

CAHu Clone

Rossi Luca

Schiaroli Davide

Tronetti Elisa

1 Abstract

L'idea è quella di sviluppare una versione digitale e distribuita di *Cards Against Humanity*, i cui riferimenti del gioco ufficiale possono essere trovati a questo link <https://cardsagainsthumanity.com>.

Gli utenti hanno a disposizione un applicativo Java che comunica con un server, che gli permetterà di interagire con gli altri client. All'apertura del programma si chiede se l'utente vuole creare una nuova lobby (stanza), oppure entrare in una lobby già esistente. Durante la partita il server si occuperà soltanto di gestire la sincronia fra i vari client che appartengono alla stessa stanza, di modo che ogni client abbia la stessa visione del turno della partita, mentre lo stato del gioco viene memorizzato dai client.

A inizio partita vengono distribuite 10 carte bianche (carte che contengono delle frasi-completamento) ad ogni giocatore e viene eletto un leader. A ogni turno viene scoperta una carta nera (carta con frase da completare) e i giocatori devono scegliere entro un tempo prestabilito una tra le carte a loro disposizione, al fine di completare nel modo che ritengono opportuno la carta nera.

Alla fine del turno il leader sceglie la carta bianca più divertente e viene assegnato un punto al proprietario della carta, il quale diventa il leader per il turno successivo.

A questo punto viene assegnata un'altra carta ai giocatori e si va avanti finché uno tra i giocatori non raggiunge un punteggio prestabilito.

2 Goal and deliverable

L'obiettivo è quello di sviluppare un applicativo Java per la parte client ed un server, sempre implementato in Java.

L'architettura sarà client-server, con modello fat-client, in cui la complessità del gioco verrà gestita principalmente dai client. Infatti si mira a mantenere il server come intermediario per lo scambio di messaggi tra client, mentre lo stato intero della partita verrà mantenuto in locale da ogni client, di modo da alleggerire il lavoro a carico del server.

Per realizzare il server abbiamo intenzione di usare la tecnologia Vert.x, modellando le specifiche API per la comunicazione HTTP attraverso Swagger.

Saranno progettati test Junit sia per il lato client che per il lato server e verranno

effettuate anche prove di gioco reali, al fine di verificare il corretto funzionamento del software prodotto.

Abbiamo pensato di utilizzare Gradle per la build automation, di modo da poter sfruttare i vantaggi offerti da questo strumento.

3 Work plan

La prima fase riguarderà la modellazione dell'intera struttura del gioco, che verrà effettuata da tutti i membri del gruppo, al fine di avere una visione completa dei componenti e di come dovranno interagire tra loro.

Una volta terminata la fase di progettazione, il lavoro potrà essere suddiviso equamente, supponendo almeno inizialmente che il 30% del lavoro sarà per la parte server e il 70% per la parte client.

Le parti implementative ancora da decidere verranno discusse più approfonditamente nel report finale insieme alla suddivisione del lavoro effettiva.