

# Progetto Sistemi Operativi - VideoGame

Tiziano Marino, matricola 1698254

Elisa Berti, matricola 1716412

Riguardo la prova del progetto di Sistemi Operativi abbiamo optato per la realizzazione di un videogame partendo da un progetto "base" fornitoci dai professori e implementando quest'ultimo sia sul lato client che sul lato server, in modo tale da offrire la possibilità all'utente di giocare anche in modalità multigiocatori.

*Compilare prima tramite il makefile.*

Il **lato Client** è stato implementato in modo tale da impostare fin da subito una connessione TCP e UDP tramite un primo pacchetto `IdPacket` con il Server, così che quest'ultimo possa effettuare l'assegnazione di un ID. Successivamente il client invia la sua texture, caricata inizialmente, e tramite una funzione `recv`, dopo aver ricevuto `surface` ed `elevation texture`, il gioco parte e vengono lanciati tre thread, i quali sono sincronizzati con la creazione del mondo del videogioco tramite dei semafori.

In particolare questi thread sono: `reciver`, `new player listener`, `update sender`.

*Per eseguire: `so_game_client <server IP address> <path to texture>`*

Per quanto riguarda il **lato Server**, esso è stato implementato in modo tale che vengano caricati inizialmente l'`elevation` e la `surface texture`, subito dopo vengono impostati i socket TCP e UDP e la `Player List` in cui vi sono memorizzati i dati sui giocatori.

Anche in questo caso vengono lanciati tre thread: `update sender thread func`, `update reciver thread func`, `delete players thread`.

Dopo aver effettuato la connessione con il client, si calcola un ID per ogni giocatore e si dedica un socket a quel client. Viene lanciato il thread `Player Handler` che si occupa dell'assegnazione dell'ID precedentemente calcolato tramite un `ID Packet`, dopo aver ricevuto la texture del giocatore aggiunge il veicolo al mondo e invia `elevation` e `surface texture` al client.

*Per eseguire: `so_game_server <path to elevation texture> <path to surface texture>`*