

# Corso di Programmazione 3

## Progetto Esame

Docenti: Proff. Angelo Ciaramella, Raffaele Montella

A.A. 2018/2019

### Studente

- **Cognome:** Cozzolino
- **Nome:** Ciro
- **Matricola:** 124/1804

### Traccia - Gioco Forza 4

Si vuole sviluppare un programma per la simulazione del gioco *Forza 4*. L'obiettivo del gioco è mettere in fila (orizzontale, verticale o diagonale) un certo numero (quattro nel gioco base) di proprie pedine.

Inizialmente il programma deve prevedere l'identificazione di un giocatore tramite il suo *nome* e *cognome*. Il giocatore può scegliere una delle tre modalità di gioco: *difesa*, *attacco* o *neutro*. Il secondo giocatore è il "computer".

La scacchiera rettangolare (si dimensioni  $6 \times 7$ ) è posta in verticale e le pedine vengono fatte cadere lungo una griglia verticale, in modo tale che una pedina inserita in una certa colonna va sempre a occupare la posizione libera situata più in basso nella colonna stessa.

In base alle modalità, il computer ad ogni turno:

- 1 - controlla il numero massimo di pedine in fila del giocatore ed inserisce una sua pedina per fermare la sequenza (modalità *difesa*)
- 2 - inserisce una pedina secondo la mossa migliore ottenuta esplorando l'*albero min-max* con *alpha-beta pruning* (vedi sotto) (modalità *attacco*)
- 3 - sceglie la strategia [1] o [2] in modo casuale (modalità *neutro*)
- 4 - sceglie una colonna in modo casuale tra quelle che hanno un numero di pedine minore (per tutte le modalità)

Il gioco termina automaticamente quando uno dei giocatori ha messo in fila quattro delle sue pedine.

Scrivere un programma per la gestione del gioco. Il programma deve visualizzare, ad ogni fine partita, l'elenco totale dei giocatori ordinato in base al punteggio effettuato (numero di vittorie). Deve prevedere, inoltre, la possibilità di sospendere e ripristinare la partita successivamente.

## Albero min-max

Per i dettagli sull'albero *min – max* (o *minimax*):

<https://it.wikipedia.org/wiki/Minimax>

## Note di sviluppo

La prova d'esame richiede la progettazione e lo sviluppo della traccia proposta.

Lo studente può scegliere di sviluppare il progetto nelle due modalità: **Applicazione Web** o **programma standalone con supporto grafico**.

Il progetto deve essere sviluppato secondo le seguenti linee:

- usare almeno **due** pattern (almeno **uno** per chi sceglie la modalità Web Application) tra i **design pattern** noti;
- attenersi ai principi della programmazione **SOLID**;
- usare il linguaggio **Java**;
- inserire sufficienti **commenti** (anche per Javadoc) e **annotazioni**;
- gestione delle **eccezioni**;
- usare i **file** o **database**.

Lo studente deve presentare una relazione sintetica (per chi usa *latex* è possibile scaricare un template dalla piattaforma e-learning). La relazione deve contenere:

- una breve descrizione dei **requisiti** del progetto;
- il **diagramma UML** delle classi;
- altri diagrammi se opportuni;
- parti rilevanti del **codice** sviluppato.

## Consegna progetto

La relazione e il codice del progetto devono essere messi a disposizione secondo le modalità ritenute più opportune (Dropbox, Google Drive, ...) entro la **data di scadenza della prenotazione on-line** dell'esame e successivamente caricati sulla piattaforma GitHub.

## Modalità di esame

La prima parte della prova di esame verterà sulla discussione del progetto. Lo studente deve preparare una **presentazione sintetica** (slide) per descrivere il progetto svolto. La seconda parte della prova verterà sulla discussione degli argomenti affrontati a lezione.