

Manuale del gioco TRIS

Alessio Recek
Matricola: VR475905

Kristian Rakipaj
Matricola: VR486817

Descrizione

Questo progetto implementa un gioco del Tris (Tic-Tac-Toe) multi-processo in C, con supporto per la modalità Bot. Il gioco utilizza memoria condivisa e semafori per gestire la sincronizzazione tra i processi del server e dei client. Il progetto comprende due componenti principali:

- **TriServer**: Il server che gestisce il gioco e coordina le azioni dei giocatori.
- **TriClient**: Il client che permette a un giocatore umano o a un bot di partecipare al gioco.

Requisiti

- Un ambiente Unix-like (Linux, macOS, ecc.)

Compilazione

Per compilare il progetto, eseguire il seguente comando nella directory del progetto:

```
make
```

Questo genererà due eseguibili:

- **TriServer**: Il server del gioco.
- **TriClient**: Il client del gioco.

1. Avvio del server

Per avviare il server, eseguire il comando seguente:

```
./TriServer <timeout> <symbol1> <symbol2>
```

timeout: Il tempo massimo (in secondi) che un giocatore ha per fare una mossa.

symbol1: Il simbolo utilizzato dal primo giocatore (es. X).

symbol2: Il simbolo utilizzato dal secondo giocatore (es. O).

Esempio:

```
./TriServer 30 X O
```

2. Connessione di un client

Per collegare un client (giocatore umano), eseguire il comando:

```
./TriClient <playerName>
```

Esempio:

```
./TriClient Giovanna
```

3. Modalità Bot

È possibile collegare un client in modalità Bot utilizzando il comando:

```
./TriClient <playerName> BOTMODE
```

Esempio:

```
./TriClient Bot BOTMODE
```

4. Modalità Bot con Fork ed Exec

```
./TriClient <playerName> "*"
```

Esempio:

```
./TriClient Giovanna "*"

Cosa succede? Il processo client corrente si chiuderà, ma avrà già in-
```

formato il server, tramite una variabile condivisa, che dovrà essere creato un processo figlio. Questo processo figlio utilizzerà la funzione `execvp` per sostituire il proprio codice con una nuova istanza del client in modalità BOTMODE. Quindi per giocare, basta ora aprire un'altra finestra del terminale e collegarsi utilizzando, ad esempio:

```
./TriClient Giovanna
```

Funzionamento del Gioco

1. **Connessione:** Il primo client che si connette sarà il giocatore 1 e utilizzerà **symbol1**. Il secondo client sarà il giocatore 2 e utilizzerà **symbol2**.
2. **Turni:** I giocatori fanno le loro mosse alternandosi. Il server gestisce i turni utilizzando semafori per sincronizzare le operazioni.
3. **Mosse:** Un giocatore può fare una mossa inserendo un numero da 1 a 9, che rappresenta una posizione sulla griglia 3x3 del Tris.
4. **Vittoria/Pareggio:** Il gioco termina quando un giocatore completa una linea (orizzontale, verticale o diagonale) oppure se la griglia si riempie senza che nessuno vinca, risultando in un pareggio.
5. **Timeout:** Se un giocatore non fa una mossa entro il tempo specificato dal timeout, l'altro giocatore vince automaticamente.
6. **Interruzione del Gioco:** Il server o i client possono essere interrotti in qualsiasi momento utilizzando **Ctrl+C**. Il server gestirà la terminazione notificando i client e liberando le risorse condivise. Una singola pressione di **Ctrl+C** sul server non lo terminerà immediatamente, ma necessiterà di una doppia pressione. Nei client invece ne basterà solo una.

Script bash

Per velocizzare il processo di testing è presente un script bash che apre in automatico i terminali (GNOME terminal). Verrà stampato un menù dove è possibile selezionare la funzione da voler testare. Nelle opzioni dove è richiesto un input umano comunque dovrete continuare voi. Prima di testarlo controllate se il file è eseguibile con:

```
ls -l test.sh
```

se non lo fosse basterà digitare:

```
chmod +x test.sh
```

dopodichè:

```
./test.sh
```

Pulizia

Per rimuovere i file eseguibili generati, eseguire:

```
make clean
```