

The Game

Progetto di Sistemi Distribuiti – A.A. 2020/2021

Simone Luzi (0000981239) – simone.luzi@studio.unibo.it

Obiettivi

- Creare una versione digitale e distribuita del gioco da tavolo «The Game»
- Realizzare un prodotto di buona fattura (qualità e manutenibilità del codice, aderenza alle best practice, presenza di documentazione, ...)
- Studiare ed utilizzare una nuova tecnologia per lo sviluppo di sistemi distribuiti
- Utilizzare strumenti di automazione del ciclo di vita del software
- Modellazione completa ed efficiente delle regole del gioco

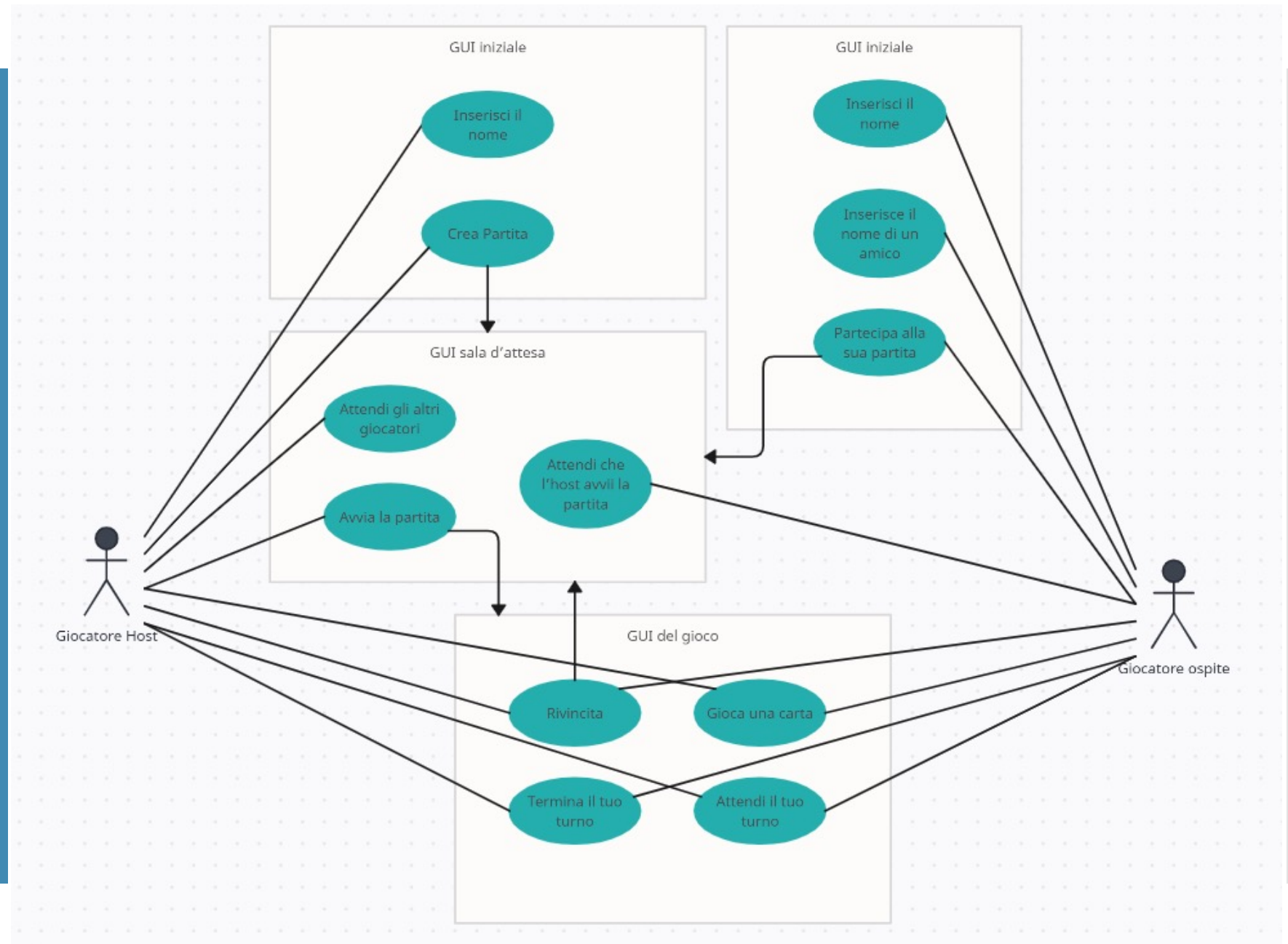
Il gioco da tavolo



Tecnologie utilizzate

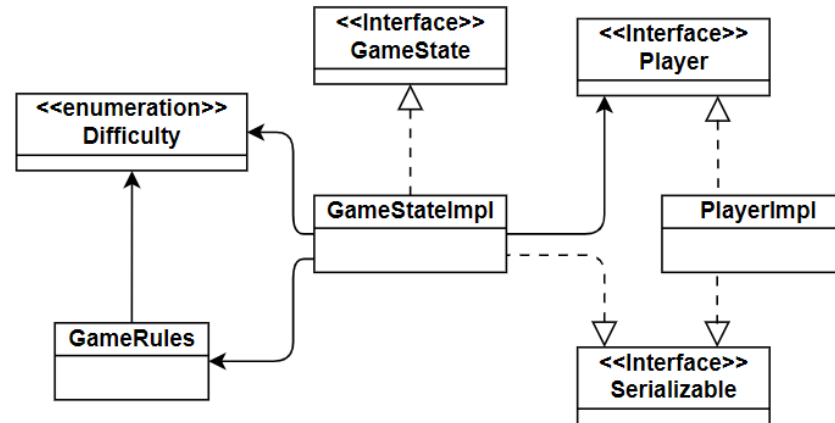
- Gradle
- Akka
- Akka Cluster
- Akka Sharding
- Checkstyle
- Shadow plugin
- Pipeline GitLab

Casi d'uso



Struttura

- Modelli utilizzati: paradigma ad attori e pattern MVC
 - Classi di Model: difficoltà, regole, stato della partita e giocatore

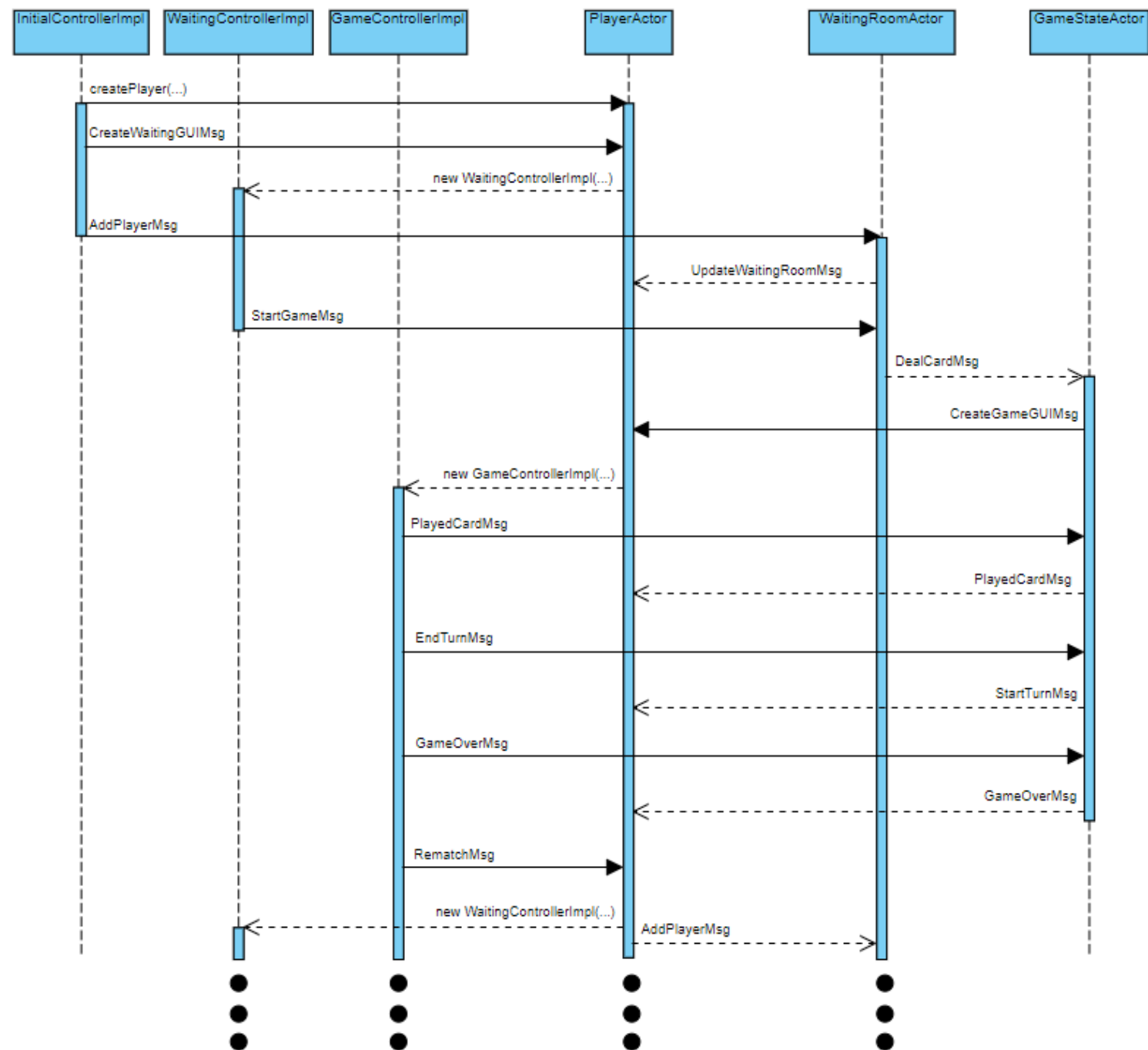


- Componenti principali: Controller, attori, cluster
 - Controller: GUI iniziale, sala d'attesa e partita
 - Attori: giocatore, sala d'attesa, gestione della partita e log di sistema
 - Cluster: creazione di un actor system con specifica configurazione

Comportamento

- I Controller ricevono le richieste dell'utente tramite la View
- L'attore relativo al giocatore crea e aggiorna le View tramite i controller, riceve gli aggiornamenti sullo stato del gioco e comunica le mosse dell'utente agli altri giocatori
- L'attore della sala d'attesa gestisce la formazione del gruppo di giocatori ed avvia la partita
- L'attore con lo stato del gioco crea e condivide in maniera consistente lo stato iniziale della partita, coordina i turni dei giocatori, comunica le loro mosse e gestisce la fine della partita

Interazioni



Gestione degli attori

- Attore classico:
 - Viene creato tramite un apposito comando
 - È ospitato all'interno del nodo nel quale è stato creato
 - È possibile inviargli messaggi tramite il suo riferimento che contiene informazioni sul nodo e sulla porta associata
- Attore in uno Shard:
 - Viene creato automaticamente quando viene inviato un messaggio ad una Shard Region in cui non è presente un attore con quell'identificativo

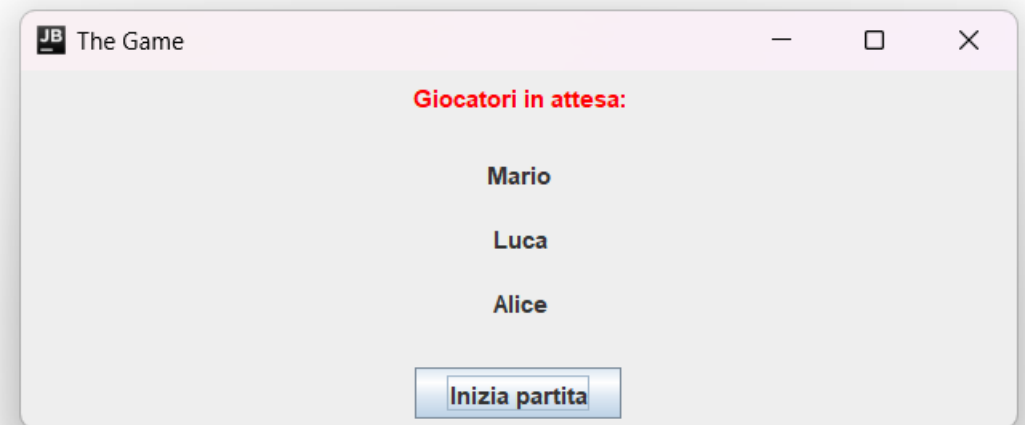
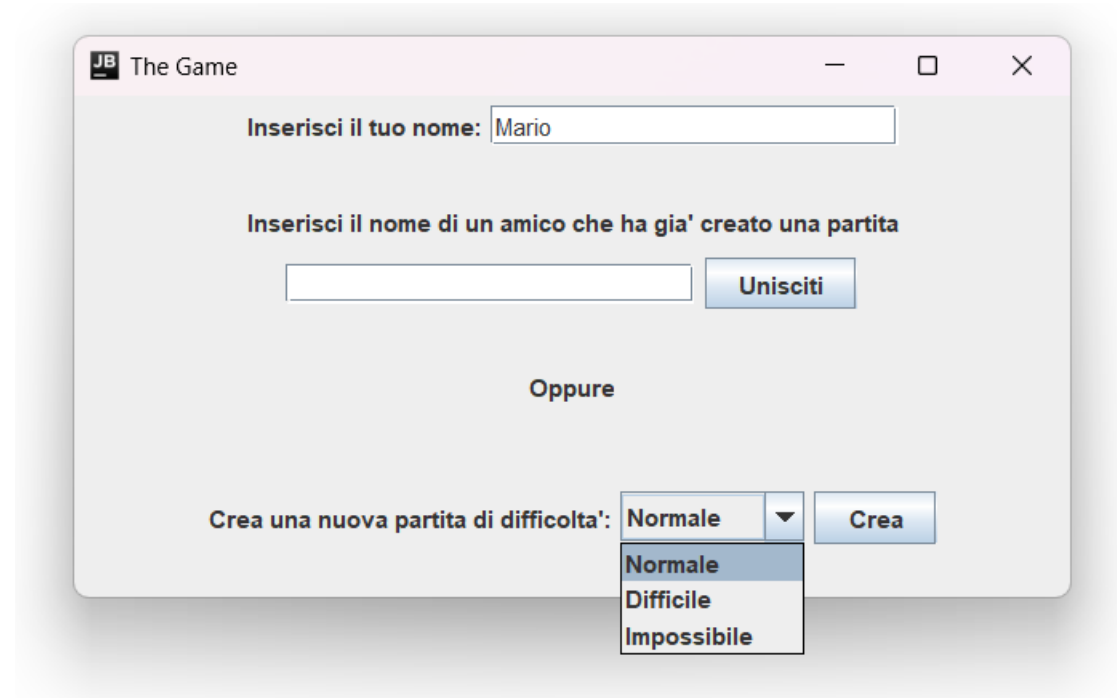
Esempi di riferimenti:

- Actor[akka://ClusterSystem/user/Luca#-1852691245]
- Actor[akka://ClusterSystem/system/sharding/Waiting/0/Luca-WaitingRoom#1171478467]
- Actor[akka://ClusterSystem@127.0.0.1:58862/user/Mario#1151566617]
- Actor[akka://ClusterSystem/system/sharding/Waiting/7/Alice-WaitingRoom#-298418361]

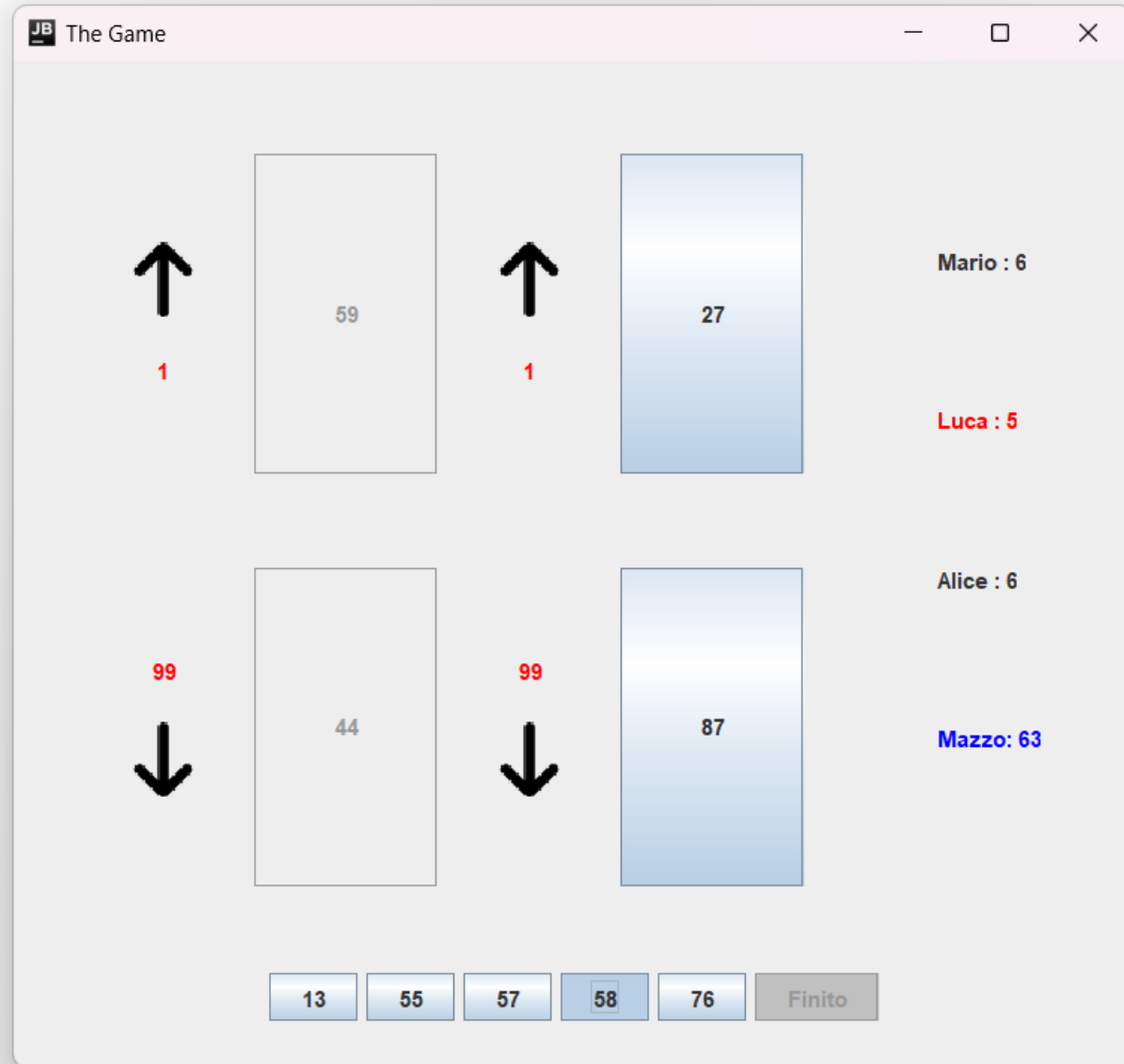
Gestione del cluster

- Utilizzare il file di configurazione nella creazione dell'actor system
- Assicurarsi di avere un nodo attivo come seed nodes in ascolto sulla porta associata
- Avviare le Shard Region prima di inviare i messaggi agli attori che ospitano
- Gestire la comunicazione asincrona tra gli attori e specificare come si devono comportare alla ricezione dei diversi tipi di messaggi

Interfacce grafiche



Interfacce grafiche



Risultati attesi e deliverable

- Una versione digitale e distribuita del gioco, giocabile facilmente sia in versione single-player che multiplayer
- Estrema facilità di esecuzione del programma grazie al Gradle Wrapper
- Un file JAR che contenga tutte le dipendenze, eseguibile avendo installato Java 17 o una versione superiore

Conclusioni

- Conoscenze acquisite:
 - Utilizzo efficace di uno strumento di build automation e di pratiche di validazione continua
 - Comprensione ed utilizzo del modello ad attori e della comunicazione asincrona
 - Gestione di un cluster Akka ed utilizzo di Akka Sharding per avere scalabilità e discovery
- Sviluppi futuri:
 - Test automatici per cluster e attori
 - Controlli sui nomi dei giocatori e gestione delle disconnessioni
 - Deploy del sistema in internet avendo seed nodes con IP fisso
 - Migliorare le GUI ed inserire l'espansione del gioco chiamata «On Fire» con le relative regole
- Considerazioni:
 - Soddisfazione per il risultato finale (Gradle, CI, modello ad attori)
 - Superamento delle difficoltà relative all'utilizzo di una nuova tecnologia per lo sviluppo di un sistema distribuito