POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

OrbTail

Design e sviluppo di un videogioco multigiocatore con interoperabilità tra ecosistemi diversi

Relatore: Prof. Pier Luca LANZI

Tesi di laurea di: Raffaele Daniele Facendola

Videogiochi

- Coinvolgimento
- Diverse piattaforme
- Condivisione
  - Cooperazione \ Competizione
  - Multigiocatore locale o online

OrbTail

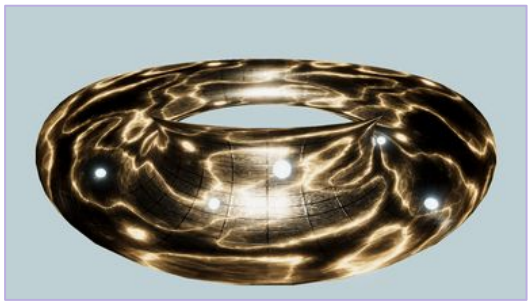
- Gioco di corse ad arena a tema fantascientifico\cyberpunk
  - Competitivo, fino a quattro partecipanti
- Multigiocatore locale ed online
- Multipiattaforma
- Cross-platform play

Veicoli

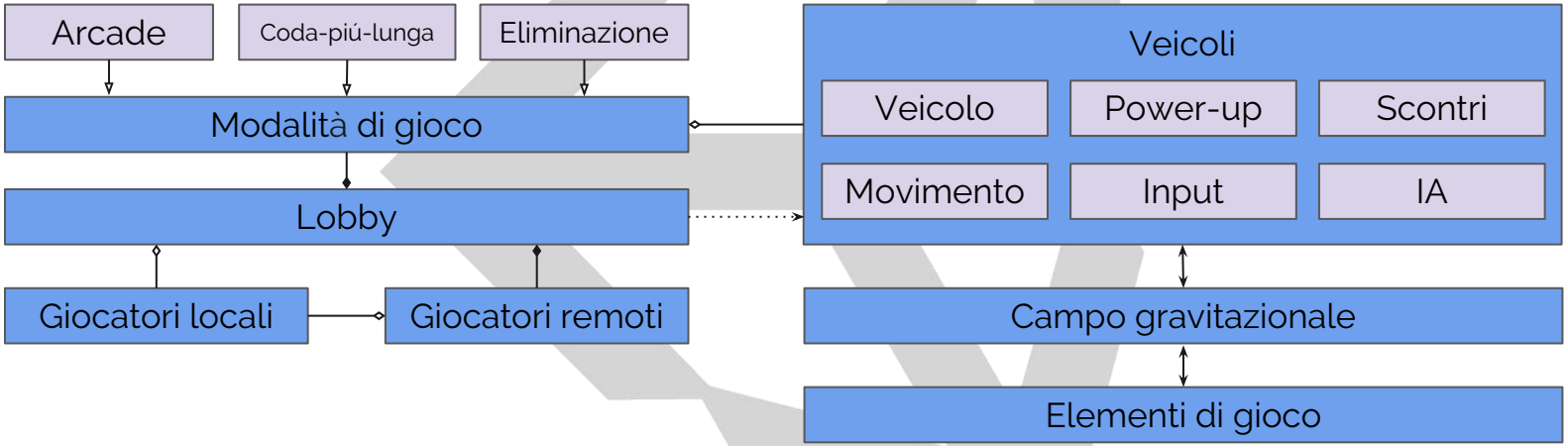
- Sei veicoli ad anti-gravitá
  - Rendering physically-based
- Stili di guida diversi
  - Velocità, accelerazione, manovrabilità

Arene

- Diverse topologie
  - Grado di sfida piú elevato
- Dimensioni limitate
  - Favoriscono gli scontri

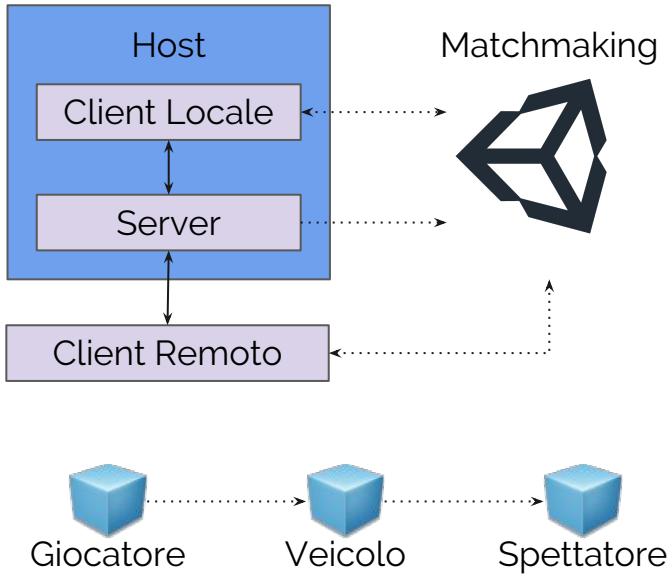


Architettura



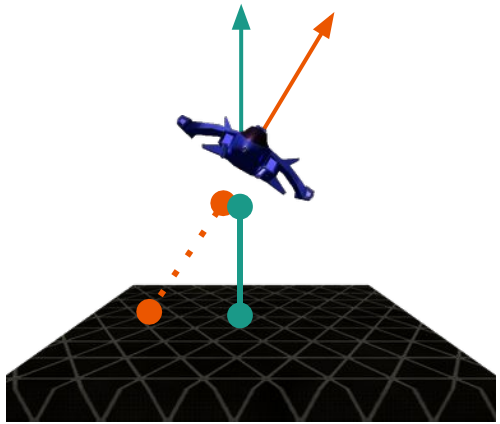
Architettura di rete

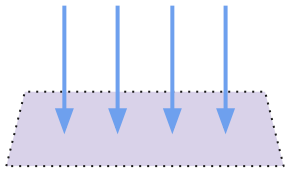
- Client-server
  - Host: autorità su IA, elementi e scontri
  - Client: autorità sui veicoli
- Sincronizzazione: RPC e replicazione
  - Dead-reckoning tramite interpolazione
- Gestione lobby tramite servizio di matchmaking di Unity

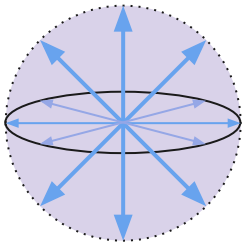


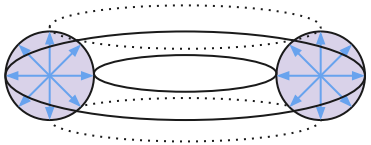
Gestione della gravità

- Posizione determinata tramite raycasting
- Fluttuazione tramite moto armonico smorzato
- Rappresentazione analitica del campo gravitazionale
  - Evita ambiguità durante il raycasting
  - Topologie complicate modellate per aggregazione





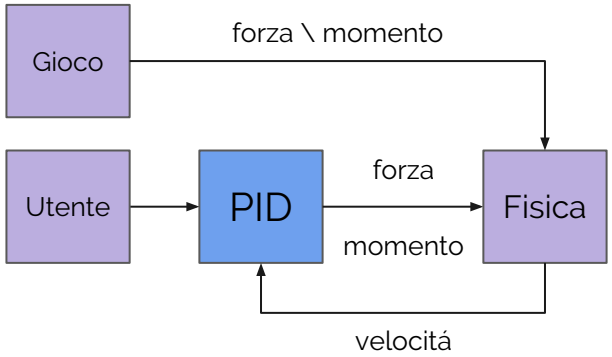


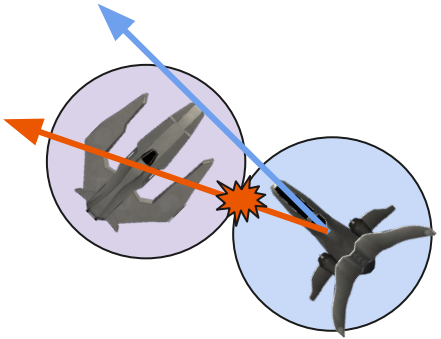


Sistema di controllo

- Controllore PID per acceleratore e sterzo
  - Riferimento determinato dall'input utente
  - Parametri influenzati dal veicolo

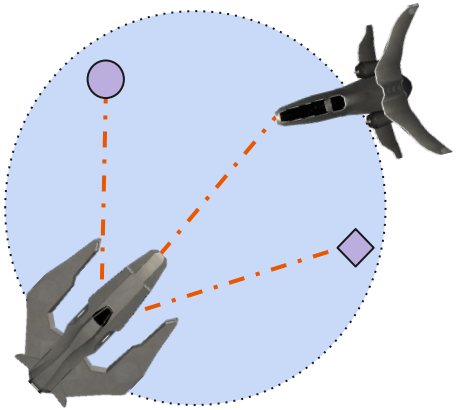
- Scontri gestiti dal motore fisico
  - Veicoli modellati come sfere
  - Danni proporzionali a velocitá e direzione



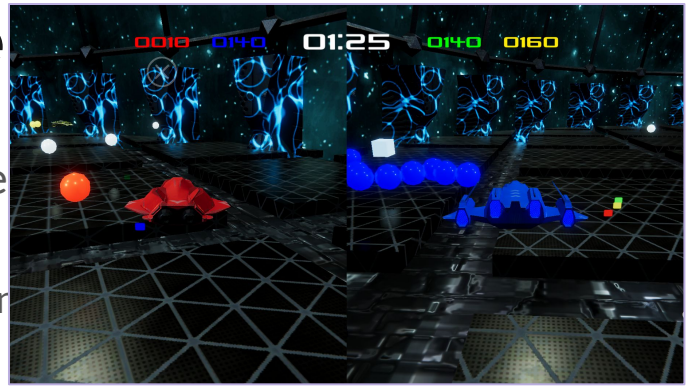


Intelligenza artificiale

- Genera gli input usati dal controllore del veicolo
- Campo visivo usato per individuare obiettivi
- Punti di controllo per aiutare la navigazione
- Strategie temporizzate per evitare l'accanimento



Multigiocatore locale



- Multigiocatore misto locale ed online
- Split-screen fino a quattro giocatori
  - Suddivisione del viewport in quadrar
- Separazione HUD 3D

Conclusioni

- Motori grafici indispensabili per lo sviluppo multipiattaforma
- Interoperabilità non adatta a tutti i tipi di design
- Architettura unica evita l’esplosione combinatoria delle implementazioni

Sviluppi futuri

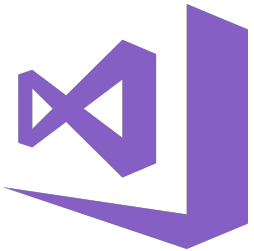
- Nuove arene, modalitá di gioco, potenziamenti, livree…
- Nuove strategie per l’intelligenza artificiale
- Distribuzione della simulazione sui client

DOMANDE?

GRAZIE!

Sviluppo

- Motore di gioco: Unity
  - Supporto multipiattaforma
  - Servizio di matchmaking
- Logiche di gioco in C#, shader in CG
- Architettura basata sul pattern entity-component



Piattaforme

Titoli analizzati

Modalitá di gioco

- Tre condizioni di vittoria
  - Punteggio più elevato
  - Coda più lunga
  - Ultimo giocatore in partita
- Potenziamenti ed armi
  - Rottura dell’equilibrio di gioco

Sviluppi futuri

- Nuove arene, modalitá di gioco, potenziamenti, livree…
- Intelligenza artificiale
  - Strategie adattabili alla modalitá di gioco corrente
  - Uso dei potenziamenti in maniera vantaggiosa
- Esperienza online
  - Algoritmi più sofisticati per gestire il dead-reckoning
  - Distribuire la simulazione di gioco