

1 Purpose

This document comprises the foundational requirements of the system.

Section 2 is quoted verbatim from Assignment Specification ITSD2018-TaskDocument. Requirements within are identified in **bold** and given a unique requirement ID.

Section 3 captures any assumed requirements which are not present in the Customer Specification.

Section 4 provides a mechanism for checking requirement coverage in generated user stories.

2 Customer Specification

2.1 Context

Top Trumps is a simple card game in which decks of cards are based on a theme. For example, race cars, dinosaurs, and even TV shows like “The Simpsons”. Within a deck each card represents an entity within that topic (e.g. T-Rex for dinosaurs or Bart Simpson for the Simpsons). Within a deck each card has the same list of characteristics. For example, dinosaurs can have a height, weight, length, ferocity, and intelligence. Each card has a value for each characteristic of the deck. The objective of the game is to “trump” your opponent by selecting a category (e.g. intelligence) and having a “better” value for your card than the opponent does in their current card.

Gameplay is as follows:

- **[C1000: There must be at least two players]. [C1010: The deck of cards is divided between the players]. [C1020: The first player takes their topmost card and selects a characteristic]. [C1030: The value of that characteristic is compared against the value for the same characteristic in the other players’ top card]. [C1040: The player with the best value for that characteristic wins the round and the winner takes all the cards from that round (including their own) and places them at the back of their deck]. [C1050: If there is a draw the cards from the round are placed in a new communal pile and a new characteristic is selected by the same player from the next card]. [C1060: The winner then takes the cards from the round and any in the communal pile]. [C1070: The winner of a round maintains the choice of category until they lose, then the choice moves to the next player]. [C1080: Players lose the game when they have no cards left; the player left with all the cards is the winner of the game].**

2.2 Aim

The aim is to build a computer program to allow a user to play top trumps against one or more AI opponents given a deck. **[C2000: The program should have two modes]:**

- **[C2010: Command Line Mode: The game is played only through command line input and output]. [C2020: In this mode, only one game can be played at a time]. [C2030: This mode should be selected via a -c flag when starting the program]:** o java -jar TopTrump.jar -c

- [C2040: Online Mode: The game should be hosted as a web service, comprised of a REST API that provides remote access to the core game functionality and one or more web-pages that enable a user to play the game]. [C2050: In this mode, multiple users should be able to play the game concurrently (e.g. in different Web browser tabs)]. [C2060: This mode should be selected via a -o flag when starting the program]: `o java -jar TopTrumps.jar -o`

2.3 Functionality

In both modes, the program needs to:

- [C3000: Enable a user to play a game of top trumps with a deck that was loaded in when the program started].
- [C3010: Store the results of past games played in a database as well as visualize that information to the user on-demand].

In command line mode only:

- [C3020: Write a test log to file that contains snapshots of the programs state as it runs].

We discuss each of these requirements in more detail below:

2.3.1 Playing Top Trumps

The program should implement the top trumps game as described in the Context section above. You can make the following assumptions to simplify the implementation. Do not add additional unnecessary functionality. This is gold-plating and will not result in a better grade. If in doubt, consult the course co-ordinator.

- [C4000: There should be one human player and up to 4 computer players (AIs)].
- [C4010: 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].
- [C4020: A deck has 5 criteria and the criteria are always positive integers between 1 and 50 (inclusive)].
- [C4030: A higher number is always better for any given characteristic].
- [C4040: There are 40 cards in a deck.txt file].
- [C4050: The first player should be selected at random].
- [C4060: A draw won't continue until the point where there are only cards in the communal pile you do not need to deal with this programmatically, just assume that should this happen you are not expected to deal with it].

[C4070: The deck we will be using is stored in a text file called 'StarCitizenDeck.txt']. [C4080: The first line of the text file should contain a list of the categories in the deck, separated by a space]. [C4090: All decks should have a 'description' category to label the individual cards]. [C4100: A description within a deck can be assumed to be unique and a single word]. [C4110: The subsequent lines contain details of one card]. [C4120: You can assume the order of the categories from the first line align with 3 the values provided for the cards and that all cards have a value for all categories in a deck]. [C4130: You can assume that all categories are single words].

For example, in a dinosaur deck (numbers bear no resemblance to reality):

```
description height weight length ferocity intelligence
TRex 6 6 12 9 9
Stegosaurus 4 3 8 1 8
Brachiosaurus 12 8 16 2 6
Velociraptor 3 5 5 12 10
Carnotaurus 5 6 7 9 8
Iguanodon 2 2 3 1 9
Megalosaurus 9 9 8 6 9
Oviraptor 8 7 4 3 2
Parasaurolophus 7 7 1 3 4
Ornithomimus 10 9 8 7 5
Protoceratops 9 5 4 7 10
Riojasaurus 6 1 4 7 7
Saurolophus 7 1 10 7 8
Styracosaurus 7 3 4 1 1
Xiaosaurus 10 6 5 7 2
```

and so forth the star citizen-based deck is provided with the Template Package that can be downloaded from Moodle.

[C4140: The program must first load all card details from the deck and shuffle them (randomly order them)]. [C4150: The program should then deal the cards between the players]. [C4160: The user should then be shown the detail from their top card (note there is no need to visualise this in a complex manner, the card details can be shown in text) and the first player is randomly selected]. [C4170: If the player is an AI player it should select a category for play, if the user is the first player they should be allowed to select a category to play the round]. [C4180: The game play should then proceed as detailed in the section Context].

2.3.2 Persistent Game Data

[C5000: Upon completion of the game, the user should automatically write the following information about the game play to a database]:

- [C5010: How many draws were there?]
- [C5020: Who won the game?]

- [C5030: How many rounds were played in the game?]
- [C5040: How many rounds did each player win?]

You should select one team members database on the yacata server, from the Database Theory and Applications course, to write to. [C5050: It is important you do not remove the username and password from the final code, as this allows us to test the software]. You should also provide details of this database (username, database name and password) in the report.

[C5060: There should also be possible, so long as a game isn't currently in progress, for the user to connect to the database and get information about previous games]. This should include the following:

- [C5070: Number of games played overall].
- [C5080: How many times the computer has won].
- [C5090: How many times the human has won].
- [C5100: The average number of draws].
- [C5110: The largest number of rounds played in a single game].
- [C5120: These values should be calculated using SQL].

2.3.3 Test Log

In addition to the functionality described above, you should implement the following to allow for program debugging when in command line mode only. [C6000: When the program is started, if a -t flag is set on the command line, then the program should write out an extensive log of its operation to a toptrumps.log file in the same directory as the program is run], e.g.: `java -jar TopTrumps.jar -c -t`

[C6010: If a toptrumps.log file already exists, your program should overwrite that file]. [C6020: Your program should print the following information to that file, separated by a line containing dashes at the appropriate times as mentioned below]:

- [C6030: The contents of the complete deck once it has been read in and constructed].
- [C6040: The contents of the complete deck after it has been shuffled].
- [C6050: The contents of the users deck and the computers deck(s) once they have been allocated. Be sure to indicate which the users deck is and which the computers deck(s) is].
- [C6060: The contents of the communal pile when cards are added or removed from it].
- [C6070: The contents of the current cards in play (the cards from the top of the users deck and the computers deck(s))].
- [C6080: The category selected and corresponding values when a user or computer selects a category].

- [C6090: The contents of each deck after a round].
- [C6100: The winner of the game].

2.3.4 Command Line Mode

[C7000: When started in command line mode, the program should ask the user whether they want to see the statistics of past games (see the Persistent Game Data section) or whether they want to play a game]. [C7010: The users choice should be obtained from standard in (System.in)]. [C7020: If they select to see statistics of past games, the program should print the associated statistics to standard out (System.out) and then ask the same question again]. [C7030: If they select to play a game, a game instance should be started, and the core game loop initialized]. [C7040: For each round, the round number, the name of the active player and the card drawn by the player should be printed to standard out]. [C7050: If the user is the active player, then it should ask the player to select a category, and obtain the users choice from standard in, otherwise the AI player should select a category]. [C7060: The program should then print to standard out the selected category, who won (or whether it was a draw), the winning card and whether the player has been eliminated (they have no cards left)]. [C7070: Rounds should continue to be played until a winner is determined (only one player has cards left)]. [C7080: Once the user has been eliminated, the remaining rounds should be completed automatically (without user input)]. [C7090: At the end of the game, the overall winner should be printed, and the Persistent Game statistics should be updated]. [C7100: The user should then be asked if they want to print the statistics of past games or play another game]. [C7110: An example of the command line output for the program is provided in the CLI.example.txt file on Moodle].

2.3.5 Online Mode

Online mode is an extension to the command line mode that provides a version of the game that users can play through their Web browser. A Web application has two main components. First, a back-end Application Programming Interface (API) that provides remote access to the game functionalities (e.g. starting a new game, drawing cards or getting players/cards for display). Second, one or more webpages that use the back-end API to enable the user to play the game. In this case, each web page should be comprised of HTML elements that are displayed and Javascript functions that connect to the API.

The webpage design is down to you, but it should display the following:

- [C8000: Upon loading the web page, the user should be presented the option to view overall game statistics (as detailed previously), or play a single game].
- [C8010: During game play, the GUI should display the contents of the users top card and, when they are the active player, allow the user to select a category to play against the computer].
- [C8020: The GUI should clearly indicate whos turn it currently is, and only allow the user to select a category when it is their turn].

- [C8030: Once the category has been selected for a round and the values compared, the GUI should display the values of the category for each player and highlight who won the round].
- [C8040: If a draw should occur, the user should be notified of this].
- [C8050: The GUI should contain an indication of how many cards are in the communal pile].
- [C8060: The GUI should contain an indication of how many cards are left in the users deck and in the computers deck(s)].
- [C8070: When the round played results in the game finishing, an indication of the overall winner should be presented to the user along with an option to update the database with the statistics of the game as previously described].

A video of an example website is provided in Moodle.

3 Assumptions

3.1 Both Modes

- [A1000: The user will specify the number of AI opponents upon starting a new game].
- [A1010: When the AI is responsible for selecting a category, it will select a value which is the highest value on it's active card].

3.2 Command Line Mode

-

3.3 Online Mode

-

4 Coverage

[See following page]

