cmpt 370 – Group F1 Requirements document

January 30, 2019

GROUP MEMBERS:

# EXECUTIVE SUMMARY

This document provides an overview of the functionality and system interactions within the card game Hanabi. The planned functionality of the game system is broken down into three categories: must-have, should-have, and might-have. These levels of functionality correspond to definitive game requirements—must-haves—and features that may be implemented—should-haves and might-haves. The game system requirements are also discussed, including software and hardware requirements for the game to run. Elements of planned functionality are described in detail through the use of system actors and actions with conditions and paths. The external entities that interact with the system and the ways they perform those interactions. Detailed images and descriptions for all planned user-interface elements are also included within the document’s appendices. This document does not delve into detailed implementation of system interactions, but provides the information required to describe the rules of the Hanabi card game.

# INTRODUCTION

## *1.1 Scope*

Hanabi is a card game played with two to five players. Hanabi is a cooperative game, i.e. a game where the players do not play against each other but work together towards a common goal. The deck of cards contains five suits (white, yellow, green, blue, and red): three 1’s, two each of 2’s, 3’s, and 4’s and one 5. The goal is to build all five fireworks by building sets of each colour in correct numerical order. The game begins with 8 available information tokens and 3 fuse tokens. The players are dealt a hand containing five cards (four for 4 or 5 players). The players are unable to see their own cards but can see every other players hand. Each turn a player can give information, discard a card, or play a card. Players lose immediately if all fuse tokens are gone, and win immediately if all 5’s have been played successfully. Otherwise the game continues until the deck is becomes empty, and for one full round after that. At the end of the game, the values of the highest cards in each suit are summed, resulting in a total score out of a possible 25 points.

## *1.2 Glossary*

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Give Information | The player points out the cards of either a given number or a given suit in the hand of another player |
| Discard a card | The player chooses to discard a card from their hand and adds it to the discard pile, then draws a card to replace it. The discarded card is out of the game and can no longer be played. Discarding a card replenishes one information token |
| Play a card | The player chooses a card from his hand and attempts to add it to the cards already played. This is successful if the card is a 1 in a suit that has not yet been played, or if it is the next number sequentially in a suit that has been played. Otherwise a fuse token is consumed and the misplayed card is discarded. Successfully playing a 5 of any suit replenishes one information token. Player draws a card regardless of outcome. |
| Fuse Token | 3 fuse tokens, players lose once they are all gone |
| Information Token | 8 information tokens, can be replenished throughout the game |
| Timeout Period |  |

## *1.3 References*

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## *1.4 Overview of Documents*

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

# PLANNED FUNCTIONALITY

The must-have functionalities for this program include multiplayer options, an intuitive graphical interface, and a client-supported game. Both human players and computer players will have the ability to play, and the program will allow for two to five players/computers to participate in a game. One of the most important functionalities of this game is the network capabilities, which will allow for human and computer connection. In addition, an artificially intelligent computer player, applying statistics, machine learning, or other techniques in order to make strong plays. A graphical user interface will be included, as well as animation of actions. The animation includes fuse burning, and/or fireworks exploding and possibly sound-effects.

Should-have functionality includes a pop-up menu. The menu would be able to do many sub tasks that a multiplayer game can have given available resources. For example, there can be an option to toggle sound effect and graphical effect if they are implemented, the ability to control time given to each player’s turn, the ability to kick or add more players in a middle of the game. Another feature that we want to implement given enough time would be the ability to have people waiting in queue to be observer to the game.

# SOFTWARE AND HARDWARE REQUIREMENTS

# 4.0 SYSTEM ACTORS AND THEIR ACTIONS

# 5 CONCLUSION

This document provided information to describe the card game Hanabi. The planned functionality of the game falls into three categories: must-have, should-have, might-have. Might-have functionalities were shown as a simple actor-action scenario description. Should-have features were presented as fully-realized actor-action scenarios, including paragraph descriptions, use-cases as numerated steps, and diagrams. Finally, must-have scenarios additionally included user-interface diagrams. With the information provided in this document, the client should have an general in-depth idea of what the program will entice.

# 6 APPENDIX A: GRAPHICAL USER INTERFACE



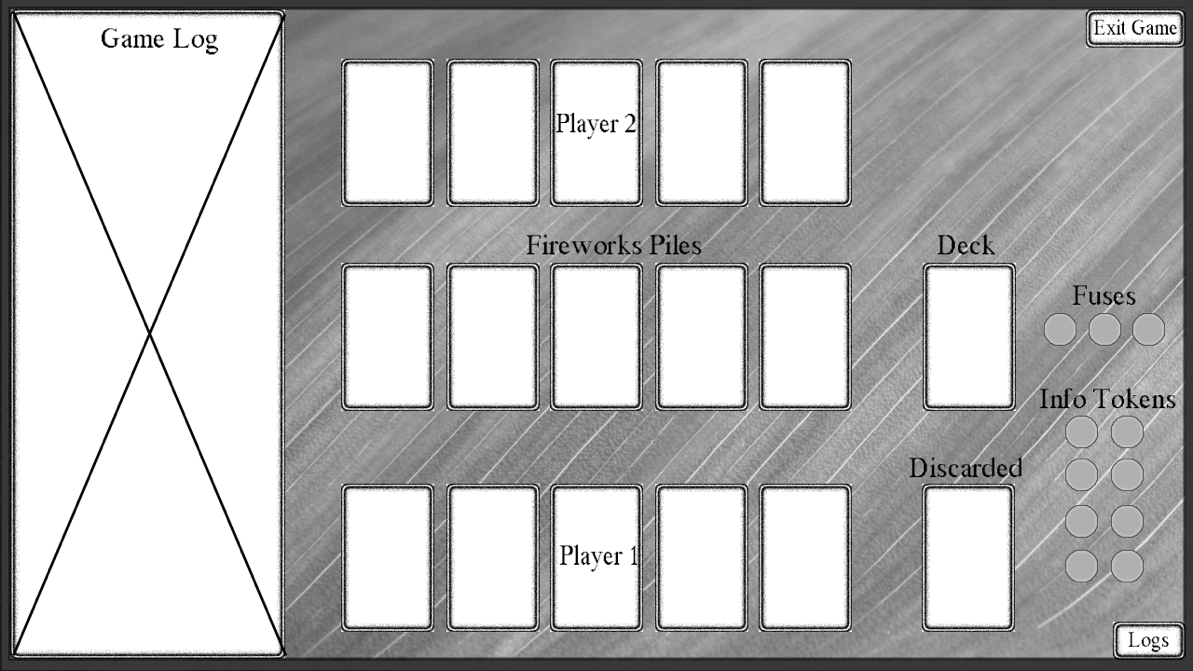
**Main Menu**

Once the user has opened the client this main menu window will appear with “CREATE GAME” to create a game of Hanabi. A “JOIN GAME” for players to join a lobby and a “QUIT GAME” to exit the client.



**Join Menu**

Once the user clicked the “JOIN GAME” button from the main menu they will be prompted to enter their NSID and a secret code to “JOIN” a server or go “BACK” or “QUIT” the game.

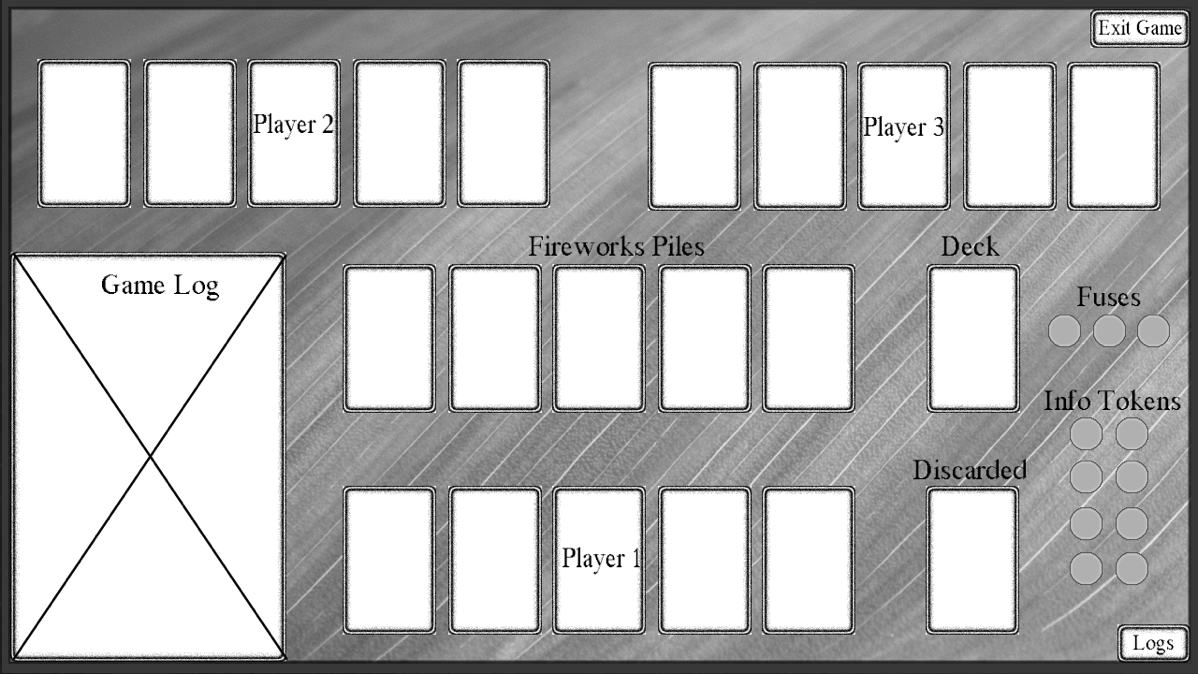


**Game Lobby**

The game queue for the host and players. A “START” button and a “QUIT” button. The host can decide when to start the game or wait until the lobby is full.

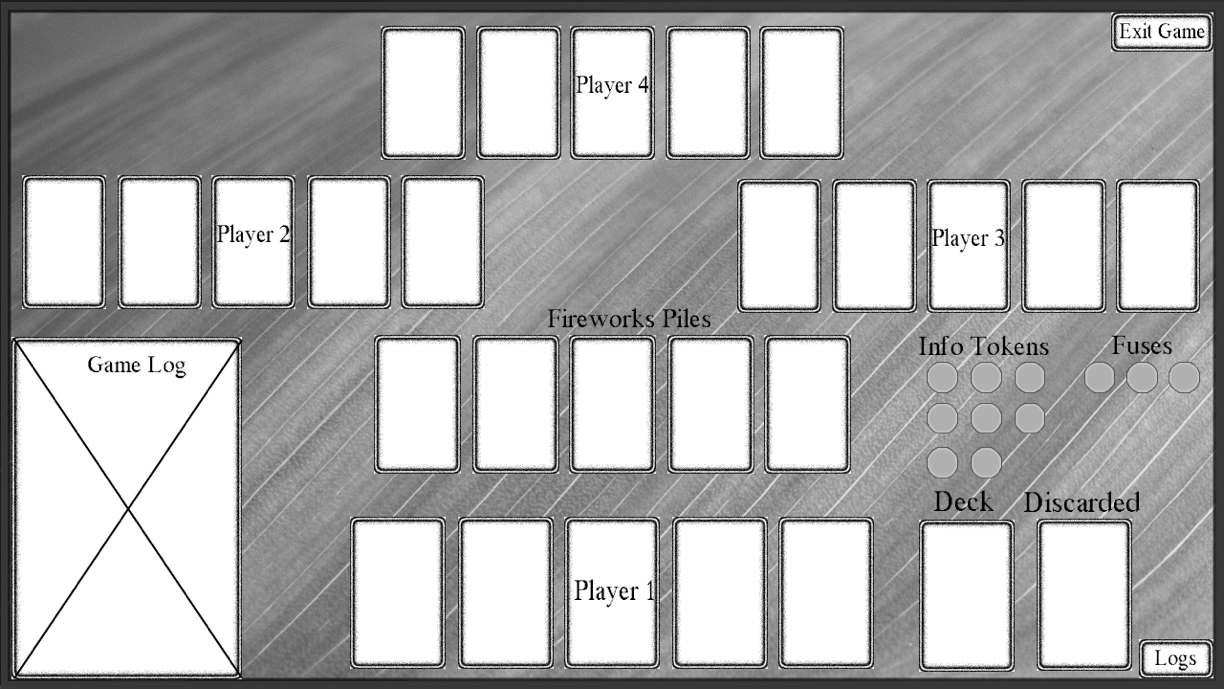
**Game Board 2-Players**

Game board for 2 Players. The players are displayed the other player’s cards while their own cards are unknown. A “EXIT GAME” button for players to leave and a “LOGS” button to toggle if a player would like to see the game logs. Fuse and information tokens are displayed and kept track of. A pile for discard and cards still in play are also displayed.



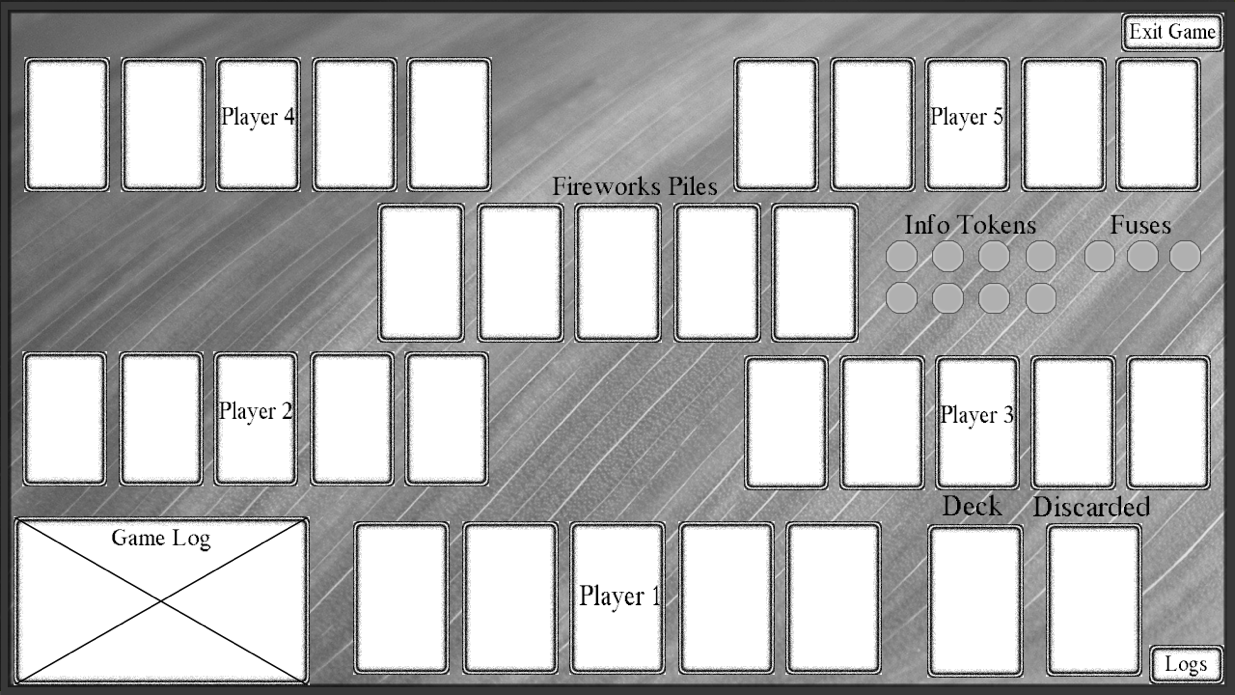
**Game Board 3-Players**

Game board for 3 players. The players are displayed the other player’s cards while their own cards are unknown. A “EXIT GAME” button for players to leave and a “LOGS” button to toggle if a player would like to see the game logs. Fuse and information tokens are displayed and kept track of. A pile for discard and cards still in play are also displayed.



**Game Board 4-Players**

Game board for 4 players. The players are displayed the other player’s cards while their own cards are unknown. A “EXIT GAME” button for players to leave and a “LOGS” button to toggle if a player would like to see the game logs. Fuse and information tokens are displayed and kept track of. A pile for discard and cards still in play are also displayed.



**Game Board 5-Players**

Game board for 5 players. The players are displayed the other player’s cards while their own cards are unknown. A “EXIT GAME” button for players to leave and a “LOGS” button to toggle if a player would like to see the game logs. Fuse and information tokens are displayed and kept track of. A pile for discard and cards still in play are also displayed.



**Game Over**

Once the game has ended the group of players will have a tallied score of what they achieved. A”MAIN MENU” button to return to the main menu page and a “QUIT” button to exit the game.