

# SCRATCH

## 3.0

### CAPITOLO 15

### I VIDEOGIOCHI



Gli argomenti di questo capitolo:

Realizziamo semplici videogiochi

Far muovere uno sprite con la tastiera

Rotazione e movimento degli sprite

con i tasti freccia

I proiettili nei videogiochi

Il progetto: Porta a spasso il tuo cane!

Il progetto: Un tiro al pallone

Il progetto: Sparatutto. V.1

Il progetto: Sparatutto con boomerang

Il progetto: Pista si corre!

Cercare sprite su Internet

Il progetto: automobilina che accelera e frena!

Il progetto: Acchiappapalla. V1

Il progetto: Acchiappapalla. V2

Il progetto: Il serpente mangiatutto

Il progetto: Fantasmio sparatutto

(....senza cloni)

Il progetto: Memorizza gli animali

della fattoria

## REALIZZIAMO SEMPLICI VIDEOGIOCHI

In questo capitolo realizzeremo qualche semplice videogiochi. Questo libro non è stato concepito per realizzare progetti di videogiochi; qui verrà indicata qualche tecnica: è importante capire il funzionamento base dei videogiochi, apprendere qualche semplice tecnica poi, utilizzando quello che hai appreso nei capitoli precedenti (costumi, variabili, cloni, blocchi personali, estensioni) potrai realizzare tutti i videogiochi che vorrai.

Spesso in un videogiochi c'è qualche sprite che devi muovere tu nello stage, oppure qualche sprite che si muove casualmente o che cade dall'alto. Vediamo come far muovere gli sprite utilizzando la tastiera e i blocchi di **Movimento**.

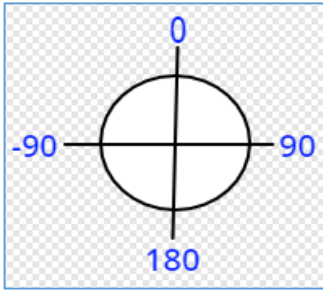
### Far muovere uno sprite con la tastiera



Osserva questi 4 script.

Con il primo codice, ogni volta che premi il tasto **a** sulla tastiera, lo sprite si sposta di 100 passi verso destra (direzione **90** gradi), dopo il suo spostamento si troverà correttamente rivolto verso destra. Vedi la figura 1.

Con il secondo codice puntiamo lo sprite verso sinistra (**-90** gradi), quando premi il tasto **b** sulla tastiera, lo sposti di 100 passi. Anche in questo caso lo sprite, durante il movimento, è rivolto correttamente verso sinistra. Vedi la figura 2. Ma questi due codici sono corretti perché ho messo all'inizio il blocco di **Movimento** **usa stile rotazione sinistra-destra**.



Il terzo script fa spostare verso l'alto (direzione **0** gradi) lo sprite di 100 passi ogni volta che premi sulla tastiera il tasto **c**. Vedi figura 3.

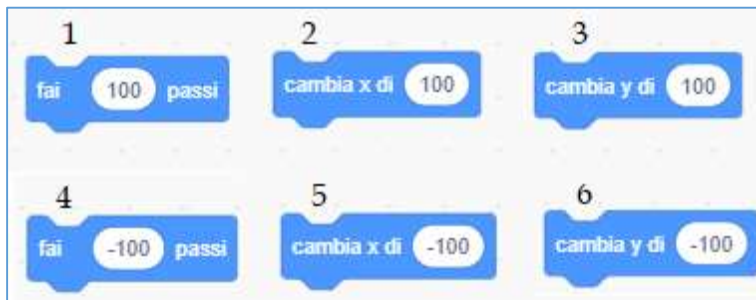
Infine, il quarto script, premendo il tasto lettera **d**, fa spostare lo sprite verso il basso (direzione **180** gradi) di 100 passi. Però in questi ultimi due script all'inizio ho inserito il blocco che permette la rotazione dello sprite: **usa stile rotazione può ruotare**,

che è l'opzione di *default*. Vedi figura 4.

È allora importante capire che puoi muovere gli sprite con i tasti utilizzando il blocco di **Situazione quando si preme il tasto** e che devi impostare una direzione verso la quale devono muoversi i tuoi sprite.

Ovviamente se non è un personaggio, ma una semplice palla per esempio, non ha molto senso inserire il blocco **usa stile rotazione**, ma rimane molto importante indicare la direzione verso la quale deve muoversi.

Ancora: **fai 100 passi** verso destra, equivale al comando **cambia x di 100** passi?



Osserviamo questi 6 blocchi che indicano, rispettivamente:

- 1) spostarsi di 100 passi verso destra, se abbiamo impostato una certa direzione, 90° oppure -90, i 100 passi saranno effettuati, rispettivamente, verso destra oppure sinistra;
- 2) spostarsi di 100 passi sempre verso destra;
- 3) spostarsi sempre verso l'alto di 100 passi;
- 4) spostarsi di 100 passi, ma anche in questo caso dipende dalla direzione impostata;
- 5) spostarsi sempre verso sinistra di 100 passi;
- 6) spostarsi sempre verso il basso di 100 passi.

### Rotazione e movimenti degli sprite con i tasti freccia

Se vogliamo costruire un videogioco in cui lo sprite (per esempio un'astronave) deve sparare contro degli alieni, abbiamo la necessità di dover ruotare la nostra astronave in direzione degli alieni. Abbiniamo i blocchi appena visti con i quattro tasti freccia (ma potrebbero rimanere anche gli stessi tasti precedenti):

## Capitolo 15: I Videogiochi



Quando si premerà uno dei quattro tasti freccia lo sprite punterà, cioè ruoterà nella direzione indicata nel blocco e si sposterà di 10 passi.

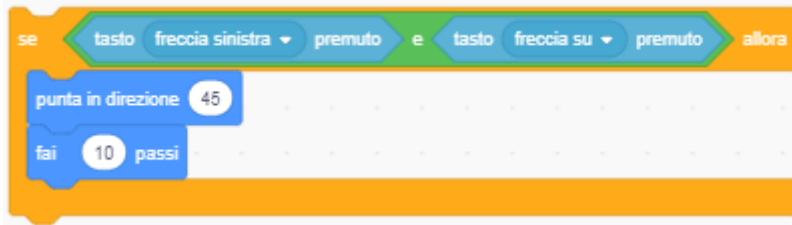
Spesso questi 4 blocchi (escluso l'*Hat Block*) sono inseriti in un ciclo **per sempre** utilizzando il **Controllo se**. Nell'esempio che vedi qui a lato c'è un tipico caso di utilizzo di questi blocchi. In un ciclo **per sempre**, se si preme uno dei quattro tasti



freccia, in nostro sprite **ruoterà** nella direzione indicata sotto il tasto freccia e si muoverà di tanti passi quanti indicati nel blocco **fai n passi**. È utile questo tipo di script per far muovere uno sprite sullo stage. Il nostro sprite possiamo muoverlo liberamente nello stage, guidato da noi attraverso i quattro tasti freccia.

I tasti freccia (destra, sinistra, su, giù) in questo esempio, ti permettono di far puntare in quattro direzioni precise:  $90^\circ$ ,  $-90^\circ$ ,  $0^\circ$  e  $180^\circ$ ; si potrebbero anche collegare tra loro per poter ruotare anche di  $45^\circ$  e di  $-45^\circ$  (o di altri gradi), utilizzando gli operatori relazionali. In questo caso abbiniamo la pressione di due tasti freccia (non sono ovviamente obbligatori i tasti freccia, vanno bene anche altri tasti).

Per far puntare lo sprite in direzione  $45^\circ$  potremmo usare uno script come quello della pagina successiva, in cui viene combinata la pressione contemporanea di due tasti (freccia sinistra e freccia su).



Se sono premuti contemporaneamente sia il tasto **freccia sinistra** sia il tasto **freccia su**, lo sprite si muoverà diagonalmente di 45 gradi e farà 10 passi. Potremmo abbinare questo movimento di 45 gradi (o di altri gradi) anche ad un solo tasto, per esempio al tasto lettera **q** (ho preso un tasto a caso tra i tasti che è possibile utilizzare con il sensore **tasto premuto**).

Ci sono altri interessanti blocchi per il movimento dello sprite.



Il primo blocco, lo abbiamo già usato, comanda allo sprite di **raggiungere** con una certa velocità (1 è una velocità bassissima), **una posizione a caso**; oppure può **scivolare verso il puntatore del mouse** (*posizione a caso e puntatore del mouse*, sono voci fisse nel menu che si apre cliccando sul triangolino); ma potrebbe anche **raggiungere un altro sprite** caricato nel progetto (come abbiamo fatto nel progetto del nastro trasportatore); nel raggiungere un altro sprite, che potremo chiamare “bersaglio”, potrebbero toccarsi: questo dipende da come si trovano centrati rispetto al centro dello stage!

Il secondo blocco **punta verso...** cambia la sua direzione in base alla posizione dell’oggetto verso il quale decidiamo di farlo puntare.

Anche qui, oltre al **puntatore del mouse** (voce fissa) potremmo far puntare il nostro sprite verso un altro sprite del progetto.



In questi altri blocchi, il primo ha due scelte fisse: **posizione a caso** e **puntatore del mouse**; cliccando sul triangolino si troveranno elencati anche tutti gli altri sprite caricati nel progetto.

Gli altri tre blocchi, sempre presenti nella categoria **Movimento**, conservano la coordinata x e la coordinata y della posizione che ha in quel momento lo sprite

attivo. L'ultimo blocco conserva la direzione dello sprite attivo, sempre riferita al momento.

Come le variabili (in effetti questi sono delle variabili), puoi mettere un segno di spunta sul quadratino che trovi accanto a quei blocchi così potrai vedere sullo stage i loro valori.



Non dimentichiamo gli importanti blocchi di rotazione.

Se uno sprite è per esempio inserito in un ciclo **per sempre**, come questo qui a lato, utilizzando i blocchi **ruota di n gradi** insieme a **quando si preme il tasto x** possiamo far ruotare lo sprite mentre si muove.

Come utilizzare questi ultimi blocchi? In mille modi! Immagina un gatto e un topolino che si seguono e poi osserva questo codice:



Sullo stage abbiamo caricato due sprite: un gatto e un topo.

Il codice a sinistra lo mettiamo al gatto. Cosa farà? Si girerà (**punterà**) sempre nella direzione del topo e in mezzo secondo scivolerà verso il topo, ovunque questo si trovi. Questo **per sempre**.

Il topo lo salviamo noi! Infatti il suo codice, che parte insieme a quello del gatto, impone sempre al topo **punta verso il puntatore del mouse** e di **raggiungi il puntatore del mouse**. Il topo, in pratica, si muoverà seguendo sempre il puntatore del mouse... e il gatto lo inseguirà di continuo!



È ovviamente possibile far saltare uno sprite, ma il discorso del salto è un po' più complicato. Questo semplice codice qui a sinistra fa saltare (parola grossa in questo caso) il nostro sprite. In realtà lo solleva dalla sua posizione di 50 passi, dopo 0.3 secondi lo sprite tornerà nella posizione in cui si trovava. Non è certo un vero e proprio salto. Qui manca la "gravità", ossia l'aumento della velocità di caduta mentre lo sprite torna giù. Questo codice appena visto potremmo inserirlo nel codice qui in basso.



Se vuoi vedere come realizzare un sistema per far camminare e saltare uno sprite in modo

realistico, poi prendere spunto da un tutorial realizzato da uno dei più bravi autori di Scratch, si tratta di Griffpatch (lo trovi su Internet facilmente).

## I proiettili dei videogiochi

Spesso nei videogiochi qualcuno spara! Lancia dei proiettili. Cosa sono questi proiettili? In realtà è uno sprite lo sparo lo realizza uno sprite che di continuo crea i suoi cloni, ridotti nelle dimensioni; questi, i cloni, sono proprio i proiettili. Normalmente questi proiettili (cloni) sono eliminati automaticamente quando toccano un bordo dello stage oppure il bersaglio.

Immaginiamo che un'astronave posta sul bordo inferiore, debba sparare dei proiettili per colpire un alieno che scende dall'alto. Cosa succederà? Useremo uno sprite come proiettile e l'astronave comanderà a quello sprite di creare per sempre dei cloni, questi verranno utilizzati come proiettili dall'astronave. I cloni saranno creati in modo continuo, occorrerà quindi qualche sistema utile per eliminarli (altrimenti avremo lo stage pieno di proiettili): normalmente si fa in modo, come abbiamo detto poco fa, che i cloni si autoeliminino quando toccano