

Corso di Grafica Computazionale

EnvyMyCar – Strutture Dati

***Docente:
Massimiliano Corsini***

Laurea Specialistica in Ing. Informatica

Facoltà di Ingegneria

Università degli Studi di Siena



Classi

- Client
- Race
- Circuit
- Player
- Car
- Street
- Tunnel
- Holes
- Buildings
- Trees



- NvmcClient (game_client.h)
 - Contiene vari dati relativi al client (ip, modalità)
 - Gestisce i messaggi
 - Contiene i metodi per inviare gli screenshots agli altri giocatori.
- UpdatePlayers() è necessario per effettuare l'update della posizione dei giocatori.



Race

- La gara(!) → contiene il circuito attualmente in uso più altre variabili di stato (e.g. tempo da cui la gara è iniziata).
- Contiene anche la lista dei Players.
- **Warning** → Ai players si accede però dalla struttura dati relativa al client.
 - Esempio: `if (!cli.players.empty()) ..`



Player

- Contiene lo stato del giocatore.
- La macchina utilizzata dal giocatore (classe Car).
- Ip del giocatore.
- Lo screenshot corrente.
- Timer di servizio.



Car

- L'automobile è caratterizzata dal modello 3D (avatar).
- Dal suo bounding box (*bbox*).
- Dai vettori velocità ed accelerazione (*vel* e *acc*).
- Dalla sua posizione ed orientamento (*frame*).



Circuit

- Il circuito su cui si sta svolgendo la gara.
- Dati a cui siete sicuramente interessati:
 - Il bounding box del circuito (***bbox***).
 - La strada.
 - Un vettore (vector) contenente tutti gli edifici del circuito.
 - Un vettore (vector) contenente tutti gli alberi del circuito.
 - La posizione del sole..



Street

- E' un vettore di coppie di punti 3D. Ogni coppia definisce il lato destro ed il lato sinistro della strada.
- Quindi, può essere visualizzata effettuando il rendering di una serie di quads.
- Esempio:
 $street[i][0], street[i][1], street[i+1][1], street[i+1][0]$ definisce un quad della strada.
- La strada contiene anche informazioni relative alle buche (holes) ed alle gallerie (tunnel).



Tunnels

- Una galleria viene descritta nello stesso modo della strada con in aggiunta un valore di altezza.
- Street memorizza le gallerie del circuito in un contenitore di tipo vector chiamato ***tunnels***.



Holes

- Una buca è rappresentata memorizzando una sequenza di vertici che ne rappresentano il bordo (ordine anti-orario) più la sua profondità.
- Street memorizza tutte le buche del circuito in un contenitore di tipo vector chiamato **holes**.



Buildings

- Un edificio è rappresentato con una serie di vertici che ne definiscono la base più una variabile per indicarne l'altezza.
- I vertici sono memorizzati in un vettore di vertici (*vector<Vertex>*).
- Potete accedere a tutti gli edifici del circuito usando *BuildingsIterator*.



Trees

- Un albero viene rappresentato con una posizione più la sua altezza.
- Potete accedere a tutti gli alberi del circuito usando ***TreesIterator***.



Conclusioni

- I dati degli elementi del circuito sono organizzati in contenitori di tipo vector.
- E' quindi sufficiente scorrere i vari contenitori e disegnare i corrispondenti elementi.
- Ovviamente si deve disegnare anche la propria automobile e gli avversari.



Domande?