

Progettare un Gioco

In questo progetto vedremo le varie fasi di sviluppo di un gioco. Useremo il modulo `pygame`, le funzioni e le istruzioni `if` per costruire una variante di Pong, uno dei primi videogiochi arcade. Pong è un gioco relativamente semplice da costruire e che offre una buona opportunità per lavorare sulle abilità che servono per costruire programmi più complicati.

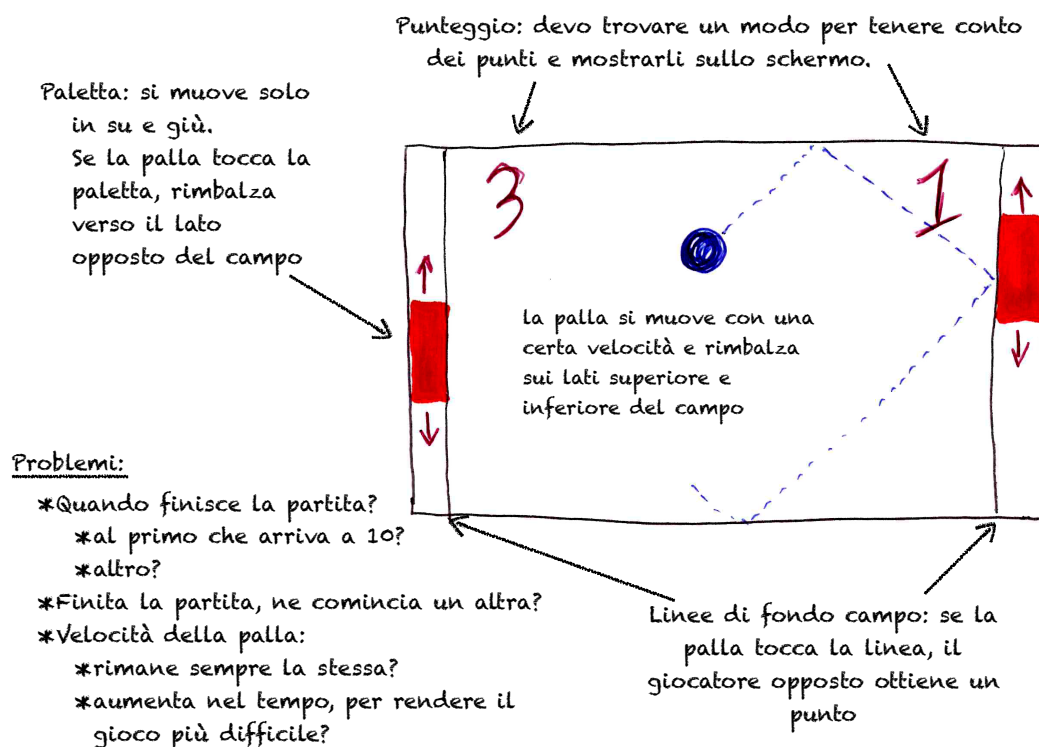
L'obiettivo è seguire un processo creativo dall'inizio alla fine per vedere come le idee prendono vita e possono essere realizzate in Python grazie alle cose che hai imparato nelle prime lezioni del corso.

Per fare ciò seguiremo una immaginaria discussione tra due persone: Michele e Alessia, che lavorano insieme e condividono idee su come progettare il gioco e sviluppare il codice. Sarete voi a scrivere il codice, ma seguire Michele e Alessia vi darà un'idea del processo di sviluppo.

Michele: "Ciao Alessia, voglio creare un gioco, mi puoi aiutare?"

Alessia: "Ma certo! Che gioco vorresti fare?"

Michele: "Vorrei fare Pong, hai presente qual'è? Ho fatto uno schizzo di quello che ho in mente."



Alessia: "Wow! Che bel disegno! Penso di aver capito cosa vuoi fare. Come posso aiutarti?"

Michele: "Da dove ci conviene iniziare?"

Alessia: "Mi sembra che la prima cosa che conviene fare sia capire come disegnare il campo di gioco. Quindi affrontiamo questo problema per primo: ci occuperemo dopo della palla, delle racchette e di come mostrare e segnare i punti."

Michele: "Okay, hai ragione. Ho qui il programma della settimana scorsa: possiamo modificarlo per aggiungere le linee di fondo campo. Ho visto che si può usare la funzione `draw.line` per disegnare delle linee."

Alessia: " `draw.line`? Che cos'è?"

Michele: "Hai presente le funzioni `draw.circle` e `draw.rect`? `draw.line` si comporta in modo simile, ma disegna una linea invece di cerchi e rettangoli."

Alessia: "Ho capito. Fammi vedere come si fa."

Esercizio 1

Modificate il codice del programma `primogioco.py` della lezione scorsa per fare in modo che disegni le due linee di fondo campo sullo schermo. Usate la funzione

```
pygame.draw.line(superficie, colore, inizio, fine, spessore)
```

per disegnare le linee. I suoi parametri sono:

- **superficie** indica la superficie (finestra) dove disegnare la linea
- **colore** è una tripla (**r,g,b**) che rappresenta il colore della linea
- **inizio** e **fine** sono due coppie di interi (**x,y**) che rappresentano le coordinate degli estremi
- **spessore** è un intero che indica lo spessore della linea.

Michele: “Ecco fatto. Ora il campo di gioco viene mostrato correttamente. Adesso cosa facciamo?”

Alessia: “Disegniamo e facciamo muovere la palla?”

Michele: “Buona idea! Questo l’abbiamo già fatto la scorsa settimana. Basta ricopiare il codice.”

Alessia: “Non proprio, ora c’è una differenza. La palla rimbalza solo sui lati superiore e inferiore dello schermo. Se tocca una linea di fondo campo invece non deve rimbalzare: il giocatore opposto ottiene un punto e la palla torna al centro del campo.”

Michele: “È vero! Quindi dobbiamo anche decidere come segnare i punti ...”

Alessia: “Per il momento forse non serve: facciamo solo tornare la palla al centro. Al punteggio ci pensiamo dopo.”

Michele: “Ok, hai ragione.”

Esercizio 2

- Aggiungete al programma il codice che disegna la palla che si muove sullo schermo. La palla deve rimbalzare sul bordo superiore e sul bordo inferiore dello schermo. Sperimentate diverse velocità iniziali per testare il vostro codice.
- Aggiungete il codice che verifica se la palla tocca le linee di fondo campo a sinistra e destra (Fate attenzione al fatto che le linee di fondo campo sono sfalsate dai bordi sinistro e destro dello schermo per la larghezza della racchetta). Quando la palla tocca una linea di fondo campo, rimettete la palla al centro del campo, orientata verso la linea di fondo campo opposta.

Alessia: “Ora la palla è fatta. Passiamo alle due racchette?”

Michele: “Va bene. Quali tasti usiamo per muovere le racchette?”

Alessia: “Possiamo usare freccia su e freccia giù per la racchetta di destra, e W e S per la racchetta di sinistra, che ne dici?”

Michele: “Per me va bene. Ti ricordi come si faceva a far muovere il rettangolo con la tastiera? Possiamo usare lo stesso metodo per muovere le due racchette in su e in giù.”

Alessia: “Sì, mi ricordo, il codice è questo qui:”

```
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
```

Michele: “Ora le racchette si muovono solo in verticale. Se togliamo i due `if` per `keys[pygame.K_LEFT]` e `keys[pygame.K_RIGHT]` abbiamo fatto la racchetta di destra.”

Alessia: “E per la racchetta di sinistra?”

Michele: "Basta cambiare un po' il codice per la racchetta di destra. Possiamo usare una nuova variabile `y_sinistra` per memorizzare la posizione verticale della racchetta di sinistra, e poi controllare se i tasti W o S sono premuti, come abbiamo già fatto per l'altra racchetta."

Alessia: "Giusto: ci serve una nuova variabile. Come facciamo a controllare i tasti? Cosa va messo al posto di `pygame.K_LEFT` e `pygame.K_RIGHT`"

Michele: "Il tasto W si chiama `pygame.K_w`, mentre il tasto S si chiama `pygame.K_s`."

Alessia: "Ottimo! Non ci resta che provare!"

Esercizio 3

- Aggiungete il codice che disegna le due racchette dietro le rispettive linee di fondo campo. I tasti `W` e `S` controllano la posizione della racchetta di sinistra mentre i tasti `↑` e `↓` controllano la posizione della racchetta di destra. Per esempio, nella nostra versione di Pong, la racchetta di sinistra si alza se si preme il tasto `W` e si abbassa se si preme `S`, mentre rimane immobile se non viene premuto nessuno dei due tasti.
- Limitate il movimento delle racchette per mantenerle all'interno dello schermo aggiungendo un controllo. In particolare, verificate se l'aggiornamento della posizione di una racchetta fa uscire la racchetta dallo schermo. Se lo fa, non consentire l'aggiornamento della posizione.

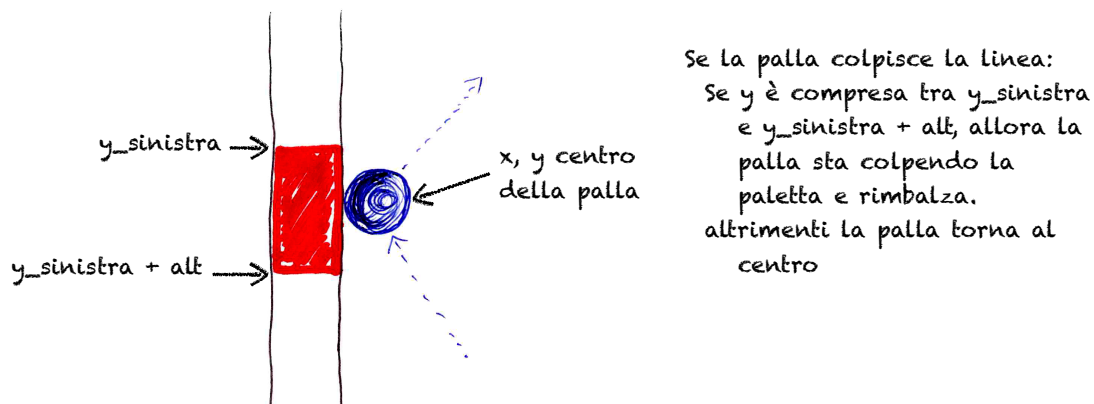
Michele: "Ora ci sono anche le racchette e si muovono. Cosa facciamo adesso?"

Alessia: "Le racchette si muovono, ma la palla non rimbalza quando colpisce una racchetta."

Michele: "È vero! Dobbiamo cambiare il codice che controlla le collisioni con le linee di fondo campo, aggiungendo un nuovo caso: se la palla colpisce una racchetta, rimbalza invece di tornare a centro campo."

Alessia: "Come facciamo a sapere quando la palla colpisce una racchetta?"

Michele: "Ho un'idea: ti faccio un disegno per spiegarla."



Alessia: "Penso di aver capito! Vediamo se funziona."

Esercizio 4

- Modificate il codice di controllo della collisione con le linee di fondo campo per verificare se la palla sta colpendo una racchetta quando tocca una linea di fondo campo. Se è così, fate rimbalzare la palla mantenendola in gioco. Questo modello di collisione elimina la possibilità che la palla colpisca il bordo superiore o inferiore delle racchette e semplifica enormemente il codice di collisione/rimbalzo.

Michele: "Ora la palla rimbalza sulle racchette. Siamo quasi alla fine, cosa manca?"

Alessia: "Non abbiamo ancora pensato al punteggio."

Michele: "Ok. Cosa ci potrebbe servire?"

Alessia: "Abbiamo bisogno di memorizzare il punteggio dei due giocatori da qualche parte. E poi dobbiamo trovare un modo per mostrarlo sullo schermo."

Michele: “Memorizzare i punti è facile: basta usare due nuove variabili. Per esempio `punteggio1` con i punti del primo giocatore e `punteggio2` con i punti del secondo giocatore. Però non so come fare per mostrare il punteggio. Possiamo scrivere sullo schermo con `pygame`?”

Alessia: “Sì, si può scrivere del testo sullo schermo. È un po’ più complicato di disegnare figure. Bisogna fare tre cose.”

Michele: “Quali sono?”

Alessia: “Come prima cosa bisogna aggiungere `import pygame.freetype` all’inizio del programma, per caricare il modulo `pygame.freetype` con le funzioni per scrivere.”

Michele: “E poi?”

Alessia: “Poi bisogna scegliere il tipo di carattere che vogliamo usare per il testo e caricarlo.”

Michele: “A me piace il ‘Comic Sans MS’. E a te?”

Alessia: “Anche a me piace. Per usarlo dobbiamo creare un oggetto di tipo ‘Font’ in questo modo:”

```
font = pygame.freetype.SysFont('Comic Sans MS', 48)
```

“Ho messo dimensione 48 così la scritta è bella grande.”

Michele: “Creato il font come si fa a scrivere?”

Alessia: “Si deve usare il metodo `render_to` dell’oggetto `font`. Ora ti faccio vedere come si fa.”

Esercizio 5

- Aggiungete due variabili `punteggio1` e `punteggio2` per memorizzare il punteggio dei due giocatori. All’inizio entrambe le variabili valgono 0. Non preoccupatevi di aggiornare il loro valore, vedremo dopo come si fa.
- Mostrate il punteggio sullo schermo. Dopo aver caricato il modulo `pygame.freetype` e aver creato l’oggetto `font` come spiegato da Alessia, potete usare il metodo

```
font.render_to(superficie, posizione, testo, colore)
```

per scrivere il punteggio. I suoi parametri sono:

- `superficie` indica la superficie (finestra) dove disegnare la linea
- `posizione` è una coppia di interi (x,y) con la posizione dove scrivere
- `testo` è la stringa da scrivere
- `colore` è una tripla (r,g,b) che rappresenta il colore del testo

Attenzione: `testo` deve essere una stringa. Per scrivere un valore intero bisogna usare la funzione `str(numero)` per convertire l’intero in una stringa.

Michele: “Bello, mi piace! Ora pensiamo a come aggiornare il punteggio e poi abbiamo finito.”

Alessia: “Questo è facile: basta aumentare di 1 la variabile `punteggio1` o la variabile `punteggio2`, a seconda di quale giocatore guadagna un punto, quando facciamo tornare la palla al centro.”

Michele: “È vero! Ora lo faccio e poi possiamo giocare!”

Esercizio 6

- Aggiornate il punteggio del gioco. Ogni volta che la palla colpisce la linea di fondo campo a sinistra o a destra (ma non una racchetta), il giocatore opposto riceve un punto e la palla viene riportata al centro del campo in modo appropriato.

Sfide!

- Per aumentare un po' la difficoltà del gioco, aumentate la velocità della palla del 10% ogni volta che colpisce una racchetta.
- Stabilite quando finisce la partita e trovate un modo per riavviare il gioco quando uno dei due giocatori ha vinto, iniziando una nuova partita.
- Aggiungete degli effetti sonori al gioco. Sul padlet

https://unipd.padlet.org/davide_bresolin/python1

potete trovare il file `pop.wav` con il suono della palla che rimbalza. Per utilizzarlo nel programma occorre come prima cosa caricare il file usando l'istruzione

```
suono = pygame.mixer.Sound("pop.wav")
```

Dopodiché il suono può essere riprodotto con l'istruzione

```
suono.play()
```

- Fate in modo che una delle due racchette venga controllata dal computer e non dalla tastiera.