

PROGETTO DI SISTEMI OPERATIVI - REPORT

WHAT

'*so_game*' è un videogioco modalità single/multiplayer con architettura client-server.

Ogni giocatore che vi partecipa possiede un veicolo (la cui immagine è scelta dall'utente stesso) che pilota all'interno di percorso di gioco.

La procedura client-side che caratterizza il progetto richiede per ogni utente una connessione al server dedicato (OS SERVER) , tramite il quale vengono effettuate diverse operazioni come la distribuzione della mappa di gioco ai vari giocatori connessi e l'interazione con le loro informazioni personali (ad esempio la posizione dei veicoli durante il gioco).

HOW

L'applicazione si divide in client-side e server-side comunicanti tramite connessioni di tipo TCP (transfer control protocol) e di tipo UDP (user datagram protocol) tramite le quali vi è passaggio di informazioni utili. Nelle righe successive sono descritti i vari step compiuti da entrambe le parti.

server_side

1. carica le immagini scelte per il gioco (la mappa del mondo e l'immagine di default del veicolo).
2. inizializza una connessione UDP e vi crea un thread dedicato (udp_thread), il quale si occupa di gestire gli aggiornamenti del gioco, ovvero le posizioni che i veicoli dei client connessi assumono durante la partita. Ogni volta che riceve un nuovo aggiornamento lo notifica a tutti gli utenti che attualmente sono in gioco.
3. inizializza una connessione TCP e si mette in ascolto sulla porta 3000;
4. per ogni utente che si connette crea un thread dedicato (client_thread), il quale si occupa di gestire, sotto specifica richiesta, le risorse necessarie al client (mappa del gioco, id, immagine del veicolo etc.)

client_side

1. inizializza una connessione TCP e si connette al server dedicato.
2. carica l'eventuale immagine scelta dall'utente, chiede al server quella di default altrimenti.
3. richiede al server le varie risorse necessarie (mappa del gioco, id, etc.)
4. costruisce il mondo di gioco con le risorse ricevute.
5. Inizializza una connessione UDP e vi crea un thread dedicato, il quale si occupa di inviare al server gli spostamenti del veicolo durante il gioco e aggiorna il proprio mondo in base alle nuove informazioni ricevute dal server riguardo le posizioni assunte dagli altri veicoli.
6. Crea un thread il quale durante la partita verifica periodicamente che il server rimanga attivo. Nel caso in la connessione si interrompa all'utente è permesso giocare in single player, eliminando quindi dal proprio mondo i veicoli degli altri giocatori.

7. Chiude tutte le connessioni quando l'utente decide di disconnettersi dal gioco.

HOW TO RUN

Compilazione

1. Aprire il terminale (ctrl+alt+t)
2. Spostarsi nella cartella `so_game/game`
3. Compilare digitando il comando `'make'`

Avvio Server Side

1. Aprire il terminale (ctrl+alt+t)
2. Spostarsi nella cartella `so_game/game`
3. Avviare il server tramite il comando `'make test_server'`

Avvio Client Side

1. Aprire il terminale (ctrl+alt+t)
2. Spostarsi nella cartella `so_game/game`
3. Avviare il client tramite il comando `'make test_client'`
4. Ripetere la procedura dal punto 1 per ogni nuovo client

Cancellazione file eseguibili

1. Aprire il terminale
2. Spostarsi nella cartella `so_game/game`
3. Digitare il comando `'make clean'`

AUTORI

Nicolanti Valerio ~ 1693224

Terenzi Francesco ~ 1702217