

# Relazione Progetto SO 2019/20

Alessandro Ampala 895430

31-01-2020

## 1 Introduzione progetto

Segue la relazione del progetto di sistemi operativi dell'A.A 19/20 riguardante la simulazione di un gioco su una scacchiera su cui le pedine di più giocatori competono per l'ottenimento di bandierine in più round. La stampa a schermo della scacchiera nei vari momenti del gioco rappresenta lo stato della scacchiera in quel momento.

- Pedine dello stesso giocatore vengono rappresentate con lo stesso carattere e i possibili caratteri di rappresentazione vanno da 'A' a '~'.
- Le bandierine vengono rappresentate con il carattere '>'.
- Le celle vuote della scacchiera vengono rappresentate con il carattere '-'.

## 2 Istruzioni per l'uso

Per compilare il progetto, recarsi nella cartella contenente il progetto ed usare make. In seguito potrebbe essere necessario utilizzare i comandi

```
chmod u+x master
chmod u+x player
chmod u+x pedina
```

per dare i permessi d'esecuzione ai file appena creati. Prima di avviare il progetto è necessario impostare le variabili d'ambiente (copia e incolla dal file conf.txt). Per avviare il programma utilizzare il comando make run.

## 3 Gestione variabili di configurazione di gioco

Le variabili di configurazione di gioco sono lette a tempo d'esecuzione da variabili d'ambiente preventivamente impostate. Queste vengono poi caricate in apposite strutture di tipo *private\_envar* e *public\_envar*.

## 4 Inizio gioco e creazione processi

All'avvio del master vengono creati tutti gli oggetti necessari a gestire il gioco stesso come vari oggetti IPCs tra cui la scacchiera. La creazione dei processi player e pedina avviene tramite fork seguita da chiamata alla funzione execve. Il passaggio delle informazioni iniziali avviene tramite argomenti alla chiamata stessa e dipende da quale eseguibile si sta cercando di avviare. Il processo master avvia i player ed i processi player avviano le loro pedine.

## 5 Disposizione pedine

Prima dell'inizio dei round i giocatori piazzano a turno una pedina alla volta per un totale di  $SO\_NUM\_P$  pedine. Questo meccanismo viene implementato tramite  $SO\_NUM\_G$  code di messaggi diverse, una per ogni giocatore:

- Il master manda un messaggio con  $mtype = metti\_pedina$  sulla message queue del giocatore a cui tocca piazzare una pedina
- - Il giocatore riceve il messaggio, piazza la pedina e spedisce sulla stessa message queue un messaggio con  $mtype = pedina\_piazzata$

In questo modo è garantito che i player piazzino a turno una pedina alla volta. L'algoritmo di disposizione pedine è implementato in modo tale che un giocatore piazza una pedina ogni  $freq/6$  celle con  $freq = (\text{numero celle totali})/SO\_NUM\_P$ . Se la cella prescelta non è disponibile si passa a quella accanto.

## 6 Gestione round

Viene descritta la gestione del round da parte del master:

- - A inizio round il processo master piazza le bandierine in modo casuale tra le celle libere. Il punteggio delle bandierine è pari a  $SO\_ROUND\_SCORE$  diviso per il numero di bandierine. Se la divisione ha resto diverso da zero, i punti rimanenti vengono distribuiti tra le bandierine.
- Viene stampato lo stato (rappresentazione della scacchiera, mosse e punti di ogni giocatore)
- Viene poi mandato un messaggio ad ogni giocatore con  $mtype = fornisci\_indicazioni$
- Si attende che tutti i giocatori mandino un messaggio con  $mtype = ready$  per far sapere che hanno fornito le indicazioni iniziali alle loro pedine e che sono pronti ad iniziare il round
- Il round ha inizio: il timer viene avviato
- - Si rilasciano  $SO\_NUM\_P * SO\_NUM\_G$  risorse su un semaforo su cui le pedine aspettano in attesa di iniziare il round
- Si attende su un semaforo settato al numero di bandierine che tutti le bandierine vengano prese (ogni volta che una pedina conquista una bandierina decrementa il semaforo)
- Il round termina e il timer viene resettato
- Si ricevono messaggi dalle pedine per sapere il player a cui assegnare i punti e si assegnano i punti ai relativi player

Se un round dura più di *SO\_MAX\_TIME* secondi, il timer scade (implementato con alarm), viene ricevuto un segnale di tipo SIGALRM e viene chiamato il relativo handler che termina il gioco, informa tutti i player che il gioco è terminato, stampa le metriche di fine gioco, rimuove gli oggetti IPCs e termina dopo aver fatto la wait su tutti i giocatori.

## 7 Indicazioni da player a pedina

La comunicazione tra player e pedina è implementata tramite pipe. Per ciascun round, il player assegna ad ogni bandierina la pedina disponibile più vicina: ne calcola il percorso e la lunghezza dello stesso. Quindi scrive sulla pipe la lunghezza del percorso (quanti byte la pedina dovrà leggere da pipe) ed il percorso stesso. Viene inoltre tenuta traccia di quali pedine hanno ricevuto indicazioni e quali sono ancora disponibili tramite una maschera. In seguito si aspetta che tutte le pedine informino il giocatore che hanno ricevuto informazioni prima di comunicare al master che il player è pronto all'inizio del round.

## 8 Pathfinding e movimento pedina

Il pathfinding è stato implementato con l'algoritmo BFS o sue varianti. Le pedine ricevono un primo percorso ad inizio round dal loro giocatore. Quando invece non hanno un percorso da seguire, oppure la bandierina che stavano cercando di raggiungere è già stata catturata, calcolano un percorso tramite la funzione *find\_nearest\_flag* che cerca la bandierina più vicina a quella pedina e, se non è ancora stata assegnata ad un'altra pedina dello stesso giocatore, ritorna il percorso e assegna quella bandierina a quella pedina tramite una maschera per ogni giocatore che tiene traccia di quali bandierine sono già state assegnate ad una pedina di quel giocatore: questo per evitare di sprecare mosse in modo tale che al massimo due pedine vadano verso la stessa bandierina. Se la pedina trova un ostacolo sul suo percorso, aspetta un tot di tempo che quell'ostacolo si sposti. Se dopo lo scadere del tempo non si è spostata, allora ricalcola un nuovo percorso dalla posizione attuale alla posizione della bandierina da raggiungere.

## 9 Cattura di una bandierina

Quando una pedina riesce a raggiungere una bandierina, comunica al master la posizione della bandierina e decrementa il semaforo legato alla cattura delle bandierine. A round terminato il master legge tutti i messaggi inviati dalle pedine e distribuisce i punteggi delle bandierine ai player che le hanno conquistate.