

Report LAB02

Nicola Modugno

prof.ssa Serena Morigi

Abstract—Questo report descrive lo sviluppo di una demo 2D interattiva basata sul progetto LAB_2 fornito, con l'aggiunta di animazioni, fisica, interfaccia grafica e meccaniche di gioco. Sono stati integrati un ciclo giorno/notte, la possibilità di trascinare la pallina con il mouse, un sistema di sbarre dinamiche, un'interfaccia utente realizzata con ImGui e diverse migliorie grafiche.

1. Introduzione

Il progetto si basa sulla traccia proposta nel laboratorio LAB_2, che richiedeva di compilare ed eseguire i progetti di base forniti e sviluppare un piccolo videogame in stile 2D. A partire dal progetto iniziale (Jumpball) sono state implementate delle regole di gioco originali che coinvolgono l'interazione con l'utente, punteggio e timer. La scena è stata arricchita con movimenti dinamici simili alla realtà come il ciclo giorno/notte, la rotazione delle pale eoliche ed effetti particellari.

2. Trascinamento della pallina e rotazione delle pale eoliche

Una delle prime funzionalità introdotte è stata la possibilità di trascinare la pallina con il mouse. Quando l'utente clicca sulla pallina, il sistema verifica la distanza tra il punto di clic e il centro della pallina. Se la distanza è inferiore al raggio, viene attivata la modalità di trascinamento. Questo comportamento è implementato nella funzione seguente: Le pale eoliche, che inizialmente erano statiche, sono state aggiornate affinché ruotassero nel tempo. Per ottenere questo comportamento, sono state decommentate le righe di codice nel file LAB2D.cpp che incrementano l'angolo di rotazione e aggiornano l'immagine sullo schermo.

3. Ciclo giorno/notte animato

Per rendere l'ambiente più dinamico, è stato aggiunto un ciclo giorno/notte. Il sole si muove lungo una traiettoria ellittica definita in forma parametrica in cui le sue coordinate vengono aggiornate secondo le formule:

$$\begin{aligned}x &= c_x + r_x \cos(t + 2\pi) \\ y &= c_y + r_y \sin(t) \\ t &\in [0, 2\pi]\end{aligned}$$

In base a queste coordinate il colore del cielo, del sole e del suo alone cambiano progressivamente grazie alla funzione di interpolazione lineare

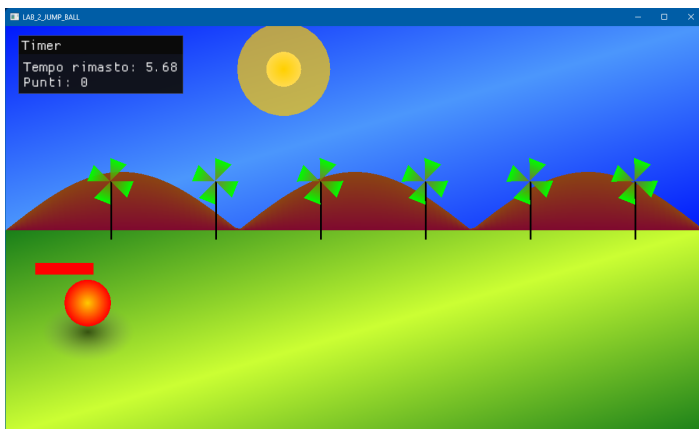


Figure 1. Demo 2D con sole, sbarre, ombra e interfaccia

4. Fisica del rimbalzo e gestione dell'ombra

Inizialmente, la pallina rimbalzava anche contro il "cielo", creando un comportamento poco realistico. Il controllo è stato corretto per limitare il rimbalzo solo al suolo. Anche l'ombra della pallina è stata migliorata: ora segue la pallina solo in orizzontale quando essa supera l'orizzonte, come mostrato in figura 3. Inoltre, la dimensione dell'ombra è regolata tramite una funzione di lerp sulla distanza tra la pallina e l'ombra così da mantenerne la scala originale quando la pallina è vicino al terreno e rimpicciolirla man mano che questa si allontana.



Figure 2. Spostamento dell'ombra

Il presente lavoro ha migliorato anche l'aspetto relativo al rimbalzo della pallina, introducendo delle nuove variabili che cambiano in base alla posizione della palla. Quando la palla è in fase di caduta la sua velocità diminuisce così da diminuire l'altezza del rimbalzo e quando supera una certa soglia si ferma

```
1 float distacco_da_terra = 0; // distacco da terra
2 float velocitaY = 0.0f; // velocit verticale
3 float gravita = -1.2f;
4
5 [...]
6
7
8 void init(){
9     [...]
10    velocitaY += gravita;
11    posy += velocitaY;
12
13    float groundLevel = height * 0.2f; // livello del
14    // prato
15
16    if (posy <= groundLevel) {
17        posy = groundLevel;
18        velocitaY *= -0.8f; // rimbalzo attenuato
19
20        // se il rimbalzo troppo piccolo, ferma tutto
21        if (fabs(velocitaY) < 1.0f)
22            velocitaY = 0;
23    }
24
25    // Limita l'altezza massima
26    if (posy > height - 80) {
27        posy = height - 80;
28        velocitaY *= -0.5f;
29    }
30
31    [...]
32 }
```

5. Sistema particellare

Il sistema particellare del progetto LAB2D è stato realizzato per simulare effetti visivi dinamici, come piccoli fuochi d'artificio. Le particelle sono memorizzate in un array dinamico di tipo `Pointxy`, inizializzato nel file `init_geometrie.cpp`. Ogni particella è un punto dotato di posizione (x,y) e colore (r,g,b,a) . Durante il gioco, le particelle vengono gestite tramite tre funzioni principali: `aggiungiParticella(vec3 posizione, vec3 direzione)` che aggiunge una nuova particella nella direzione desiderata, codificando la direzione nel colore (r,g) , `aggiungiFlussoColorato(vec3 posizione)` che genera un flusso circolare di particelle attorno a un punto dato, iterando su un angolo che va da 0 a 2π e `muoviParticella(int index)` che aggiorna posizione e trasparenza della particella, decodificando la direzione a partire dai canali r e g . Le particelle vengono animate ogni frame dalla funzione `aggiornaParticelle()`, che aggiorna tutte le particelle attive e rimuove quelle con $a \leq 0$. Infine, le particelle vengono rese visibili attraverso un controllo nella funzione `update()` che verifica la presenza di particelle attive e lo stato della partita:

6. Sbarre dinamiche e sistema di punteggio

Sono state aggiunte delle sbarre che appaiono casualmente e premiano l'utente quando vengono colpite e il loro rendering è gestito nella funzione `drawScene()` in `LAB2D.cpp`. La creazione avviene utilizzando la funzione `disegna_pianoxy()` passando come parametro il puntatore all'oggetto `Pointxy` della Sbarra dichiarato in `init_geometrie.cpp`. Queste, inoltre, dispongono anche di un loro VAO e VBO che vengono popolati nella funzione `init()`. La sua posizione cambia ogni volta che questa viene colpita.

7. Interfaccia grafica con ImGui

Per visualizzare il punteggio e il tempo rimanente è stata utilizzata la libreria `ImGui` e realizzata un'apposita funzione `my_interface()` nel file `Gui.cpp` che viene richiamata nel ciclo principale del gioco. L'interfaccia è semplice, e non-collapsible. Inoltre, grazie all'integrazione con `ImGui`, viene mostrata una finestra popup al centro della finestra con il messaggio "Game Over" che include anche il totale dei punti accumulati.



Figure 3. Spostamento dell'ombra