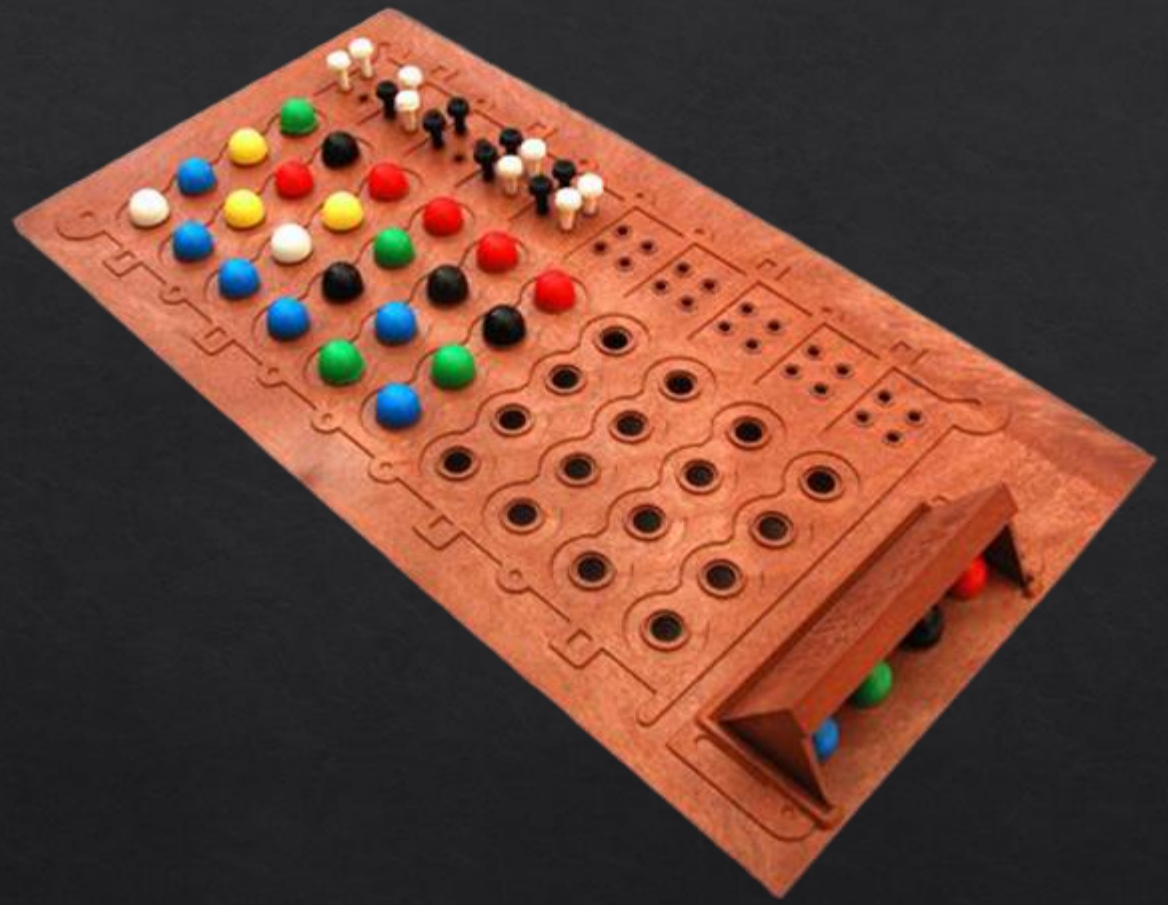


Mastermind Online

Sistemi Distribuiti

Andrea Negri, A.A. 2022/2023

Regole di gioco

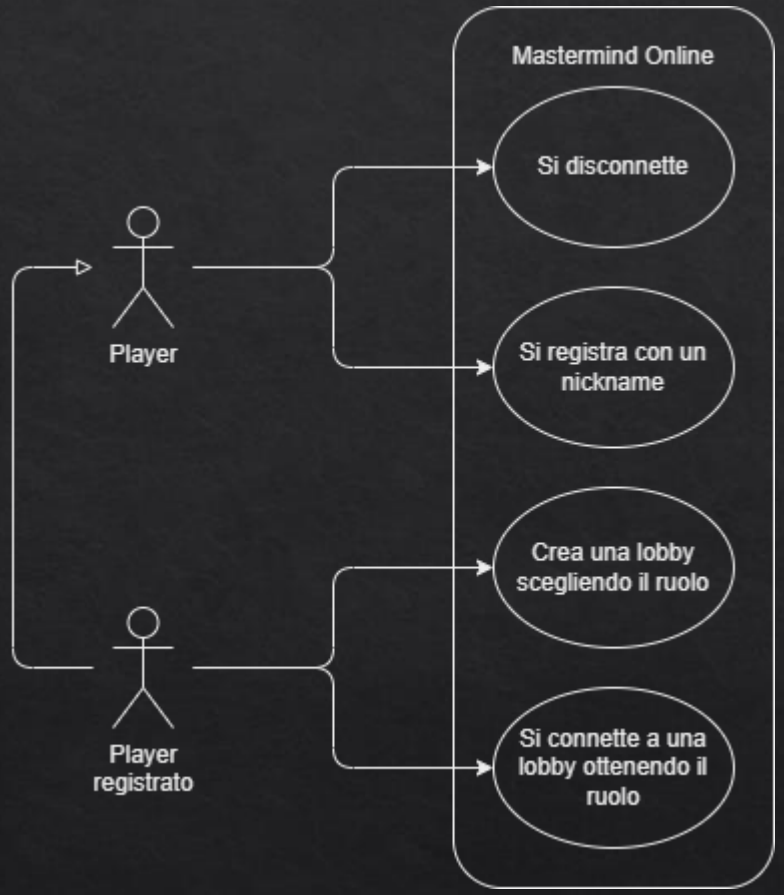
Il progetto consiste nella realizzazione di una **versione online** del gioco da tavolo **Mastermind**.

- Ogni partita è formata da **2 giocatori** che si sfideranno. Un giocatore ricopre il ruolo di codificatore (encoder) mentre l'altro ricopre il ruolo di decodificatore (decoder)
- Il codificatore sceglie un codice segreto di 4 cifre decimali, e il decodificatore ha 9 tentativi per indovinarlo
- Per ogni tentativo effettuato, il decodificatore ottiene degli indizi sulla correttezza del tentativo, in particolare il numero di cifre giuste al posto giusto e il numero di cifre giuste al posto sbagliato
- Se il decodificatore indovina il codice entro i 9 tentativi vince, altrimenti vince il codificatore

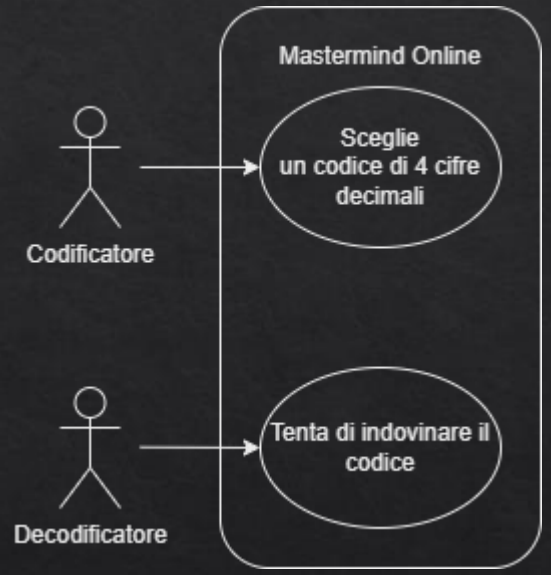
Obiettivi: generali

- Avere le stesse regole della versione fisica
- Consentire ai giocatori di creare le lobby, o unirsi ad esse, per sfidarsi
- Implementare una versione **distribuita** di Mastermind

Obiettivi: scenari



Casi d'uso delle funzionalità di connessione



Casi d'uso delle funzionalità di gioco

Analisi

Requisiti di connettività

Ogni **giocatore** deve:

- Potersi connettere al server inserendo un **nickname** univoco
- Poter creare una **lobby** (scegliendo il ruolo) o connettersi a una già esistente (ottenendo il ruolo mancante)
- Giocare correttamente la **partita**

Analisi

Requisiti di gioco

Riprendendo le regole di gioco:

- Il codificatore a inizio partita sceglie un codice segreto
- Scelto il codice il decodificatore ha 9 tentativi per indovinarlo: se ci riesce vince, altrimenti vince il codificatore
- Per ogni tentativo vengono forniti gli indizi al decodificatore

Analisi
Requisiti di sistema

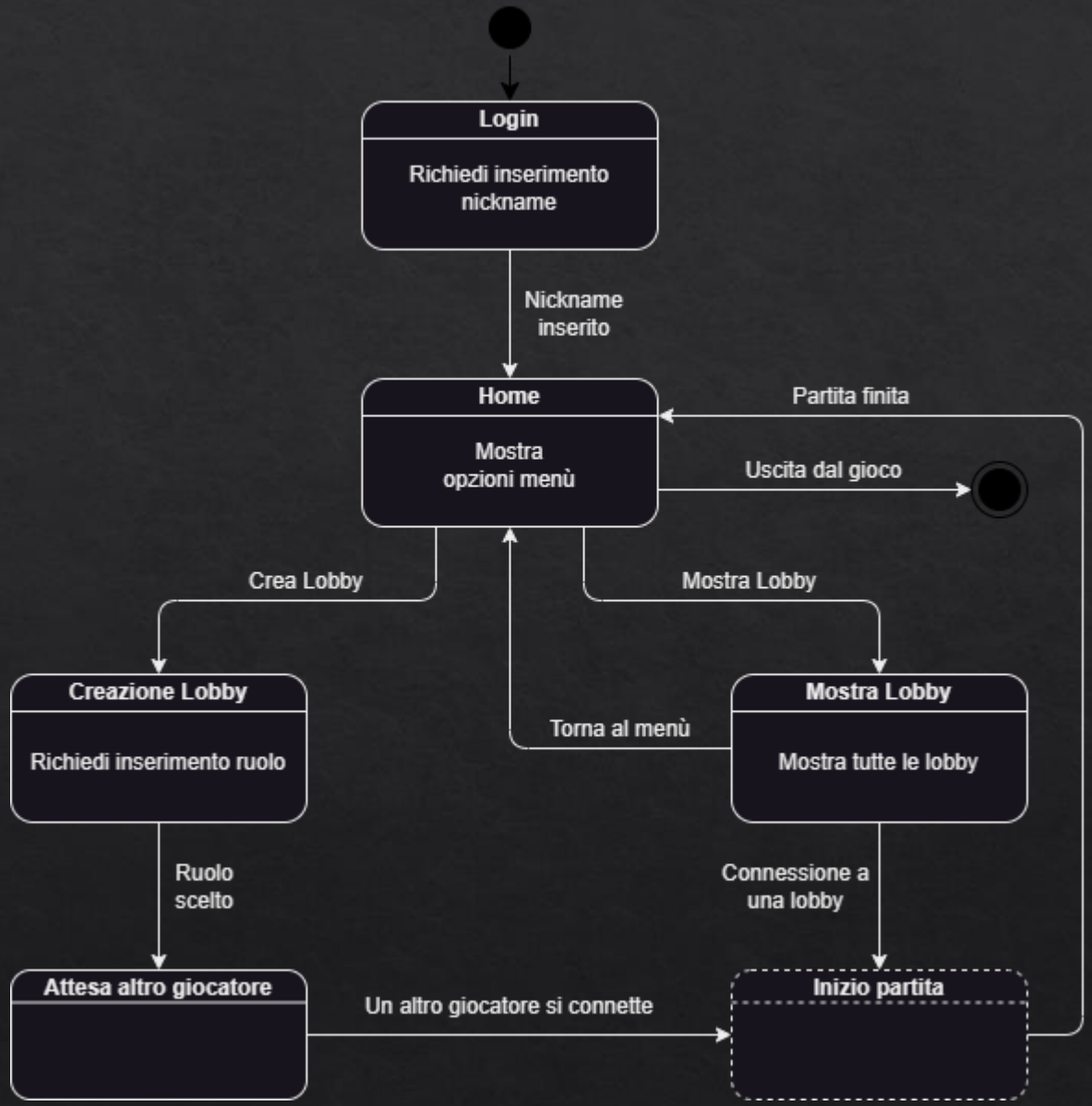
- Architettura **client-server**
- Server:
  - **Web Service** con **API ReST**
  - Implementato tramite framework **Javalin**
  - Utilizzo del protocollo **HTTP**
  - Gestisce le lobby e le partite
- Client:
  - Utilizza una **CLI** per interagire con il sistema

Design

Struttura

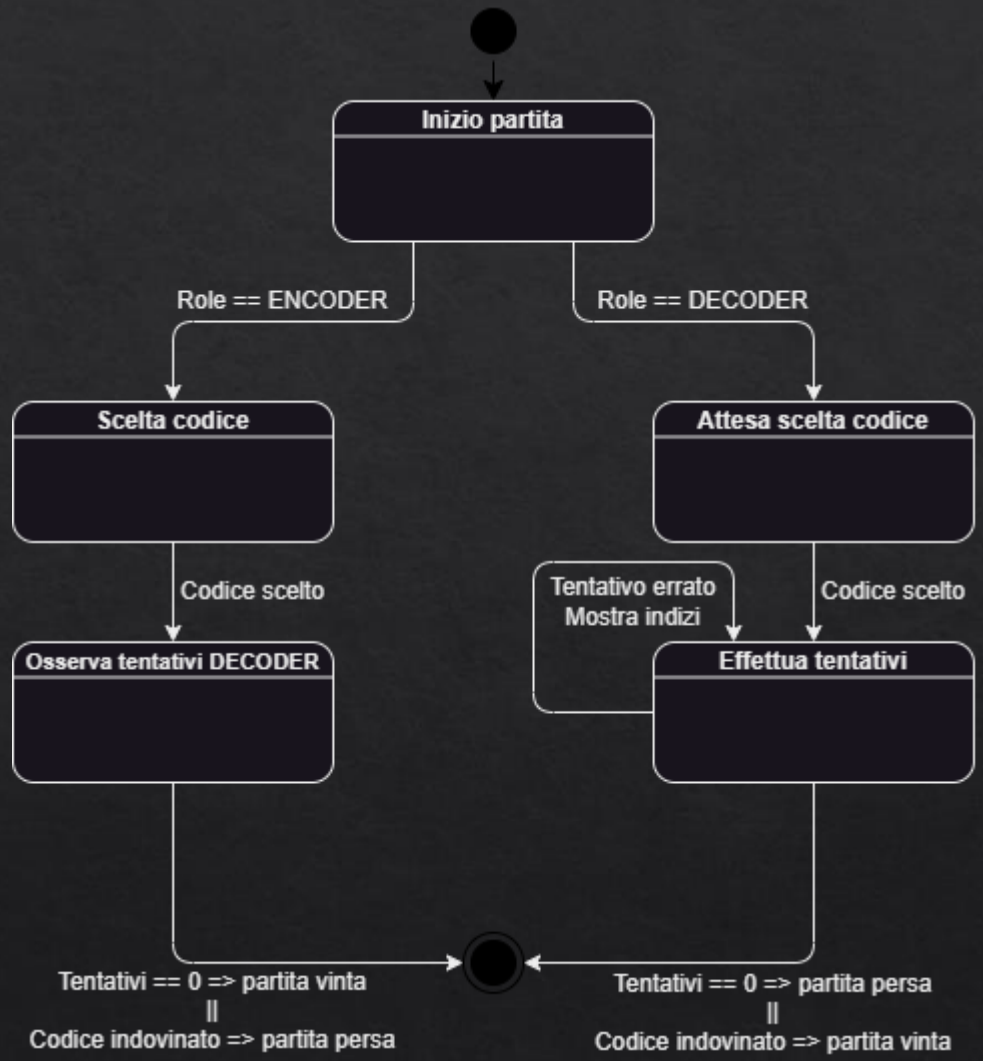
Design

Comportamento: stati della lobby



Design

Comportamento: stati della partita



Design

Comportamento: timer

Il sistema prevede l'utilizzo di alcuni **timer** per gestire i casi di giocatori inattivi:

- Il codificatore ha 45 secondi per scegliere il codice, finiti i quali, se non ha scelto, il decodificatore viene disconnesso automaticamente, mentre il codificatore rimane solo in lobby. Una volta scelto il codice, viene disconnesso anche lui
- Scelto il codice il decodificatore ha 45 secondi per effettuare un tentativo, se non lo fa, il codificatore viene disconnesso, e il decodificatore rimane solo in lobby. Una volta effettuato un tentativo, viene disconnesso anche lui

Design

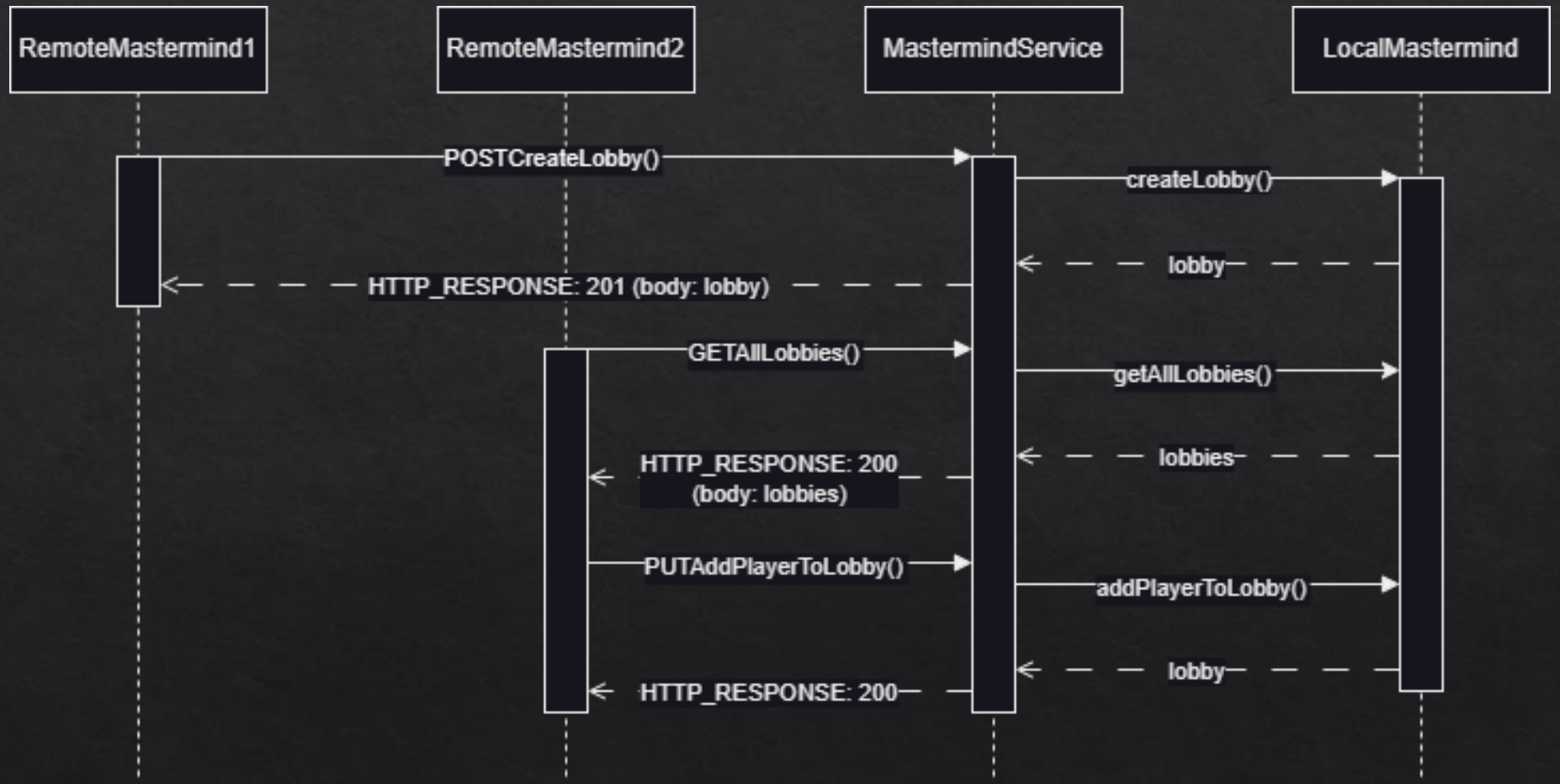
Comportamento: heartbeat

Nel sistema è stato implementato un meccanismo di **heartbeat** per gestire i casi di disconnessioni non volute:

- Nel momento in cui un nuovo giocatore si connette al server, il client inizia a inviare un segnale ogni **5 secondi** per comunicare che è ancora online
- Il server tiene traccia della connettività dei client grazie ai segnali ricevuti
- Se il server non riceve alcun segnale da un determinato client per **10 secondi** lo considera disconnesso (il relativo giocatore viene eliminato e disconnesso da un eventuale lobby a cui partecipava)

Design

Interazioni: creazione, visualizzazione e connessione di una lobby



Implementazione

Moduli del progetto

Il progetto consiste di **4** moduli:

- **Client**: contiene la CLI utilizzata dai giocatore per interagire con il sistema
- **Common**: contiene gli elementi del modello, serializzatori e deserializzatori
- **Server**: contiene il Web Service dotato di API ReST grazie al quale i giocatori possono connettersi e giocare
- **Test**

Implementazione

Client

Si divide in **2** componenti principali:

- **MastermindClient**: contiene CLI tramite la quale l'utente può interagire con il sistema
- **RemoteMastermind**: è l'implementazione dell'interfaccia **Mastermind** lato client. Viene utilizzata dal **MastermindClient** per contattare il server attraverso chiamate HTTP

Implementazione

Common

Contiene:

- Elementi del **modello**
- Gestore del meccanismo di **Heartbeat**
- Interfaccia **Mastermind** e sua implementazione locale (**LocalMastermind**)
- **Serializzatori** e **deserializzatori**
- Elementi di utility

Implementazione

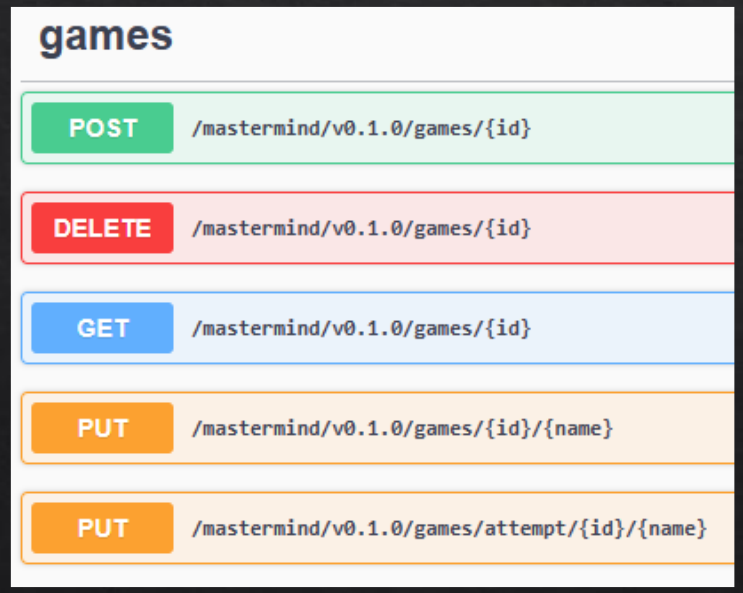
Server

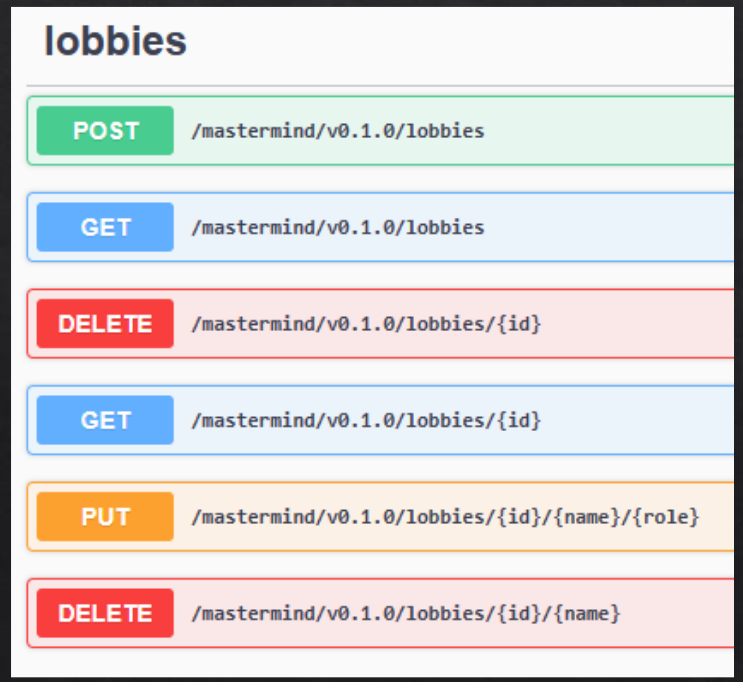
- **Web Service** dotato di **API ReST**
- Lavora su 3 risorse (ognuna con un proprio Controller e Api):
  - **Game**
  - **Lobby**
  - **Player**
- Agisce sulla versione locale del gioco, cioè **LocalMastermind**

Implementazione

Server: Swagger

Le rotte disponibili sono state documentate con **Swagger** e consultabili, una volta avviato il server, dal percorso *http://[host]/[port]/doc/ui*





Implementazione
Test e Autovalidazione

Nel modulo test sono presenti appunto i **test** che verificano la correttezza:

- Dei serializzatori e deserializzatori
- Della logica delle partite (sia locale che in rete)
- Gestione delle lobby (sia in locale che in rete)
- Dei giocatori (sia in locale che in rete)

Inoltre, sono state sfruttate le **CI/CD Pipeline** di GitLab affinché, ad ogni push, vengano eseguiti in automatico i test presenti.

Deployment

- Run **Server**

```
./gradlew server:run
```

- Run **Client**

```
./gradlew --console plain client:run
```

Esempi d’uso

Login, menù principale

```
> Task :client:run
Benvenuto! Inserisci un nickname:
Andrea
```

```
---------------MENU---------------
Seleziona una delle voci presenti
1) Crea una lobby
2) Visualizza lobby disponibili
3) Esci
```

Esempi d’uso

Creazione lobby

```
1

Seleziona un ruolo:
1) Codificatore
2) Decodificatore
1

---------------------------------
Lobby 1 creata.
In attesa di un altro giocatore...
```

Esempi d’uso

Visualizzazione lobby

```
2

--------------LOBBIES--------------
Lobby 1 - Ruolo mancante: DECODER - Player collegato: Andrea

R) Refresh lista
B) Torna al menu
Inserisci codice lobby a cui connettersi:
```

Esempi d’uso

Inizio partita

```
---------INIZIO GAME---------
Sei il codificatore
Inserisci il codice da far indovinare (ricorda, deve essere composto da 4 cifre DECIMALI (0-9):
1463
```

```
---------INIZIO GAME---------
Sei il decodificatore
Attendi che il codificatore scelga un codice...
```

Esempi d’uso

Tentativo effettuato

```
Codice 1463 impostato
Il decodificatore ha provato il codice 4523
```

```
Il codificatore ha scelto il codice
Tentativi rimasti: 9
Inserisci il tuo tentativo (ricorda, deve essere composto da 4 cifre DECIMALI (0-9):
4523
Hai provato il codice 4523
Cifre giuste al posto giusto: 1
Cifre giuste al posto sbagliato: 1
----------------------------------------
Tentativi rimasti: 8
Inserisci il tuo tentativo (ricorda, deve essere composto da 4 cifre DECIMALI (0-9):
```

Esempi d’uso

Vittoria decodificatore

```
Mi dispiace, hai perso :(
```

```
Tentativi rimasti: 8
Inserisci il tuo tentativo (ricorda, deve essere composto da 4 cifre DECIMALI (0-9):
1463
COMPLIMENTI! HAI VINTO!
```

Esempi d’uso

Vittoria codificatore

```
Codice 1973 impostato
Il decodificatore ha provato il codice 4564
Il decodificatore ha provato il codice 7897
Il decodificatore ha provato il codice 1232
Il decodificatore ha provato il codice 4562
Il decodificatore ha provato il codice 3258
Il decodificatore ha provato il codice 1594
Il decodificatore ha provato il codice 3575
Il decodificatore ha provato il codice 4568
COMPLIMENTI! HAI VINTO!
```

```
Tentativi rimasti: 1
Inserisci il tuo tentativo (ricorda, deve essere composto da 4 cifre DECIMALI (0-9):
1595
Mi dispiace, hai perso :(
```

Grazie per l’attenzione