

**Relazione progetto LAM 2018-2019**

**Matteo Conti**

## **INDICE**

- Introduzione
  - Scopo dell'Applicazione
- Architettura generale (TODO)
  - MVC
    - Carta (TODO)
    - Giocatore (TODO)
    - Gestore Gioco (TODO)
    - Gestore Database (TODO)
    - Gestore Sessione (Multipeer) (TODO)
    - Gestore Social (Facebook) (TODO)
  - Funzionalità (TODO)
  - Librerie Utilizzare (TODO)
  - Requisiti Tecnici (TODO)
- Progettazione (TODO)
  - Difficoltà e Soluzioni (TODO)
  - Aspetti Rilevanti per lo Sviluppo (TODO)
  - Use Case (TODO)
  - Idee per Estensioni (TODO)
- Conclusione e Commenti Finali (TODO)

## **INFORMAZIONI GENERALI**

Studente: *Matteo Conti*

Matricola: *0000802576*

Corso: *laboratorio di applicazioni mobile*

Anno Accademico: *2018/2019*

Data: *1 Febbraio 2020*

## **Introduzione**

Il progetto consiste nell'applicazione pratica delle conoscenze acquisite durante lo svolgimento del corso "Laboratorio di Applicazioni Mobile" presso l'Università di Bologna.

Lo scopo dell'applicazione è la riproduzione del funzionamento del gioco di carte italiano, chiamato Briscola, su dispositivi aventi sistema operativo iOS (piattaforma Apple). La versione attuale del gioco prevede la presenza di due giocatori e di due modalità di gioco. Nella prima modalità (single-player) un giocatore viene controllato da un emulatore basato su un semplice algoritmo a stati che determina la carta da giocare in base allo stato attuale del gioco. Sono state pienamente implementate tutte le regole e caratteristiche della Briscola, e quindi il comportamento dell'emulatore seguirà una strategia di gioco che rispecchia la logica dello stesso gioco. Inizieremo discutendo le caratteristiche e l'architettura generale del progetto, illustrando le funzionalità e l'implementazione alla base di ciascuna di esse. Proseguiremo illustrando le scelte attuate durante la fase di progettazione, i problemi e le difficoltà incontrate, evidenzieremo alcuni dettagli che hanno inciso sulle tempistiche di consegna e mostreremo i flussi di dati presenti all'interno dell'applicazione. Infine termineremo proponendo alcune possibili implementazioni, aggiungendo alcuni commenti relativi alla curva di apprendimento e le tecnologie a disposizione.

### **1. Scopo dell'Applicazione**

Lo scopo era quello di creare due ambienti di gioco: il primo (con la presenza di un giocatore simulato) serviva per simulare una sorta di fase iniziale di "training" (anche se non c'è un vero trainer che corregge le mosse fatte dall'utilizzatore dell'applicazione), mentre il secondo serviva per testare i propri progressi confrontandosi con altri giocatori.

## **Architettura generale (TODO)**

Il pattern utilizzato nella realizzazione di questa applicazione è il Model-View-Controller (MVC). L'importanza di questo pattern consiste nella separazione della logica di presentazione dei dati rispetto alla logica di gestione di questi. Per questo progetto infatti, sono stati creati diversi modelli di dati corrispondenti alle varie entità in gioco (Carte, Giocatori, ..), tutti gestiti da diverse librerie di "gestione" (Handlers), le quali astraggono completamente il flusso di creazione/modifica dei dati rispetto alla loro presentazione (gestita all'interno dei Controllers).

## **2. MVC**

All'interno dell'applicazione esiste un `NavigationController` attraverso il quale è possibile muoversi attraverso la varie view che compongono l'applicazione. Esistono infatti 4 schermate:

1. **Menù**: da qui è possibile impostare le varie opzioni di gioco e passare sia alla schermata di gioco, sia a quella "Social".
2. **Gioco**: in questa schermata è possibile avviare il gioco e passare alla schermata dei "Risultati"
3. **Risultati**: qui è possibile vedere il risultato della partita una volta terminata e decidere di salvarla.
4. **Social**: da qui è possibile vedere la lista di tutte le partite salvate e, per ognuna, decidere se condividerla su Facebook.

Di seguito verranno riportati i principali modelli di dati utilizzati e le librerie a supporto di questi.

### **2.1. Carta (TODO)**

Gli attributi principali sono il tipo e il numero.

### **2.2. Giocatore (TODO)**

contiene una variabile che identifica le carte che ha in mano e altre informazioni come il tipo (locale, emulatore), il nome e le carte conquistate presenti nel suo mazzo.

**2.3. Gestore Gioco (TODO)**

.. .. .

**2.4. Gestore Database (TODO)**

.. .. .

**2.5. Gestore Sessione (Multipeer) (TODO)**

.. .. .

**2.6. Gestore Social (Facebook) (TODO)**

.. .. .

**3. Funzionalità (TODO)**

.. .. .

**4. Librerie Utilizzare (TODO)**

.. .. .

**5. Requisiti Tecnici (TODO)**

.. .. .

**Progettazione (TODO)**

.. .. .

**6. Difficoltà e Soluzioni (TODO)**

.. .. .

**7. Aspetti Rilevanti per lo Sviluppo (TODO)**

.. .. .

**8. Use Case (TODO)**

.. .. .

**9. Idee per Estensioni (TODO)**

.. .. .

**Conclusione e Commenti Finali (TODO)**

.. .. .