

# Manuale d'uso tris

Francesco Tiefenthaler  
Denis Gallo  
Rakib Haque

## 1 Il gioco

Il progetto implementa il gioco del tris, dove due giocatori si sfidano nel completare per primo una riga, colonna o diagonale con il proprio simbolo, in una matrice 3x3. Il tutto costituito tramite un modello client server dove i due client giocano e il server arbitra la partita.

## 2 Avvio del gioco

Dopo la compilazione tramite il comando "make" verranno generati i due eseguibili triCLient e triServer. Avviare il server come segue:

```
./triServer time X O
```

Dove **time** indica il tempo massimo in secondi per inserire la mossa, se posto a 0 non esiste limite. Se il giocatore supera il tempo prefissato regala il turno all'avversario.

**X O** indicano i simboli che utilizzeranno i giocatori, possono essere inseriti a piacere.

Una volta avviato il server occorre avviare i due client nel seguente modo:

```
./triClient nomeUtente
```

**nomeUtente** rappresenta il nome del giocatore che avvia il client. Verrà attribuito il simbolo corrispondente inserito all'avvio del server.

Nel caso si volesse giocare contro un avversario casuale utilizzare:

```
./triClient nomeUtente \*
```

## 3 Regole aggiuntive

Le regole del tris rimangono quelle classiche, tuttavia questa implementazione del gioco prevede un sistema punitivo per l'utente sbadato. Ogni volta che l'utente inserisce delle coordinate non valide nella matrice di gioco (Cella occupata o indici errati) cede automaticamente il turno all'avversario.

## 4 Implementazione

### 4.1 triServer

Il server ha il compito principale di creare, distruggere la memoria condivisa e i semafori, arbitrare la partita e determinarne il vincitore.

All'avvio, il server dopo aver generato la memoria, semafori e coda di messaggi attende i seguenti segnali:

- **SIGUSR1** che segnala l'arrivo dei client

- **SIGUSR2** che segnala la modalità gioco automatico. in tal caso il server farà una fork e exec del client in modo da giocare contro un bot

A partita iniziata, il server manda e riceve il segnale **SIGUSR1** per alternare il turno tra i vari giocatori. Al termine invece, il server inserisce in memoria condivisa un flag per determinare il vincitore o il pareggio.

Viene utilizzato tra client e server anche il segnale **SIGALARM** per segnalare la fine anticipata della partita, sia allo scadere del timer sia alla terminazione di uno dei due client.

## 4.2 triClient

Il client attende i segnali:

- **SIGUSR1**: è il proprio turno, viene chiesta all'utente la mossa e viene inserita in memoria (Se valida, altrimenti si perde il turno)
- **SIGUSR2** Partita terminata per motivi esterni

Nel caso della doppia pressione di ctrl + c, i client vengono uccisi dal server che procede a liberare la memoria e arrestarsi. Nel caso del gioco automatico il client avversario non farà altro che generare coordinate casuali fino alla terminazione della partita

## 4.3 Memoria condivisa

Presenta la matrice di gioco, i pid dei processi coinvolti, i simboli che contrassegnano la mossa, il timer per il tempo massimo della giocata e delle flag sullo stato della partita. Sia server che i due client possono leggerci e scriverci sopra. La memoria condivisa è una risorsa critica il cui accesso è arbitrato tramite semafori P e V (Wait and Signal)