

Dipartimento di Ingegneria e Scienze dell’Informazione e Matematica

CORSO DI LAUREA IN INFORMATICA

Insegnamento Laboratorio di programmazione ad oggetti

Scacchi AFP

|  |  |  |  |
| --- | --- | --- | --- |
| **Membri del team** | | | |
| **Cognome e nome** | **Matricola** | | **Indirizzo e-mail** |
| Andre Larivera | 258892 | | andre.larivera@student.univaq.it |
| Falleroni Francesco | 261164 | | francesco.falleroni@student.univaq.it |
| Paolo Celani | 272163 | | paolo.celani@student.univaq.it |
|  | |  | | |
|  | | | | |
|  | |  | | |

A.A. 2022/2023

Sommario

[Capitolo 1 – Caso di studio 2](#_Toc38393257)

[1.1 – Descrizione dell’applicazione 2](#_Toc38393258)

[1.2 - Funzionalità 3](#_Toc38393259)

[1.3 Domain Model 4](#_Toc38393260)

# Capitolo 1 – Caso di studio

## 1.1 – Descrizione dell’applicazione

L’applicazione che abbiamo sviluppato si tratta di una rappresentazione del gioco degli scacchi, sia tra giocatori umani che artificiali (CPU). Il gioco creato (“Scacchi AFP”) permette di vivere una esperienza (per quanto facilitata dalla mancanza di operazioni come la promozione del pedone o gli arrocchi) del gioco effettivo in vita reale. Gli utenti che possono giocare sono due tipologie:

* Giocatore 🡺 rappresenta l’utente umano
* CPU 🡺 il computer che è in grado di giocare “come” un essere umano

in entrambe le tipologie l’utente è in grado di effettuare una Mossa (lo spostamento di un pezzo qualsiasi sulla scacchiera) e un Attacco/Presa in cui può mangiare un pezzo del giocatore avversario. Le opzioni di scelta delle partite all’interno del nostro gioco sono “Giocatore vs Giocatore”, “Giocatore vs CPU” e “CPU vs CPU”.

## 1.2 – Funzionalità

RIMUOVERE DESCRIZIONE TESTO SOTTOSTANTE

Fornire una lista completa ed esaustiva dei requisiti funzionali della vostra applicazione organizzandoli per tipologie di persone che sono state identificate nel capitolo precedente.

1. Giocatore:
   1. deve poter scegliere se effettuare una partite con un altro utente oppure contro una CPU
   2. scegliere un colore (Bianco/Nero)
   3. sapere quando si trova sotto scacco/scacco matto/in stallo
2. CPU (può fare qualsiasi cosa che può fare il giocatore tranne la scelta del colore dato che avrà il colore opposto del giocatore nel caso di una partita tra Giocatore - CPU)
3. l’utente deve essere in grado di poter interrompere la partita in qualsiasi momento tramite l’opzione SALVA PARTITA, che salverà lo stato degli attuali pezzi sulla scacchiera, e successivamente essere in grado di riprendere a giocare
4. poter visualizzare le partite salvate in base al numero di pezzi sulla scacchiera, caselle libere oppure numero di mosse totali fatte nella partita

## 1.3 – Domain Model

RIMUOVERE DESCRIZIONE TESTO SOTTOSTANTE

Mostrare un class diagram che rappresenta il domain model della applicazione contenente entità e relazioni tra esse. Le entità e le relazioni derivano dalle funzionalità descritte nella sezione precedente.