MSc IT+ Masters Team Project

Team 14

Matthew Henderson – 2031944H
Christos Dritsoulas - Mousiadis- 2351485D
Kaikan Wu - 2327942W
Parvir Chomber - 2352834C
Christopher Barnes - 2031986B

Introduction

The assignment specification was to design a game of top trumps that can be played both in the command line and in an online browser. The game will read in a pre-supplied file containing details of top trump cards. It will allow users to specify how many AI players they wish to play against, and then allows users to play a complete a game against these AI opponents. Furthermore, statistics for each game will be saved to a database and can be displayed to users when requested. A test log file can also be generated for the command line version if the user requests it. This test log contains information about the gameplay of each round.

The project must be completed within 5 weeks and will be managed with the agile "scrum" approach.

Requirements Gathering

As part of the requirements gathering process, each developer was involved in an initial brainstorming meeting. Each developer suggested actors they believed to be involved in the system. They then identified user stories for each of these actors and the story points they believed would be required to complete each of the stories. These were compared and then compiled into a complete log of user stories for the system.

In the initial brainstorming the following user roles were identified: human player, AI player, returning player. These can be seen on the actor cards seen below. The user stories for each of these actors then follow.

Role – Player
Frequency of use – Little
Knowledge of game – low
Computer expertise – Unknown
Goals – To play a complete game of top trumps

Role – Returning player
Frequency of use – Often
Knowledge of game – High
Computer expertise – Unknown
Goals – To play a complete game of top trumps and to track statistics of games played.

Role – AI player
Frequency of use – N/A
Knowledge of game – N/A
Computer expertise – N/A
Goals – To select the best attribute on their current card when it is their turn to play.

Player Story: Start a new game Story points - 0.5	Acceptance Tests: Upon selecting start game, a new game is started and the user is able to play.
Returning Player Story: View game statistics Story points - 2	Acceptance Tests: Upon selecting statistics from the command line mode, they are printed in the correct format to the command line. Upon selecting statistics from the online mode, they are displayed in the browser in the correct format. Statistics match those stored in the database.
Player Story: Select number of opponents Conversation: The maximum number of opponents should be 4. Story points - 2	Acceptance Tests: When the number of opponents is selected, the correct number of opponents participate in the game. Incorrect formats such as strings and doubles produce an informative error message and re-prompt the player for an input Numbers integers 1-4 produce an informative error message and re-prompt the player for an input

Player Story: Quit midgame Story points – 0.5	Acceptance Tests: When the user leaves a game part of the way through, the programme is terminated.
Player Story: Save statistics to database Conversations: Statistics should only be saved for completed games Story points – 1	Acceptance Tests: Statistics saved to the database are correct. Statistics are only saved for completed games, not those which the user quits part of the way through.
Player Story: Select command line mode. Story points – 0.5	Acceptance Tests: Upon selecting command line mode, the command line version is started and the user is able to play.

Player Story: Select online mode Story points – 0.5	Acceptance Tests: Upon selecting online mode, the browser version is started and the user is able to play.
Player Story: Select the attribute to compare Story points – 0.5	Acceptance Tests: The user is prompted to enter which attribute to select when it is their turn. Any unexpected inputs display an error message and re-prompt user for input. Correct user inputs are recognised by the game.
AI Story: Select the attribute to compare Conversation The AI should always select the best attribute on their card. Story points – 0.5	Acceptance Tests: When it is the AI's turn, an attribute should be automatically selected. The attribute selected is the highest, or joint highest on the card.

Player Story: Compare attribute Story points – 2	Acceptance Tests: The player with the highest attribute is selected as the winner of the round. User is informed. Draws are recognised. User is informed and game proceeds accordingly. The correct attributes are compared on every card.
Player Story: Play new game Conversation: The player should be allowed the option to play a new game once the last has finished. Story points – 0.5	Acceptance Tests: Once a game has finished, the option to play a new game is displayed. Selecting new game will start a completely new game from scratch.
Player Story: Deal cards Story points – 2	Acceptance Tests: Cards are distributed from the deck to individual players. Cards are distributed in an order that accurately reproduces dealing in a real card game.

Player Story: Shuffle Story points – 2	Acceptance Tests: The cards in the deck are sorted into a completely random order. The cards in the deck for the previous game should not match the order of the current game's deck.
Player Story: Load in cards Story points – 0.5	Acceptance Tests: All cards are loaded from the file to the deck. All cards have the same stats as in the file. No file handling errors are produced.
Player Story: Send cards to the communal pile Story points – 1	Acceptance Tests: Cards are only sent the communal pile when the round is a draw. All cards in play are sent to the communal pile. Only the correct cards are in the communal pile.

Player

Story:
Select winner

Conversation:
Assume that a winner would always be found. A draw is highly unlikely.

Acceptance Tests:

Once a game is completed, the correct winner is displayed on the screen.

The winner should match the value displayed in the database.

The winner should be the last player with remaining cards.

Player

Story:
Display results

Acceptance Tests:

The results of the game are displayed on the screen, including winner and how many rounds were there in total.

Results are shown for the game which had just been played.

Results match the values stored in the database.

Player

Story:
Assign cards based on winner

Story points – 1

Acceptance Tests:

When a round is won, all cards from the other player's hand and the communal pile (if cards are present) are given to the round winner.

Player Story: Print to the log Conversations: Users should only be able to choose to print to the test log in the command line mode. Story points – 4	Acceptance Tests: A complete log is printed for a game after the user selects print log mode. The testlog correctly matches the game that has just been played. Print log mode cannot be selected from the online mode.
Player Story: Play a command line game. Story points – 4	Acceptance Tests: The player can start a command line version of the game when prompted to enter “-c”
Player Story: Play an online game. Story points – 8	Acceptance Tests: The player can start the online version of the game when prompted to enter “-o”

Non-Functional Requirements

Database Story: The database will have the correct columns and rows to store the data Conversation: The database must be relational for integrity of statistics	Acceptance Tests: The correct data is inserted into the appropriate tables The correct data is retrieved from the appropriate tables
GUI Story: The GUI for the online version is responsive and has the appropriate layout and functions to allow for the user to select an attribute	Acceptance Tests: The user can select an attribute to compare against the AI players

Release Plan

The programme must be completed in 5 weeks. There will be 2 sprints of length 2 weeks, and one week for contingency. There is a total of 35.5 story point to be completed in this time, thus the average velocity of the project should be 17.75 story points per sprint. For this project, a story point will be defined as one ideal working day. With 5 developers working on the project, this equates to ~3.5 ideal working days each per sprint.

In keeping with the scrum process there will be a review and retrospective at the end of every iteration. The purpose of these is to assess and demonstrate the progress of the project, and to look for improvements. There will also be "daily scrums" at each of the team meetings. Team meetings are scheduled for Monday and Wednesday.

User stories have been split into must-haves and should-haves. These can be seen below.

Must-have stories

- Select command line mode
- Play a new game
- Command line display
- Start game
- Load in cards
- If draw, cards to communal pile
- Select winner
- If there is a winner, cards go to winner
- Shuffle
- Deal cards
- Select attribute
- Compare attribute

Should-have stories

- Select number of opponents
- Quit midgame
- AI selects the number which is the highest
- Play an online game
- View statistics
- Print to the log
- Save stats
- Select online mode
- Display results

Iteration Plan

The iteration plan can be seen below. Must-have user stories have been assigned to the earlier sprint, ensuring that they are not entered into the iteration backlog until prerequisites have been completed. The first sprint consists of 19 story points and the second of 16.5 story points. User story cards were then picked from the first iteration pile by developers based on the skills and interests of the developer.

Iteration 1 – 19 story points

- Select command line mode (0.5 point)
- Play a new game (0.5 point)
- Command line display (4 points)
- Start game (0.5 point)
- Select number of opponents (2 points)
- Load in cards (0.5 point)
- If draw, cards to communal pile (1 point)
- Select winner (1 point)
- If there is a winner, cards go to winner (1 point)
- Shuffle (2 points)
- Deal cards (2 points)
- Select attribute (0.5 point)
- Compare attribute (2 points)
- Quit midgame (0.5 point)
- AI selects the number which is the highest (1 point)

Iteration 2 – 16.5 story points

- Play an online game (8 points)
- View statistics (2 points)
- Print to the log (4 points)
- Save stats (1 point)
- Select online mode (0.5 point)
- Display results (1 point)

Review and Retrospective

Iteration 1

After the first sprint, only 13.5 story points had been completed of the planned 19. The following user stories were added to the backlog for the second iteration.

- Command Line Display (4 points)
- If draw, cards to communal pile (1 point)
- Quit midgame (0.5 point)

At the retrospective, the difference between the predicted velocity and the actual velocity was attributed to a miscalculation of the story points for some of the user stories. It was also acknowledged that developers had not been working at the ideal velocity laid out in the release plan.

Going forward, it was suggested that more team scrums would improve communication, and therefore improve the velocity of the developers. A discussion was also had about any previously unidentified user stories. However, all developers agreed that there were no more user stories to be added at this stage.

Iteration 2

11.5 story points were achieved in the second sprint. This left a backlog of 10.5 story points to be completed before the final deadline. The uncompleted user stories can be seen below.

- Play an online game (8 points points)
- View statistics (2 points)
- Quit midgame (0.5 point)

This discrepancy in velocity was mainly attributed to the delay in starting the “Play an online game” user stories. This delay was caused by the prerequisite user stories remaining unfinished after the first iteration. In retrospect it was recognised that a spike should have been planned for this user story, since it could be considered an “epic” at 8 story points.

It was agreed that the contingency week would be used to finish the two user requirements “play an online game” and “view statistics”. With the final “Quit midgame” user story to be completed only if time allowed. Therefore, a velocity of 10 user points was need for this contingency sprint.

Contingency Sprint

In the final contingency sprint only one user story was left uncompleted, this was the “quit midgame” story. This was due to time constraints. It was decided that this was an acceptable user story to miss as it was only a should-have story. Furthermore, if the user requires to quit the game they could simply terminate the programme.

In the final retrospective the team reflected on what could have improved the projects progress as a whole. It was decided that the epic user story that was not dealt with using a spike was a major factor in being 10 points behind after the second iteration. Furthermore, it was suggested by several team members that several of the user stories points allocations were poorly judged.

However, overall most developers were happy with the communication within the team and believed that team meeting and retrospectives helped greatly with the overall progress on the project.

Iteration Burndown Chart

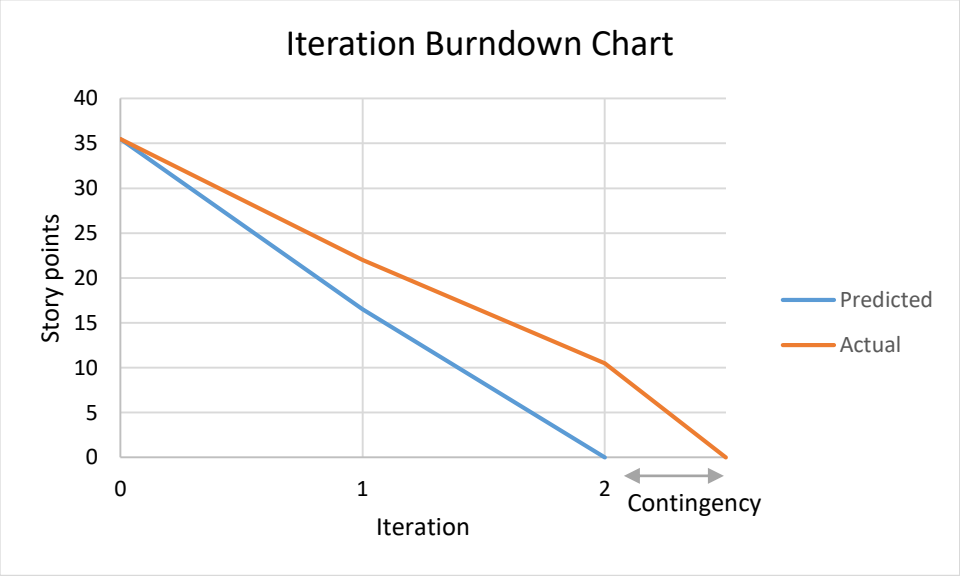


Figure 1. Iteration Burndown Chart

Assumptions

- It was assumed that draws, which may result in the final round of a game, if the round results in a draw while the remaining active players have no cards remaining, therefore leaving no clear winner, would be extremely rare and so no contingencies were put in place to handle this condition.
- It is assumed that the format of the deck file will not change in the future.
- It is assumed that the player wanting to play the online version of Top Trumps will have the latest version internet browser.
- It is assumed that the AI players number is read from TopTrumps.json.

Testing

Command Line Version

Test case	**Result**
Attempted to enter an invalid number of AI players.	If number outside given range, a message is displayed to keep the value between 1 and 4. If a String, double or punctuation is entered, a message is displayed to enter an integer.
Entering an invalid attribute when it is the players turn.	If any input does not match the attribute choices given by the program, a message is displayed to tell the user to enter a valid attribute.
When prompted the user can view statistics by pressing “s” or not view them and continue to AI selection	Statistics are displayed by pressing “s” Any other input continues the game to AI selection.
User can start a brand new game once the current game has been completed	Users can start a new game by pressing “y” Any other input terminates the program.
Users can play a game correctly with one, two, three or four AI players	The results of the games played are correct

Testlog mode is activated and printed in the correct format	The testlog file is displayed in the correct format and is correct
There is no deck file in the package	Error message is shown displaying error loading file
The deck file is not in the correct format	Error message is shown displaying error reading from the file

Web Application Version

Test case	**Result**
Users can start a new game via the home page	If the user clicks the "START A NEW GAME" button, they can play a new top trumps game in another webpage.
Users can view the overall statistics via the home page	If the user clicks the "VIEW OVERALL STATISTICS" button, they can see the overall statistics from database.
Users can return to the home page when they finishing the game.	If the user clicks the "GAME FINISHED" button when the game has a final winner, the webpage will be redirected to home page.
Users can see the current game information when they playing the game.	In the game page, when the game started, the current game information will show above the button.

Deficiencies:

1. To choose the number of AI players for the online version, the code in the JSON file would need to be changed. It is currently set to 4.

Screenshots:

1. Home page of the online version
2. Game initial page of the online version
3. Game is in progress
4. Game is finished, with a final winner
5. The overall statistics pages of the online version(without database connected)

Database Details:

Database Name: m_17_2352834c

Username: m_17_2352834c

Password: 2352834c

Database Schema:

Game(GameNo, TotalRounds, TotalDraws,Winner)

Human(GameNo, RoundsWon)

AI1(GameNo, RoundsWon)

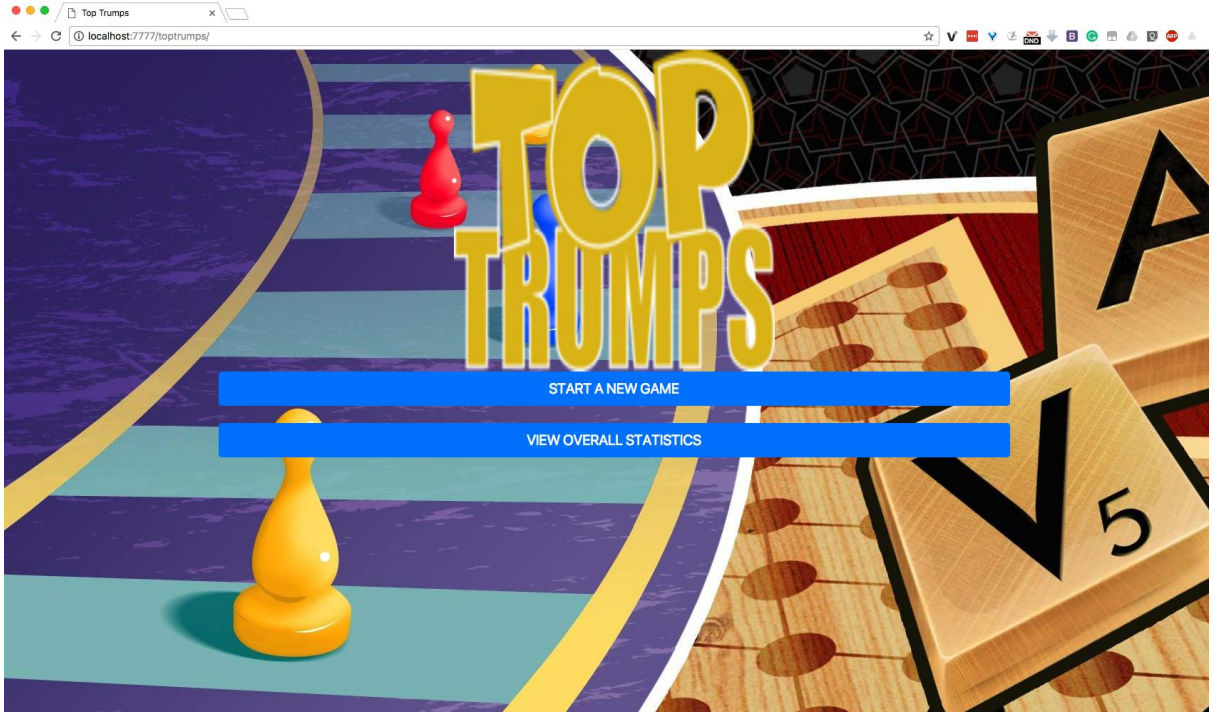
AI2(GameNo, RoundsWon)

AI3(GameNo, RoundsWon)

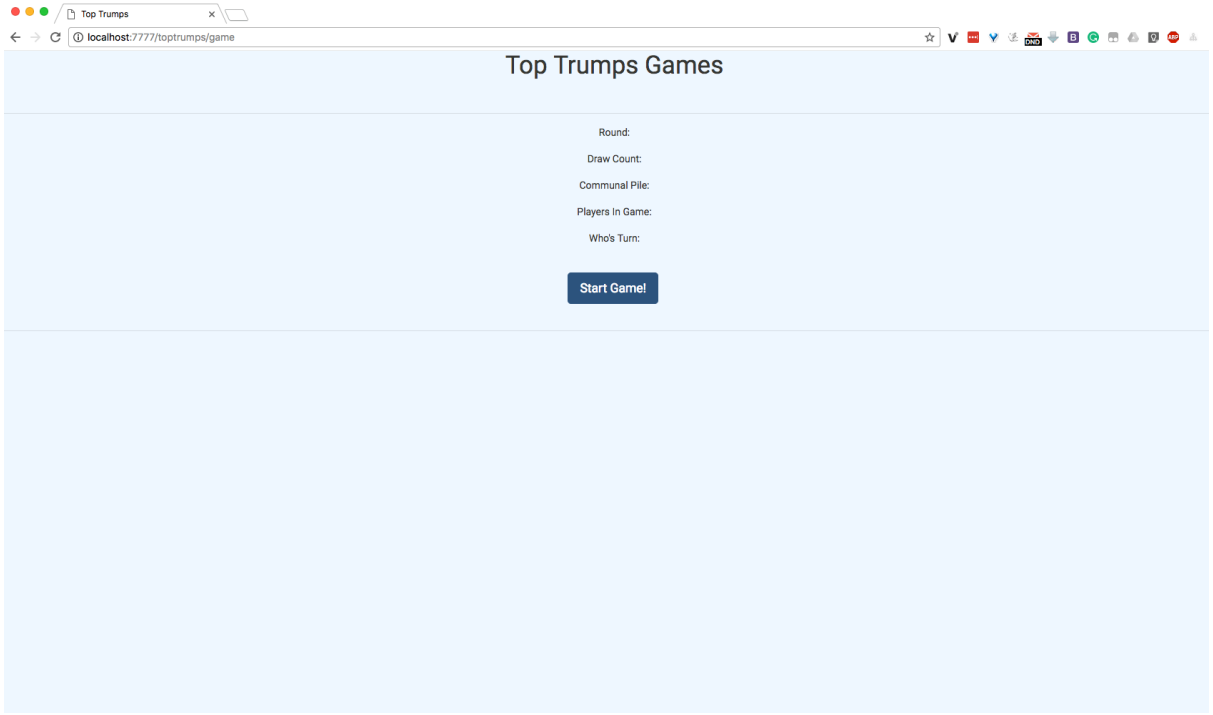
AI3(GameNo, RoundsWon)

Appendix:

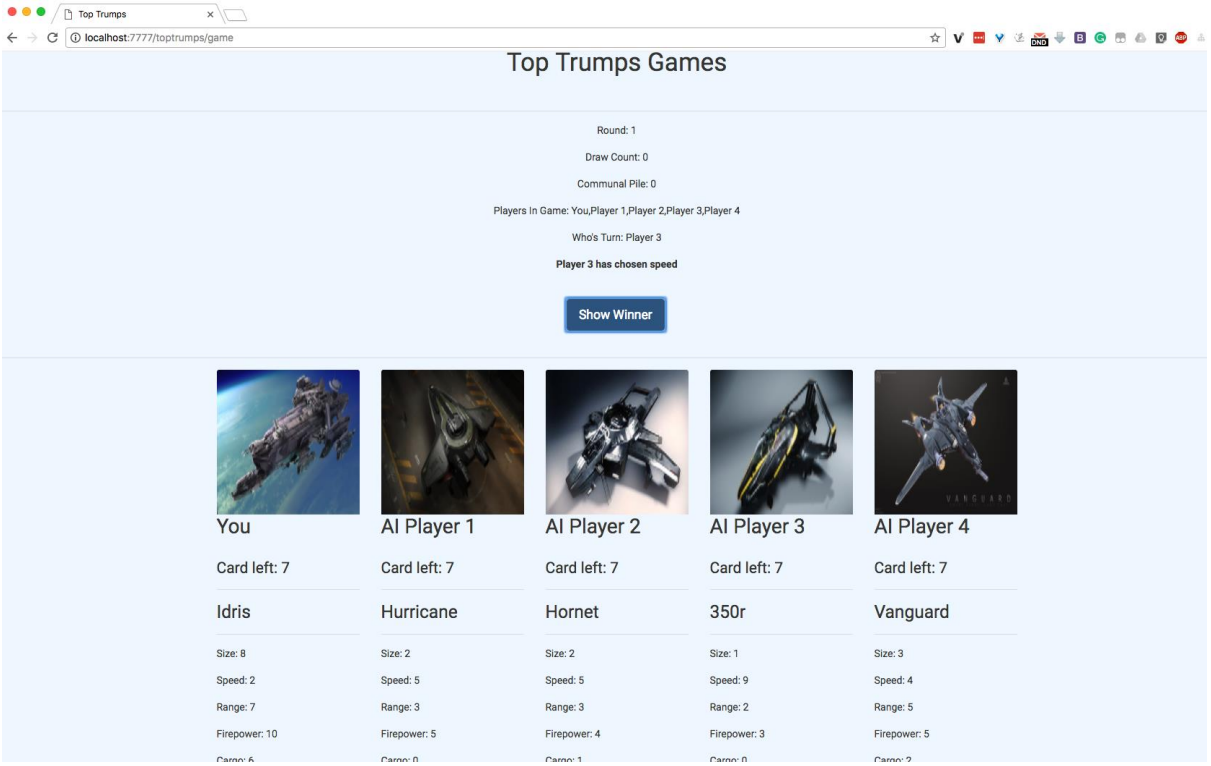
1. Home page of the online version



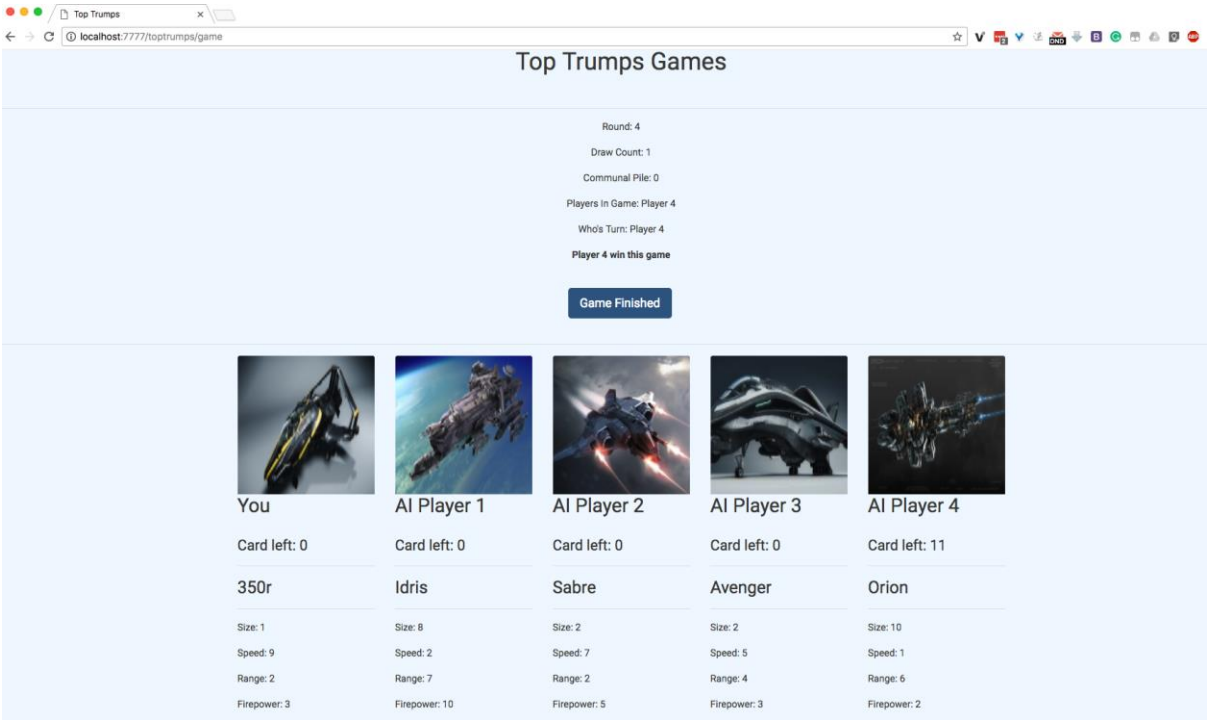
2.Game initial page of the online version



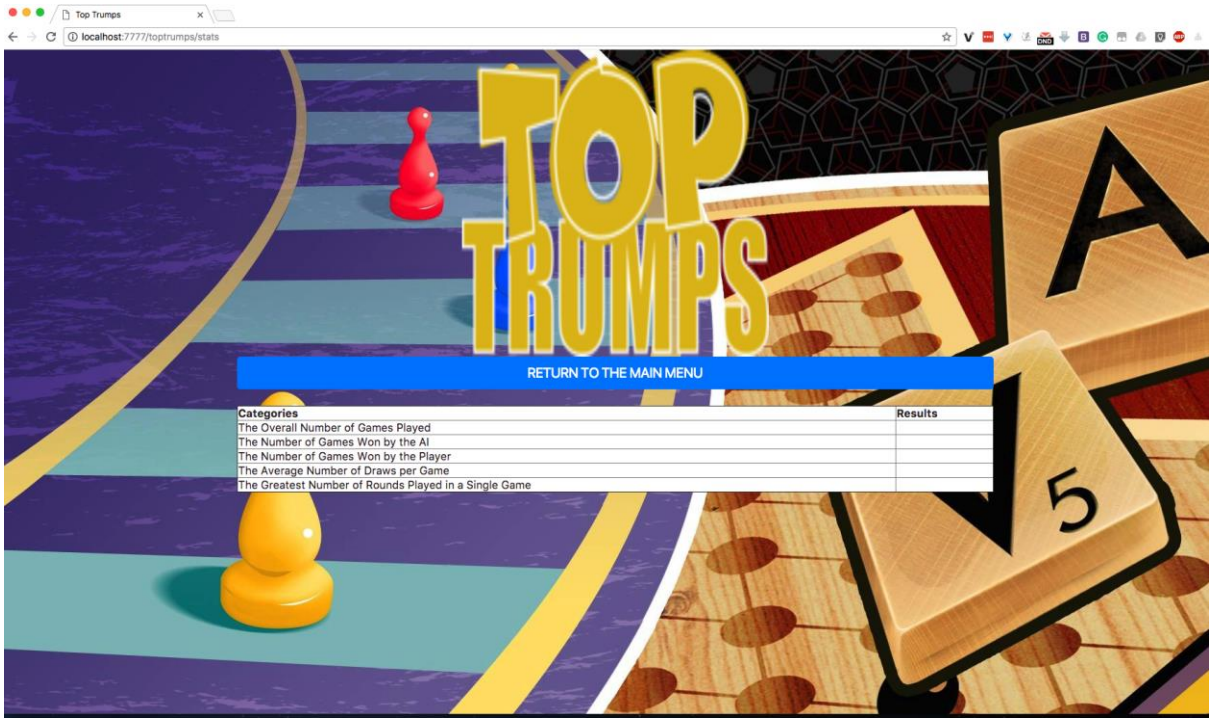
3. Game is in progress



4. Game is finished, with a final winner



5. The overall statistics pages of the online version



UML Diagram: Game Logic