

# FONDAMENTI DI COMPUTER GRAPHICS LM

## GIOCO-ANIMAZIONE, CAMERA PATH & COLLISION DETECTION

---

Questa esercitazione puo' essere eseguita sia in ambiente Windows che Linux. L'esercitazione propone due scenari di gioco, sceglierne uno da personalizzare ed arricchire come proposto.

### 1. Terrain



Figure 1: Scenario 1: Il territorio

Scaricare i file necessari dalla pagina WEB del docente. L'archivio contiene un programma principale (`terrain.c`) per la navigazione virtuale di un territorio (vedi fig. 1); una directory `texture-map` contenente alcuni file immagine in formato `.bmp` per la generazione di heightfield e texture, e un file `Gauss.c` per la risoluzione di sistemi lineari con metodo di Gauss (se si sceglie di creare un percorso interpolante). Compilare ed eseguire il programma `terrain.c` fornito.

La navigazione e' interattiva tramite i tasti frecce e 'a'/'z' per accelerare/decelerare la velocita'. Il tasto 'h' disattiva/attiva la visualizzazione della pagina di help.

A partire dal template fornito si richiede di realizzare la navigazione del territorio di un oggetto mesh lungo un percorso definito dall'utente.

Cosa deve essere realizzato in dettaglio:

- (a) **Definizione del percorso**  
Inserimento in una finestra 2D di una curva spline o Bézier a tratti per punti. I punti inseriti potranno essere interpolati o gestiti come punti di controllo della curva.
- (b) **Definizione dell'oggetto**  
Caricamento di un oggetto mesh che si dovrà muovere lungo la curva. Per esempio l'aereo `x-wing.m` fornito nei file di progetto. **OPZIONALE** gestione collisioni con il territorio.
- (c) **Controllo della camera**  
Attivare al comando 'v' la navigazione su curva percorso dell'oggetto caricato in modalità 'lookahead' nella quale la camera segue l'oggetto a distanza offset da esso lungo il percorso. Le quote 'z' del percorso potranno essere definite costanti o calcolate come offset dalla mesh del territorio.
- (d) **Modifiche al territorio, ottimizzazione**  
Il territorio è creato da un heightfield (immagine a livelli di grigi) che può essere cambiata (si veda directory texture-map). Ottimizzare mediante quad-strip la resa del territorio. Attivare al comando 't' le texture opportune per territorio e superficie acqua. Creare il cielo: caricare una texture map sky tra quelle fornite e associarla ad una sfera contenente la scena.
- (e) **OPZIONALE**: Modificare il progetto aggiungendo strategie di gioco. Per esempio creare più percorsi per aggiungere oggetti/aerei in scena e dare la possibilità di portare l'osservatore da un aereo all'altro. Se due aerei si trovano al di sotto della distanza di sicurezza attivare un'azione relativa (es. distruzione/attacco/un aereo cambia rotta e segue l'altro). Attivare la fantasia e proporre una propria strategia di gioco!

Buon divertimento!

## 2. Tunnel

Sviluppare a partire dal file `tunnel.c` un gioco in cui un oggetto (a vostra scelta) deve percorrere un tunnel (vedi fig. 2) evitando collisioni con ostacoli (da voi definiti) o con le pareti del tunnel stesso.

Il gioco si sviluppa in due parti: nella prima parte il giocatore definisce il percorso interattivamente posizionando/spostando punti percorso (PP) che vengono poi interpolati tramite o una spline interpolante o una Bézier cubica a tratti, nella seconda parte l'oggetto segue il percorso pre-definito ed interpreta una pre-stabilita strategia di gioco.

Lo sviluppo deve seguire le seguenti specifiche:

- (a) **Definizione del percorso**  
Predisporre nel programma `tunnel.c` una serie di marker equispaziati (che chiameremo PP) lungo una poligonale che segue il centro del tunnel. Quindi permettere al giocatore lo spostamento (selezione/modifica) interattivo dei PP per modificare il percorso. Si devono prevedere due modalità attivabili tramite i tasti 's' (seleziona), 'm' (modifica):
  - Selezione PP  
In modalità selezione attraverso i tasti 'n', 'p' ("next", "previous") deve essere possibile cambiare la selezione corrente del PP (punto percorso).
  - Modifica PP  
In tale modalità deve essere possibile modificare la posizione del PP correntemente selezionato traslandolo sul piano perpendicolare alla direzione del tunnel tramite i tasti direzionali (freccia su, giù, sinistra, destra).



Figure 2: Scenario 2: Il tunnel

- (b) **Controllo della camera** Implementare le seguenti modalita' di controllo della camera:
- Camera mobile:  
Durante l'esecuzione del gioco, cioe' quando l'oggetto segue il percorso definito, la camera deve seguire la posizione dell'oggetto con visualizzazione in prima o terza persona. La direzione di vista puo' essere definita tramite il vettore tangente alla spline percorso, tramite look ahead oppure definendo il centro di interesse sugli ostacoli.
  - Camera statica:  
In modalita' di selezione e modifica nodo la camera deve essere posizionata in modo da visualizzare il nodo correntemente selezionato.
- (c) **Collision Detection**  
Gestire le collisioni dell'oggetto con il tunnel e con gli ostacoli approssimando la geometria dei modelli (oggetto, tunnel, ostacoli) tramite bounding boxes o bounding spheres.
- (d) **OPZIONALE:** Modificare il progetto aggiungendo strategie di gioco e varianti. Per esempio creare ostacoli dinamici (mobili, intelligenti, distruggibili tramite un meccanismo di shooting, etc.), permettere la modifica run-time del percorso, etc. Attivare la fantasia e proporre una propria strategia di gioco!