

METODI PYGAME – Usati per la creazione di Flappy Bird

Introduzione

Che cos'è la pygame?

La Pygame è una libreria di Python che fornisce un insieme di funzionalità per la creazione di giochi e app. Pygame è stato progettato per essere facile da usare e fornisce diversi strumenti per la creazione di giochi, come ad esempio il supporto di immagini, suoni e animazioni. Inoltre, Pygame è molto versatile e supporta diverse piattaforme, tra cui Windows, MacOS e Linux.

Ecco i metodi della pygame utilizzati per la creazione del gioco Flappy Bird

- **pygame.init():** Questo metodo è fondamentale e deve essere inserito all'inizio del file. Esso infatti inizializza i moduli della pygame e permette di utilizzare quindi tutti i metodi.

Ecco un esempio di utilizzo di questo metodo:

```
import pygame
pygame.init()
```

- **pygame.display.set_caption(titolo):** Questo metodo imposta il titolo della finestra del gioco. Il parametro titolo è una stringa che rappresenta il titolo da visualizzare.

Ecco un esempio di utilizzo di questo metodo:

```
pygame.display.set_caption('Flappy Bird - Galeasso')
```

- **pygame.image.load(percorso dell'immagine):** Questo metodo carica un'immagine, che possiamo salvare in un oggetto. Il parametro che richiede è il percorso dell'immagine da caricare.

Ecco un esempio di utilizzo di questo metodo:

```
sfondo=pygame.image.load('./immagini/sfondo.png')
```

- **pygame.transform.flip(oggetto, True/False, True/False):** Questo metodo viene utilizzato per capovolgere un'immagine. Il primo parametro è l'immagine, i parametri successivi sono due booleani che indicano se l'immagine deve essere capovolta orizzontalmente o verticalmente. Nell'esempio seguente, l'immagine ha come valori booleani *False* e *True*, quindi verrà capovolta verticalmente.

Ecco un esempio di utilizzo di questo metodo:

```
tubo_su=pygame.transform.flip(tubo_giu, False, True)
```

- **pygame.mixer.Sound(percorso del suono):** Questo metodo carica un suono che possiamo salvare in un oggetto. Il parametro che richiede è il percorso del suono da caricare.

Ecco un esempio di utilizzo di questo metodo:

```
suono_gameover = pygame.mixer.Sound('./suoni/hit.mp3')
```

- **pygame.display.set_mode(*larghezza, altezza*)**: Questo metodo crea una finestra per il gioco con le dimensioni specificate. I parametri *larghezza* e *altezza* sono interi che rappresentano la larghezza e l'altezza della finestra in pixel.
Ecco un esempio di utilizzo di questo metodo:
`SCHERMO=pygame.display.set_mode((288,512))`
- **SCHERMO.blit(*oggetto dell'immagine caricata, (x, y)*)**: Questo metodo disegna un'immagine sulla finestra di gioco (in questo caso la finestra di gioco è *SCHERMO*). Il parametro *oggetto dell'immagine caricata* è un oggetto immagine che è stato caricato con *pygame.image.load* e i parametri *x* e *y* sono interi che rappresentano le coordinate sulla superficie in cui disegnare l'immagine.
Ecco un esempio di utilizzo di questo metodo:
`SCHERMO.blit(sfondo,(0,0))`
- **.get_width() / .get_height()**: Questi metodi restituiscono la larghezza e l'altezza di un'immagine o di una superficie.
Ecco un esempio di utilizzo di questo metodo:
`uccello_lato_dx=uccellox+uccello.get_width()-tolleranza`
`uccello_lato_giu=uccelloy+uccello.get_height()-tolleranza`
- **pygame.font.Font(*percorso del font, dimensione del font*)**: Questo metodo carica un font e restituisce un oggetto font che può essere utilizzato per disegnare testo nel gioco. Il parametro *percorso del font* è una stringa che rappresenta il percorso del file del font e il parametro *dimensione del font* è un intero che rappresenta la dimensione del font in pixel.
Ecco un esempio di utilizzo di questo metodo:
`stile = pygame.font.Font(None, 50)`
- **stile.render(*testo , anti_aliasing, colore*)**: Questo metodo disegna il testo specificato su una superficie. Il parametro *testo* è una stringa che rappresenta il testo da disegnare, il parametro *anti_aliasing* è un flag booleano che specifica se abilitare o meno l'anti-aliasing, e il parametro *colore* è una tupla di tre interi che rappresenta il colore RGB del testo.
Ecco un esempio di utilizzo di questo metodo:
`testo = stile.render(str(punteggio), True, (255, 255, 255))`
- **pygame.display.update()**: Questo metodo aggiorna la finestra del gioco per visualizzare le modifiche effettuate (ad esempio, le immagini o il testo disegnati).
Ecco un esempio di utilizzo di questo metodo:

```
def aggiorna():
    pygame.display.update()
    ...
```
- **pygame.time.Clock().tick(FPS)**: Questo metodo gestisce il frame rate del gioco, che rappresenta il numero di volte in cui l'immagine viene aggiornata al secondo. Il parametro *FPS* è un intero che rappresenta il numero di fotogrammi al secondo che si devono visualizzare sullo schermo.
Ecco un esempio di utilizzo di questo metodo:

```
def aggiorna():
    pygame.display.update()
    pygame.time.Clock().tick(FPS)
```

- **suono.play()**: Questo metodo riproduce un suono. Può essere utilizzato per riprodurre un suono caricato con *pygame.mixer.Sound*.

Ecco un esempio di utilizzo di questo metodo:

```
suono_gameover.play()
```

- **pygame.event.get()**: Questo metodo cattura l'evento che è accaduto in un istante nel gioco (ad esempio, un tasto che è stato premuto o una finestra che è stata chiusa).

Ecco un esempio di utilizzo di questo metodo:

```
for evento in pygame.event.get():
```

```
...
```

- **evento.type == pygame.KEYDOWN and evento.key == pygame.K_UP**: Questa espressione verifica se un tasto è stato premuto (*pygame.KEYDOWN*) e se il tasto premuto è la freccia su (*pygame.K_UP*). Questa espressione può essere utilizzata per gestire gli input dell'utente dalla tastiera e per fare muovere, ad esempio, il nostro personaggio.

Ecco un esempio di utilizzo di questo metodo:

```
if(evento.type == pygame.KEYDOWN and evento.key == pygame.K_UP):  
    uccello_vely=-10
```

- **pygame.QUIT**: Questa costante viene utilizzata per verificare se un evento è un evento di uscita (ad esempio, se l'utente ha chiuso la finestra del gioco).

Ecco un esempio di utilizzo di questo metodo:

```
if evento.type == pygame.QUIT:
```

```
...
```

- **pygame.quit()**: Questo metodo termina la sessione di Pygame. Di solito viene richiamato per chiudere la finestra di gioco in seguito al click sulla X.

Ecco un esempio di utilizzo di questo metodo:

```
if evento.type == pygame.QUIT:  
    pygame.quit()
```