

Corso Di Laurea In Informatica

Anno Accademico 2024/2025

Progettazione e sviluppo di un sistema concorrente e distribuito (Client-Server) per la gestione di partite di Tris

Antonio Esposito

Matricola N86005375

antonio.esposito167@studenti.unina.it

Raffaele Giustiniani

Matricola N86003051

ra.giustiniani@studenti.unina.it

Indice

1. [Introduzione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#Capitolo1) 4
2. Progettazione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
   1. Strutture dati . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
   2. Funzionalità principali . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
      1. Funzionalità Server-side . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
      2. Funzionalità Client-side . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
3. Implementazione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
   1. Funzioni . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
      1. Funzioni Server-side . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
      2. Funzioni Client-side . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
   2. Gestione dei thread . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
   3. Gestione degli errori . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
4. Docker . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
5. Note d'uso . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x

Capitolo 1

Introduzione

L’elaborato si propone di descrivere le scelte di progettazione ed implementazione di un sistema Client-Server per la gestione partite di tris in modalità concorrente.

Lo sviluppo in ogni sua fase è stato supportato dall’utilizzo della piattaforma GitHub per la condivisione dei file ed organizzazione del flusso di lavoro.

Come scelte preliminari si è deciso di utilizzare il linguaggio C per l’implementazione di entrambi i lati Client e Server, l’ambiente d’esecuzione è definito tramite Docker Compose e l’avvio è automatizzato da uno dei due script file forniti, a seconda del sistema operativo dell’host (Windows o Unix).

Capitolo 2

Progettazione

Come primo passo sono stati individuati i macro-elementi principali del sistema da definire, i giocatori e le partite. Analizzandone i possibili stati e fasi del loro “corso di vita” è stato possibile produrre uno scheletro delle principali funzionalità e strutture dati utili, suddividendone le responsabilità tra Client e Server.

2.1 Strutture dati

2.2 Funzionalità principali

2.2.1 Funzionalità Server-side

2.2.2 Funzionalità Client-side

Capitolo 3

Implementazione

In questo capitolo si discutono alcune specifiche sull’implementazione del sistema progettato, quali l’implementazione di funzioni, la gestione dei thread tramite segnali e la gestione di errori.

3.1 Funzioni

3.1.1 Funzioni Server-side

3.1.2 Funzioni Client-side

3.2 Gestione dei thread

3.3 Gestione degli errori

Capitolo 4

Docker

Capitolo 5

Note d’uso