TEAM KAIZEN REPORT

Top Trumps MSc Group Project

GUID:	Name:
2146919D	Florin Danila Casian
2387688L	Xiaoyu Lu
2147974P	Luca Palkovits
2411100P	Radu Petrescu
2351028T	Maaz Bin Tariq

FEBRUARY 18, 2019

TEAM KAIZEN

Top Trumps MSc Group Project

Contents

Project Summary:

- Top Trumps is a simple card game in which decks of cards are based on a theme.
- Our team used a single deck. The deck theme involves **DOTA2** Characters.
- Each deck has a list of characteristics. Our characteristics are:
    - *Attack;*
    - *Health;*
    - *Strength;*
    - *Agility;*
    - *Intelligence;*
- The aim of the program is to allow the user to play the game in both command line and online mode versus 1 to 4 other AI players.
- **The program meets all functionality and specifications conveyed in the assignment requirements document. All user stories beside would like to have’s have been covered.** More details are presented below.

User Roles

- **The User / Human Player**
  - This is the human player who uses the program on a laptop. The role is bi-partite involving game specific use as a player (example: playing the game per se) and general use as a user (example: viewing database statistics).
- **The AI player(s)**
  - This is the AI player that must make smart decisions while playing the game with the human player. The role only involves gameplay, there is no involvement in general use.
- **The Robot / Dealer**
  - The robot is simply an imaginary dealer that is responsible for anything involving card automations or game logic within the program.

User Stories[1]

#	Story	Conversations	Priority[2]	E_time[3]	A_time
1	As a user I want to be able to choose command line (cmd) gameplay mode.	Done initially via cmd flag -c. Command line processing is already present in the package.	MUST	0.2	0.1
2	As a user I want to load a deck of cards.	We assume for the main program there is only one deck of cards. There is no need for file chooser. The deck .txt file resides in the home folder of the program.	MUST	1	2
3	As a dealer I want to be able to shuffle deck of cards.	This can be done using Collections shuffle.	MUST	0.2	0.1
4	As a human or AI player I want to receive fair share of cards.	There are 40 cards. Special rule must be implemented for 3 players (40%3<>0).	MUST	0.2	0.1
5	As a human player I want to see the details of the top card.	Iterate attributes of card in System.out. Easy to implement.	MUST	0.2	0.1

[1] User story cards are not scanned as these have been brainstormed and amalgamated in Microsoft Excel initially based on the input of all team members. Cards written in *Italics red font* did not make it into the final product version. E_time means estimated time in story points. A_time is actual time in story points.
[2] For simplicity, any cmd functionality is a MUST, any database functionality is a SHOULD, any web functionality is a COULD. Everything else is a WOULD LIKE TO HAVE (hereby 'WLTH').
[3] Time is expressed in story points. **A story point involves one ideal day.** Assumption: smallest unit is 0.1 story points.

#	Story	Conversations	Priority[2]	E_time[3]	A_time
6	As a human player I want to choose card attribute.	Choose an attribute by number input through System.in.	MUST	0.2	0.1
7	As an AI player I want to choose best option for top card.	From an array list of attributes choose max. Could implement hash set, maybe.	MUST	0.2	0.1
8	As a dealer I want to choose random first player.	Can use random integer from 0 to 4 or use Collections.shuffle.	MUST	0.2	0.1
9	As a human player I want to see relevant game data.	Printed in System.out. The data should involve a toString() method involving explanatory text as well. Relevant game data includes: round number, active player, card chosen, card attributes, chosen card and the human player's remaining card count.	MUST	5	6
10	As a dealer I want to determine who wins each round.	Core game logic can be implemented in the controller.	MUST	1	1
11	As a dealer I want to determine who wins the game.	There should be a method that keeps checking if the game is terminated via a loop determined by a Boolean value.	MUST	0.2	0.1
12	As a dealer I want to keep track of the common pile.	Game logic should keep track of common pile whenever a round ends in a draw.	MUST	0.2	0.1
13	As a human player I want to see who wins the round, the game and details of the common pile.	Can be implemented via the view in cmd mode. After each action and at the end of the game System.out prints the relevant outcomes.	MUST	0.2	0.1
14	As a human or AI player I want to start the next round if I won the previous one.	Use a if(winning_player==true) construct perhaps.	MUST	0.2	0.1
15	As a human or AI player I want to receive or lose cards in/from my pack, if I win or lose, respectively.	Add() or remove() methods can be used. Array can be used to store the cards that are in play each round. These are subsequently appended to the winner's pack.	MUST	0.2	0.1

#	Story	Conversations	Priority[2]	E_time[3]	A_time
16	As a human or AI player I want to append the cards that I won to the back of my pack.	Array list could be used with an add() method.	MUST	0.2	0.1
17	As a dealer I want to make sure that once a player is eliminated, they do not participate in any more rounds.	Each player has a Boolean variable initiated with true allowing them to participate in rounds. When eliminated, variable should change to false.	MUST	0.2	0.1
18	As a human player I want to choose number of AI opponents.	System.in with an integer input from 1 to 4.	MUST	0.2	0.1
19	As a user I want to receive an output of gameplay details.	Test log. File Writer is used to generate .txt file with output similar to the cmd view class.	SHOULD	1	1.5
20	As a user I want to store specific persistent game data at the end of the game.	Postgresql jar used – a Connection class based on the jar can be used to manage all database connections and transactions.	SHOULD	2	1.5
21	As a user I want to retrieve persistent data on past games.	Connection class is used. Specific cmd flag is used upon program startup. Select and to string can be used to retrieve data in console or web view.	SHOULD	2	1.5
22	As a user I want to choose between cmd and online mode.	Done using cmd flags as per initial project template (Moodle).	COULD	0.5	0.2
23	As a user I want to have functional “Start Game” and “Show Statistics” buttons for online mode.	2 clickable buttons in browser.	COULD	0.5	0.3
24	As a user I want to have a button framework for choosing number of AI players online.	Could be either clickable buttons or combo-box.	COULD	0.5	1
25	As a user I want to see card graphics for each player.	Card static images used. Card could be a picture or an iteration of attributes combined with a picture.	COULD	3	1

#	Story	Conversations	Priority[2]	E_time[3]	A_time
26	As a user I want to have buttons for choosing attributes online.	Could be either clickable buttons or combo-box.	COULD	3	1
27	As a user I want to have a 'Next Round' button online.	Single button.	COULD	0.5	.5
28	As a user I want to see text representation for round and game outcomes, as well as database statistics.	Section of screen displaying text (simple html implementation).	COULD	3	11
29	*As a user I would like to choose a relevant deck from command line or online mode.*	*Could have more decks, graphical assets and choice option in cmd (Scanner/FileChooser) or online (ComboBox).*	*WLTH*	*2*	*0**
30	*As a user I would like to choose between AI difficulties.*	*Easy is for random attribute choice. Medium is for best attribute choice. Hard is for smart distribution-knowledge attribute choice (example: AI chooses most probable statistical win based on overall attribute distribution).*	*WLTH*	*2*	*0**

*Not implemented in final product.

Planning and review reports

Sprints Overview

Sprint #	Week #	Timeframe	Ideal days per week*
1	1	7 – 13 Jan	5
	2	14 – 20 Jan	5
	3	21 – 27 Jan	5
2	4	28 – 3 Feb	5
	5	4 – 10 Feb	5
	6	11 – 17 Feb	5

* One story point corresponds to an ideal day.

Planning and review report Sprint 1

General

Sprint Dates:	7 January 2019 – 27 January 2019
Sprint ideal days/story point capacity:	15 points
Planned velocity:	15 points
Actual velocity:	15 points

Team Info

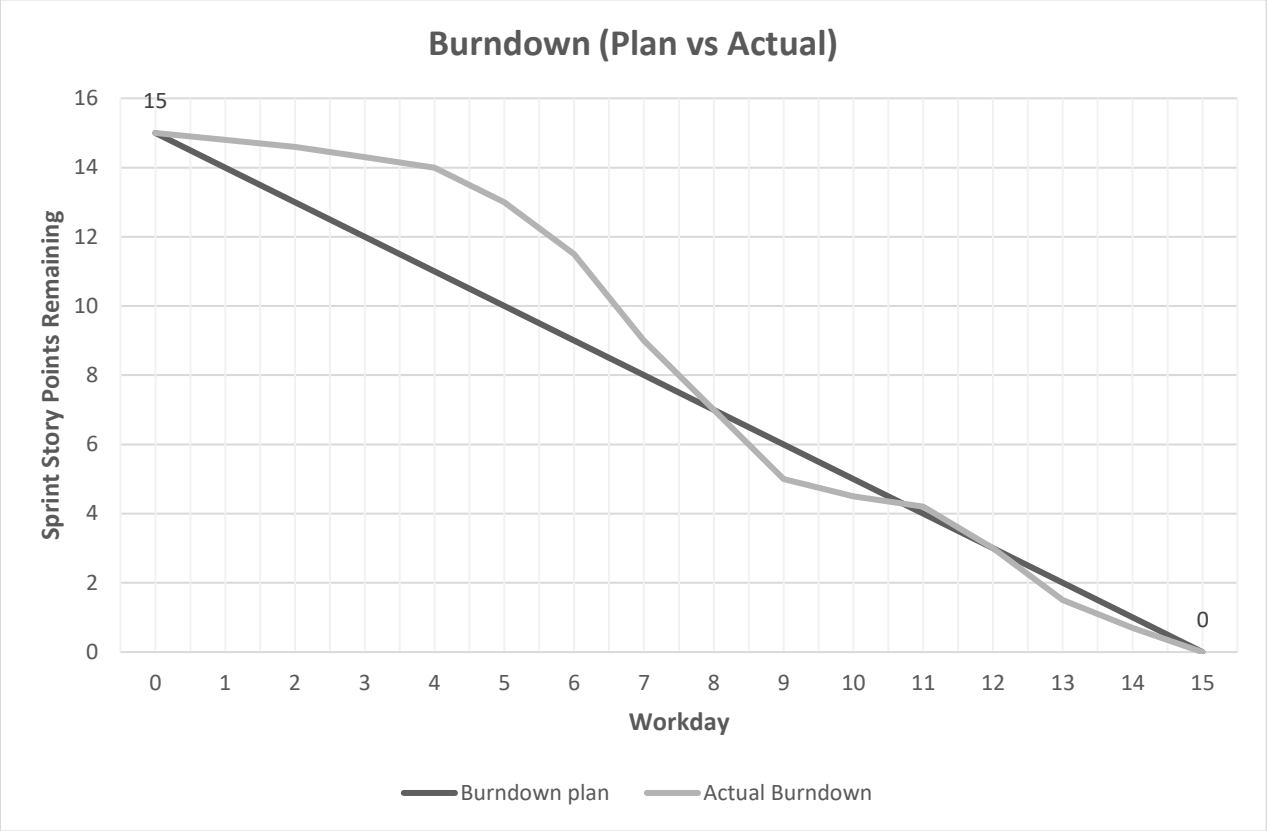
Role*	Name	Effort Proportion
Java Developer and Tester	Casian	20%
Java and JavaScript Developer	Xiaoyu	20%
Java Developer and Tester	Luca	20%
Integration Developer and Database Administrator **	Radu	20%
Java and JavaScript Developer	Maaz	20%

*Documentation and research were done as required by all team members.
**Planning Manager and Scrum Master.

Completion and timing*

User Story #	Complete	Story points (planned)	Story points (actual)*
1	Yes	0.2	0.1
2	Yes	1	2
3	Yes	0.2	0.1
4	Yes	0.2	0.1
5	Yes	0.2	0.1
6	Yes	0.2	0.1
7	Yes	0.2	0.1
8	Yes	0.2	0.1
9	Yes	5	6
10	Yes	1	1
11	Yes	0.2	0.1
12	Yes	0.2	0.1
13	Yes	0.2	0.1
14	Yes	0.2	0.1
15	Yes	0.2	0.1
16	Yes	0.2	0.1
17	Yes	0.2	0.1
18	Yes	0.2	0.1
19	Yes	1	1.5
20	Yes	2	1.5
21	Yes	2	1.5

*Working on a user story involves all stages: research, development, testing and documentation. At the end of each working day each member of the team was asked how much they contributed and an estimate was made.



Project Backlog after Sprint*

#	Type	Status	Estimate	Spent
1	User Story	Done	0.2	0.1
2	User Story	Done	1	2
3	User Story	Done	0.2	0.1
4	User Story	Done	0.2	0.1
5	User Story	Done	0.2	0.1
6	User Story	Done	0.2	0.1
7	User Story	Done	0.2	0.1
8	User Story	Done	0.2	0.1
9	User Story	Done	5	6
10	User Story	Done	1	1
11	User Story	Done	0.2	0.1
12	User Story	Done	0.2	0.1
13	User Story	Done	0.2	0.1

#	Type	Status	Estimate	Spent
14	User Story	Done	0.2	0.1
15	User Story	Done	0.2	0.1
16	User Story	Done	0.2	0.1
17	User Story	Done	0.2	0.1
18	User Story	Done	0.2	0.1
19	User Story	Done	1	1.5
20	User Story	Done	2	1.5
21	User Story	Done	2	1.5
22	User Story	In Progress	0.5	0
23	User Story	In Progress	0.5	0
24	User Story	In Progress	0.5	0
25	User Story	In Progress	3	0
26	User Story	In Progress	3	0
27	User Story	In Progress	0.5	0
28	User Story	In Progress	3	0
29	User Story	In Progress	2	0
30	User Story	In Progress	2	0

* No new items have been added as a result of the sprint. All existing items remain on backlog.

Planning and review report Sprint 2

General

Sprint Dates:	28 January 2019 – 17 February 2019
Sprint ideal days/story point capacity:	15 points
Planned velocity:	15 points
Actual velocity:	11 points

Team Info

Role*	Name	Effort Proportion
Java Developer and Tester	Casian	20%
Java and JavaScript Developer	Xiaoyu	20%
Java Developer and Tester	Luca	20%
Integration Developer and Database Administrator **	Radu	20%
Java and JavaScript Developer	Maaz	20%

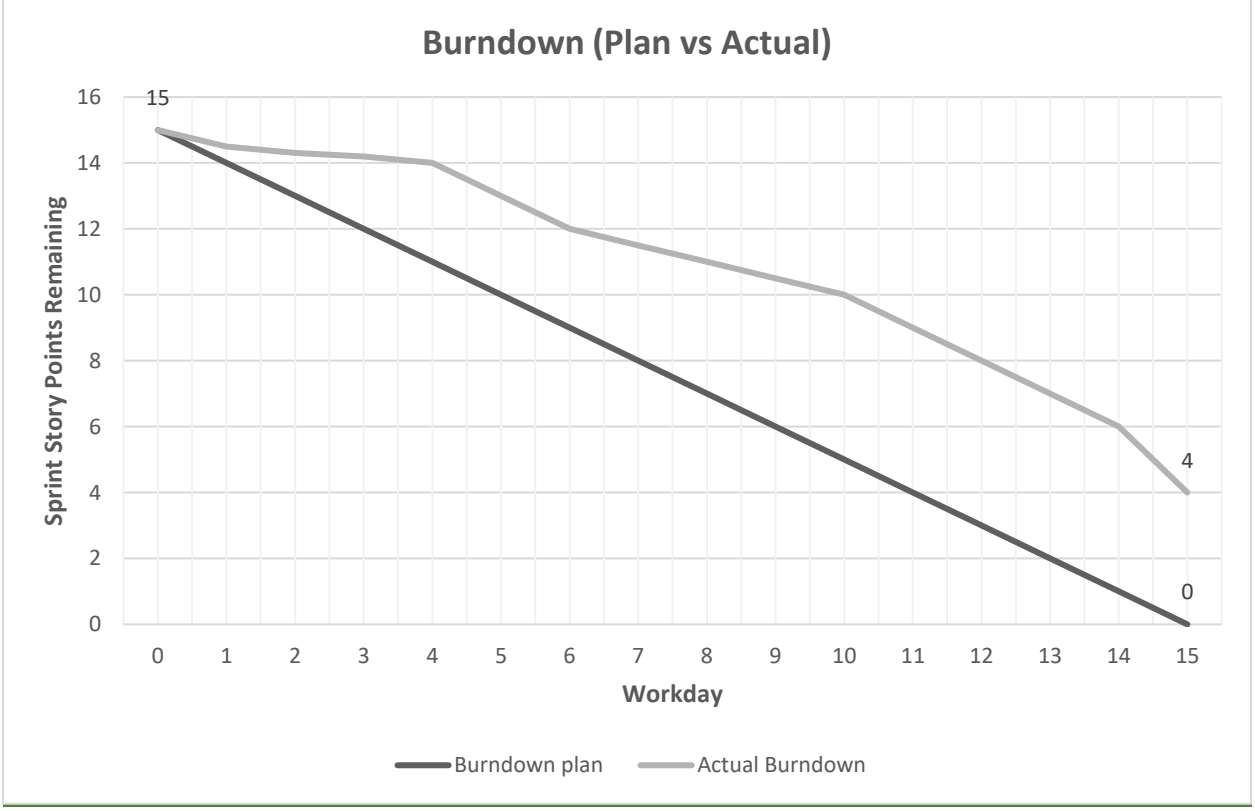
*Documentation and research were done as required by all team members.

**Planning Manager and Scrum Master.

Completion and timing*

User Story #	Complete	Story points (planned)	Story points (actual)*
22	Yes	0.5	0.2
23	Yes	0.5	0.3
24	Yes	0.5	1
25	Yes	3	1
26	Yes	3	1
27	Yes	0.5	.5
28	Yes	3	11
29	No (not implemented)	2	0
30	No (not implemented)	2	0

*Working on a user story involves all stages: research, development, testing and documentation. At the end of each working day each member of the team was asked how much they contributed and an estimate was made.



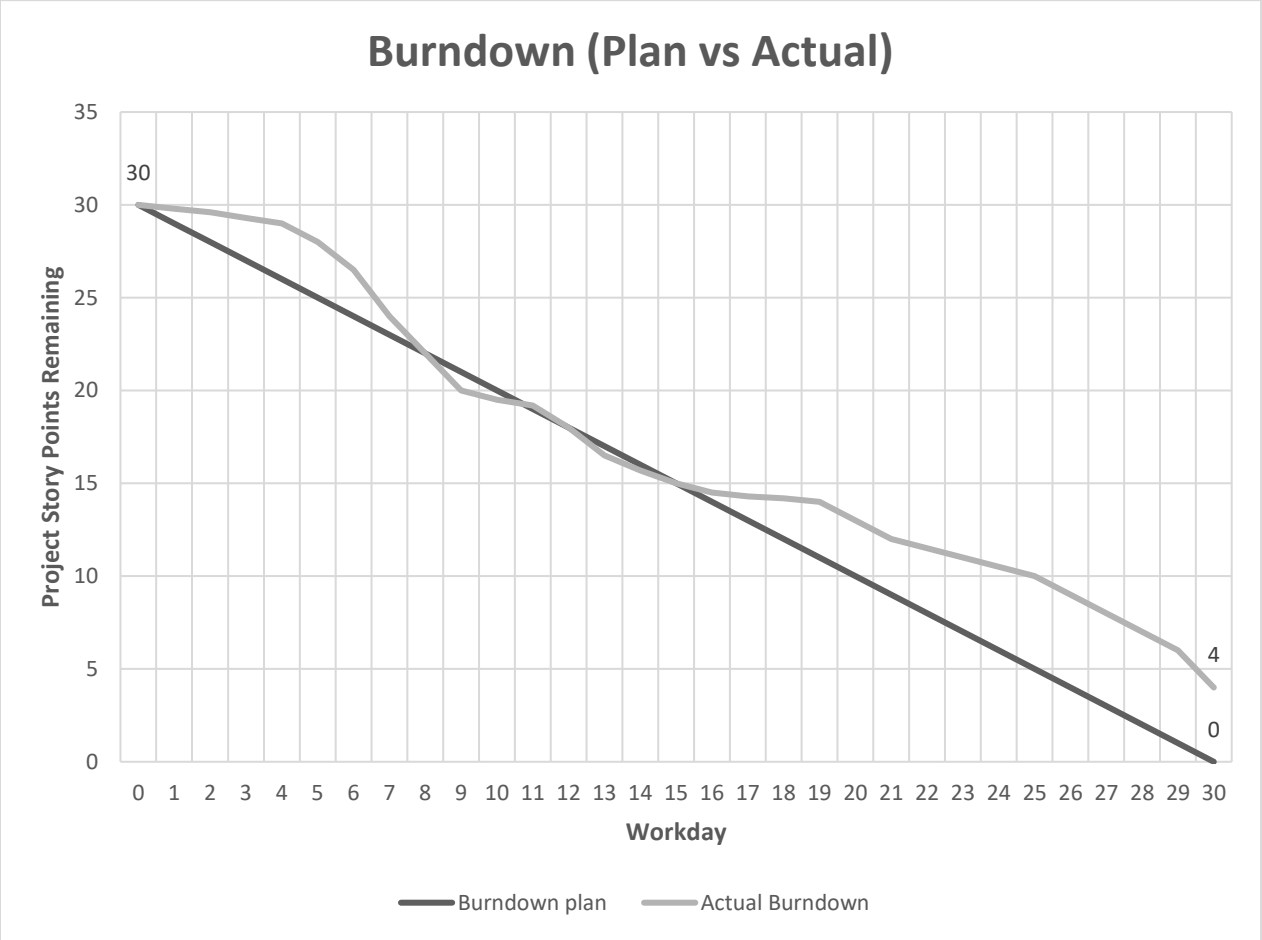
Project Backlog after Sprint*

#	Type	Status	Estimate	Spent
1	User Story	Done	0.2	0.1
2	User Story	Done	1	2

#	Type	Status	Estimate	Spent
3	User Story	Done	0.2	0.1
4	User Story	Done	0.2	0.1
5	User Story	Done	0.2	0.1
6	User Story	Done	0.2	0.1
7	User Story	Done	0.2	0.1
8	User Story	Done	0.2	0.1
9	User Story	Done	5	6
10	User Story	Done	1	1
11	User Story	Done	0.2	0.1
12	User Story	Done	0.2	0.1
13	User Story	Done	0.2	0.1
14	User Story	Done	0.2	0.1
15	User Story	Done	0.2	0.1
16	User Story	Done	0.2	0.1
17	User Story	Done	0.2	0.1
18	User Story	Done	0.2	0.1
19	User Story	Done	1	1.5
20	User Story	Done	2	1.5
21	User Story	Done	2	1.5
22	User Story	Done	0.5	0.2
23	User Story	Done	0.5	0.3
24	User Story	Done	0.5	1
25	User Story	Done	3	1
26	User Story	Done	3	1
27	User Story	Done	0.5	.5
28	User Story	Done	3	11
29	*User Story*	*Removed*	*2*	*0*
30	*User Story*	*Removed*	*2*	*0*

* No new items have been added as a result of the sprint. *The highlighted items have been removed from the scope of the project (story #s 29 and 30).*

Project Burndown Chart



Conclusion:

All user stories with priorities above "Would Like To Have"(WLTH) have been completed. The 4 point discrepancy at the end of the project (workday 30) represent user stories 29 and 30 – these are the WLTHs that have been removed from the scope of the project.

Assumptions

- There should be one human player and up to 4 computer players (AIs).
- Assume game has at least 2 players. Human player cannot therefore play alone as game logic would not make sense.
- If a deck does not divide equally between the players, then some players may have less cards. For example, if there are 3 players and 40 cards, then two players receive 13 cards and one player receives 14.

- A deck has only 5 criteria and the criteria are always positive integers between 1 and 50 (inclusive).
- A higher number is always better for any given characteristic.
- There are 40 cards in a DotaDeck.txt file.
- The first player should be selected at random.
- A draw won't continue until the point where there are only cards in the communal pile – there is no need to deal with this from a programming perspective.
- There is only one operational deck of cards present in the project folder.
- Assume “You” in cmd refers to human player.
- Assume cards go to the back of the winning deck.
- Collections.shuffle() is a good enough shuffle for the purpose of the cards and can therefore be implemented.
- Assume any user (human or AI) can only see and interact with the top card of the deck. Players cannot therefore choose between cards.
- Assume AI is smart and chooses the best numerical option.
- Assume fast forwarding the GUI (cmd or otherwise) is appropriate once human player has lost and has no more choices to make. This avoids the potential tedious nature of the game.
- Assume the round winner starts next round.
- In the event of a round draw, all cards go into the common pile. As soon as the following round ends in a win, the winning player gets the cards from the current round as well as any from the common pile (also referred to as the communal pile).
- Assume cards are always appended to the back of the pack.
- An eliminated player (human or AI) cannot participate in further rounds in the current game.
- Assume there is no multiplayer – multiple players in the game.
- Assume between 1 and 4 AI players.
- Assume no JUnit is needed for documentation.
- Assume card names are unique.
- Assume card pictures are appropriate representations for the online GUI.
- Assume the player does not refresh or close the web page during online mode. Only buttons are used. Failure to do so may result in unpredictable behaviour.

- Assume the following general outline/wireframe is sufficient from a user experience perspective concerning the online game mode:

<table>
<tr><td>URL:</td><td colspan="3">http://...topTrumps/game...</td></tr>
<tr><td colspan="4">Header
Game logic text 1</td></tr>
<tr><td rowspan="2">Game buttons</td><td>Card 1</td><td>Card 2</td><td>Card 3</td></tr>
<tr><td>Card 4</td><td>Card 5</td><td>Game logic text 2</td></tr>
</table>

- Assume fading effect is sufficient to highlight a card in the online mode GUI.

Testing

Testing has involved manual testing, inspecting the GameLog.txt (-t flag) file versus actual game running and JUnit testing (relevant classes are TTModelTest and TTControllerTest in the test folder). Testing was done extensively while coding as well as after each version of the software artifact was produced. The team found it helpful to structure the tests based on the user stories involved.

The following is a summary of all the tests undertaken for the project.

Story	Test[4]	Appendix
As a user I want to be able to choose cmd gameplay mode.	Manual test passed - Printscreen of actual functionality.	**1.1**
As a user I want to load a deck of cards.	Junit test passed - Class: TTModelTest – method: testCardsRead(). Since we assume there is only one deck, no further manual testing with dummy .txt files is needed.	**1.2**
As a dealer I want to be able to shuffle deck of cards.	Manual testing passed – print screen from test mode to check if the deck is shuffled.	**1.3**
As a human or AI player I want to receive fair share of cards.	Junit test passed - Class: TTModelTest – method: testCardsDistributed(). The method checks if the cards are distributed equally among players and if the rule concerning 3 players applies.	**1.4**
As a human player I want to see the details of the top card.	Manual test passed – Printscreen of top card in cmd and online mode.	**1.5**
As a human player I want to choose option for top card.	Manual test passed – Printscreen of top card attribute chosen in cmd and online mode.	**1.6**
As an AI player I want to choose best option for top card.	Junit test passed - Class: TTControllerTest – method: playRoundTest(). The test checks if the AI player chooses the highest attribute for the card. This has also been tested in cmd and online mode in repeated practices.	**1.7**
As a dealer I want to choose random first player.	Manual test passed – Printscreen of first player chosen randomly in both cmd and online mode. In the print screen in the Appendix below, AIPlayer4 is randomly chosen as the first player.	**1.8**
As a human player I want to see relevant game data.	Manual test passed – all relevant game data is displayed as per requirements in both cmd and online mode. Repeated manual tests have been made with a checklist of the functionality before the tester. The checklist has been repeatedly iterated over.	**1.9**
As a dealer I want to determine who wins each round.	Junit test passed - Class: TTControllerTest – method: playRoundTest(). This involves general round functionality. Manual testing has also been performed for cmd and online mode.	**1.7**
As a dealer I want to determine who wins the game.	Junit test passed - Class: TTControllerTest – method: playRoundTest(). This involves general game functionality. Manual testing has also been performed for cmd and online mode.	**1.7**

[4] All JUnit tests can be found in the test directory. The relevant classes are TTModelTest and TTControllerTest. The test methods within these are connected and therefore some functionality may be tested by more than one method.

Story	Test[4]	Appendix
As a dealer I want to keep track of the common pile.	Junit test passed - Class: TTControllerTest – method: playRoundTest().This involves general game functionality. Manual testing has also been performed for cmd and online mode.	**1.7**
As a human player I want to see who wins the round, the game and details of the common pile.	Manual test passed – Print screens have been provided of the functionality for the round, game and draw situation for both cmd and online modes.	**1.10**
As a human or AI player I want to start the next round if I won the previous one.	Manual test passed - Printscreen in cmd and online mode presented in appendix.	**1.11**
As a human or AI player I want to receive or lose cards in/from my pack, if I win or lose, respectively.	Manual test passed - Printscreen in cmd and online mode presented in appendix.	**1.11**
As a human or AI player I want to append the cards that I won to the back of my pack.	Junit test passed - Class: TTControllerTest – method: playRoundTest(). This involves general round functionality. Manual testing has also been performed for cmd and online mode.	**1.7**
As a dealer I want to make sure that once a player is eliminated, they do not participate in any more rounds.	Manual testing passed – Printscreens provided for both cmd and online mode.	**1.12**
As a human player I want to choose number of AI opponents.	Junit test passed - Class: TTModelTest – method: testNumberOfAIsChoice(). Simply checks if the number of AI's has been correctly chosen. This has also been tested manually for both cmd and online mode.	**1.13**
As a user I want to receive an output of gameplay details.	Manual testing passed – printscreen of test mode file provided. This has been checked against cmd.	**1.14**
As a user I want to store specific persistent game data at the end of the game.	Manual testing passed. Database select has been performed after game has been played on university Yacata server.	**1.15**
As a user I want to retrieve persistent data on past games.	Manual testing passed. Beside the sql presented under appendix 1.15, printscreens of retrieved persistent data have been obtained in cmd and online mode. (duplicates in printscreens are due to some Yacata testing – can be ignored)	**1.16**
As a user I want to choose between cmd and online mode.	This has been manually checked via -c and -o flags as well as in most print screens above.	**n/a**
As a user I want to have functional "Start Game" and "Show Statistics" buttons for online mode.	Manual testing passed – printscreens under appendix 1.16.	**1.16**
As a user I want to have a button framework for choosing number of AI players online.	Manual testing passed – printscreens under appendix 1.13.	**1.13**

Story	Test[4]	Appendix
As a user I want to see card graphics for each player.	Manual testing passed – all online mode printscreens contain card graphics.	**1.13**
As a user I want to have buttons for choosing attributes online.	Manual testing passed – buttons present, printscreens provided.	**1.11**
As a user I want to have next round button online.	Manual testing passed – buttons present, printscreens provided.	**1.11**
As a user I want to see text representation for round and game outcomes, as well as database statistics.	Manual testing passed – all details under specifications are present. Printscreens have been provided.	**1.10**

Deficiencies

- No missing functionality has been identified. The program works as expected.
- There is no incorrect behaviour.
- The following potential areas of improvement have been identified:
    - Accessing the /game page in online mode without first going to the topTrumps/ page does not work as no cards are initialized and therefore displayed. This is expected behaviour, but perhaps a default session could solve this.
    - Using an incognito tab in Chrome is important in case preset settings are used from an earlier version of the code. This has happened numerous time when using Chrome without incognito mode.
    - It is very statistically improbable to get a draw due to a the structure of the deck. This is in expected – any attributes may be modified.
    - Refreshing goes back one round on the /game page in online mode – we assume user does not refresh. Perhaps some cookies could be used to improve this and save session/instance details.

Appendix 1 – Screenshots Main Functionality & Testing

The following is a representation of the normal functionality of the game:

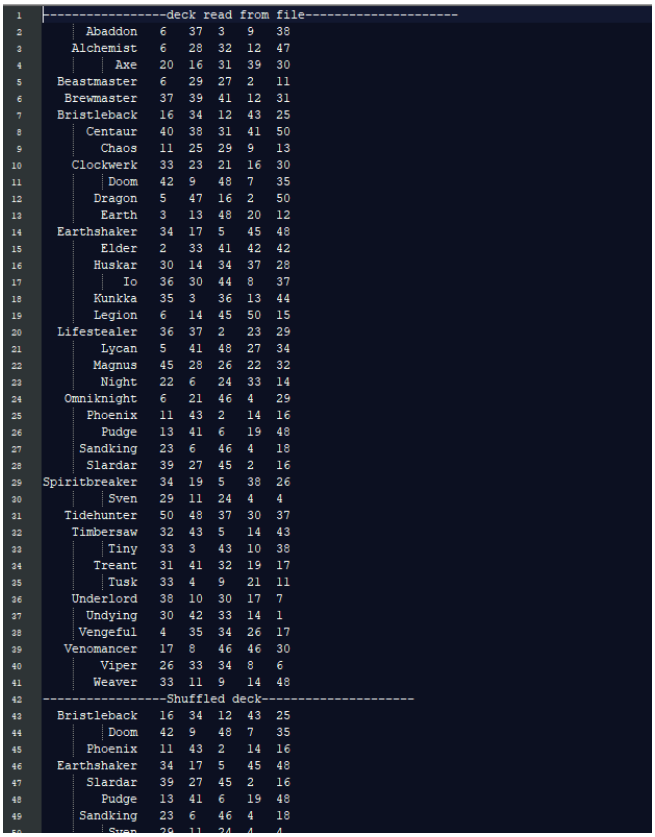
1. Command line mode – get statistics view:

```
Jar found.
Controlling your database.
Game_id is: 44

Total games:   43
Computer wins: 34
Human wins:    9
Average draws: 4
Max rounds:    219

Do you want to see past results or play a game?
   1: Print Game Statistics
   2: Play game
   0: Quit
```

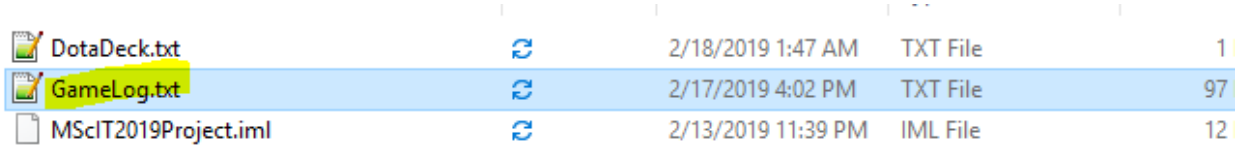
2. Command line mode – play game with 4 AI players, as chosen by human player:

```
   > Agility: 19
   > Intelligence: 17
You now have 2 card(s) in your deck.
Round 7
Round 7: Players have drawn their cards
AIPlayer1 Picked characteristic 1: Attack.
AIPlayer2          Dragon   5   47  16  2   50
AIPlayer3          Legion   6   14  45  50  15
AIPlayer1       Underlord   38  10  30  17  7
AIPlayer4         Phoenix   11  43  2   14  16
You         Treant   31  41  32  19  17
Round7: AIPlayer1 won this round!
The winning card was 'Underlord':
   > Attack: 38 <--
   > Health: 10
   > Strength: 30
   > Agility: 17
   > Intelligence: 7
1
You Drew 'Earthshaker'
   > Attack: 34
   > Health: 17
   > Strength: 5
   > Agility: 45
   > Intelligence: 48
You now have 1 card(s) in your deck.
Round 8
Round 8: Players have drawn their cards
AIPlayer1 Picked characteristic 1: Attack.
AIPlayer2         Undying   30  42  33  14  1
AIPlayer3     Lifestealer   36  37  2   23  29
AIPlayer1            Sven   29  11  24  4   4
AIPlayer4       Alchemist   6   28  32  12  47
You   Earthshaker   34  17  5   45  48
Round8: AIPlayer3 won this round!
The winning card was 'Lifestealer':
   > Attack: 36 <--
   > Health: 37
   > Strength: 2
   > Agility: 23
   > Intelligence: 29
1
You lose!
```

3. Command line mode – test log (actual .txt):

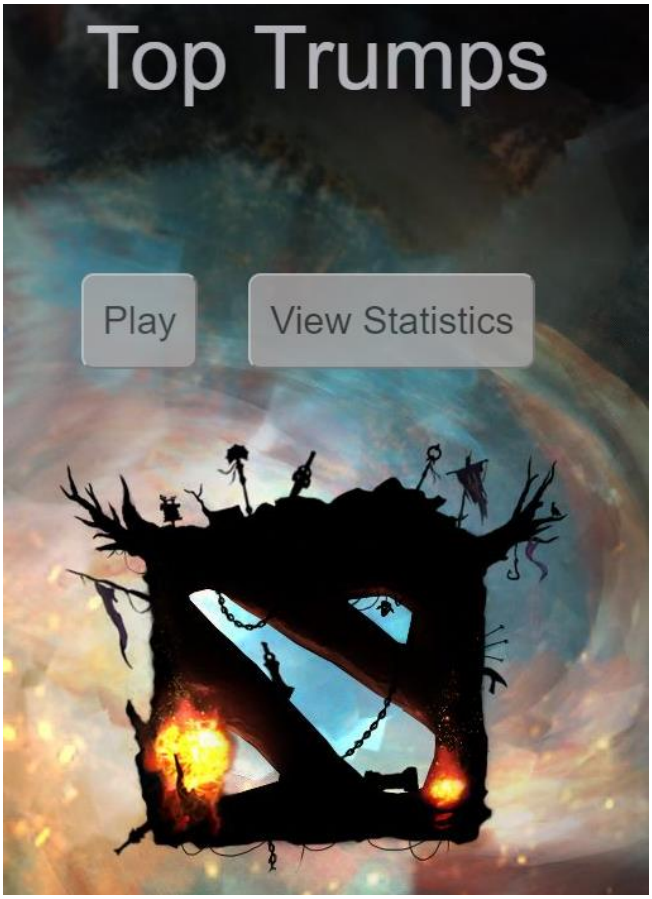


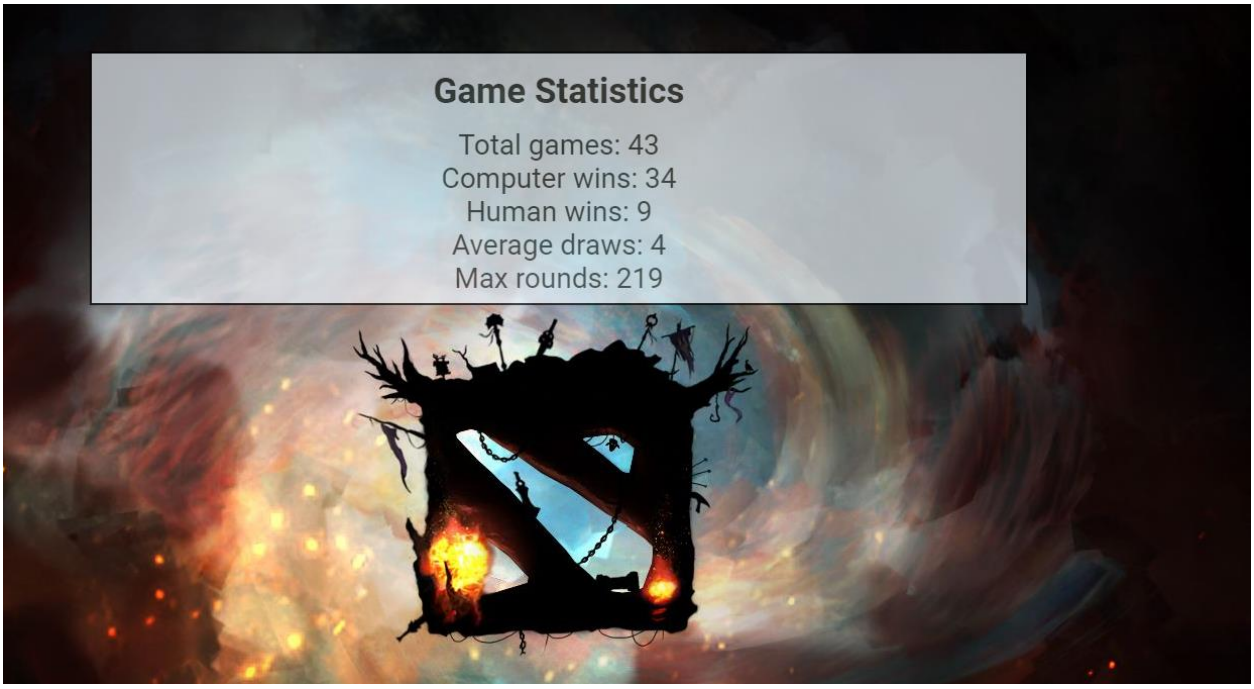
```
----------deck read from file----------
     Abaddon   6  37  3  9  38
   Alchemist   6  28  32  12  47
        Axe   20  16  31  39  30
  Beastmaster   6  29  27  2  11
   Brewmaster   37  39  41  12  31
  Bristleback   16  34  12  43  25
     Centaur   40  38  31  41  50
       Chaos   11  25  29  9  13
   Clockwerk   33  23  21  16  30
        Doom   42  9  48  7  35
      Dragon   5  47  16  2  50
       Earth   3  13  48  20  12
  Earthshaker   34  17  5  45  48
       Elder   2  33  41  42  42
      Huskar   30  14  34  37  28
          Io   36  30  44  8  37
      Kunkka   35  3  36  13  44
      Legion   6  14  45  50  15
  Lifestealer   36  37  2  23  29
       Lycan   5  41  48  27  34
      Magnus   45  28  26  22  32
       Night   22  6  24  33  14
   Omniknight   6  21  46  4  29
     Phoenix   11  43  2  14  16
       Pudge   13  41  6  19  48
    Sandking   23  6  46  4  18
     Slardar   39  27  45  2  16
Spiritbreaker   34  19  5  38  26
       Sven   29  11  24  4  4
   Tidehunter   50  48  37  30  37
    Timbersaw   32  43  5  14  43
        Tiny   33  3  43  10  38
      Treant   31  41  32  19  17
       Tusk   33  4  9  21  11
   Underlord   38  10  30  17  7
     Undying   30  42  33  14  1
    Vengeful   4  35  34  26  17
  Venomancer   17  8  46  46  30
       Viper   26  33  34  8  6
      Weaver   33  11  9  14  48
----------Shuffled deck----------
  Bristleback   16  34  12  43  25
        Doom   42  9  48  7  35
     Phoenix   11  43  2  14  16
  Earthshaker   34  17  5  45  48
     Slardar   39  27  45  2  16
       Pudge   13  41  6  19  48
    Sandking   23  6  46  4  18
```



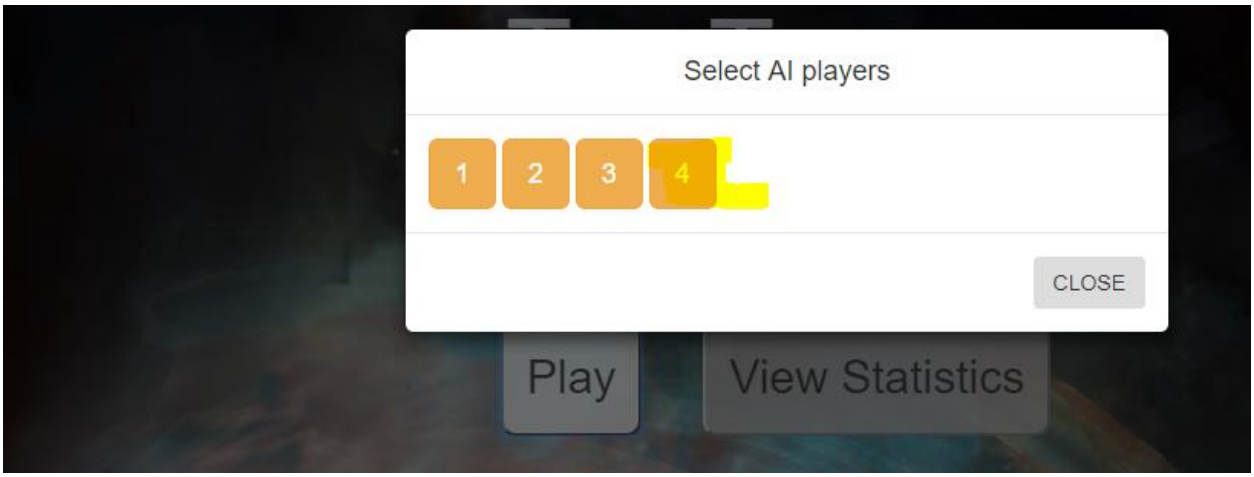
DotaDeck.txt		2/18/2019 1:47 AM	TXT File	1
GameLog.txt		2/17/2019 4:02 PM	TXT File	97
MScIT2019Project.iml		2/13/2019 11:39 PM	IML File	12

4. Online mode – get statistics view:



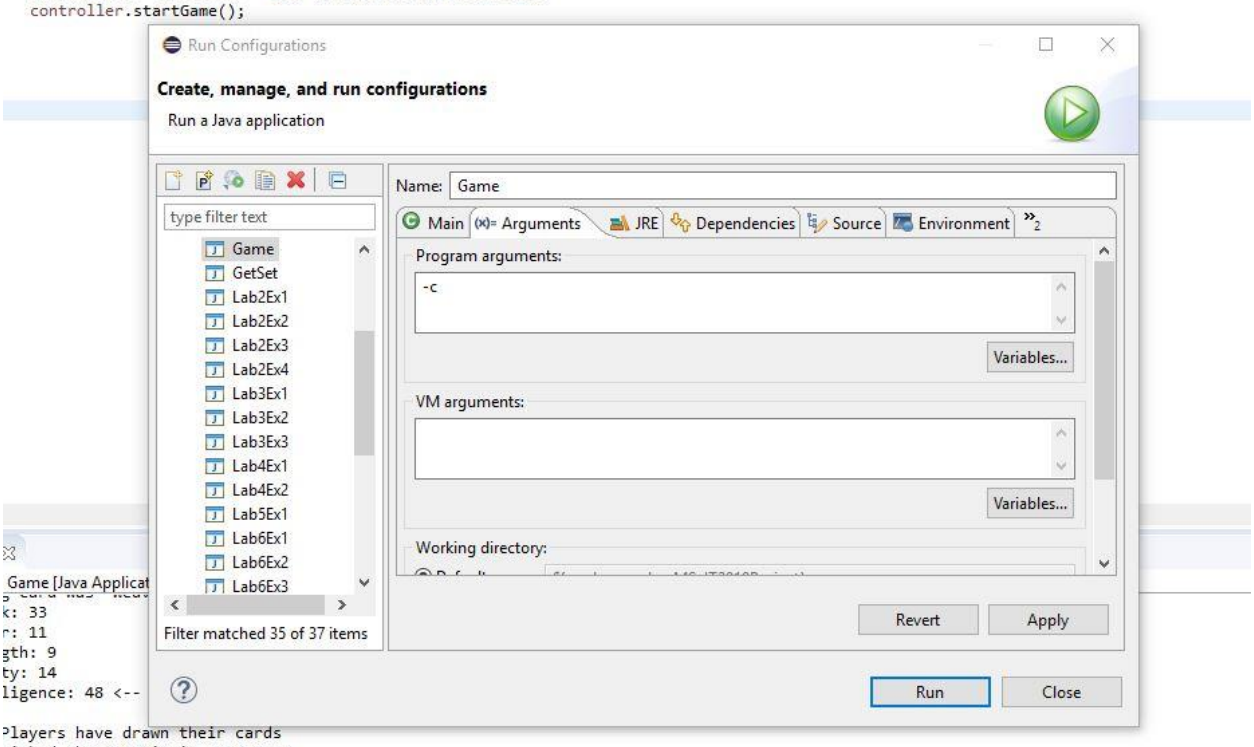


5. Online mode – play game:



1.1 As a user I want to be able to choose cmd gameplay mode.

```
TTModel model = new TTModel();
TTView view = new TTView(model);
TTController controller = new TTController(model,view);
controller.startGame();
```




```
Players have drawn their cards
Picked characteristic 2: Armour.
```

```
9          TTModel model = new TTModel();
10         TTView view = new TTView(model);
11         TTController controller = new TTController(model,view);
12         controller.startGame();
13
14     }
15
16
17
18
19 }
20
```

Console

```
Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 13:42:29)

Enter the number of AI players
```

1.2 As a user I want to load a deck of cards.

```
@Test
public void testCardsRead() {

    // testModel.readCards( new File("DotaDeck.txt"));
    assertEquals( expected: false, testModel.getCards().isEmpty(), message: "Check that Card objects are create
    assertEquals( expected: 6, testModel.getAttributeNames().length,
            message: "Check that method reads attributes and stores them into array of size 5");
    assertEquals( expected: 40, testModel.getCards().size(), message: "Check that exactly 40 cards are read");
}
```

1.3 As a dealer I want to be able to shuffle deck of cards.

GameLog.txt

```
----------------deck read from file----------------------
      Abaddon   6   37  3   9   38
    Alchemist   6   28  32  12  47
         Axe    20  16  31  39  30
  Beastmaster   6   29  27  2   11
   Brewmaster   37  39  41  12  31
  Bristleback   16  34  12  43  25
      Centaur   40  38  31  41  50
        Chaos   11  25  29  9   13
    Clockwerk   33  23  21  16  30
         Doom   42  9   48  7   35
       Dragon   5   47  16  2   50
        Earth   3   13  48  20  12
  Earthshaker   34  17  5   45  48
        Elder   2   33  41  42  42
       Huskar   30  14  34  37  28
           Io   36  30  44  8   37
       Kunkka   35  3   36  13  44
       Legion   6   14  45  50  15
  Lifestealer   36  37  2   23  29
       Lycan    5   41  48  27  34
      Magnus    45  28  26  22  32
        Night   22  6   24  33  14
   Omniknight   6   21  46  4   29
      Phoenix   11  43  2   14  16
       Pudge    13  41  6   19  48
    Sandking    23  6   46  4   18
     Slardar    39  27  45  2   16
Spiritbreaker   34  19  5   38  26
        Sven    29  11  24  4   4
  Tidehunter    50  48  37  30  37
   Timbersaw    32  43  5   14  43
        Tiny    33  3   43  10  38
      Treant    31  41  32  19  17
        Tusk    33  4   9   21  11
  Underlord     38  10  30  17  7
    Undying     30  42  33  14  1
   Vengeful     4   35  34  26  17
  Venomancer    17  8   46  46  30
       Viper    26  33  34  8   6
      Weaver    33  11  9   14  48
----------------Shuffled deck----------------------
      Weaver    33  11  9   14  48
  Tidehunter    50  48  37  30  37
       Night    22  6   24  33  14
   Timbersaw    32  43  5   14  43
  Lifestealer   36  37  2   23  29
      Kunkka    35  3   36  13  44
    Sandking    23  6   46  4   18
       Lycan    5   41  48  27  34
  Earthshaker   34  17  5   45  48
  Underlord     38  10  30  17  7
   Vengeful     4   35  34  26  17
        Sven    29  11  24  4   4
      Huskar    30  14  34  37  28
       Chaos    11  25  29  9   13
     Phoenix    11  43  2   14  16
```

1.4 As a human or AI player I want to receive fair share of cards.

```
@Test
public void testCardsDistributed() {
    // test that the total number of players is within constraints
    int numberOfPlayers = testModel.getAIPlayerNumber() + 1;
    if (numberOfPlayers > 5 || numberOfPlayers < 2) {
        fail("Wrong number of players (bad user input)");
    }

    ArrayList<Player> players = testModel.getPlayers();
    if (numberOfPlayers == 3) {
        int playersWith13Cards = 0;
        int playersWith14Cards = 0;
        for (Player p : players) {
            if (p.getPlayerCards().size() == 14) {
                playersWith14Cards++;
            } else {
                playersWith13Cards++;
            }
        }
        assertTrue( condition: playersWith14Cards == 1,  message: "there is just 1 player with more cards");
        assertTrue( condition: playersWith13Cards == 2,  message: "the other 2 players have just 13 cards");
    } else {
        for (Player p : players) {
            assertTrue( condition: p.getPlayerCards().size() == 40 / numberOfPlayers,
                    message: "otherwise each players gets 40/total players cards");
        }
```

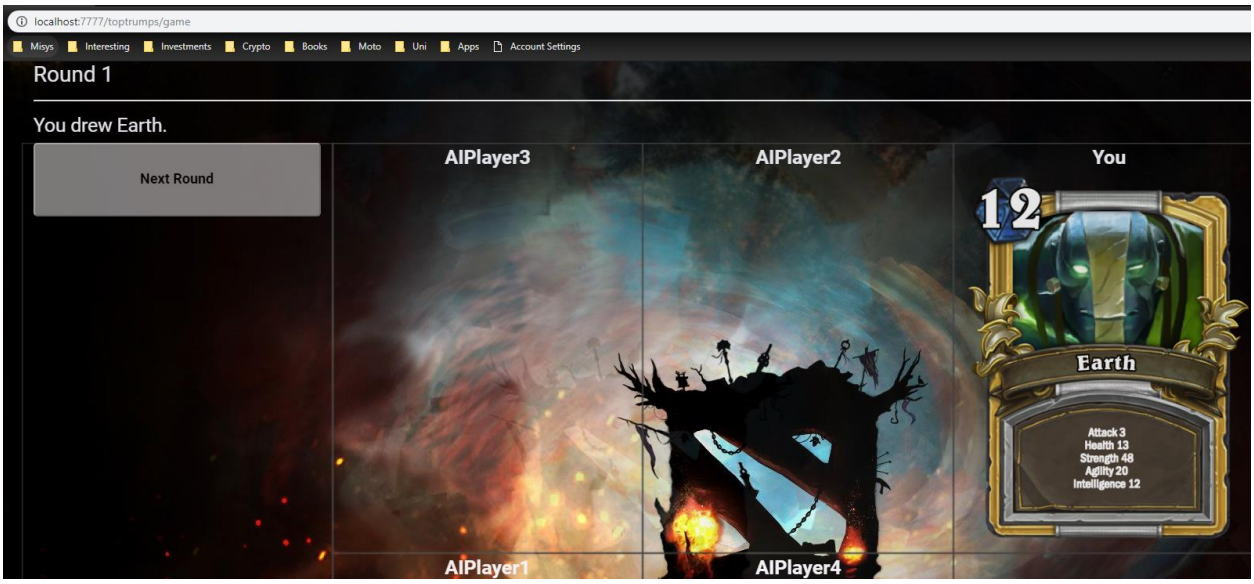
1.5 As a human player I want to see the details of the top card.

```
9           TTModel model = new TTModel();
10          TTView view = new TTView(model);
11          TTController controller = new TTController(model,view);
12          controller.startGame();
13
14      }
15
16
17
```

Console

Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:21:40)

```
Enter the number of AI players
1
You Drew 'Bristleback'
   > Attack: 16
   > Armour: 34
   > Strength: 12
   > Agility: 43
   > Intelligence: 25
You now have 20 card(s) in your deck.
Round 1
Round 1: Players have drawn their cards
AIPlayer1 Picked characteristic 1: Attack.
Round1: AIPlayer1 won this round!
The winning card was 'Tidehunter':
   > Attack: 50 <--
   > Armour: 48
   > Strength: 37
   > Agility: 30
   > Intelligence: 37
```

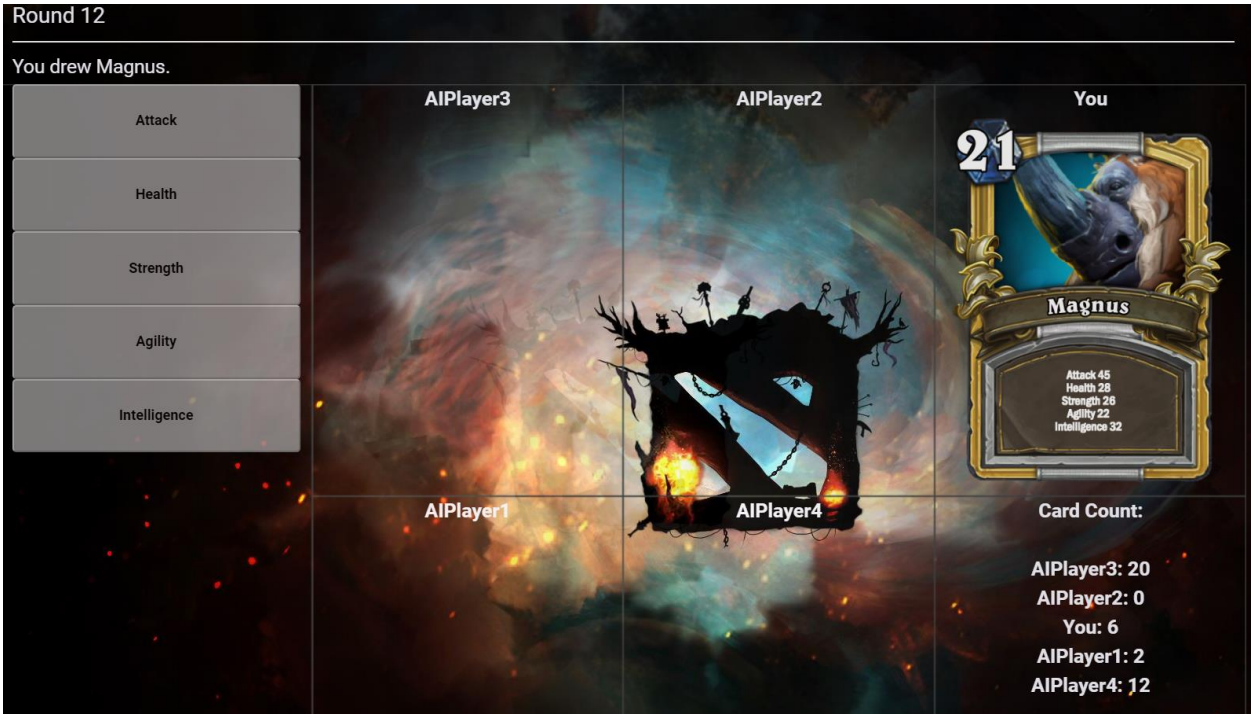


1.6 As a human player I want to choose option for top card.

```
10          TTView view = new TTView(model);
11          TTController controller = new TTController(model,view);
12          controller.startGame();
13
14      }
15
16
17
```

Console

```
Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:56:26)
   > Intelligence: 16

You Drew 'Treant'
   > Attack: 31
   > Armour: 41
   > Strength: 32
   > Agility: 19
   > Intelligence: 17
You now have 15 card(s) in your deck.
Round 2
Round 2: Players have drawn their cards
It is your turn to select a category, the categories are:
   1:Attack
   2:Armour
   3:Strength
   4:Agility
   5:Intelligence
2
You Picked characteristic 2: Armour.
Round2: AIPlayer1 won this round!
The winning card was 'Timbersaw':
   > Attack: 32
   > Armour: 43 <--
   > Strength: 5
```



1.7 As an AI player I want to choose best option for top card.

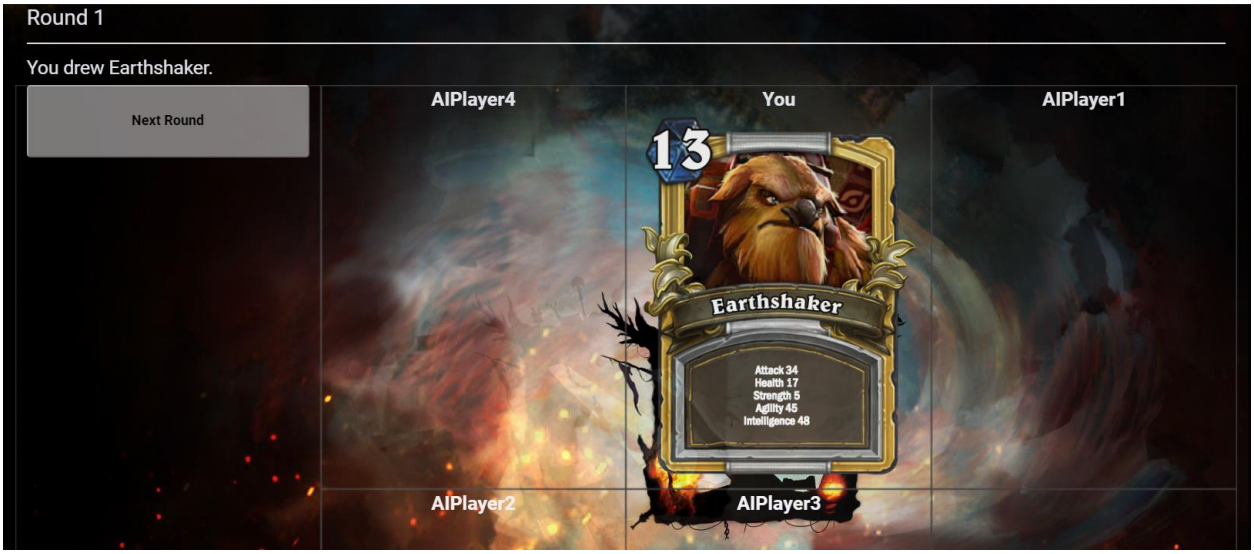
```
@Test
void playRoundTest() {
    testController.getModel().setHumanPlayerEliminated(true);

    // set the first player in the list pointing to the highest attribute of their
    // topmost card
    int currentAttribute = testController.getHighestAttributeIndex( playerIndex: 0);
    testController.getModel().setIndexOfCurrentAttribute(currentAttribute);

    //take all the cards in play this round and add them to the cardsThisRound arrayList
    ArrayList<Card> cardsThisRound = new ArrayList<~>();
```

1.8 As a dealer I want to choose random first player.

```
9           TTModel model = new TTModel();
10          TTView view = new TTView(model);
11          TTController controller = new TTController(model,view);
12          controller.startGame();
13
14      }
15
16
17
```

Console

```
Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:48:56)

Enter the number of AI players
4
You Drew 'Huskar'
   > Attack: 30
   > Armour: 14
   > Strength: 34
   > Agility: 37
   > Intelligence: 28
You now have 8 card(s) in your deck.
Round 1
Round 1: Players have drawn their cards
AIPlayer3 Picked characteristic 5: Intelligence.
Round 1: This round was a Draw, common pile now has 5 cards
```

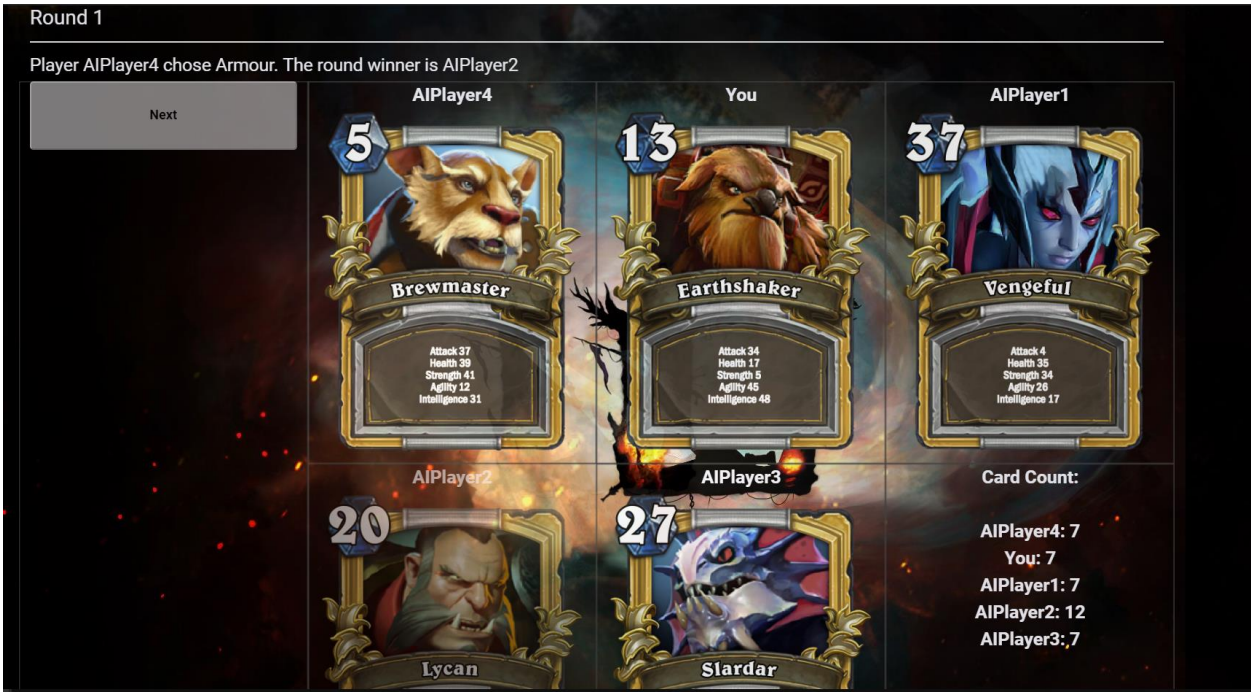


1.9 As a human player I want to see relevant game data.

```
9          TTModel model = new TTModel();
10         TTView view = new TTView(model);
11         TTController controller = new TTController(model,view);
12         controller.startGame();
13
14     }
15
16
17
```

Console

```
<terminated> Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:48:56)
   > Agility: 7
   > Intelligence: 35
Round 41
Round 41: Players have drawn their cards
AIPlayer3 Picked characteristic 3: Strength.
Round41: AIPlayer3 won this round!
The winning card was 'Tiny':
   > Attack: 33
   > Armour: 3
   > Strength: 43 <--
   > Agility: 10
   > Intelligence: 38
Round 42
Round 42: Players have drawn their cards
AIPlayer3 Picked characteristic 1: Attack.
Round42: AIPlayer3 won this round!
The winning card was 'Magnus':
   > Attack: 45 <--
   > Armour: 28
   > Strength: 26
   > Agility: 22
   > Intelligence: 32
Round 43
Round 43: Players have drawn their cards
AIPlayer3 Picked characteristic 1: Attack.
Round43: AIPlayer3 won this round!
```



1.10 As a human player I want to see who wins the round, the game and details of the common pile.

```
//      int i = 0;
//      i = (int) (Math.ceil(Math.random()*5));
//      System.out.println(i);
        TTModel model = new TTModel();
        TTView view = new TTView(model);
        TTController controller = new TTController(model,view);
        controller.startGame();

    }
```

```
Console
Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:21:40)
You now have 20 card(s) in your deck.
Round 1
Round 1: Players have drawn their cards
AIPlayer1 Picked characteristic 1: Attack.
Round1: AIPlayer1 won this round!
The winning card was 'Tidehunter':
   > Attack: 50 <--
   > Armour: 48
   > Strength: 37
   > Agility: 30
   > Intelligence: 37

You Drew 'Kunkka'
   > Attack: 35
   > Armour: 3
   > Strength: 36
   > Agility: 13
   > Intelligence: 44
You now have 19 card(s) in your deck.
Round 2
```

```
        TTView view = new TTView(model);
        TTController controller = new TTController(model,view);
        controller.startGame();

    }
```

Console

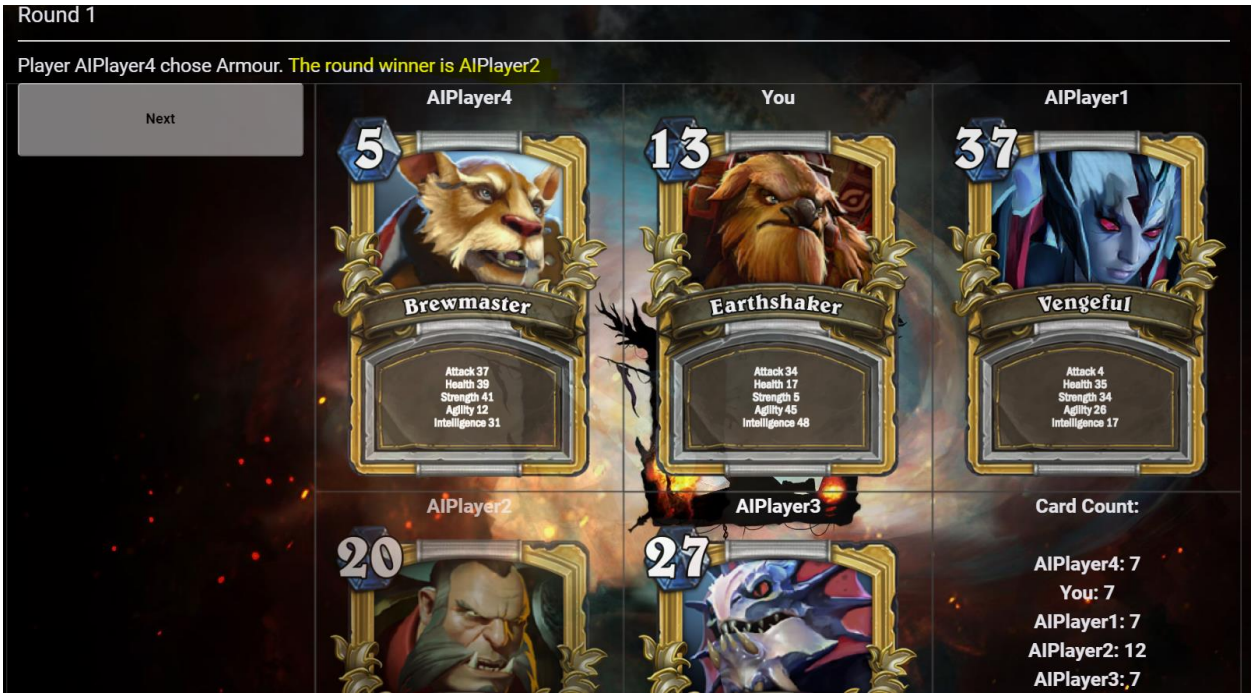
<terminated> Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:21:40)

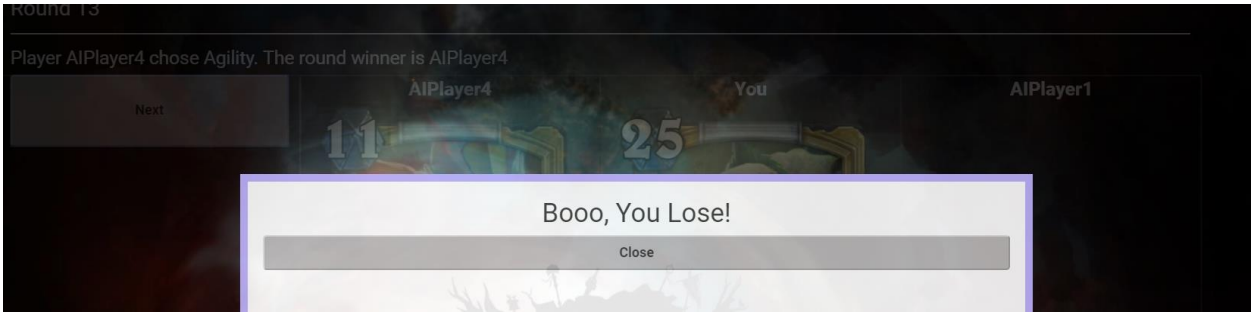
```
   > Attack: 32
   > Armour: 43
   > Strength: 5
   > Agility: 14
   > Intelligence: 43
You now have 1 card(s) in your deck.
Round 32
Round 32: Players have drawn their cards
AIPlayer1 Picked characteristic 5: Intelligence.
Round32: AIPlayer1 won this round!
The winning card was 'Alchemist':
   > Attack: 6
   > Armour: 28
   > Strength: 32
   > Agility: 12
   > Intelligence: 47 <--

Game Over! The Final winner is AIPlayer1!
Scores:
   AIPlayer1: 26
   You: 6
```

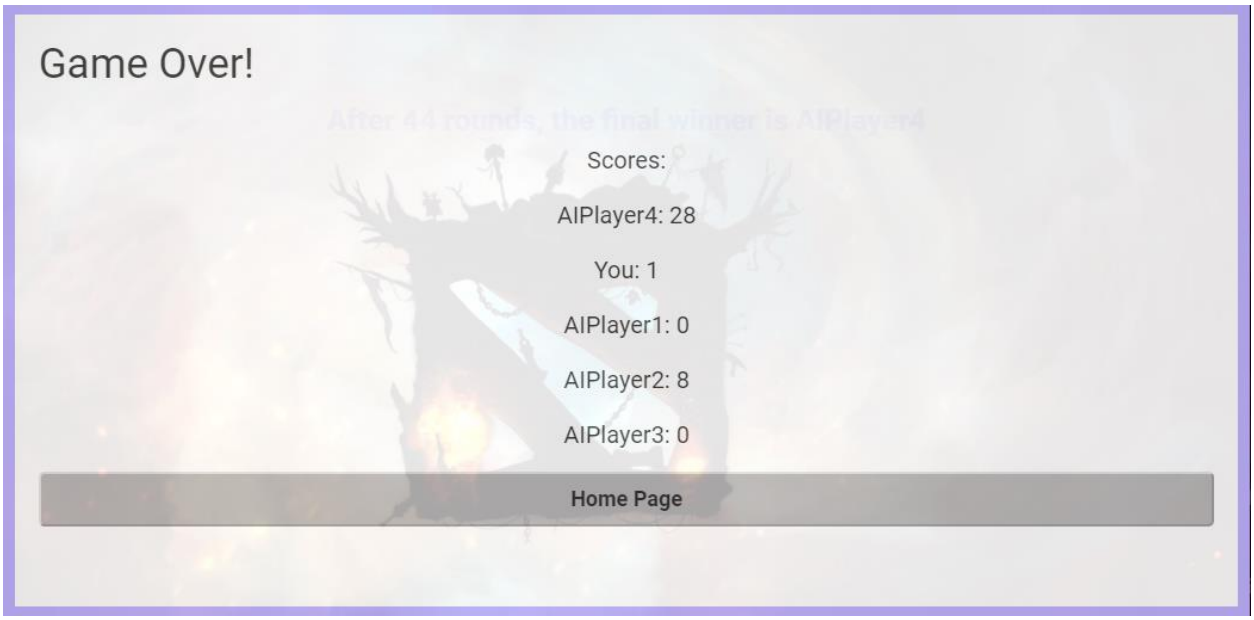
```
10          TTView view = new TTView(model);
11          TTController controller = new TTController(model,view);
12          controller.startGame();
13
14      }
15
16
17
```

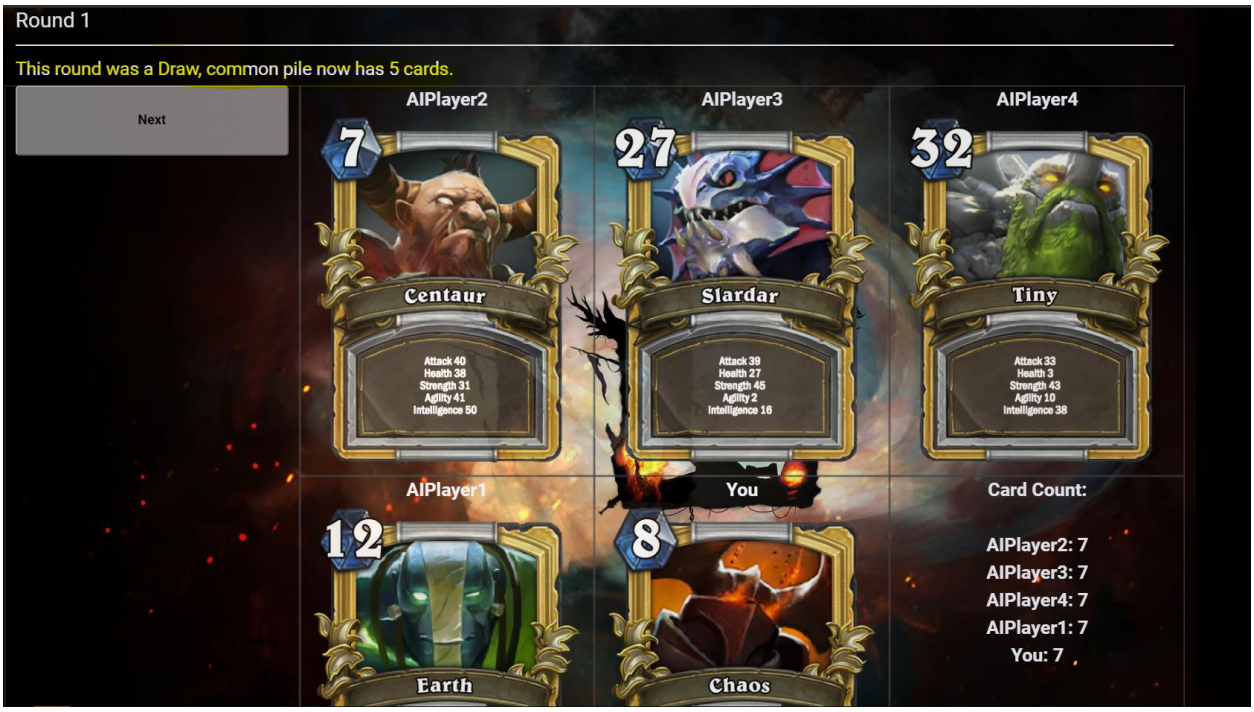
Console

```
Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:35:42)
You Drew 'Timbersaw'
   > Attack: 32
   > Armour: 43
   > Strength: 5
   > Agility: 14
   > Intelligence: 43
You now have 2 card(s) in your deck.
Round 24
Round 24: Players have drawn their cards
AIPlayer1 Picked characteristic 2: Armour.
Round 24: This round was a Draw, common pile now has 3 cards

You Drew 'Io'
   > Attack: 36
   > Armour: 30
   > Strength: 44
   > Agility: 8
   > Intelligence: 37
You now have 1 card(s) in your deck.
Round 25
Round 25: Players have drawn their cards
```




Round 13
Player AIPlayer4 chose Agility. The round winner is AIPlayer4
Next
AIPlayer4
You
AIPlayer1
11
25
Booo, You Lose!
Close


Game Over!
After 44 rounds, the final winner is AIPlayer4
Scores:
AIPlayer4: 28
You: 1
AIPlayer1: 0
AIPlayer2: 8
AIPlayer3: 0
Home Page


Round 1
This round was a Draw, common pile now has 5 cards.
Next
AIPlayer2
AIPlayer3
AIPlayer4
7
27
32
Centaur
Slardar
Tiny
Attack 40
Health 38
Strength 31
Agility 41
Intelligence 50
Attack 39
Health 27
Strength 45
Agility 2
Intelligence 16
Attack 33
Health 3
Strength 43
Agility 10
Intelligence 38
AIPlayer1
You
Card Count:
12
8
Earth
Chaos
AIPlayer2: 7
AIPlayer3: 7
AIPlayer4: 7
AIPlayer1: 7
You: 7

1.11 As a human or AI player I want to start the next round if I won the previous one.

```
11          TTController controller = new TTController(model,view);
12          controller.startGame();
13
14      }
15
16
17
```

Console

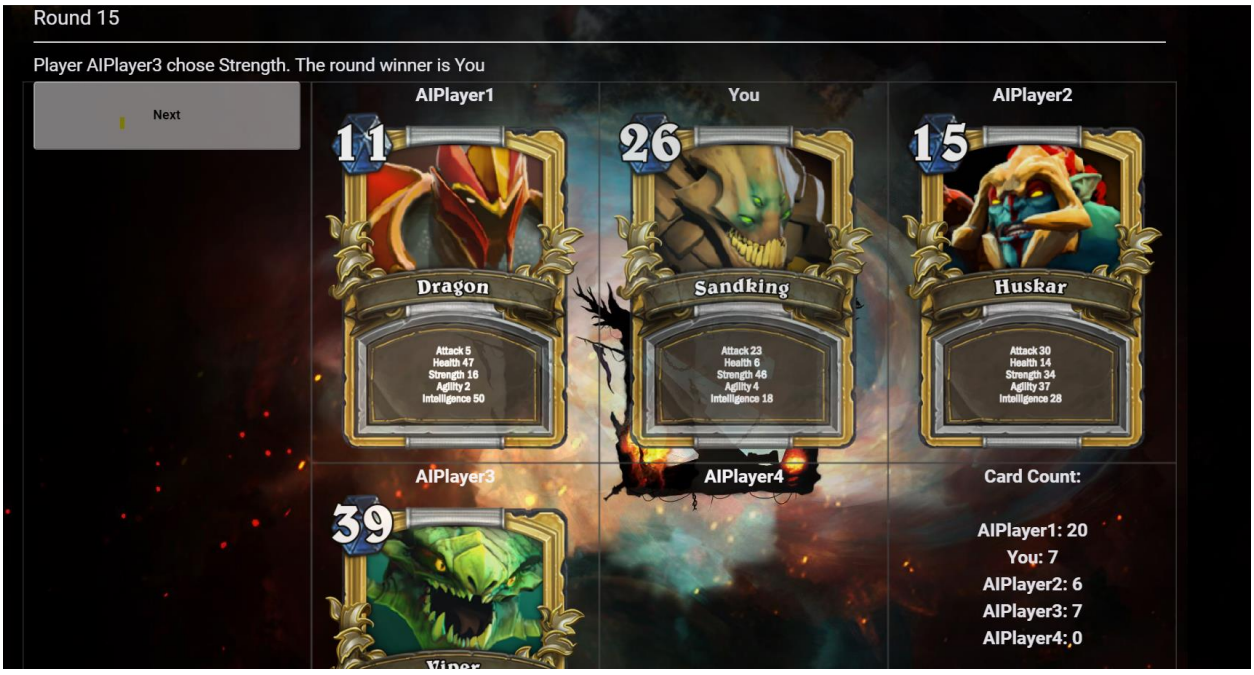
Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:35:42)

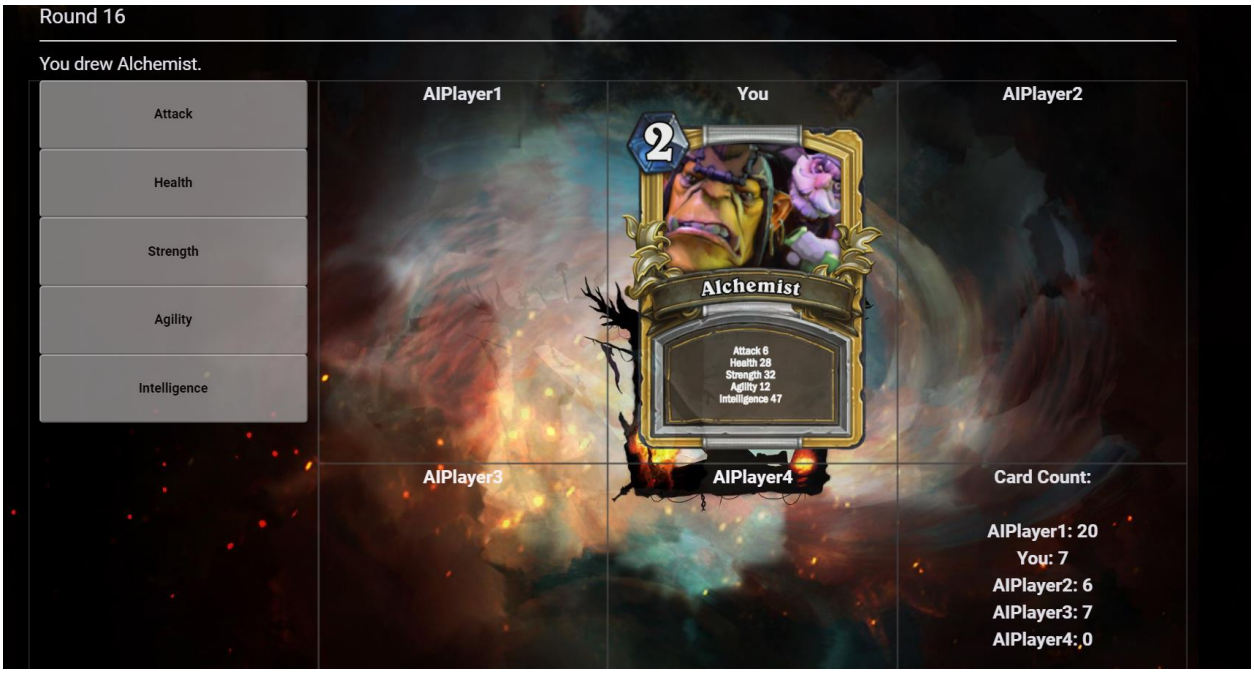
```
   > Strength: 24
   > Agility: 4
   > Intelligence: 4
You now have 5 card(s) in your deck.
Round 21
Round 21: Players have drawn their cards
AIPlayer1 Picked characteristic 3: Strength.
Round21: AIPlayer1 won this round!
The winning card was 'Sandking':
   > Attack: 23
   > Armour: 6
   > Strength: 46 <--
   > Agility: 4
   > Intelligence: 18

You Drew 'Abaddon'
   > Attack: 6
   > Armour: 37
   > Strength: 3
   > Agility: 9
   > Intelligence: 38
You now have 4 card(s) in your deck.
Round 22
Round 22: Players have drawn their cards
AIPlayer1 Picked characteristic 3: Strength.
```

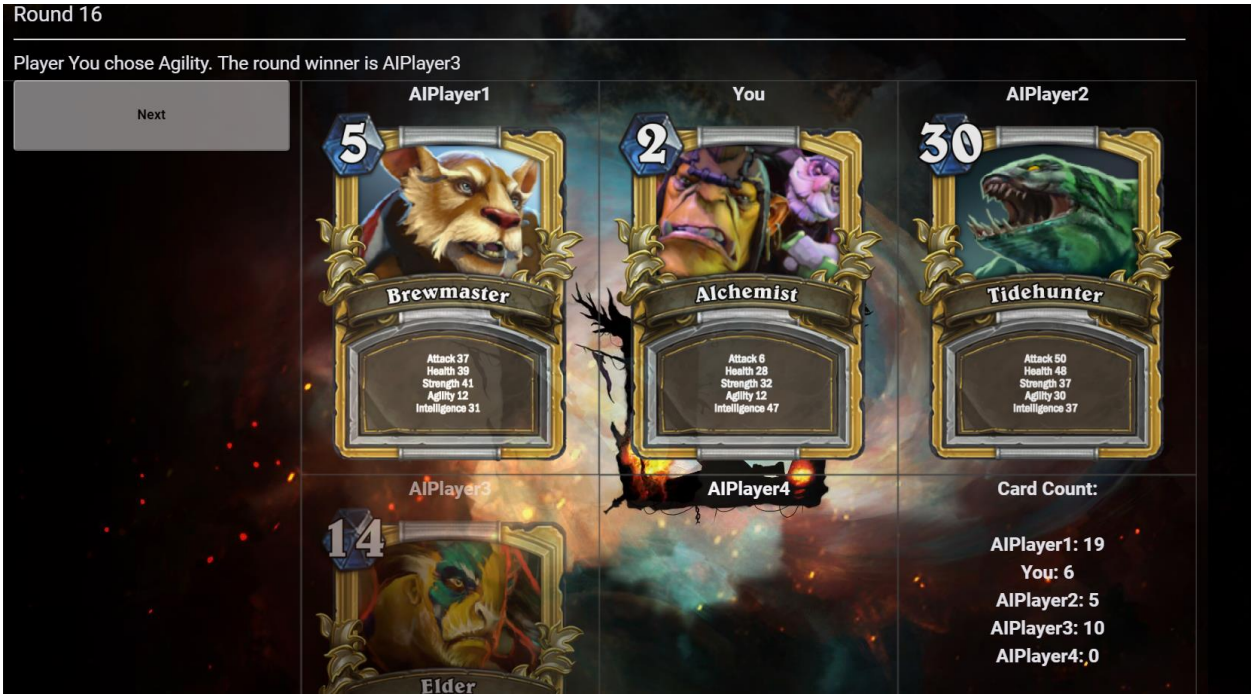
```
        TTController controller = new TTController(model,view);
        controller.startGame();

    }
```

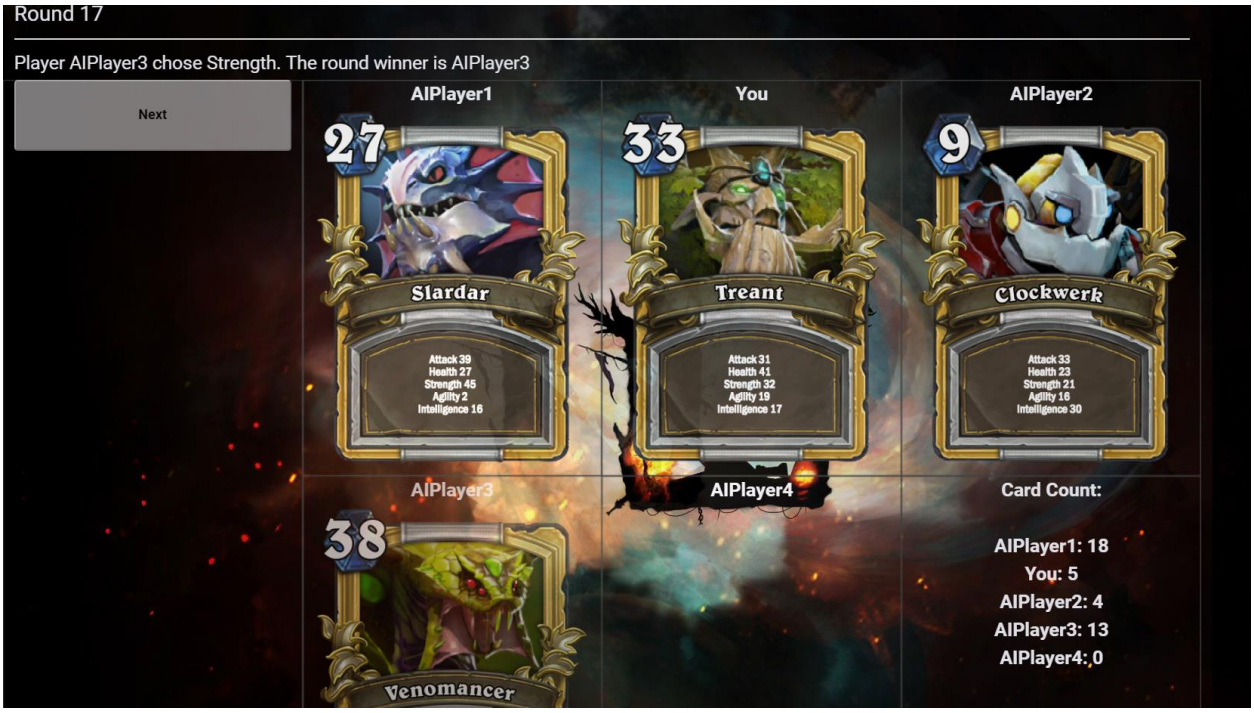
Console

```
Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:39:45)
Round 1
Round 1: Players have drawn their cards
AIPlayer2 Picked characteristic 2: Armour.
Round1: You won this round!
The winning card was 'Dragon':
   > Attack: 5
   > Armour: 47 <--
   > Strength: 16
   > Agility: 2
   > Intelligence: 50

You Drew 'Tiny'
   > Attack: 33
   > Armour: 3
   > Strength: 43
   > Agility: 10
   > Intelligence: 38
You now have 16 card(s) in your deck.
Round 2
Round 2: Players have drawn their cards
It is your turn to select a category, the categories are:
   1:Attack
   2:Armour
   3:Strength
```




Round 16
You drew Alchemist.
Attack
Health
Strength
Agility
Intelligence
AIPlayer1
You
AIPlayer2
2
Alchemist
Attack 6
Health 28
Strength 32
Agility 12
Intelligence 47
AIPlayer3
AIPlayer4
Card Count:
AIPlayer1: 20
You: 7
AIPlayer2: 6
AIPlayer3: 7
AIPlayer4: 0


Round 16
Player You chose Agility. The round winner is AIPlayer3
Next
AIPlayer1
You
AIPlayer2
5
Brewmaster
Attack 37
Health 39
Strength 41
Agility 12
Intelligence 31
2
Alchemist
Attack 6
Health 28
Strength 32
Agility 12
Intelligence 47
30
Tidehunter
Attack 50
Health 48
Strength 37
Agility 30
Intelligence 37
AIPlayer3
AIPlayer4
Card Count:
14
Elder
AIPlayer1: 19
You: 6
AIPlayer2: 5
AIPlayer3: 10
AIPlayer4: 0



1.12 As a dealer I want to make sure that once a player is eliminated, they do not participate in any more rounds.

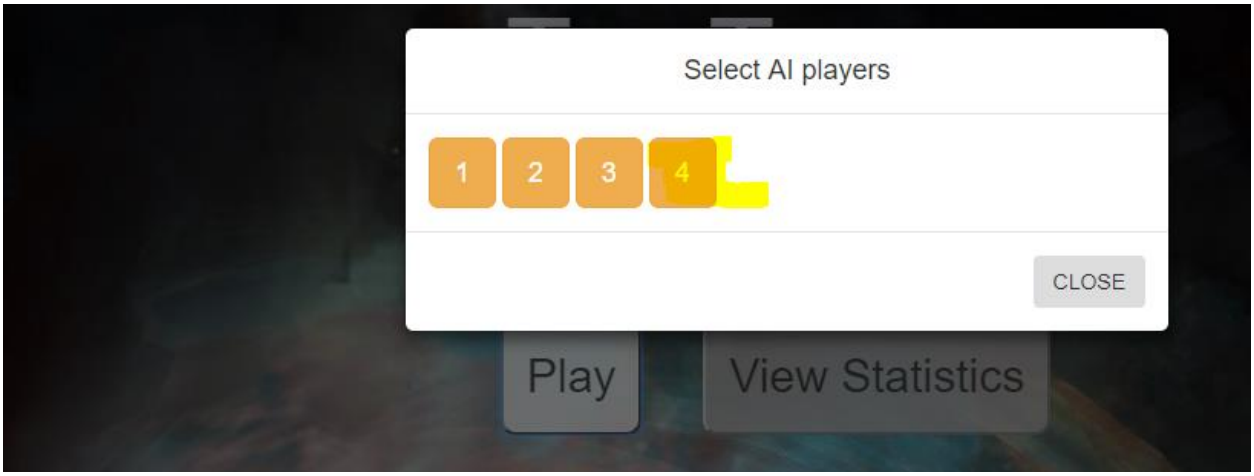
```
        controller.startGame();

    }
```

Console

```
<terminated> Game [Java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15 Feb 2019, 12:39:45)
   > Armour: 19
   > Strength: 5
   > Agility: 38 <--
   > Intelligence: 26

You Drew 'Earthshaker'
   > Attack: 34
   > Armour: 17
   > Strength: 5
   > Agility: 45
   > Intelligence: 48
You now have 1 card(s) in your deck.
Round 50
Round 50: Players have drawn their cards
AIPlayer2 Picked characteristic 5: Intelligence.
Round50: AIPlayer2 won this round!
The winning card was 'Dragon':
   > Attack: 5
   > Armour: 47
   > Strength: 16
   > Agility: 2
   > Intelligence: 50 <--

You lose!
```

1.13 As a human player I want to choose number of AI opponents.

```
@Test
public void testNumberOfAIsChoice() {
    assertEquals(testModel.getAIPlayerNumber(), actual: testModel.getPlayers().size() - 1,
            message: "test that the number of players in the game corresponds to the player's choice");
}
```

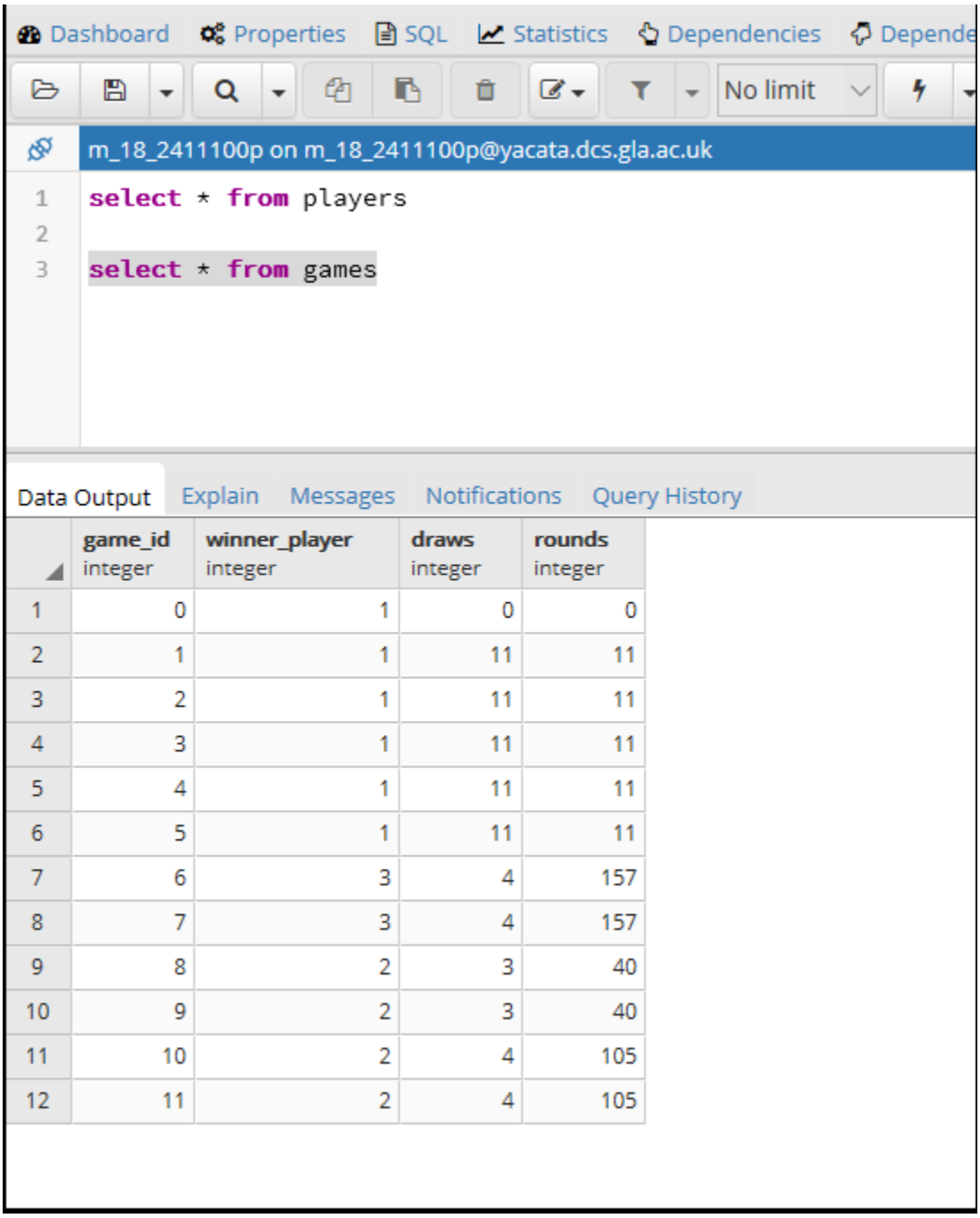


1.14 As a user I want to receive an output of gameplay details.

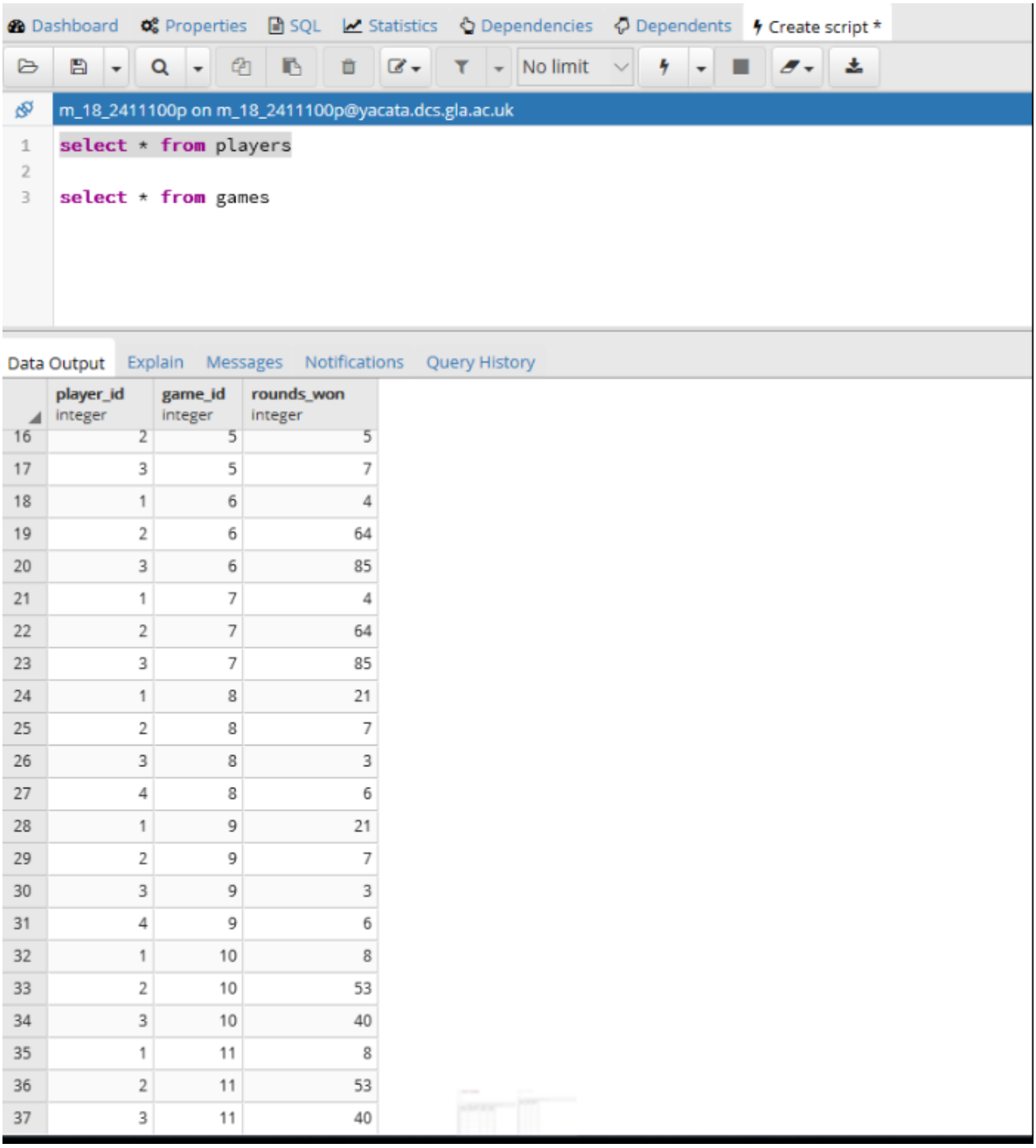
GameLog.txt

```
   Venomancer   17  8   46  46  30
        Tusk    33  4   9   21  11
    Alchemist   6   28  32  12  47
      Magnus    45  28  26  22  32
----------------Attribute selected---------------------
Attribute 4: Agility
----------------Cards in play---------------------
AIPlayer4:     Timbersaw    32  43  5   14  43
      You:          Sven    29  11  24  4   4
AIPlayer2:   Beastmaster    6   29  27  2   11
AIPlayer1: Spiritbreaker    34  19  5   38  26
AIPlayer3:         Viper    26  33  34  8   6
----------------Player decks---------------------
AIPlayer4:
  Lifestealer   36  37  2   23  29
       Kunkka   35  3   36  13  44
     Sandking   23  6   46  4   18
        Lycan   5   41  48  27  34
   Tidehunter   50  48  37  30  37
       Weaver   33  11  9   14  48
  Earthshaker   34  17  5   45  48
    Clockwerk   33  23  21  16  30
     Undying    30  42  33  14  1
   Brewmaster   37  39  41  12  31
    Underlord   38  10  30  17  7
        Elder   2   33  41  42  42
  Bristleback   16  34  12  43  25
      Slardar   39  27  45  2   16
You:
       Huskar   30  14  34  37  28
        Chaos   11  25  29  9   13
      Phoenix   11  43  2   14  16
         Doom   42  9   48  7   35
AIPlayer2:
       Dragon   5   47  16  2   50
           Io   36  30  44  8   37
         Tiny   33  3   43  10  38
   Omniknight   6   21  46  4   29
AIPlayer1:
       Pudge    13  41  6   19  48
     Centaur    40  38  31  41  50
    Abaddon     6   37  3   9   38
       Earth    3   13  48  20  12
       Night    22  6   24  33  14
    Vengeful    4   35  34  26  17
      Treant    31  41  32  19  17
      Legion    6   14  45  50  15
         Axe    20  16  31  39  30
   Timbersaw    32  43  5   14  43
        Sven    29  11  24  4   4
  Beastmaster   6   29  27  2   11
Spiritbreaker   34  19  5   38  26
        Viper   26  33  34  8   6
AIPlayer3:
   Venomancer   17  8   46  46  30
```

1.15 As a user I want to store specific persistent game data at the end of the game.



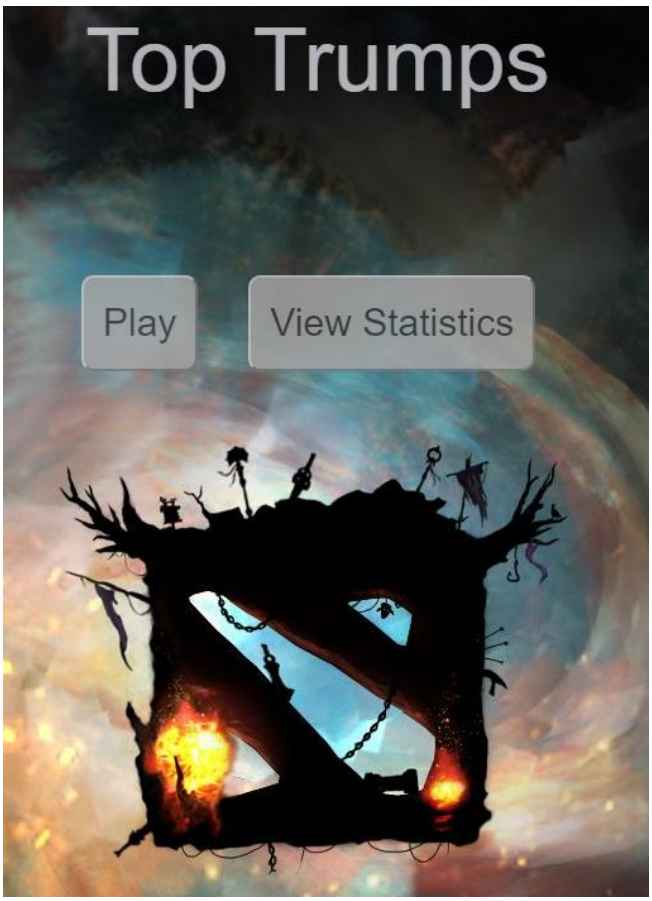
	game_id integer	winner_player integer	draws integer	rounds integer
1	0	1	0	0
2	1	1	11	11
3	2	1	11	11
4	3	1	11	11
5	4	1	11	11
6	5	1	11	11
7	6	3	4	157
8	7	3	4	157
9	8	2	3	40
10	9	2	3	40
11	10	2	4	105
12	11	2	4	105

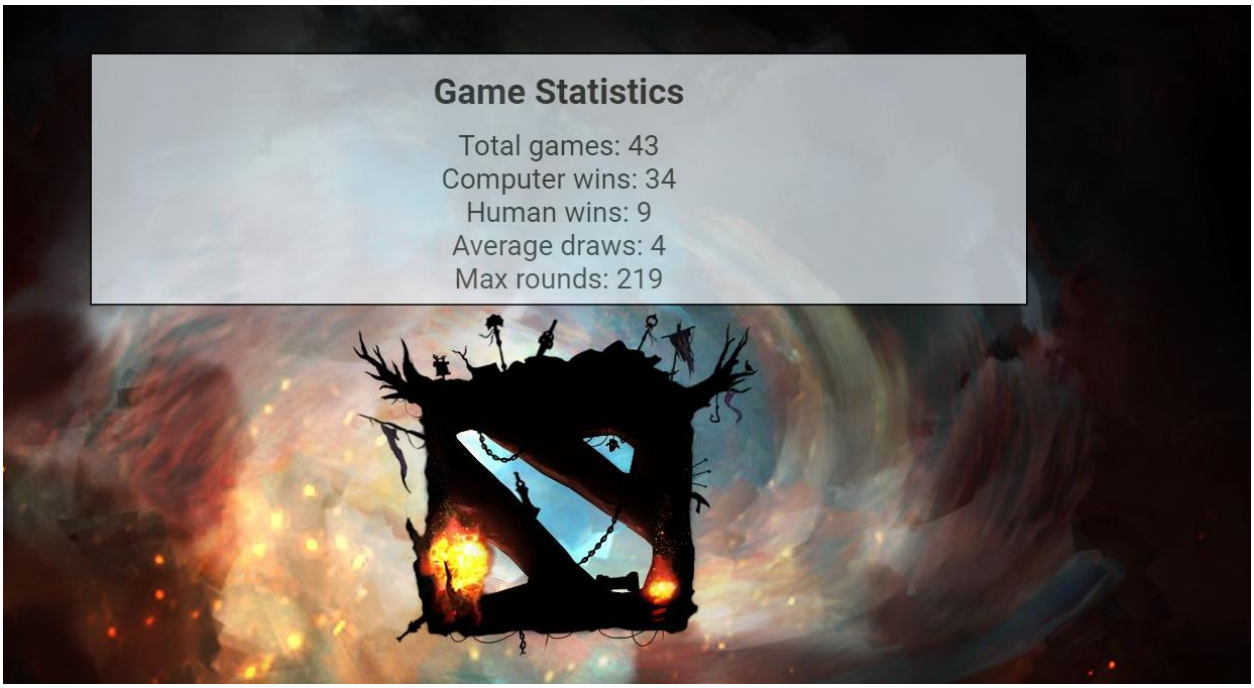


	player_id integer	game_id integer	rounds_won integer
16	2	5	5
17	3	5	7
18	1	6	4
19	2	6	64
20	3	6	85
21	1	7	4
22	2	7	64
23	3	7	85
24	1	8	21
25	2	8	7
26	3	8	3
27	4	8	6
28	1	9	21
29	2	9	7
30	3	9	3
31	4	9	6
32	1	10	8
33	2	10	53
34	3	10	40
35	1	11	8
36	2	11	53
37	3	11	40

1.16 As a user I want to have functional “Start Game” and “Show Statistics” buttons for online mode.

```
Do you want to see past results or play a game?
   1: Print Game Statistics
   2: Play game
   0: Quit

Enter the number for your selection: 1
Jar found.
Controlling your database.
Game_id is: 44

Total games:   43
Computer wins: 34
Human wins:    9
Average draws: 4
Max rounds:    219
```





1.17

Appendix 2 – References

- Initial game screen picture has been retrieved from: https://ro.pinterest.com/pin/761038037000999099/?lp=true - for personal and academic use only.
- All card pictures have been retrieved from: https://dota2.gamepedia.com/Heroes - for personal and academic use only.
- The following web page has been used to generate the cards from the pictures above: http://www.hearthcards.net/ - for personal and academic use only.