

Metodi py.game utilizzati per il videogioco “Flappy Bird”

INIZIALIZZAZIONE

`pygame.init()`

- Lo utilizzo per inizializzare la py.game
- Non necessita di parametri

IMPORTAZIONI

`pygame.image.load()`

- Lo utilizzo per caricare un'immagine
- Necessita come parametro il percorso del file immagine
- `nomeImmagine = pygame.image.load('cartella/nomeFile.estensione')`
`sfondo = pygame.image.load('immagini/sfondo.png')`

DISPLAY

`pygame.display.set_mode()`

- Lo utilizzo per impostare lo schermo (area di gioco)
- Necessita come parametri una tupla contenente i 2 valori corrispondenti a larghezza ed altezza del riquadro di gioco che si vuole andare a creare. I due valori devono essere espressi in pixel (px)
- `nomeDisplay = pygame.display.set_mode((larghezza, altezza))`
`SCHERMO = pygame.display.set_mode((288, 512))`

`SCHERMO.blit.()`

- Lo utilizzo per disegnare gli oggetti sullo schermo, questi devono essere inseriti in ordine di sovrapposizione (prima lo sfondo, poi lo sprite etc)
- `nomeDisplay.blit(oggetto, (x, y))`
`SCHERMO.blit(sfondo, (0, 0))`

`pygame.display.update()` e `pygame.time.Clock().tick(FPS)`

- Le utilizzo nella funzione `aggiorna()`, la prima non necessita di parametri mentre la seconda richiede come parametro la costante in cui è salvato il numero di frame per secondo (FPS)

RIPDOUZIONE EFFETTI SONORI

`pygame.mixer.Sound()`

- Lo utilizzo per caricare file audio (effetti sonori)
- Necessita come parametro il percorso del file audio
- `nomeAudio = pygame.mixer.Sound(join("audio", "nomeFile.estensione"))`
`soundPunto = pygame.mixer.Sound(join("audio", "soundPunto.wav"))`

`suono.play()`

- Lo utilizzo per riprodurre un suono importato precedentemente
- Non richiede parametri
- `mioSuono.play()`
`soundGameOver.play()`

SCRITTURA DI UN TESTO

`pygame.font.SysFont()`

- Lo utilizzo per impostare un font per inserire successivamente un testo
- Necessita come parametri il nome del font e la dimensione
- `nomeFont = pygame.font.SysFont("Font", dimensione)`
`fnt = pygame.font.SysFont("Times New Roman", 24)`

`fnt.render()`

- Lo utilizzo per scrivere il testo all'interno di una casella creata precedentemente nella mia finestra di gioco
- Richiede come parametri il testo da inserire, il valore booleano True e il colore che si vuole dare al testo
- `nomeTesto = mioFont.render("testo", True, "colore")`
`scoreText = fnt.render("Score: " + str(score), True, "yellow")`

CONTROLLO INTERAZIONI

- Verifico se è stato rilevato un evento (`evento.type`)
- Se corrisponde ad un tasto premuto (`pygame.KEYDOWN`)
- E se il tasto corrisponde ad uno specifico (`evento.key` e `pygame.K_UP`)
- `if(evento.type == pygame.KEYDOWN and evento.key == pygame.K_UP):`
 #azione da eseguire

`pygame.quit()`

Controllo se l'utente ha cliccato la "X" per chiudere la finestra, se sì → chiudo

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