



# **Costruisci il tuo primo videogioco**

con Python e Pygame Zero

Guida ufficiale per la Masterclass di Orientamento

A cura degli studenti del corso 4 + 2

ITI MARCONI

Docenti: Gennaro - Teani

---

“Il futuro non si predice. Si programma.”

# 1 · Benvenuto nel laboratorio digitale ☒

## [ OBIETTIVO DEL LAB ]

In questa masterclass scoprirai che con poche righe di Python puoi creare un videogioco completo.  
Vedrai come funziona davvero un gioco:

- un personaggio controllabile (il tuo razzo),
- un nemico che ti attacca (l'asteroide),
- un punteggio che sale,
- e un vero GAME OVER.

Tutto questo usando **Pygame Zero**, una libreria pensata proprio per chi inizia.

## 2 · Preparazione ambiente

### [ MATERIALE TECNICO ]

- Python installato sul computer.
- Pygame Zero (installa da terminale con `pip install pgzero`).
- Le immagini `razzo.png` e `asteroide.png` nella cartella `images/`.
- (Opzionale) un suono `esplosione.wav` nella cartella `sounds/`.

### [ AVVIARE IL GIOCO ]

Apri un terminale (Prompt dei comandi) nella cartella del progetto e digita:

1 `pgzrun gioco.py`

### [ RISULTATO ]

Se vedi una finestra che si apre, stai già facendo girare un video-gioco creato da te.

## 3 · Struttura del progetto

[ ORGANIZZA COSI' I FILE ]

```
GiocoPygameZero/  
    ├── gioco.py  
    ├── images/  
    |   ├── razzo.png  
    |   └── asteroide.png  
    └── sounds/  
        └── esplosione.wav
```

[ ATTENZIONE ]

I nomi dei file immagine devono combaciare con quelli usati nel codice. Se l'attore si chiama Actor("razzo"), il file deve chiamarsi razzo.png.

## 4 · Step 1 – La finestra di gioco

### [ COSA FA QUESTO BLOCCO ]

Crea la finestra del gioco (lo "spazio" in cui tutto succede).

```
1 LARGHEZZA = 600
2 ALTEZZA = 400
3
4 WIDTH = LARGHEZZA      # richiesto da Pygame Zero
5 HEIGHT = ALTEZZA
6
7 def draw():
8     screen.fill("black")
```

### [ RISULTATO ]

Se lanci questo codice, vedrai una finestra nera 600x400. Questa è la tua "galassia".

## 5 · Step 2 – Il tuo razzo

### [ CREAZIONE DEL GIOCATORE ]

Aggiungiamo l'attore principale: il razzo pilotato dal giocatore.

```
1 razzo = Actor("raazzo")      # usa images/raazzo.png
2 razzo.x = LARGHEZZA / 2
3 razzo.y = ALTEZZA - 50
4
5 def draw():
6     screen.fill("black")
7     razzo.draw()
```

### [ ESPERIMENTO IN AULA ]

Cambia razzo.y e rilancia il gioco. Chiedi: "sta salendo o scendendo?" (Capite insieme come funzionano le coordinate.)

## 6 · Step 3 – Muovi il razzo

### [ OBIETTIVO ]

Far muovere il razzo con le frecce sinistra/destra. Questa è la prima volta che il gioco "risponde" al giocatore.

```
1 velocita_razzo = 5
2
3 def update():
4     if keyboard.left:
5         razzo.x -= velocita_razzo
6     if keyboard.right:
7         razzo.x += velocita_razzo
8
9     # Limiti dello schermo
10    if razzo.x < 0:
11        razzo.x = 0
12    if razzo.x > LARGHEZZA:
13        razzo.x = LARGHEZZA
```

### [ NOTA ]

update() viene chiamata 60 volte al secondo. Ora il razzo risponde ai tasti ↎ ↎. È vivo.

## 7 · Step 4 – Il nemico cade

### [ ASTEROIDE NEMICO ]

Creiamo l'asteroide che scende verso di te.

```
1 from random import randint
2
3 asteroide = Actor("astroide")    # usa images/astroide.png
4 asteroide.x = randint(0, LARGHEZZA)
5 asteroide.y = 0
6
7 velocita_astroide = 3
8
9 def update():
10     # ... (qui sopra c'è già il movimento del razzo) ...
11
12     # movimento dell'astroide
13     asteroide.y += velocita_astroide
14
15     # se esce dal basso, ricompare dall'alto
16     if asteroide.y > ALTEZZA:
17         asteroide.y = 0
18         asteroide.x = randint(0, LARGHEZZA)
```

### [ RICORDA DI DISEGNARLO ]

Oltre a muoverlo, dobbiamo anche disegnarlo nello schermo:

```
1 def draw():
2     screen.fill("black")
3     razzo.draw()
4     asteroide.draw()
```

# 8 · Step 5 – Punteggio e collisione

## [ LOGICA DI PARTITA ]

Aggiungiamo le variabili per punteggio, record e la logica di collisione.

```
1 gioco_finito = False
2 punteggio = 0
3 miglior_punteggio = 0
4
5 def update():
6     global gioco_finito, punteggio, miglior_punteggio
7
8     if not gioco_finito:
9         # se l'asteroide è stato schivato
10        if asteroide.y > ALTEZZA:
11            asteroide.y = 0
12            asteroide.x = randint(0, LARGHEZZA)
13            punteggio += 1
14            if punteggio > miglior_punteggio:
15                miglior_punteggio = punteggio
16
17        # controllo collisione
18        if razzo.collidirect(astroide):
19            gioco_finito = True
20            try:
21                sounds.esplosione.play()
22            except:
23                pass
```

## [ SPIEGAZIONE ]

Ora il gioco tiene il conto dei punti e rileva le collisioni. Nella prossima sezione aggiungeremo la grafica del punteggio e la possibilità di ricominciare.

# 9 · Step 6A – Disegno e messaggi di fine partita

## [ TESTO A SCHERMO ]

Mostriamo il punteggio, il record e i messaggi di fine partita.

```
1 def draw():
2     screen.fill("black")
3     razzo.draw()
4     asteroide.draw()
5
6     # Punteggio e record
7     screen.draw.text(f"Punteggio: {punteggio}",
8                      (10, 10), color="white", fontsize=30)
9     screen.draw.text(f"Record: {miglior_punteggio}",
10                      (LARGHEZZA - 180, 10), color="yellow", fontsize=30)
11
12     # Messaggi di fine partita
13     if gioco_finito:
14         screen.draw.text("GAME OVER",
15                           center=(LARGHEZZA/2, ALTEZZA/2 - 20),
16                           color="red", fontsize=60)
17         screen.draw.text("Premi R per ricominciare",
18                           center=(LARGHEZZA/2, ALTEZZA/2 + 30),
19                           color="white", fontsize=32)
```

## [ SPIEGAZIONE ]

La funzione `draw()` disegna tutto ciò che vedi a schermo. In caso di `game_over`, mostra il messaggio e invita al riavvio.

# 10 · Step 6B – Riavvio e reset del gioco

## [ OBIETTIVO ]

Aggiungiamo la funzione `on_key_down()` per ripartire dopo il GAME OVER.

```
1 def on_key_down(tasto):
2     global gioco_finito, punteggio, asteroide
3
4     if tasto == keys.R and gioco_finito:
5         gioco_finito = False
6         punteggio = 0
7
8         # Posizione iniziale del razzo e dell'asteroide
9         razzo.x = LARGHEZZA / 2
10        razzo.y = ALTEZZA - 50
11
12        asteroide.x = randint(0, LARGHEZZA)
13        asteroide.y = 0
```

## [ VERIFICA IN AULA ]

Premi R dopo un GAME OVER per verificare se il gioco riparte. Se funziona, avete costruito un ciclo di gioco completo! ☺

# 11 · Il gioco è tuo

## [ SFIDE CREATIVE ]

- **Modalità difficile:** aumenta `velocita_asteroide` da 3 a 8.
- **Cambia atmosfera:** usa `screen.fill("midnightblue")`.
- **Doppio asteroide:** crea un nuovo Actor("asteroide2").
- **Effetti sonori:** aggiungi `esplosione.wav` in `sounds/`.
- **Messaggio personale:** cambia "GAME OVER" con "SEI STATO DISTRUTTO ☹".

"Il codice è la lingua del futuro. Impara a parlarla."