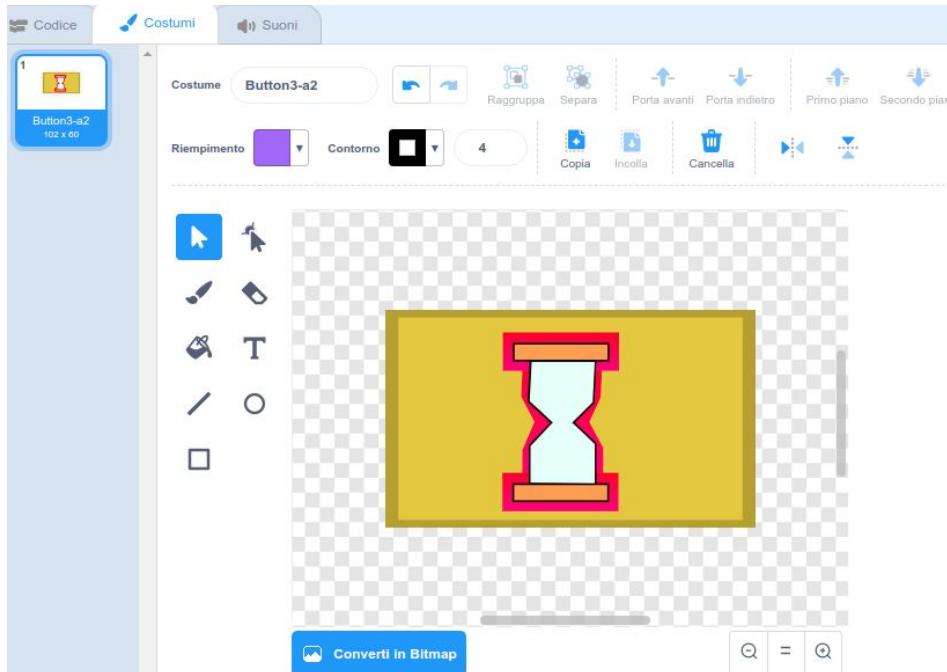




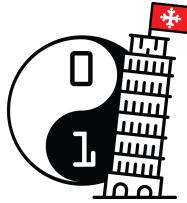
# Upgrade Danni al Secondo

Aggiungiamo un nuovo **Sprite** chiamato *UpgradeDps*.

Cliccando su questo bottone aumentiamo i danni automatici inflitti ai nemici ogni secondo (DPS).



Per iniziare copiamo tutto il codice dallo **Sprite UpgradeClick**. Vedremo che basterà cambiare soltanto alcune variabili.



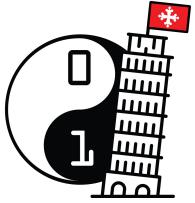
# Upgrade Danni al Secondo

The Scratch script consists of two main sections:

- Top Section (when green flag clicked):** Sets initial values and handles mouse input.
  - Initializes position: `vai a x: 181 y: -105`.
  - Initializes cost variable: `costoUpgradeDPS = 1`.
  - Initializes luminosity effect: `porta effetto luminosità a 0`.
  - Initializes damage per second: `cambia dannoPerSecondo di 1`.
  - Updates money variable: `porta denaro a denaro - costoUpgradeDPS`.
  - Updates cost variable: `porta costoUpgradeDPS a arrotonda costoUpgradeDPS * fattoreDiScalaDPS`.
  - Waits until mouse button is released: `attendi fino a quando non pulsante del mouse premuto`.
  - Otherwise (altrimenti):
    - Decreases luminosity: `porta effetto luminosità a -50`.
- Bottom Section (per sempre loop):** Applies damage over time.
  - Waits 1 second: `attendi 1 secondi`.
  - Updates enemy health: `porta Nemico HP a Nemico HP - dannoPerSecondo`.

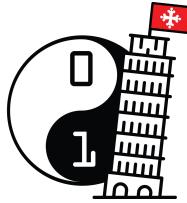
Sostituiamo le variabili con quelle appena create per il nuovo potenziamento.

Aggiungiamo un ciclo che applicherà il danno al secondo.



# Sfide per voi!

1. Personalizzare
2. Aggiungere altri mostri
3. Cambiare sfondo ogni Livello
4. Aggiungere Effetti Sonori
6. Aggiungere nuovi Upgrade
7. Boss di fine livello



# Io resto a casa e programmo!



Ministero della Salute

#IORESTO ACASA

ESCI SOLO PER ESIGENZE ESSENZIALI



PROTEZIONE CIVILE  
Presidenza del Consiglio dei Ministri  
Dipartimento della Protezione Civile