

## "Creazione di un gioco sparatutto 2D"

### Obiettivi dell'esercitazione:

1. Imparare a inizializzare una finestra di gioco utilizzando FreeGLUT.
2. Utilizzare funzioni di disegno di base per creare forme geometriche per rappresentare il giocatore, i nemici e i proiettili.
3. Implementare la gestione del giocatore e dei nemici.
4. Gestire le collisioni tra i proiettili del giocatore e i nemici.
5. Aggiungere uno sfondo animato utilizzando un fragment shader.
6. Utilizzare la libreria "glm" per gestire le animazioni del giocatore e dei proiettili.
7. Utilizzare curve interpolanti di Hermite per definire la forma dei nemici.
8. Utilizzare curve interpolanti di Hermite per definire la forma del giocatore.
9. Memorizzare le geometrie del giocatore, dei nemici e dei proiettili su Vertex Buffer Objects (VBOs) per una gestione efficiente dei dati grafici.
10. Aggiungere funzionalità come punti.

### Istruzioni:

1. Inizializzazione del progetto:
  - o Crea un nuovo progetto C++.
  - o Assicurati di avere FreeGLUT, la libreria "glm" e la libreria freetype per la gestione del testo installati e configurati nel tuo ambiente di sviluppo.
2. Inizializzazione della finestra:
  - o Inizializza una finestra di gioco utilizzando FreeGLUT.
  - o Imposta il titolo, le dimensioni e la modalità della finestra.
3. Creazione del giocatore:
  - o Utilizza curve interpolanti di Hermite per definire la forma del giocatore.
  - o Memorizza la geometria del giocatore su un Vertex Buffer Object (VBO).
4. Utilizzo di curve interpolanti di Hermite per la forma dei nemici:
  - o Definisci curve interpolanti di Hermite che rappresentano la forma dei nemici
  - o Memorizza la geometria dei nemici su VBOs.
5. Gestione dei proiettili:
  - o Consentire al giocatore di sparare proiettili verso l'alto quando preme un tasto.
  - o Creare oggetti geometrici per rappresentare i proiettili e gestirne la posizione e il movimento mediante le trasformazioni di modellazione realizzate mediante le funzioni della libreria glm.
  - o Memorizzare la geometria dei proiettili su VBOs.
6. Collisioni:
  - o Implementare il rilevamento delle collisioni tra i proiettili del giocatore e i nemici.
  - o Gestisci la rimozione dei nemici colpiti e il conteggio dei punti.
7. Aggiunta di uno sfondo con un fragment shader:
  - o Crea un fragment shader che disegnerà uno sfondo animato.
  - o Carica il fragment shader e utilizzalo per disegnare lo sfondo sulla finestra.
8. Aggiunta di funzionalità extra:
  - o Aggiungi punti
  - o Aggiungi una schermata di game over quando il giocatore perde tutte le vite.
9. Testing e debugging:
  - o Testa il gioco per assicurarti che funzioni correttamente.

- Risolvi eventuali bug o problemi di collisione.
10. Ottimizzazione:
- Assicurati che il gioco sia ottimizzato per prestazioni fluide.
11. Documentazione:
- Documenta il tuo codice in modo chiaro e completo.

Esempio

