

CA400 - Final Year Project

User Manual

Title: Heads Up Computer Poker

Student: Neil Saccardo

Student Number: 13332546

Supervisor: Dr. David Sinclair

Blog Link:

<https://gitlab.computing.dcu.ie/saccarn2/2017-ca400-saccarn2/blob/master/docs/blog/blog.md>

Date Finished: 22/05/2017

Table of Contents

Prerequisites	3
Project set up	3
Steps	3
Downloading project	3
Setting Up Data_Processing	3
Running System	4
Validation Module	5
User Guide - UI	6

Prerequisites

The following must be downloaded and installed to run the project.

- Java 8 <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>
- NodeJS (version 6.x) <https://nodejs.org/en/> (ensure NPM is also installed).
- MongoDB - <https://www.mongodb.com/download-center#community>
- Maven - <https://maven.apache.org/download.cgi>

Unfortunately the data used for clustering appears to have been taken down from the University of Alberta website. The data used as part of the project in a zip file on DCU school of computing webspace, found here:

<http://student.computing.dcu.ie/~saccarn2/pokerUofA/pokerdata.zip>

Download and extract this zip file into an appropriate directory.

Project set up

Note: MongoDB must be running as a service on port 27017 (default MongoDB port) in the background.

Steps

Downloading project

- Use git to clone repository **or** download zip file and extract to desired directory.
 - Use this to clone repo using http:
<https://gitlab.computing.dcu.ie/saccarn2/2017-ca400-saccarn2.git>
 - Download zip file from here :
<https://gitlab.computing.dcu.ie/saccarn2/2017-ca400-saccarn2/tree/master>

Setting Up Data_Processing

- Go to Data_Processing directory
- Using an editor, open com.saccarn.poker.dataprocessing.GameAnalyser.
- Go to constructor with one argument of the class- looks like this:

```
/* Change the directory to point where your data is kept */  
public GameAnalyser() { dir = new File("C:\\Data\\test\\nolimit\\"); }
```

- Change the directory path to point where the poker data was extracted to.
- Using the command line, cd to Data_Processing directory
- Run mvn install

- Using the command line, cd to AI_Agent directory
- Run mvn exec:java -Dexec.mainClass="com.saccarn.poker.dbprocessor.DataLoader"
- Run mvn exec:java -Dexec.mainClass="com.saccarn.poker.dbprocessor.PlayerTypeClusterer"

Running System

- Using the command line, cd into AI_Agent directory.
- Run mvn install
- Using the command line (cmd1), cd into Poker_Server directory
- Run mvn install
- Open new command line (cmd2), cd into the ui/server directory.
- Run npm install
- Run npm install gulp -g . This installs gulp globally.
- With cmd1 run 'mvn exec:java' in the Poker_Server directory.
- With cmd2, in the ui/server directory run 'gulp'.

Wait a few seconds, and open up localhost:3000 in the browser. You should be presented with the login page of the web app.

Note:

The web app listens for connections on port 3000 - this can be changed by editing the PORT_NUM variable in the app.js file in ui/server.

The java server listens for connections on port 3500. This can be changed by changing the port variable in the Server class in the Poker_Server module. The JAVA_PORT variable in app.js of ui/server must also be changed to match this number.

Unit Tests

The junit tests of every java module can be run by going into the module directory using the command line and running:

- mvn test

The output to the console will include how many tests were run, how many were successful and how many failures there were.

Validation Module

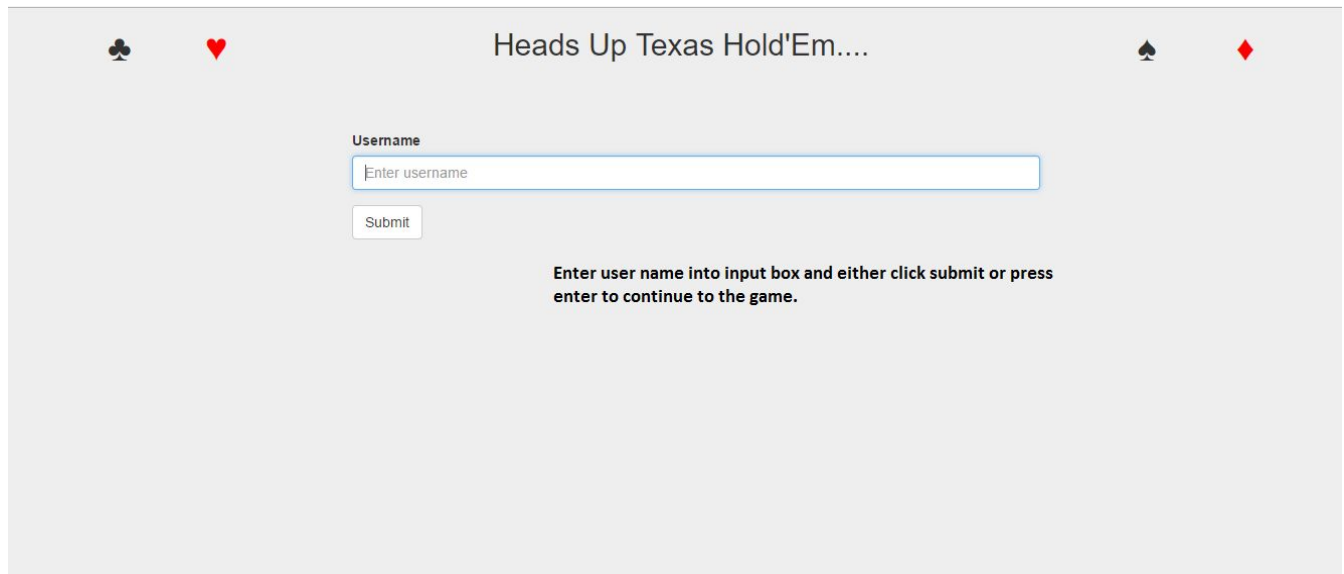
To run the classes in the validation module

- Go the validation module directory using the command line
- To run opponent model tests
 - Run mvn exec:java
-Dexec.mainClass="com.saccarn.poker.opponentmodels.TEST_NAME" where
TEST_NAME is the name of test class.
- To run bet pass parameter model tests
 - Run mvn exec:java -Dexec.mainClass="com.saccarn.poker.testsbetparams.TEST_NAME"
where TEST_NAME is the name of test class.
- To run 'common hand' tests
 - Run mvn exec:java -D
exec.mainClass="com.saccarn.poker.testcommonhandvalues.T_N" where T_N is the
name of test class.
- To run 'hand potential' tests
 - Run mvn exec:java -Dexec.mainClass="com.saccarn.poker.testshandpotential.T_N"
where T_N is the name of test class.
- To run pre flop tests
 - Run mvn exec:java -Dexec.mainClass="com.saccarn.poker.testspreflopvalues.T_N"
where T_N is the name of test class.

User Guide - UI

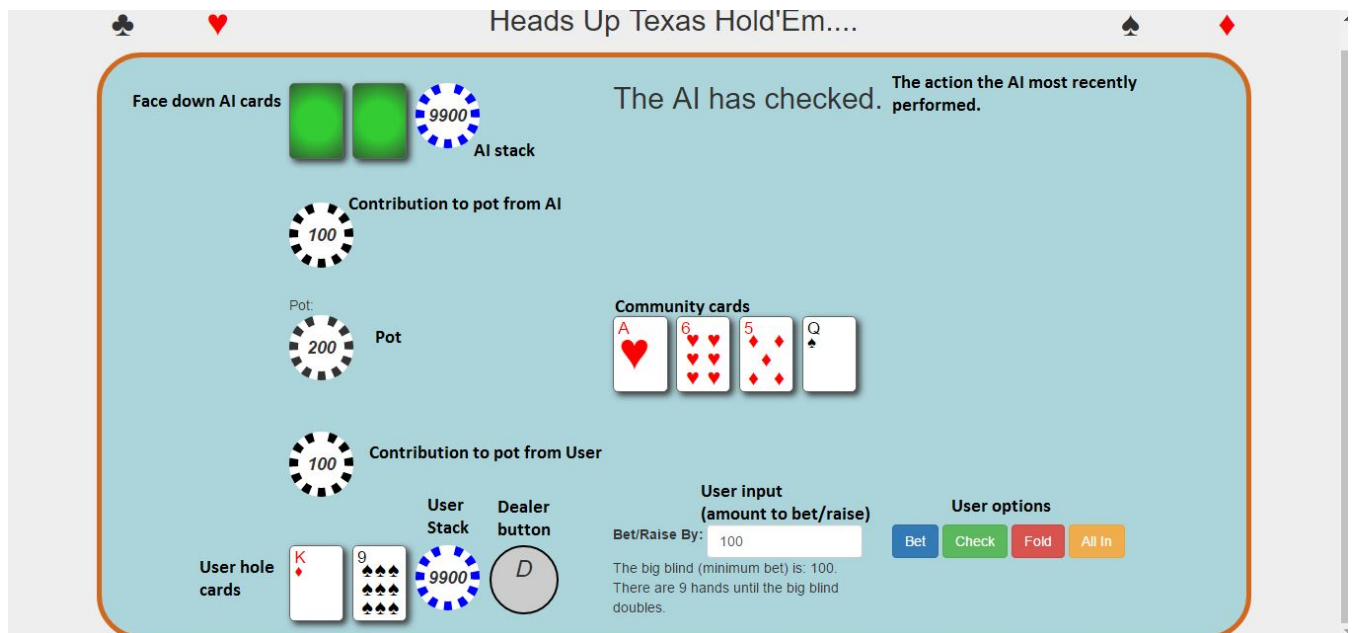
Some annotated screen shots of the UI are included as follows to show the important game objects/features that a user must understand in order to play the game:

Logging in:



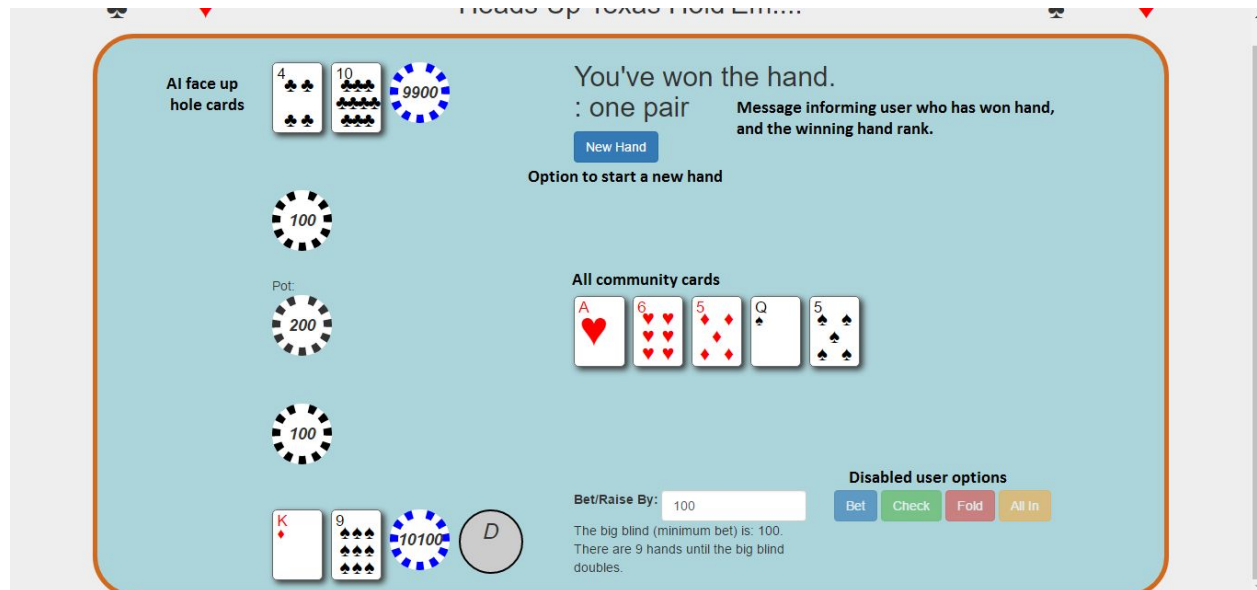
The screenshot shows the login interface for 'Heads Up Texas Hold'em....'. At the top, there are four suit icons: Club, Heart, Spade, and Diamond. Below them is a 'Username' label followed by a text input field containing the placeholder 'Enter username'. A 'Submit' button is positioned below the input field. A message below the button reads: 'Enter user name into input box and either click submit or press enter to continue to the game.'

Playing the game:



The screenshot displays the game interface for 'Heads Up Texas Hold'em....'. The interface is divided into several sections within a light blue rounded rectangle. At the top left, 'Face down AI cards' are shown as two green rectangles. Next to them is the 'AI stack' with a blue chip icon and the value '9900'. Below the AI cards is the 'Contribution to pot from AI' shown as a black chip icon with '100'. The 'Pot' is shown as a black chip icon with '200'. Below the pot is the 'Contribution to pot from User' shown as a black chip icon with '100'. The 'User hole cards' are shown as the King of Diamonds (K♦) and the 9 of Spades (9♠). The 'User Stack' is shown as a blue chip icon with '9900'. The 'Dealer button' is a grey circle with the letter 'D'. The 'Community cards' are the Ace of Hearts (A♥), the 6 of Hearts (6♥), the 5 of Diamonds (5♦), and the Queen of Spades (Q♠). The 'User input (amount to bet/raise)' is shown as a text input field with '100'. Below the input field, there is a message: 'The big blind (minimum bet) is: 100. There are 9 hands until the big blind doubles.' To the right of the input field are four buttons: 'Bet' (blue), 'Check' (green), 'Fold' (red), and 'All In' (orange). At the top right, there is a message: 'The AI has checked. The action the AI most recently performed.'

Showdown:



The video walkthrough supplied as part of this project also details some of the typical actions user would want to perform.