



Quizzone - il codice per vincere

Game Design & Data Analysis con Pygame Zero

 III Liceo Scientifico Biella - Scienze Applicate

 Python Biella Group



Obiettivi del Progetto

In questa lezione analizzeremo un sistema di Quiz Evoluto che integra:

1. **Manipolazione Dati:** Utilizzo di `Po1ars` e `csv` per la lettura ed il salvataggio dei dati in un CSV.
2. **Nuova Gestione Stati di Gioco:** Gestione della fase di input (Nickname) e del Game Loop.
3. **Analisi del Tempo:** Misurazione del tempo di risposta in millisecondi tramite il delta-time (`dt`).



Polars: Caricamento e Shuffle

Leggiamo i dati in modo casuale usando Polars.

```
def carica_e_mischia():  
    # Caricamento ultra-veloce del CSV  
    df = pl.read_csv("domande.csv", truncate_ragged_lines=True)  
  
    # Rimescolamento dell'intero DataFrame (fraction=1.0)  
    df_shuffled = df.sample(fraction=1.0, shuffle=True)  
  
    # Conversione in lista di dizionari per Pygame Zero  
    lista_domande = df_shuffled.to_dicts()
```




La Fisica del Tempo: `update(dt)`

Il gioco distingue tra due tipi di tempo:

1. **Tempo Logico (`tick`)**: Scandisce i secondi del timer (es. 15 secondi per rispondere).
2. **Tempo Reale (`update`)**: Misura quanto è stato veloce il giocatore.

```
def update(dt):  
    global millisecondi_trascorsi  
    if not game_over and not entering_name:  
        # dt è il tempo (frazione di secondo) tra due frame  
        millisecondi_trascorsi += int(dt * 1000)
```

Il calcolo della barra del timer usa una proporzione lineare:



Gestione Input: Nickname

Il gioco inizia in uno stato di "attesa input". Usiamo l'evento `on_key_down` per costruire una stringa dinamica:

```
if entering_name:
    if key == keys.BACKSPACE:
        nome_utente = nome_utente[:-1] # Rimuove l'ultimo carattere
    elif key == keys.RETURN:
        start_game() # Passa alla fase di gioco
    else:
        # Aggiunge il carattere digitato
        nome_utente += key.name
```




Le fasi di gioco

Struttura degli Stati

- `entering_name = True`: Menu iniziale.
- `entering_name = False`: Quiz attivo.
- `game_over = True`: Schermata risultati.



Salvataggio Risposte

Il cuore del progetto è la **persistenza dei dati**. Ogni risposta viene salvata in un file unico per giocatore: `nome_risposte.csv`.

nome_utente	id_domanda	numero_risposta	tempo_risposta (ms)
...	42	2	1450
...	12	1	890

```
def salva_risposta(nome, id_domanda, numero_risposta, tempo_risposta):  
    with open(nome_file_risposte, mode="a", newline="") as f:  
        writer = csv.writer(f)  
        writer.writerow([nome, id_domanda, numero_risposta, tempo_risposta])
```




UI: Styled Rectangles & Textbox

Per ottenere un look migliore, il codice utilizza una funzione di disegno personalizzata per creare un effetto ombra:

```
def draw_styled_rect(rect, color):  
    # Disegna l'ombra (spostata di 4 pixel)  
    screen.draw.filled_rect(Rect(rect.x + 4, rect.y + 4, rect.w, rect.h), COLOR_SHADOW)  
    # Disegna il rettangolo principale  
    screen.draw.filled_rect(rect, color)
```

`screen.draw.textbox`: Uno strumento potente di Pygame Zero che adatta automaticamente il testo all'interno di un `Rect`, gestendo a capo e centraggio.



Logica di Navigazione (Collisioni)

Quando l'utente clicca, dobbiamo capire quale opzione ha scelto. Usiamo un ciclo `enumerate` sulla lista di `answer_boxes`:

```
def on_mouse_down(pos):  
    for i, box in enumerate(answer_boxes):  
        if box.collidepoint(pos):  
            numero_risposta = i + 1  
            salva_risposta(nome_utente, id_domanda, numero_risposta, tempo_ms)  
            prossima_domanda()
```

***Nota:** Il sistema è progettato per essere "cieco". Non sa se la risposta è corretta; l'analisi della correttezza verrà fatta in una fase di **Post-Processing** sui dati salvati.*

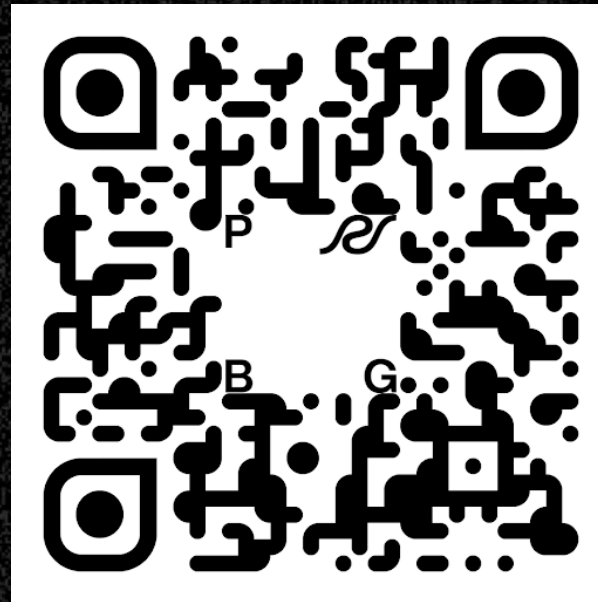


Possibili Evoluzioni

1. **Analisi Real-time:** Usa Polars durante il `game_over` per calcolare la media dei tempi di risposta e mostrarla al giocatore.
2. **Effetti Sonori:** Aggiungi un feedback audio per il click o per il tempo che scorre (tic-toc).
3. **Difficoltà Dinamica:** Accorcia `TEMPO_DOMANDA` man mano che il giocatore procede.



Grazie per l'attenzione...



"C'è sempre qualcosa da imparare per migliorarci e crescere...insieme!"