

Perudo online

Presentazione progetto di Sistemi Distribuiti

Enrico Lumini
`enrico.lumini@studio.unibo.it`

Alma Mater Studiorum - Università di Bologna

December 2023

Table of Contents

- 1 Goal & Requirements
- 2 System Architecture & Technical details
- 3 Disconnection management
- 4 Usage examples

Table of Contents

- 1 Goal & Requirements
- 2 System Architecture & Technical details
- 3 Disconnection management
- 4 Usage examples

Main goal

L'obiettivo del progetto è quello di realizzare una versione distribuita del gioco da tavolo "*Perudo*" che permetta la presenza di molteplici partite in contemporanea.

Summary game rules

Il gioco è composto da 30 dadi di 6 colori diversi

- Ogni giocatore prende i 5 dadi dello stesso colore.
- Ogni giocatore agita i propri dadi e guarda in segreto i propri valori.
- Il primo giocatore dichiara un valore e una quantità di dadi.

Examples

“Al tavolo, ci sono almeno 5 dadi di valore 2”

- Il giocatore alla sua sinistra può **dubitare** o **rilanciare**, nel caso in cui ritenga rispettivamente falsa o vera la dichiarazione del giocatore precedente.
- A seguito di un dubbio i dadi vengono svelati e il giocatore che ha sbagliato perde un dado.
- Vince la partita l'ultimo giocatore che rimane in possesso di almeno un dado.

Requirements

Functional requirements (some)

- Possibilità di creare una lobby oppure unirsi ad una già esistente.
- Possibilità per gli utenti che si sono uniti ad una lobby di uscire prima dell'inizio della partita.
- Terminazione della partita a seguito del crash o dell'abbandono di un giocatore a partita avviata.
- Chiusura dell'applicazione, previa notifica, a seguito del crash del server.

Non-functional requirements (some)

- L'applicazione deve essere intuitiva e semplice da utilizzare.
- L'architettura deve permettere di gestire più partite in contemporanea.
- Casi di disconnessione improvvisa degli utenti o crash del server devono essere gestiti nella maniera più consona.

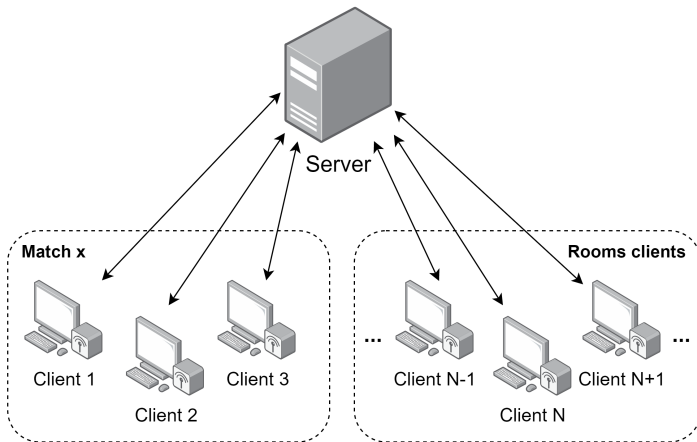
Table of Contents

- 1 Goal & Requirements
- 2 System Architecture & Technical details**
- 3 Disconnection management
- 4 Usage examples

System Architecture

Application structure

L'architettura scelta per questo progetto è stata di tipo **client-server**



Technologies

- Utilizzo di **stream socket** per la comunicazione tra *client* e *server*.
- *Client* e *server* scritti in **Java**.
- Utilizzo di `java.net.Socket` lato *client*.
- Utilizzo di `java.net.ServerSocket` lato *server*.
- Utilizzo del framework **libGDX**^a per la realizzazione dell'interfaccia grafica.

^alibGDX documentation

Table of Contents

- 1 Goal & Requirements
- 2 System Architecture & Technical details
- 3 Disconnection management**
- 4 Usage examples

Disconnection management

Disconnection circumstance and related management

- **Utente normale pre partita** → L'utente viene riportato alla schermata iniziale.
- **Creatore della lobby pre partita** → Tutti gli utenti all'interno della lobby vengono riportati alla schermata iniziale e la lobby viene eliminata.
- **Utente qualsiasi durante la partita** → La partita viene terminata e tutti gli utenti vengono riportati alla schermata iniziale. La lobby viene eliminata.
- **Crash del server** → Tutti gli utenti connessi vengono notificati e l'applicazione viene chiusa.

Table of Contents

- 1 Goal & Requirements
- 2 System Architecture & Technical details
- 3 Disconnection management
- 4 Usage examples**

Lobby creation

User must

L'utente deve specificare il **nome** e la **dimensione** della lobby oltre allo **username** che lo identificherà durante la partita.



Enter the game!

Lobby join

User must

L'utente deve specificare solo il suo **username**. Se quest'ultimo dovesse essere già utilizzato l'utente verrà notificato.

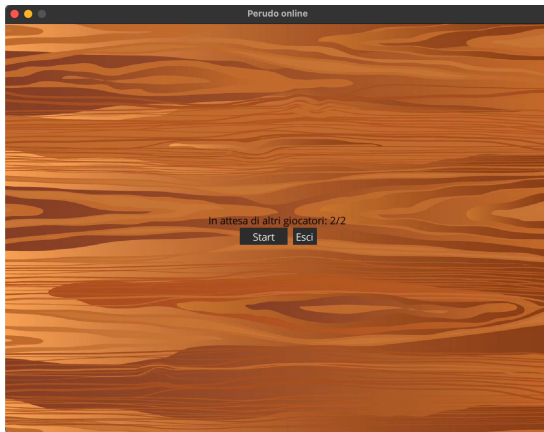


Enter the game!

During match I

User must

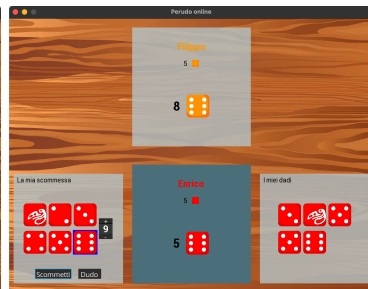
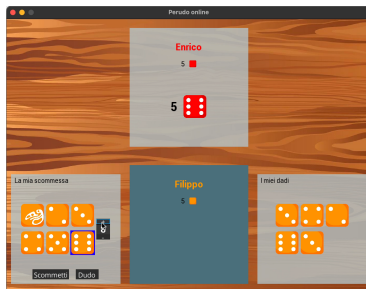
Il primo giocatore di turno può solo **scommettere**.



During match II

User must

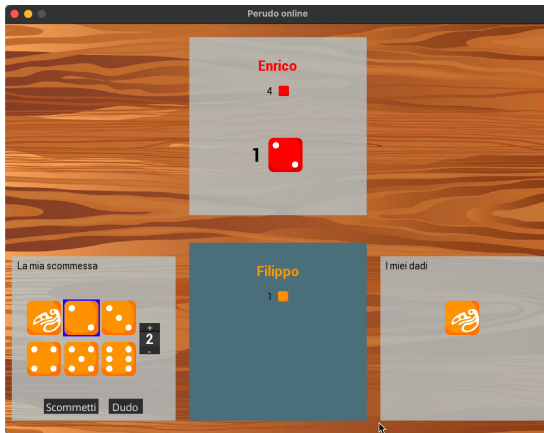
A seguito di una prima scommessa è possibile **rilanciare** oppure **dubitare**. A seguito di un dubito il giocatore che ha sbagliato perde un dado e un nuovo turno ha inizio.



Match end

What happens

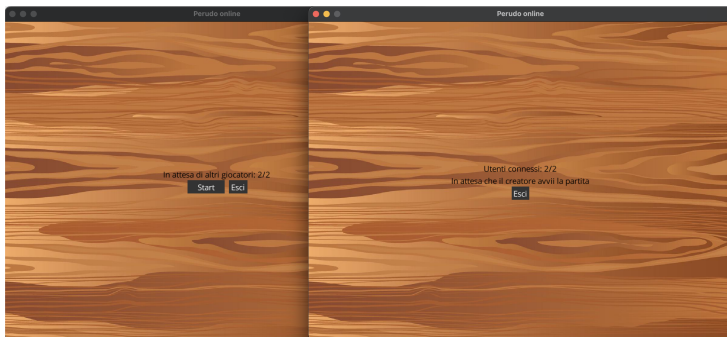
Al termine della partita ogni giocatore viene notificato e riportato alla schermata iniziale.



Pre match disconnection

What happens

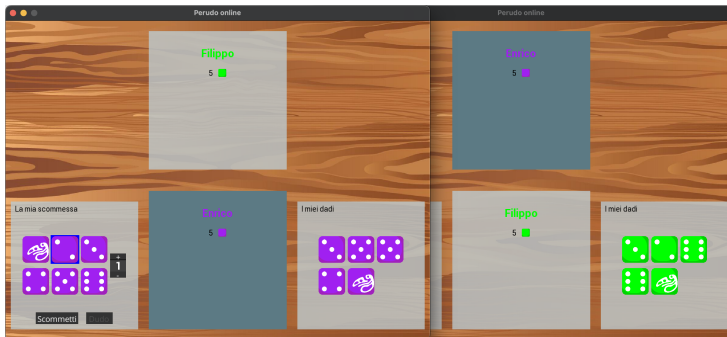
Se è un utente normale ad uscire prima che la partita sia stata avviata quest'ultimo viene riportato alla schermata iniziale. Se è il creatore della lobby ad uscire tutti gli utenti connessi alla corrispondente lobby vengono riportati alla schermata iniziale e la lobby viene eliminata.



During match disconnection

What happens

Se un utente qualsiasi si disconnette durante una partita in corso tutti i giocatori vengono riportati alla schermata iniziale, la partita viene interrotta e la lobby eliminata.



Server crash

What happens

Dopo alcuni secondi dal crash del server il client si accorge dell'irraggiungibilità di quest'ultimo e notifica l'utente. In seguito l'applicazione viene terminata.

