

Pygame: Movimenti di base e gestione dei tasti

pygame è un modulo Python progettato per la creazione di giochi 2D. Ha una sintassi molto semplice e intuitiva che ci consente di creare rapidamente giochi potenti.

Creare una finestra

Dopo aver caricato il modulo pygame è una buona idea iniziarlo chiamando la funzione `pygame.init()`. Dopo aver inizializzato pygame possiamo creare una finestra che conterrà il nostro gioco. Crea un file chiamato `primogioco.py` che contiene quanto segue:

```
import pygame
pygame.init()

# Definizione delle costanti con la dimensione della finestra
LARG_FINESTRA = 640
ALT_FINESTRA = 480
# Questa riga crea una finestra di larghezza LARG_FINESTRA, altezza ALT_FINESTRA
finestra = pygame.display.set_mode((LARG_FINESTRA, ALT_FINESTRA))
```

Eseguendo il programma dovrete veder apparire una finestra intitolata “pygame window”. Non vi preoccupate se non riuscite a chiuderla: vedremo tra poco come modificare il programma per renderlo reattivo agli eventi. Questo è fantastico, ma vorremmo dare al nostro gioco un nome più creativo di “pygame window”. Per fare ciò possiamo aggiungere quanto segue al programma.

```
pygame.display.set_caption("Il mio primo gioco")
```

Prima di proseguire, provate ad eseguire di nuovo il programma. Il titolo della finestra dovrebbe essere “Il mio primo gioco”: ora è molto meglio!

Disegnare sullo schermo

Vedremo ora come possiamo disegnare delle figure geometriche sullo schermo e come possiamo spostarle usando la tastiera. Iniziamo definendo il ciclo principale, o ciclo di gioco del programma. Tutti i programmi che devono gestire eventi, come i giochi e le app del vostro cellulare, hanno una serie di istruzioni che viene eseguita continuamente. Queste istruzioni sono responsabili di attività come la verifica di eventi (come eventi da tastiera o dal mouse), lo spostamento di oggetti, l'aggiornamento dello schermo e la fine del gioco. Nel nostro gioco useremo un ciclo **while** per ripetere queste istruzioni.

All'interno del ciclo inseriremo un ritardo in modo da poter controllare la velocità del gioco. Inizieremo anche controllando un evento specifico: la chiusura della finestra di gioco.

Traduzione ed adattamento del “General Pygame Tutorial” di Tech With Tim. <https://techwithtim.net/tutorials/game-development-with-python/pygame-tutorial/>

```

import pygame
pygame.init()

LARG_FINESTRA = 640
ALT_FINESTRA = 480
finestra = pygame.display.set_mode((LARG_FINESTRA, ALT_FINESTRA))
pygame.display.set_caption("Il mio primo gioco")

continua = True

while continua:
    pygame.time.delay(20) # Questo ritarda il gioco per un certo ammontare di millisecondi.
    # Nel nostro caso inserendo 20 millisecondi, o 0.02 secondi di ritardo

    for event in pygame.event.get(): # Questo scorre l'elenco di tutti gli eventi
    # della tastiera o del mouse.
        if event.type == pygame.QUIT: # Verifica se si fa clic sul pulsante
    # di chiusura della finestra
            continua = False          # Termina il ciclo di gioco

pygame.quit() # Se usciamo dal ciclo questo viene eseguito e termina il gioco

```

Ora possiamo disegnare un rettangolo sullo schermo. Iniziamo definendo alcune variabili per rappresentare la sua posizione, le dimensioni e la velocità di spostamento. Inserite il codice seguente prima dell'inizio del ciclo **while**:

```

x = 50
y = 50
larg = 40
alt = 60
vel = 5

```

Disegneremo il rettangolo nel ciclo principale in modo che venga costantemente ridisegnato ad ogni ripetizione del ciclo. Inserite queste due righe di codice all'interno del ciclo **while**. Ricordatevi di indentare correttamente il codice!

```

pygame.draw.rect(finestra, (255,0,0), (x, y, larg, alt))
pygame.display.update()

```

La funzione `pygame.draw.rect` disegna un rettangolo e richiede tre parametri: la superficie dove va disegnato il rettangolo, il colore del rettangolo, ed una quadrupla con la posizione e le dimensioni del rettangolo. Nel nostro caso la superficie è `finestra`, il colore è la tripla di valori RGB `(255,0,0)` che corrisponde al colore rosso, mentre l'ultimo parametro è la quadrupla `(x, y, larg, alt)`. La funzione `pygame.display.update()` aggiorna il contenuto dello schermo, rendendo visibile il rettangolo.

Muovere gli oggetti sullo schermo

Ora possiamo iniziare a controllare gli eventi in modo da poter spostare il rettangolo usando la tastiera. Modificate `primogioco.py` in modo da ottenere qualcosa del genere.

```
import pygame
pygame.init()

LARG_FINESTRA = 640
ALT_FINESTRA = 480
finestra = pygame.display.set_mode((LARG_FINESTRA, ALT_FINESTRA))
pygame.display.set_caption("Il mio primo gioco")

x = 50
y = 50
larg = 40
alt = 60
vel = 5





continua = True
while continua:
    pygame.time.delay(20)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            continua = False

    keys = pygame.key.get_pressed() # Keys ci permette di sapere quali tasti sono premuti.
    # Possiamo controllare se un tasto e' premuto in questo modo.
    if keys[pygame.K_LEFT]: # pygame.K_LEFT e' il tasto "freccia a sinistra"
        pass # pass e' una istruzione che non fa nulla.
    if keys[pygame.K_RIGHT]: # pygame.K_RIGHT e' il tasto "freccia a destra"
        pass
    if keys[pygame.K_UP]: # pygame.K_UP e' il tasto "freccia su"
        pass
    if keys[pygame.K_DOWN]: # pygame.K_DOWN e' il tasto "freccia giu'"
        pass
    pygame.draw.rect(finestra, (255,0,0), (x, y, larg, alt))
    pygame.display.update()

pygame.quit()
```

Il comando **pass** non fa nulla: se provate ad eseguire il programma vi accorgete che il rettangolo non si muove. Per poter muovere il rettangolo dobbiamo cambiare il valore delle variabili `x` e `y`. Questo ci porta al *sistema di coordinate*. In `pygame` l'angolo in alto a sinistra dello schermo è `(0,0)` e in basso a destra `(larg, alt)`. Questo significa che per spostarci verso l'alto dobbiamo sottrarre alla coordinata `y` del nostro personaggio e per spostarci verso il basso dobbiamo sommare alla coordinata `y`.

Eliminate le istruzioni **pass** presenti all'interno delle istruzioni **if** e sostituitele con delle istruzioni che modifichino i valori di `x` e `y` in modo che il rettangolo si muova quando premete i tasti    . Il rettangolo si deve muovere di `vel` pixel per ogni iterazione del ciclo di gioco.

Cancellare lo schermo

Se eseguite la vostra soluzione dell'esercizio precedente spostate il rettangolo, otterrete molto probabilmente qualcosa di simile alla figura qui sotto, dove possiamo vedere la scia lasciata dai rettangoli precedenti.



Per risolvere questo problema, dobbiamo semplicemente cancellare lo schermo prima di disegnarne il rettangolo nella nuova posizione. Possiamo usare il metodo `fill(colore)` dell'oggetto `finestra` per farlo.

```
import pygame
pygame.init()

LARG_FINESTRA = 640
ALT_FINESTRA = 480
finestra = pygame.display.set_mode((LARG_FINESTRA, ALT_FINESTRA))
pygame.display.set_caption("Il mio primo gioco")

x = 50
y = 50
larg = 40
alt = 60
vel = 5
continua = True
while continua:
    pygame.time.delay(20)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            continua = False
    keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT]:
        x -= vel
    if keys[pygame.K_RIGHT]:
        x += vel
    if keys[pygame.K_UP]:
        y -= vel
    if keys[pygame.K_DOWN]:
        y += vel
    finestra.fill((0,0,0)) # riempie lo schermo di nero
    pygame.draw.rect(finestra, (255,0,0), (x, y, larg, alt))
    pygame.display.update()

pygame.quit()
```

Esercizi

Quella che segue è una lista di esercizi che migliorano il programma che avete scritto aggiungendo nuove caratteristiche.

1. Modificate il programma in modo da mantenere il rettangolo sempre visibile e all'interno dello schermo. Per esempio, impedendo il movimento verso l'alto quando il rettangolo tocca il bordo superiore della finestra, impedendo il movimento verso il basso quando il rettangolo tocca il bordo inferiore, e così via. Suggesto: aggiungete ulteriori controlli sui valori delle coordinate `x` e `y` all'interno o dopo le istruzioni **if** per la gestione dei tasti.
2. La funzione `pygame.draw.circle(superficie, colore, centro, raggio, spessore)` disegna un cerchio. I suoi parametri sono:
 - `superficie` indica la superficie (finestra) dove disegnare il cerchio
 - `colore` è una tripla (`r,g,b`) che rappresenta il colore del cerchio
 - `centro` è una coppia di interi (`x,y`) che rappresenta le coordinate del centro del cerchio
 - `raggio` è un intero che rappresenta il raggio del cerchio in pixel
 - `spessore` indica lo spessore della linea che forma la circonferenza. Se è `0` allora disegna un cerchio pieno.

Aggiungete un cerchio di colore giallo (`255,255,0`) e di raggio `20` al gioco. All'inizio il cerchio si trova al centro dello schermo e si muove da solo verso l'alto e verso destra, senza che sia necessario premere alcun tasto. Ad ogni iterazione del ciclo principale il cerchio si muove di `5` pixel lungo ciascuna delle coordinate. Il cerchio deve rimanere all'interno dello schermo e deve rimbalzare quando tocca uno dei bordi, cambiando la sua direzione di movimento come se fosse una biglia che si muove su un tavolo da biliardo.