

Elaborato S.O

Membri gruppo + matricola

June 2023

Indice

1 Obiettivo del progetto	1
2 Funzionamento Server	1
3 Funzionamento Client	2
4 Segnali	2
5 Scelte progettuali	3

1 Obiettivo del progetto

L'obiettivo del progetto è la realizzazione di Forza 4 utilizzando le System Call in linguaggio C. Struttura progetto

- Forza4
 - exec → eseguibili
 - inc → librerie ".h"
 - obj → file ogg ".o"
 - src → codice ".c"
 - Makefile → script per la compilazione

2 Funzionamento Server

Per il funzionamento dell'applicazione si devono aprire 3 diversi terminali su cui nel primo si lancia il Server con successivi paramenti, ossia le righe e colonne della griglia del gioco e i 2 gettoni ovvero **X O**, in questo modo:

```
$ .F4Server riga colonna gettone1 gettone2
```

il Server possiamo dire che è colui che arbitra la partita infatti segnala quando i 2 giocatori si sono connessi, quando si abbandona, quando gli input sono errati, decreta chi vince, chi perde o il pareggio, quando si avvia una partita casuale, controlla la matrice.

3 Funzionamento Client

Mente negli altri 2 terminali lanciamo i 2 Client ossia i giocatori, con successivi paramenti ossia nome giocatore (obbligatorio) e con via opzionale l'asterisco (*) se si voglia fare una partita contro un BOT (Server), in questo modo:

- → 2 giocatori:

```
$ .F4Client nomeGiocatore
```

- → contro il server:

```
$ .F4Client nomeGiocatore *
```

successivamente parte il gioco vero e proprio, ogni giocatore deve inserire il numero della colonna dove il gettone verrà inserito. il gioco finirà solo quando un giocatore avrà inserito in orizzontale/verticale/diagonale 4 suoi gettoni di fila.

- .F4Client numeroColonna

4 Segnali

Con i segnali abbiamo gestito la parte del timer, ossia il giocatore ha un tot di secondi per fare la giocata, se non dovesse eseguire la giocata finisce il gioco e decreta il vincitore.

Successivamente un altro segnale è dovuto all'abbandono ossia al Ctrl-C, quando un qualsiasi giocatore preme 2 volte consecutivamente Ctrl-C, si perde per abbandono e si ferma automaticamente la partita proclamando il vincitore.

Segnali Client:

- quit → il client ha terminato spontaneamente e quindi invia questo segnale al server per informarlo.
- term → segnale ricevuto dal server per indicare che il client deve terminare.
- timeOut → il client ha perso troppo tempo per fare la mossa quindi invia questo segnale al server per informarlo.

- gioco casuale → quando il client vuole giocare contro la CPU informa il server inviandogli questo segnale.

Segnali Server

- first Ctrl-C → intenzione del server di terminare.
- second ctrl-c → quando il server termina.
- term client → un client si è scollegato.
- casual game → segnale ricevuto dal client per indicare che vuole iniziare una partita casuale.
- time out → segnale ricevuto dal client che ha impiegato troppo tempo a fare la mossa.

5 Scelte progettuali

- Quando il server termina, invia due segnali ai giocatori per terminare pure loro.
- Quando uno dei due giocatori esce dalla partita in modo inaspettato, informa il server prima di uscire e il server termina la partita con la vincita a tavolino dell'altro giocatore.
- Quando un giocatore impiega troppo tempo a fare una mossa la partita viene terminata con la vincita a tavolino dell'altro giocatore.
- Se entra in partita un terzo client, questo viene fatto uscire.
- Se un primo giocatore vuole iniziare una partita normale, ma il secondo giocatore vuole fare una partita casuale e quindi attiva il suo client con l'opzione '*, allora questo secondo viene fatto terminare.