



# Introduzione a SCRATCH

## *Lezione 3*

**Stefano Lacaprara**

**stefano.lacaprara@pd.infn.it**

**INFN Padova**

Padova, 28 febbraio 2013



- **Oggi: continuiamo a consolidare quello che abbiamo imparato**
  - **Rivediamo gli esercizi per casa che vi ho proposto la volta scorsa**
  - **Altri esercizi via via piu' difficili**
    - **Cerchiamo di capire meglio l'uso delle variabili**



- **Ho aggiornato la pagina web del corso**
  - <http://www.pd.infn.it/~lacaprar/ProgettoScuola/Scratch/>
  - **Slides della prima lezione**
  - **Slides seconda lezione**
    - **Soluzione degli esercizi proposti**



- **Online Summer Scratch PD: June 3 - July 12**
  - The ScratchEd Team is pleased to announce that we will be hosting a six-week, online course for teachers, beginning June 3, supported through Google's CS4HS initiative.
    - **Interested?**  
Join the course mailing list for updates!
- **European Scratch Conference a Barcellona 25-27 luglio**
  - <http://scratch2013bcn.org/>
- **Scratch day: 18 maggio 2013**
  - <http://day.scratch.mit.edu/>
    - **Per ora: 4 eventi in Spagna, 1 in repubblica Ceca.**



#### A Scratch Conference in Europe

Connecting Worlds is the first Scratch conference held in Europe, where educators, researchers, developers and other members of the worldwide Scratch community are gathering at Citilab-Cornellà in Barcelona to celebrate and share the creative possibilities of Scratch.

#### Theme: Connecting Worlds

Proposals are needed for a variety of sessions, ranging from presentations and workshops to panel discussions. If it interests you, and you want to share it with others, submit a proposal.

#### Who is it for?

The aim of the conference is to bring together the inhabitants of the different Scratch worlds, ranging from the world of classroom use through to the world of hobbyists, with emphasis ranging from Scratch as a programming language to Scratch as a general educational environment, and with motivations ranging from the artistic to the technological.

We want your proposals for:

- Posters
- Demonstrations
- Presentations
- Panel Discussions
- Workshops

Deadline: 28th Feb. 2013



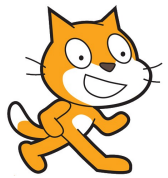
For more information:

Visit the website, or email to

[info@scratch2013bcn.org](mailto:info@scratch2013bcn.org)

Conference Website: [www.scratch2013bcn.org](http://www.scratch2013bcn.org)





## 1. Il gatto che balla

### 1. Movimenti semplici di uno sprite

#### 1. Cambiamento di aspetto con interazione (g/p) o suono

## 2. Il gatto e il drago

### 1. Due sprite: movimento e aspetto

#### 1. Azioni condizionate dalla posizione, o interazione con colori o interazione tra sprite

## 3. Gatto e ballerina

### 1. Due sprite, interazione tramite messaggi

### 2. Gestione dello sfondo



## 4. Corsa delle macchine

4. Gestione de sprite e sfondo
5. Uso numeri casuali per movimento
6. Condizioni basate su posizione sprite
  1. Introduzione alle variabili (oggi la vediamo meglio)
  2. Scommessa: input da utente e uso risposta

## 2. Pipistrello (con una p!) che vola

1. Movimento, interazione con il mouse

## 3. Cane e gatto

1. Interazione tra sprite basata su distanza e non contatto.
  1. Uso di condizioni su variabili basata su posizione



- **Disegnare con una matita**
  - Muovere uno sprite con il mouse
  - Uso degli strumenti di scrittura
- **Compagnia di danza**
  - Gestione diversi sprite
  - Comandare diversi sprite usando bottoni sullo sfondo
    - Interazione con il mouse
    - messaggistica

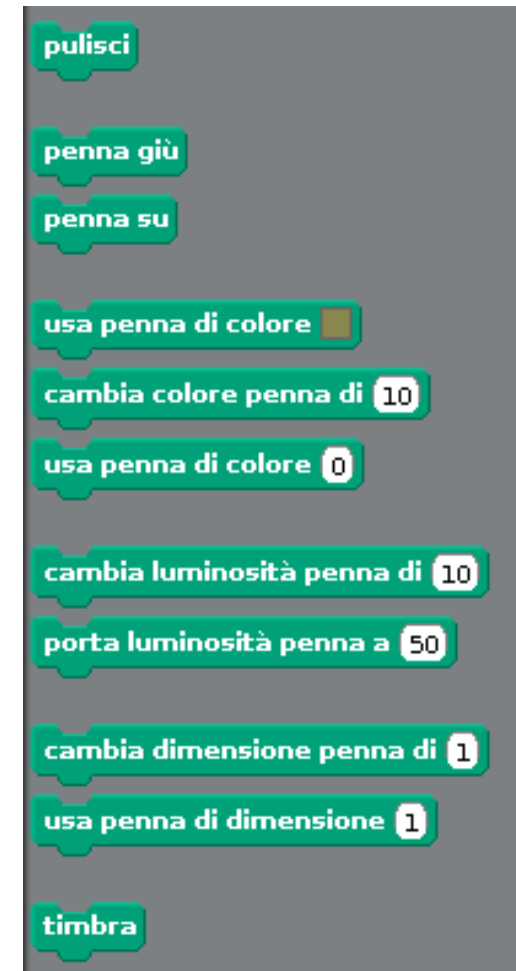


- **Su uno sfondo che mostra gli assi cartesiani, voglio disegnare usando una matita comandata dal mouse.**
- **Analisi:**
  - Sfondo con assi (e' disponibile tra gli sfondi standard)
  - Matita (**sprite**)
    - Da scegliere tra quelle disponibili
    - La matita si deve muovere con il mouse
      - Posizione segue mouse
    - La matita deve scrivere **quando** premo il bottone del mouse (**condizione**)
      - Per scrivere, uso la libreria delle azioni [**penna**]
- **Opzionale:**
  - **Aggiungiamo la possibilita' di cancellare tutto se premo spazio**
    - **Azione condizionale**





- **Posso usare gli sprite per scrivere sullo sfondo**
  - **Penna giu'/su** per cominciare (finire) di scrivere
  - Posso scegliere il **colore**, la **luminosita'** e la **dimensione** della penna
  - La punta della penna e' nel centro dello sprite
    - **Cosi' come definito dallo sprite**
      - Si puo' vedere/cambiare usando l'editor di immagini dello sprite stesso (costumi/modifica)
  - Posso **pulire** quello che ho scritto
  - **Timbra** copia l'immagine dello sprite sullo sfondo, come se fosse un timbro.





- **Ho un prato verde. Se faccio click con il mouse su un punto, un fiore appare in quel punto.**
- **Analisi:**
  - (e' piu' complicato di quanto puo' sembrare ad un primo esame)
  - Mi serve uno sfondo con il prato verde (**sfondo**)
    - All'inizio e' bene pulire il prato da tutti i fiori
  - **Quando** clicco sullo sfondo **allora** appare un fiore (**condizione**)
  - Fiore:
    - E' uno sprite
      - Lo sprite deve apparire a comando
      - Lo sprite deve essere posizionato dove e' il mouse (**movimento, posizione, interazione con il mouse**)
      - Voglio molti fiori, non solo uno
        - Invece di fare molti sprite (quanti?) preferisco che quando appare lo sprite la sua immagine venga **stampata** sullo sfondo (**Penna->Timbra**)



- **Quando** clicco sullo sfondo **allora** appare un fiore (**condizione**)
- Quando clicco su uno sprite (lo sfondo) un altro sprite deve fare qualcosa (timbrare)
  - Gli sprite si devono parlare (**messaggi**)
  - Quando (condizione) clicco su sfondo
    - **Sfondo manda messaggio**
      - Fiore e' in ascolto del messaggio
      - Quando riceve il messaggio fa la sua azione
        - Si mette dove e' il mouse
        - Timbra



- **Il computer sceglie un numero casuale tra 1 e 100, l'utente deve indovinare il numero.**
  - Per rendere le cose piu' facili all'utente, il computer indicherà se il numero è troppo alto o troppo basso
  - Quando il numero viene indovinato, si deve indicare la vittoria
  - E magari anche quanti tentativi sono stati necessari



- **Devo muovermi di un numero di passi casuale, altrimenti la corsa non e' interessante.**

- Dovrei lanciare un dado e sulla base del risultato del dado muovere l'auto di 1,2,...,6 passi

- SCRATCH ha un “dado” in (operatori)



numero a caso tra 1 e 10

- Restituisce un numero casuale (pseudo-) tra 1 e 10 o tra N e M a nostra scelta

- Notate la forma del mattoncino (numero casuale ...)

- E' la forma che posso inserire in



fai 10 passi



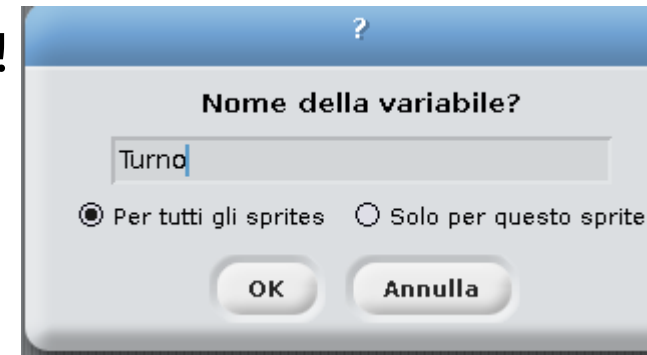
fai numero a caso tra 1 e 10 passi



- E' una variabile
  - Una variabile e' un numero che puo' variare a seconda di determinate condizioni.
  - Fai [10] passi:
    - [10] e' una costante: ho scritto 10 e non cambiera' finche' non lo cambiero' a mano.
  - Fai [N] passi:
    - Il risultato dipende da quanto vale N. E N potrebbe cambiare nel corso del programma.
    - Nel nosto caso, [numero casuale] e' evidentemente una quantita' che puo' cambiare ogni volta che eseguo quella istruzione.



- Dalla libreria **Variabili** [nuova variabile]
  - Si apre una finestra:
    - Nome della variable
      - Mia scelta, mettiamo un nome significativo!
    - Per tutti gli sprite/solo questo sprite
      - Chi puo' usare la variabile
        - tutti o solo questo sprite
      - Per noi il turno e' qualcosa che e' lo stesso per tutti gli sprite, attiene a tutto il gioco.
  - Appaiono nuovi elementi in **Variabili**
    - Per mostrare la variabile sullo stage
    - Per dare un valore alla variabile
    - Per cambiare il valore della variabile (es +42)  
Var=Var+42
    - Per mostrare o nascondere la variabile



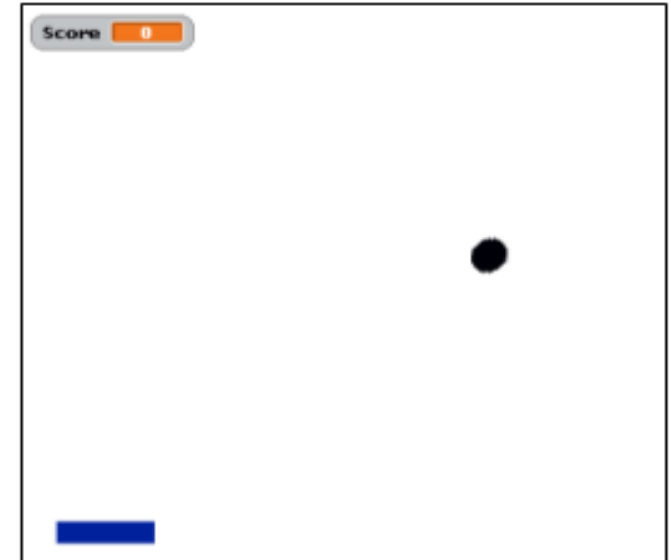


- All'inizio devo scegliere un numero casuale tra 1 e 100
  - E ricordarmi che numero ho scelto (**Variabile**)
- L'utente deve poter immettere un numero (**Chiedi numero**)
  - Devo confrontare il numero di input con quello casuale (**confronto tra variabili**)
  - **Se** il numero e' piu' alto(basso) **allora** devo dire "e' piu' alto(basso)" (**azione condizionale**)
  - **Se** il numero e' giusto **allora** ho vinto
    - Annuncia vittoria in modo appariscente (**costume, suono, ...**)
- Devo **ripetere** l'immissione **finche'** non ho vinto (**Iterazione condizionale**)





- **Una pallina si muove sullo schermo, rimbalzando sulle pareti.**
  - in basso un giocatore (barretta blu) viene mosso con il mouse.
  - Se la barretta intercetta la pallina, questa rimbalza
    - Il punteggio aumenta di 1
    - La pallina rimbalza con un angolo casuale tra -45 e +45 gradi verso l'alto
  - Se la pallina tocca il bordo inferiore, perdo una vita.
    - Se perdo 3 vite, il gioco e' finito
  - Dopo N colpi, la velocita' della pallina aumenta





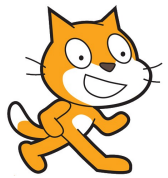
- **Movimento della pallina**
  - Rimbalzare sulle pareti
  - Se tocca il bordo inferiore allora gioco finito (**azione condizionale**)
    - Ricordatevi di come ci siamo accorti che le auto della corsa arrivavano al traguardo!
- **Movimento della barretta**
  - Movimento solo orizzontale!
  - La posizione lungo x deve essere quella del puntatore del mouse
- **Punteggio**
  - Inizialmente e' 0 (**variabile da inizializzare**)
  - Aumenta di 1 ogni volta che la barretta colpisce la pallina (**condizione, variable che cambia valore**)



- **Programmate una scacchiera di Tris tra due persone. Le due persone giocano a turno su uno schema standard, mettendo le solite X e O nelle nove caselle con un click del mouse.**
- **Analisi:**
  - **Mi serve una scacchiera con lo schema vuoto**
  - **Devo far alternare i due giocatori, prima uno e poi l'altro**
    - **Segnalando a chi tocca giocare**
  - **Quando un giocatore fa click su una casella vuota, in quella casella compare il suo simbolo (X/O)**
    - **Se invece la casella e' gia' occupata non succede nulla**
- **Ha alcuni elementi interessanti!**



- **Mi serve una scacchiera (sfondo)**
- **Mi servono le pedine**
  - La pedina puo' essere una X o un O o nessuna (**costume!**)
  - Quando click del mouse su una pedina “vuota”, la pedina deve diventare X o O a seconda di chi sta giocando
    - **Mi concentro su UNA pedina, poi la copiero' nove volte nelle varie caselle!**
  - **Se** il turno e' di X allora diventa X, **altrimenti** diventa O (**azione condizionale!**)
- **Devo alternare I giocatori, un turno per volta**
  - Ogni volta che uno gioca, il turno passa all'altro.



- **Devo fare una certa azione quando la mia variabile ha un certo valore**

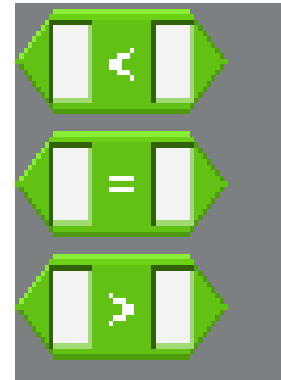
- **Se e' il turno di X diventa X, altrimenti 0 (condizione!)**

- **Mi serve una condizione che dipende dalla variabile**

- Da libreria **Operatori**

- **Se (turno) = X e' vera, altrimenti e' falsa**

- 



- **Devo ricordarmi di cambiare turno!**

- Dovrei anche controllare di non aver gia' giocato su quella casella, cioe' che il costume sia null!

- **Sugg: condizione su numero costume**

- 





- **Metto una pedina in ogni casella.**
  - La pedina e' uno sprite che ha tre **costumi**
    - Vuoto (bianco) - X - O
  - Quando faccio click su una pedina, cambio costume (**Interazione con il mouse e cambiare aspetto**)
    - Se e' il turno di X diventa X, altrimenti O (**condizione!**)
- **Turno:**
  - All'inizio e' il turno di un giocatore.
    - Quando lui gioca, il turno passa all'altro giocatore.
  - Uso una variabile che rappresenta a chi tocca (es turno=1 o 2)
    - E' una variabile nuova:
      - devo crearla,
      - inizializzarla (decidere di chi e' il primo turno)
      - modificarla (ogni volta che cambia il turno)



- **Quando tutte le caselle sono piene, lo sfondo cambia.**
- **Analisi:**
  - Devo accorgermi che tutte le caselle sono “piene”
  - Per esempio, conto quanti turni sono passati
    - Quando arrivo a 9, allora le caselle sono tutte piene
    - Aggiungo una variabile che conta I turni giocati
    - All'inizio e' 0 e aumenta di 1 ogni volta che viene fatta una mossa valida
      - Altre soluzioni sono possibili!
- **Sfida (tosta): siamo capaci di capire chi ha vinto, o se nessuno ha vinto?**
  - Devo analizzare bene cosa vuol dire vincere
    - tre elementi uguali allineati in orizzontale, verticale o obliquo.
      - Lo lascio come (non) facile esercizio per lo studente!



- **Una pallina cade come se fosse soggetta alla gravita', e rimbalza sul pavimento**
- **Analisi:**
  - Esercizio di fisica:
    - serve conoscere il moto uniformemente accelerato
  - Una pallina (**sprite**) all'inizio la mettiamo in alto a sinistra sullo sfondo
    - Si sposta (**muove N passi**) dentro un ciclo (il tempo)
      - Ogni iterazione, si sposta di X in orizzontale e Y in verticale
      - I valore di X e Y dipendono dalla velocita' lungo X e Y
    - Si muove secondo le leggi della gravita'
      - La velocita' lungo x (orizzontale) rimane costante
      - La velocita' lungo y (verticale) all'inizio e' 0
        - Dopo ogni passo, la  $V_y$  aumenta di una quantita' costante (accelerazione di gravita')
    - Mi muovo in modo indipendente lungo X e lungo Y





- Partiamo da un moto non accelerato, cioè senza la gravità.
  - La velocità è costante sia in x che in y
- Faccio due cicli, uno per x e uno per y. Ad ogni iterazione:
  - $X' = X + v_x$ 
    - Cambia X di  $v_x$
  - $Y' = Y + v_y$ 
    - Cambia Y di  $v_y$
- Definisco due **variabili**  $v_x$  e  $v_y$  e gli **assegno** dei valori iniziali.
- Per “vedere” il movimento, stampo **[timbro]** la posizione della pallina ad ogni iterazione
  - come se fosse una foto stroboscopica.



- **Facciamo rimbalzare la pallina sul pavimento:**
  - Disegno sullo sfondo il pavimento di un dato colore
  - **Quando** la pallina **tocca il colore** del pavimento
    - **Rimbalza: cosa significa?**
      - Rimbalzare vuol dire che la sua velocità lungo y viene invertita:  $v'_y = -v_y$  (**assegnare variabile a valore, operatore su variabile:  $-v_y = 0 - v_y$** )
        - Mentre quella lungo x rimane la stessa
    - **Se voglio anche emulare la parte non elastica dell'urto, allora la velocità dopo l'urto sarà un po' più piccola di quella prima dell'urto,**
      - oltre che cambiata di segno



- **Mettiamo la gravita':**
  - La gravita' fa aumentare la velocita' lungo y di una costante (l'accelerazione di gravita' g) verso il basso ad ogni iterazione
    - $v'_y = v_y - g$
  - Mi serve una terza iterazione che fa semplicemente questo.
- **Metto tutto insieme, e ho una pallina che cade sotto gravita' e rimbalza!**



- **Lo sprite segue il puntatore del mouse, ma non si avvicina mai troppo al puntatore stesso.**
- **Analisi:**
  - **Movimento** verso la **posizione del mouse**
  - Se mi **avvicino** troppo allora mi fermo
    - Avvicino: se **distanza** e' piu' piccola di X (**condizione su variabile**)



- **Disegno una linea rossa sullo sfondo. Lo sprite si muove seguendo la linea che ho disegnato**
- **Analisi:**
  - Disegnare una linea rossa (**facile**)
  - Segui la linea
    - Questo non e' per niente facile, occorre una buona analisi prima di partire.
      - Se foste voi a dover seguire una linea disegnata per terra, come fareste?
        - Provate a analizzare le vostre azioni in modo da capire l'algoritmo che usate per un tale compito
        - Tenete presente che alcune azioni che per voi sono possibili (es guardare avanti e vedere in che direzione va la linea) non sono possibili per uno sprite.
        - Provate a pensare se potete emulare questa azione (guardare avanti) usando quello che lo sprite sa fare (muoversi, condizioni, ...)



- **Simuliamo una nevicata. I fiocchi scendono, oscillando a destra e sinistra mentre scendono. Quando arrivano sul terreno, si accumulano su di esso o sulla neve che c'era prima.**

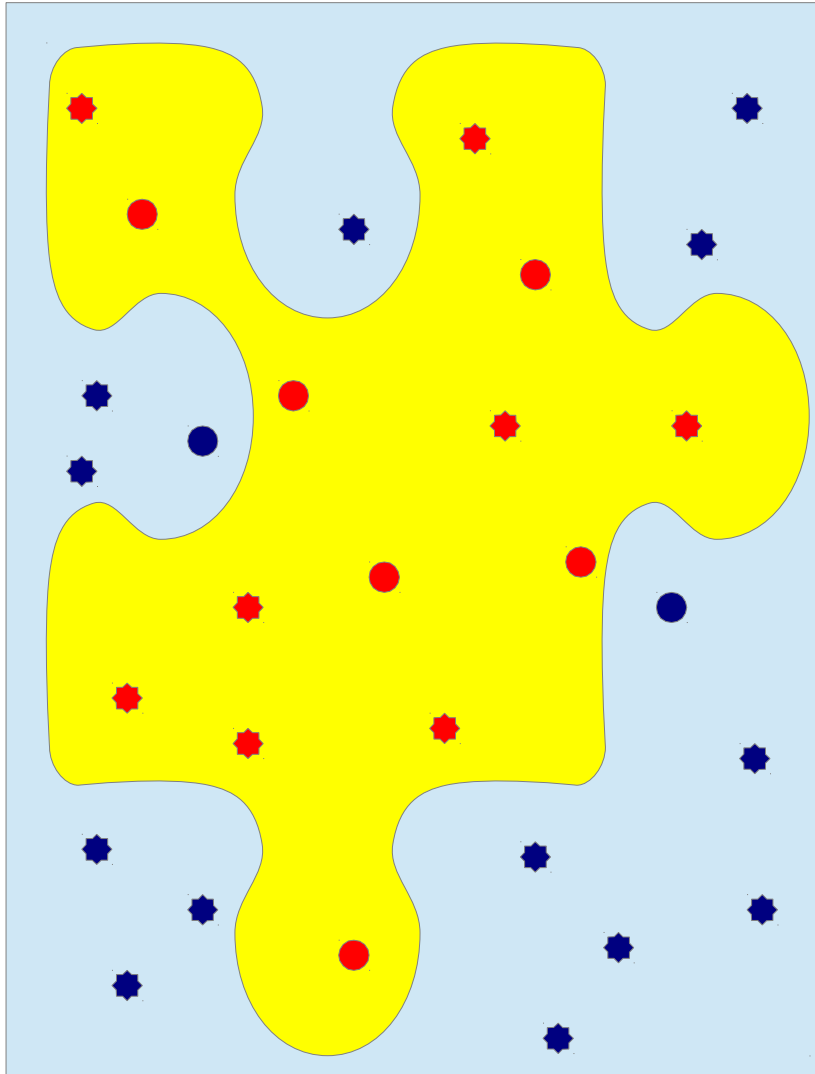


- **Un labirinto: un personaggio parte da un punto e deve arrivare ad un traguardo, senza attraversare le pareti del labirinto.**
  - Il movimento del mio personaggio e' gestito o dalle frecce o dal mouse
  - Mettiamo anche un cronometro per vedere quanto ci mette ad arrivare al traguardo
  - Se tocca le pareti, torna alla partenza
  - Magari mettiamo piu' di un labirinto tra cui scegliere all'inizio del gioco
    - Ci sono moltissimi esempi nella pagina di scratch, compreso uno mio (InfnPadova)



- **Problema: data una superficie di forma qualsiasi (non regolare) stimare la sua area A.**
  - **Metodo MonteCarlo**
    - Disegno la superficie per terra
    - La racchiudo dentro una superficie S piu' grande di area nota (es un rettangolo, o un cerchio o quello che voglio)
    - Lancio un buon numero di sassi dentro la superficie piu' grande.
      - Conto quanti di questi sassi sono dentro la superficie di cui voglio calcolare l'area
      - L'area delle due superfici sara' proporzionale al numero di sassi che sono dentro ciascuna.
        - $A/S = \# \text{sassi in } A / \# \text{sassi in } S$
        - $A = S * (\# \text{sassi in } A / \# \text{sassi in } S)$





- **Voglio calcolare l'area A della superficie gialla.**
  - La racchiudo dentro il rettangolo azzurro
    - Che ha area nota  $S=(base*altezza)$
  - Lancio (tanti) sassi in modo casuale
    - Ho lanciato 30 sassi
    - 14 sono dentro la superficie gialla
  - $A=S*14/30 \sim 1/2*S$
- **E' un metodo approssimato, tanto piu' preciso quanti piu' sassi uso!**
  - Permette di calcolare l'area di una superficie generica.
- **Implementare questo metodo in scratch e' abbastanza facile**
  - Numeri casuali per lanciare i sassi
    - Input di quanti sassi lanciare
  - Variabile (#sassi)
  - Variabile (#sassi dentro)
    - Occorre capire quando un sasso e' dentro o e' fuori
  - Area nota e' lo sfondo (sappiamo base e altezza, in pixel)