

# Hanabi Client: Software Requirements Specification

## CMPT 370 Group F1

Minh Hoang – mph952

Sharjeel Humayun – shh738

Donovan Lavoie – dbl599

Faiz Qureshi – faq588

Georgi Nikolov – gnn308

Jason Wong – jaw845

## **Table of Contents**

1. Introduction .....	3
1.1 Purpose .....	3
1.2 Scope .....	3
1.3 Glossary .....	3
1.4 Overview of Document .....	3
2. Overall Description.....	4
2.1 Product Perspective .....	4
2.2 Product Functions .....	4
2.3 User Characteristics .....	5
3. External Interface Requirements .....	6
3.1 User Interfaces .....	6
3.2 Hardware Interfaces .....	7
3.3 Software Interfaces .....	7
3.4 Communication Interfaces.....	7
4. Functional Requirements.....	8
4.1 Close Client.....	8
4.2 Create Game .....	8
4.3 Join Game (Human).....	10
4.4 Join Game (AI) .....	11
4.5 Discard a Card .....	12
4.6 Play a Card.....	13
4.7 Give Information .....	14
4.8 Leave Game.....	16
4.9 View Discard Pile.....	16
4.10 View Action Log.....	17
5. Non-functional Requirements .....	18
5.1 Performance Requirements.....	18
5.2 Reliability Requirements .....	18
6. Conclusion .....	19
Appendix A: Example Graphical User Interface Screens.....	20

## **1. Introduction**

### **1.1 Purpose**

This document describes the software requirements for a Client program that lets a user play Hanabi. It describes the larger Hanabi game system that it is a component of and the actors and actions of that system. It also describes the requirements from external interfaces and functional and non-functional requirements.

### **1.2 Scope**

The program, the Hanabi Client, is meant to let 1 user play a game of Hanabi with other Players across a network. They can make a game, let other Players (whether human or AI) join that game, and then let them make plays and see the plays of other Players as the game goes on. It will not handle communication directly between Players, as it will connect to a server which handles Player communication for it.

### **1.3 Glossary**

<b>Term</b>	<b>Definition</b>
Client/Hanabi Client	The program described in this document
Inactive Game	A game where Players are waiting for more Players to join
Active Game	A game with Players actively playing Hanabi and making move
Firework	A pile of Hanabi cards of the same color, in ascending numerical order
Game ID	A value that identifies a Player's game
Token	A secret value that is required to join a game
Prompt	A window that appears to display information and/or get input from the Player
U of S	University of Saskatchewan
AI	Artificial Intelligence
GUI	Graphical User Interface
CLI	Command Line Interface

### **1.4 Overview of Document**

After this introduction, the Overall Description section gives a description of the product and the context it exists in and an overview of the product's functionality, essentially describing the informal requirements. The External Interfaces section then goes into the requirements for the product's user interface, as well as the requirements on the hardware, software, and communications systems that interact with the product. This leads into the Functional Requirements, where the product's functions and use cases are described in detail, and the Non-functional Requirements, where other requirements for performance and security are mentioned.

## 2. Overall Description

This section now gives a high-level overview of the Hanabi Client and game system. It describes Hanabi itself briefly as well as what the system that lets a Player play Hanabi and the role the Hanabi Client plays in it. It then describes the actors and actions of the Hanabi Client, before briefly stating the characteristics of its users.

### 2.1 Product Perspective

Hanabi is a cooperative card game played with two to five players where the players cannot see the cards in their hands but instead see the other player's hands. They must communicate with each other to play cards in the right order to build fireworks in 5 different colors. Each turn a player can give information to another player about their hand (consuming an information token), discard a card (to regain an information token), or play a card to build a firework. The game continues until 3 mistakes are made in building the fireworks (represented by remaining fuse tokens), until all 5 fireworks are finished, or everyone gets 1 turn after the draw pile is empty. The players score is then the total of the highest values in each firework, resulting in a total score out of a possible 25 points.

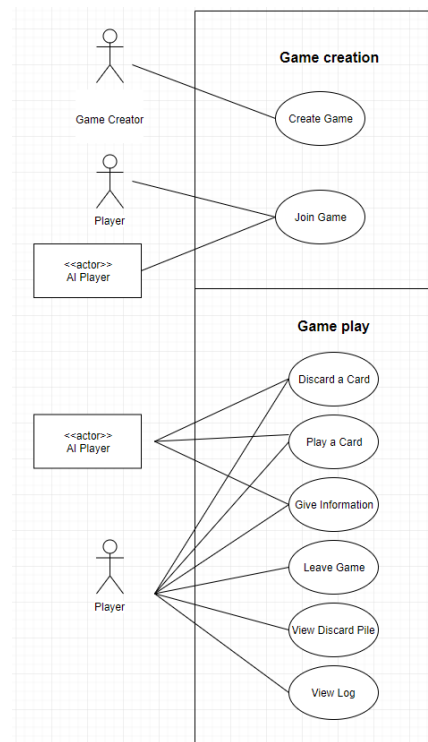
The Hanabi Client will let a Player play this game with other Players over a network connection. Each Player must be running their own copy of the Client on their computer and then connect to a server that handles communication between the Players to join a game that one of them has created. This server already exists and is not described further in these requirements. When enough Players have joined the game, it presents a Player with the information they need to play Hanabi: the other Player's hands, the fireworks piles, the Information and Fuse tokens (out of 8 and 3, respectively), the Discard pile, and the information they have been told about their own hands.

### 2.2 Product Functions

The actors and actions of the Hanabi Client can be seen in the UML diagram to the right. The UML diagram splits the game and its actors and actions into two states: Game Creation and Game Play.

For the Game Creation state, there are 3 actors: The Game Creator, the human Player, and the AI Player. The Game Creator performs the "Create Game" action to create a new Hanabi game for a given number of Players (from 2 to 5). The Game Creator will also be the first human Player in the game. The Game Creator is then given the Game ID and a Token for the new game, which other players can get from them and use to perform the "Join Game" action, which lets them join the same game as the Game Creator.

Either the Game Creator or Players have the option to add AI Players to the game. This can be done within the Client at the press of a button while Players wait for the game to start. It can



also be done via a command line interface (CLI), where they start another Client process and give it the Game ID and Token as arguments.

Once enough Players have joined the game, the Client moves into its second state: Game Play. At any time during this state, human Players can perform the “Leave Game” action to disconnect from the game and possibly join another one. They can also perform “View Discard Pile” and “View Log” actions to look at the pile of discarded cards or expand and minimize a log with all the plays made during the game. AI Players, however, cannot perform any of these 3 actions.

During a Player’s turn, whether human or AI, they can choose one of 3 actions to make a play: “Discard a Card”, “Play a Card”, or “Give Information”. Discarding a card removes one of the Player’s cards from their hand and draws a new one for them but can only be done if there are less than 8 Information Tokens. Playing a card can be done regardless of token count and places a card from the Player’s hand onto a firework pile of the same color; if it doesn’t start a new pile or numerically continue an existing one, then it is discarded instead, and a fuse is lost. Giving information partially reveals either the number or color of some of another Player’s cards but consumes 1 Information token.

### 2.3 User Characteristics

The Client’s human actors consist of a single user class based on their access privileges on the U of S: U of S Computer Science students or faculty. This is because the server component is part of the Department of Computer Science’s network, and therefore requires U of S credentials for a computer science student or faculty member to connect to it. The Client also must run on the department’s Linux computers, tuxworld, which also require those credentials. There are no other user classes for the Hanabi Client.

### **3. External Interface Requirements**

With an overview of the Hanabi Client given, this section is the start of the specific requirements for the Client. It goes over the User, Hardware, Software, and Communications interfaces that the Client must have or work with to let the Player play Hanabi.

#### **3.1 User Interfaces**

For human Players, the Hanabi Client is presented a graphical user interface (GUI) which primarily takes mouse inputs from a Player via buttons, as well as keyboard inputs so the user can fill in text boxes. It also commonly uses windowed prompts that appear when a button is selected to present more buttons or text boxes where action parameters can be specified. When creating AI Players, however, it will also use a command line interface (CLI) where a Player types in a command (described in Section 4.4) to start another process which handles the AI Player.

The states of the GUI are broken up into 4 screens: The Main Menu, Lobby, Game, and Game Over screens. The Main Menu screen is what a Player sees when they start up the Client or leave a game, and it displays 3 buttons for creating a game, joining a game, and closing the client. When the “Create Game” or “Join Game” buttons are selected, a prompt appears asking for the information required for those actions, which is entered by the Player into text boxes. Those prompts have buttons for “Create” or “Join”, “Back”, and “Quit”; “Back” cancels the action and goes back to the Main Menu while “Quit” cancels it and closes the Client.

Once a game is created but before it starts the Lobby screen is shown, which shows which Player slots have been filled so far both as a fraction of current over maximum Player slots and as blank avatars with names. It also displays the Game ID and Token for the game in the corner, which the Players can give to other Players so that they can join the game as well. AI Players can also be added here with the “Add Computer” buttons. The “Start” and “New Game” buttons let the Game Creator start the game early with fewer Players or close the game and start a new one. The “Back” button lets any Player disconnect from the game.

The Game screen is then shown once enough Players have joined a game for it to start. It has several different possible appearances depending on the total number of Players in the game, but they all share similar visual and interactive features. Each Player’s hand is displayed as a row of 4 or 5 cards. The Player only sees the backs of their cards but can see the front of the other Player’s cards, displaying a number and color. The color and number of the Player’s own cards are revealed as information is given to them by other Players. The Firework piles are then displayed near the center face up, and the Information and Fuse tokens are displayed off to the side and greyed out when they are used during game play. The Discard and Draw piles are also shown off to the side and are face down. The corners of the screen then have an Action Log, which is either a single line or a box that displays multiple lines or game actions, a button for toggling the size of the Action Log, and a button for leaving the game after the Player confirms that in a prompt.

While minimizing and expanding the Action Log and leaving the game are done with buttons, other game actions are taken by selecting the cards themselves. A Player can view the Discard pile at any time by selecting it’s visual, which displays a prompt with the cards in the Discard pile. The main actions of playing and discarding cards and giving information are done by selecting a card visual in one of the

Player's hands. If it is one of the Player's own cards, a prompt appears asking whether to play or discard it, though the Discard button will be greyed out and unresponsive if there are all 8 Information tokens available. If it is one of the other Player's cards and there is at least 1 Information token available, then a similar prompt appears asking whether to tell them about the color or number of that card and similar cards in their hand.

When the game ends, whether because of a Player disconnecting from the game or natural end conditions like running out of fuses or finishing all 5 fireworks piles, then the screen displays a Game Over screen. This screen only shows a prompt with the game's score and a button that disconnects the Player from the game and takes them back to the Main Menu screen.

Appendix A shows example screens of what the GUI described above will look like.

### 3.2 Hardware Interfaces

The Hanabi Client will support the Linux computers in the U of S Department of Computer Science, tuxworld. This usage will include running on physical machines with Linux in the computer labs as well as running it on tuxworld via a remote connection. It will require that a mouse, keyboard, and display be available on the user's machine so that human Players can give mouse-based input, type in arguments for creating and joining games, and see the Client's GUI.

### 3.3 Software Interfaces

The Hanabi Client is constrained to being written with Java 8; this necessitates that the user's machine must have the Java Runtime Environment (JRE 8) installed for the Client to run. This is the Client's only specific software requirement.

### 3.4 Communication Interfaces

To communicate with the pre-existing server component of the overall Hanabi system, the user's system will need a network connection that can access the U of S network where the server resides. The specific network connection that exists for this purpose will not matter to the Hanabi Client, so long as server communication is possible and reliable.

## **4. Functional Requirements**

With the Hanabi Client's interface requirements described, this section contains a list of the Hanabi Client's functional requirements. Each function is briefly described before the main sequence of actions for that function's use case is given, along with any alternative sequences that can occur when using it. Any preconditions, postconditions, and error sequences for the function are also given.

### **4.1 Close Client**

Description: At the Main Menu screen, the Player decides to close the Client and selects the "Quit Game" option. The Client then closes.

Preconditions:

- The Player is at the Main Menu screen

Postconditions:

- The Client is no longer running

Main Sequence:

1. The Player selects the "Quit Game" option
2. The Client closes and stops running

Alternative Sequences: None

Error Sequences: None

### **4.2 Create Game**

Description: At the Main Menu screen, a Game Creator decides to create a new game and selects the "Create Game" option. A new "Game Settings" prompt appears where they give the number of total Players (from 2 to 5), the time-out period (in seconds, at least 1), and their own U of S NSID. The Game Creator selects the "Create" option in the "Game Settings" prompt. A new game is then made with the specified number of Player slots and time-out period.

While in an inactive game, the Main Menu screen changes to a Lobby screen that displays the Game ID and Token for the game along with the Player slots that have been filled. The Game Creator then gives these to any human Players who want to join the game and waits for them to join. Once enough Players have joined, the game starts and the screen changes to the Game screen.

Alternatively, the Game Creator can choose to create another game after making a game but before playing it and selects the "New Game" option on the Lobby screen. They are given a prompt asking if they are sure they want to start a new game, where they select the "Yes" option. They then disconnect from the old game, the screen changes back to the Main Menu, and the process for starting a new game proceeds the same as creating a game from the Main Menu.

Preconditions:

- The Game Creator is at the Main Menu screen



#### Postconditions:

- A new game is created with a specified Player count and time-out period
- The Game Creator is in an inactive game

#### Main Sequence:

1. The Game Creator selects the "Create Game" option on the Main Menu screen
2. A new "Game Settings" prompt appears, displaying buttons and text boxes for the Player to give the number of Players, the time-out period, and their NSID
3. The Game Creator selects a button for the Player count and types the time-out period and their NSID into the corresponding text boxes
4. The Game Creator selects the "Create" option in the "Game Settings" prompt
5. The game is created
6. The Main Menu screen changes to the Lobby screen, which displays the Game ID and Token for the new game
7. The Game Creator creates any AI Players they want and make them join the game with the Game ID and Token
8. The Game Creator gives the Game ID and Token to other human Players and waits for them to join the game
9. When enough Players have joined the game, the game starts screen changes to the Game screen

#### Alternative Sequences:

- During Step 8, if the Game Creator selects the "Create Different Game" option on the Lobby screen before the game starts, then the following sequence occurs, after which the Main Sequence begins at Step 2:
  1. A prompt appears asking the Game Creator if they want to create another game
  2. The Game Creator selects the "No" option on that prompt
  3. The game is closed, and the Game Creator disconnects from the game
  4. The screen changes to the Main Menu screen
- If enough AI Players are made and join the game during Step 7, then Step 8 is skipped and the game starts as in Step 9.

#### Error Sequences:

- During Steps 3-4, if the Game Creator instead selects the "Back" option, then the prompt closes and the screen goes back to the Main Menu.
- When creating a game from the Lobby screen (see the first Alternative Sequence), if the Game Creator instead selects the "No" option on the confirmation prompt, the prompt closes and Step 8 of the Main Sequence continues.
- If the game settings that are given during Steps 3-4 are invalid (too few or too many Players, less than 1 second time-out period, or non-numeric values), then the "Game Settings" prompt stays, and a message appears saying that the game settings are invalid. Steps 3 and 4 are then repeated.

- If the new game fails to be created during Step 5, then the screen changes to the Main Menu screen and a prompt appears saying that the game could not be created.
- If it takes longer than 10 times the time-out period for enough players to join the game during Step 8, then the game is automatically closed, and the Game Creator is disconnected from the game. The screen changes to the Main Menu screen and a prompt appears saying that it took too long for enough Players to join.
- At the Lobby screen, if the Game Creator selects the “Back” button, then they will disconnect from the game and go back to the Main Menu.
- If any of the other Players disconnects from the game at any time, then the game is closed, and the Player is also disconnected from the game. The screen changes to the Main Menu screen and a prompt appears saying that a Player disconnected, and the game was closed.

### 4.3 Join Game (Human)

Description: A Player decides to join a game. They contact a Game Creator and get the Game ID and Token for the game they have created, and then select the “Join Game” option. A new “Specify Game” prompt appears where they enter the Game ID, Token, and their U of S NSID. The Player selects the “Join” option, and they are connected to the same game as the Game Creator. The Player then waits for enough other Players to join the game, whereupon the game starts and the screen changes to the Game screen.

Preconditions:

- The Player is at the Main Menu screen

Postconditions:

- The Player is in an inactive game

Main Sequence:

1. The Player gets the game’s Game ID and Token from the Game Creator
2. The Player selects the “Join Game” option on the Main Menu screen
3. A new “Specify Game” prompt appears, displaying blank text boxes for the Game ID, Token, and their NSID
4. The Player types the Game ID, Token, and their NSID into the corresponding text boxes
5. The Player selects the “Join” option in the “Specify Game” prompt
6. The Player is connected to the game
7. The screen changes to the Lobby screen
8. The Player waits for enough other Players to join the same game
9. When enough Players have joined the game, the game starts screen changes to the Game screen

Alternative Sequences: None

Error Sequences:

- During Steps 4-5, if the Player instead selects the “Back” option, then the prompt closes and the screen goes back to the Main Menu.

- During Steps 4-5, if the Player instead selects the “Quit” option, then the Client closes.
- If the Player fails to connect to the game during Step 6, then the screen changes to the Main Menu screen and a prompt appears saying that they could not join the game.
- If it takes longer than 10 times the time-out period since the game’s creation for enough players to join the game during Step 8, then the game is automatically closed, and the Game Creator is disconnected from the game. The screen changes to the Main Menu screen and a prompt appears saying that it took too long for enough Players to join.
- At the Lobby screen, if the Player selects the “Back” option, then they will disconnect from the game and go back to the Main Menu.
- If any of the other Players disconnects from the game at any time, then the game is closed, and the Player is also disconnected from the game. The screen changes to the Main Menu screen and a prompt appears saying that a Player disconnected, and the game was closed.

#### 4.4 Join Game (AI)

Description: While in an inactive game, a Game Creator or Player decides to add an AI Player to their game. They select the “Add Computer” option on the Lobby screen and a new AI Player is created, takes the Game ID and Token, and joins the game. The AI Player then waits for enough Players to join the game, when it starts, and the AI waits for and responds to requests for a move until the game ends.

Alternatively, they open a command line terminal and start another Client process, giving the Game ID and Token of the game as arguments. An AI Player is then made and joins the game.

Preconditions:

- The Game Creator or Player is in an inactive game

Postconditions:

- An AI Player is created and joins the game

Main Sequence:

1. The Game Creator or Player selects the “Add Computer” option
2. An AI creator is created, gets the Game ID and Token from the Client, and joins the game
3. The AI Player waits until enough Players have joined the game
4. When enough Players have joined the game, the AI Player waits for its turn

Alternative Sequences:

- If the Game Creator or Player instead creates an AI from the command line, then sequence then becomes:
  1. The Game Creator or Player starts up a command line terminal
  2. The Game Creator or Player, using the Game ID <gid> and Token <tok> starts another Client process with the following syntax
    - `hanabiClient -g <gid> -t <tok>`
  3. In the new Client process, an AI Player is made and joins the game
  4. The AI Player waits until enough Players have joined the game
  5. When enough Players have joined the game, the AI Player waits for its turn

#### Error Sequences:

- If the process fails to connect to the game during Step 2 of the Main Sequence or Step 3 of the Alternative Sequence, then the process will end, and the game will not have an AI Player join in.
- If it takes longer than 10 times the time-out period since the game's creation for enough players to join the game while the AI waits for other Players, then the game is automatically closed, the AI Player is disconnected from the game, and its process closes.
- If any of the other Players disconnects from the game at any time, then the game is closed, the AI Player is disconnected from the game, and its process closes.

#### 4.5 Discard a Card

Description: During their turn of a game, a Player decides to discard a card and then chooses which of their own cards to discard. That card is discarded, an information token is earned, and a new card is drawn for the Player. The other Players are told which card was discarded before play passes to the next Player.

#### Preconditions:

- The Player has the current turn
- There are less than 8 information tokens

#### Postconditions:

- The Player is waiting for their next turn
- There is at least 1 information token
- The discard pile has 1 more card

#### Main Sequence:

1. The Player is prompted to take an action
2. The Player selects one of their cards to discard
3. A prompt appears asking the Player whether they will Play or Discard that card
4. The Player selects the "Discard" option
5. The other Players are told about which card was discarded
6. The game state changes:
  - A. The Player's hand loses their discarded card
  - B. The discard pile has the discarded card placed on top
  - C. The information token count is increased by 1
  - D. The Player draws a card and their hand gains the drawn card
  - E. The draw pile has the top card removed
7. The Player waits for their next turn, and the next Player takes their turn

#### Alternative Sequences:

- If the Player takes longer than the time-out period (specified by the Game Creator) to do Steps 1-4, then an AI will make an action for the Player. They will select one of the Player's cards to discard and discard it for them, thereby skipping Steps 2-4.

- During Step 4, if the Player instead selects the “Cancel” option, then the prompt disappears, and the Main Sequence returns to Step 1.
- If the draw pile is empty after drawing a card, then all Players get 1 more turn before the game ends.
- If the draw pile is empty before drawing a card, then no card is drawn, and Steps 6.D and 6.E are skipped. The Player will get no more turns before the game ends.
- After Step 6, if all Players have taken their last turn, then the game ends. The sequence from Step 7 then becomes:
  7. The screen changes to the “Game Over” screen, displaying the game’s score in a prompt
  8. The Player selects the “Main Menu” option in the prompt
  9. The Player leaves the game, going back to the Main Menu screen
- At Step 8 of the previous alternative, if the Player selects the “Quit” option instead, then Step 9 becomes:
  9. The Player leaves the game, and the Client closes

#### Error Sequences:

- If another Player disconnects from the game at any point during the main sequence, then the game ends prematurely. The sequence will then jump to Step 7 of the last alternative sequence.

#### 4.6 Play a Card

Description: During their turn of a game, a Player decides to play a card and then chooses which of their own cards to play. If the played card starts the new firework pile for its color or continues its color’s firework pile, then it goes on that pile and the Player draws a card; otherwise it is discarded, a fuse token is lost, and the Player still draws a card. The other Players are told about this action before play passes to the next Player.

#### Preconditions:

- The Player has the current turn
- There is at least 1 fuse token

#### Postconditions:

- The Player is waiting for their next turn
- One of the fireworks piles or the discard pile has 1 more card

#### Main Sequence:

1. The Player is prompted to take an action
2. The Player selects one of their cards to play
3. A prompt appears asking the Player whether they will Play or Discard that card
4. The Player selects the “Play Card” option on the Game screen
5. The other Players are told about which card was played
6. The game state changes:

- A. The Player's hand loses their played card
  - B. The played card is placed on top of its color's firework pile
  - C. The Player draws a card and their hand gains the drawn card
  - D. The draw pile has the top card removed
7. The Player waits for their next turn, and the next Player takes their turn

#### Alternative Sequences:

- If the Player takes longer than the time-out period (specified by the Game Creator) to do Steps 1-3, then an AI will make an action for the Player. They will select one of the Player's cards to play and play it for them, thereby skipping Steps 2-4.
- During Step 4, if the Player instead selects the "Cancel" option, then the prompt disappears, and the Main Sequence returns to Step 1.
- In Step 6.B, if the played card was a 5 and there weren't 8 information tokens at the start of the action, then another step is added in changing the game state:
  - 6.E. The information token counter is increased by 1
- If the played card doesn't start a new firework pile (i.e. it is not a 1 in a color that doesn't have a firework pile yet) and doesn't continue an existing firework pile (i.e. isn't the next number of its firework pile), then Step 6.B becomes:
  - 6.B.i. The discard pile has the played card placed on top
  - 6.B.ii. The fuse counter is decremented by 1
- If the draw pile is empty after drawing a card, then all Players get 1 more turn before the game ends.
- If the draw pile is empty before drawing a card, then no card is drawn, and Steps 5.D and 5.E are skipped. The Player will get no more turns before the game ends.
- After Step 6, if all Players have taken their last turn or there are 0 fuses left, then the game ends. The sequence from Step 7 then becomes:
  - 7. The game's score is calculated and displayed to the Player in a prompt
  - 8. The Player selects an "OK" option in the prompt
  - 9. The Player leaves the game, going back to the Main Menu screen
- At Step 8 of the previous alternative, if the Player selects the "Quit" option instead, then Step 9 becomes:
  - 9. The Player leaves the game, and the Client closes

#### Error Sequences:

- If another Player disconnects from the game at any point during the main sequence, then the game ends prematurely. The sequence will then jump to Step 6 of the last alternative sequence.

### 4.7 Give Information

Description: During their turn of a game, a Player decides to give information about another Player's cards, and then selects both the Player and the card property (either a color or a number) to tell them about. One information token is used up and the given property is revealed to the other Player about their hand. The other Players are told about which Player was given information and what that information was. Play then moves to the next Player.

#### Preconditions:

- The Player has the current turn
- There is at least 1 information token.

#### Postconditions:

- The Player is waiting for their next turn
- There is 1 less information token

#### Main Sequence:

1. The Player is prompted to take an action
2. The Player selects one of the other Player's cards to give information about
3. A prompt appears asking the Player whether to give information about the Color or Number of the selected card
4. The Player selects either the "Color" or "Number" option in the prompt
5. The other Players are told about which Player was given information and what that information was
6. The other Player has a property about some of their hand revealed to them based on the information given to them
7. The game state changes:
  - A. The information token count decreased by 1
8. The Player waits for their next turn, and the next Player takes their turn

#### Alternative Sequences:

- If the Player takes longer than the time-out period (specified by the Game Creator) to do Steps 1-4, then an AI will make an action for the Player. Instead of giving information, they will select one of the Player's cards to discard and discard it for them, thereby jumping to Step 5 of the "Discard a Card" Main Sequence (in Section 4.5).
- During Step 4, if the Player instead selects the "Cancel" option, then the prompt disappears, and the Main Sequence returns to Step 1.
- After Step 7, if all Players have taken their last turn, then the game ends. The sequence from Step 8 then becomes:
  8. The game's score is calculated and displayed to the Player in a prompt
  9. The Player selects an "OK" option in the prompt
  10. The Player leaves the game, going back to the Main Menu screen
- At Step 9 of the previous Alternative Sequence, if the Player selects the "Quit" option instead, then Step 10 becomes:
  10. The Player leaves the game, and the Client closes

#### Error Sequences:

- If another Player disconnects from the game at any point during the main sequence, then the game ends prematurely. The sequence will then jump to Step 7 of the last alternative sequence.

#### 4.8 Leave Game

Description: Anytime during a game, a Player decides to leave the game and selects the “Exit Game” option. They are disconnected from the game and are taken back to the Main Menu screen

Preconditions:

- The Player is in an inactive or active game
- The Player is not an AI Player

Postconditions:

- The Player is no longer in a game
- The Player is on the Main Menu screen

Main Sequence:

1. The Player selects the “Exit Game” option on the Game screen
2. The Player disconnects from the game
3. The Player is taken to the Main Menu screen

Alternative Sequences: None

Error Sequences: None

#### 4.9 View Discard Pile

Description: Anytime during a game, a Player decides to view the cards in the discard pile and selects the Discard Pile. A prompt appears with a listing of the cards discarded so far in the game. The Player then selects “OK” in the prompt to close it.

Preconditions:

- The Player is in an active game
- The Player is not an AI Player

Postconditions:

- None

Main Sequence:

1. The Player selects the Discard Pile on the Game screen
2. A new prompt appears with the current list of discarded cards displayed
3. The Player selects the “OK” option, closing the prompt

Alternative Sequences: None



Error Sequences:

- If the game ends while the prompt is open, it will close automatically before the screen changes to the Game Over screen.

#### 4.10 View Action Log

Description: Anytime during a game, a Player decides to view the actions taken during the game and selects the “Log” option. The Action Log then expands on the Game screen and displays a list of all the actions taken so far in the game. The Player then selects the “Log” option again to minimize the Action Log again.

Preconditions:

- The Player is in an active game
- The Player is not an AI Player
- The Action Log is not already expanded

Postconditions:

- None

Main Sequence:

1. The Player selects the “Log” option on the Game screen
2. The Action Log expands within the empty space around it on the Game screen, displaying the current list of game actions
3. The Player selects the “Log” option again, minimizing the Action Log

Alternative Sequences: None

Error Sequences: None

## **5. Non-functional Requirements**

Besides the interface and functional requirements of the Hanabi Client, there are some other non-functional requirements that exist to make game play possible and enjoyable. This section goes over those non-functional requirements.

### **5.1 Performance Requirements**

With regards to performance, the AI Players should not be a cause of delay when playing the game; they should be able to make a play for themselves or another Player within 1 second. Changes to the game state that the Player sees should also not delay the game and should happen within 0.5 second.

### **5.2 Reliability Requirements**

For reliability, the overall Hanabi game system is also expected to process Player moves 1 at a time and in their proper order so that every Player has an up to date and correct view of the game state during their turn. It should also use a network connection that is reliable enough that a Player will not disconnect from a game once they join due to connection issues, since a game is closed when any Player disconnects, making reconnections not possible.

Another reliability concern is ensuring that other Players are told about the correct move. This is done by preventing Players from making invalid Plays, like discarding a card when there are all 8 Information tokens. This prevention involves greying out action buttons and making them unresponsive when they are invalid.

## **6. Conclusion**

This document described the software requirements for a Hanabi Client program. It described the wider Hanabi game system it exists in, which includes a server for Player communication, and its actors and actions. It also went over the User, Hardware, Software, and Communications interface requirements that it needs to function. Finally, it listed the functional and non-functional requirements of the system in detail to let human and AI Players play a full game of Hanabi over a network connection.

## Appendix A: Example Graphical User Interface Screens



Fig. A.1: The Main Menu screen, shown when the Player opens the client. "Create Game", "Join Game", and "Quit Game" buttons let the Player start a new game, join an existing game, or close the Client.



Fig. A.2: The Create Game prompt, shown when the Game Creator selects the "Create Game" option. The buttons let them select the number of Players for the game and the text box lets them state the time-out period for the game. The "Create" and "Back" buttons create a game with the given settings or cancel the action and take the Game Creator back to the Main Menu.



Fig. A.3: The Join Game prompt, shown when the Player selects the “Join Game” option. The text boxes let them input the Game ID and secret Token. The “Join”, “Back”, and “Quit” options let the Player join a game with the ID and Token given, cancel and go back to the Main Menu, or cancel and close the Client.



Fig. A.4: The Game Lobby screen, displaying the game queue for the host and players. “Start” and “Quit” buttons let the Player start the game with fewer Players or leave the game and go back to the Main Menu. The “New Game” button lets a Game Creator forcefully and the current game and create a new game from the lobby. The “Add Computer” buttons lets any Player add an AI Player to the game.

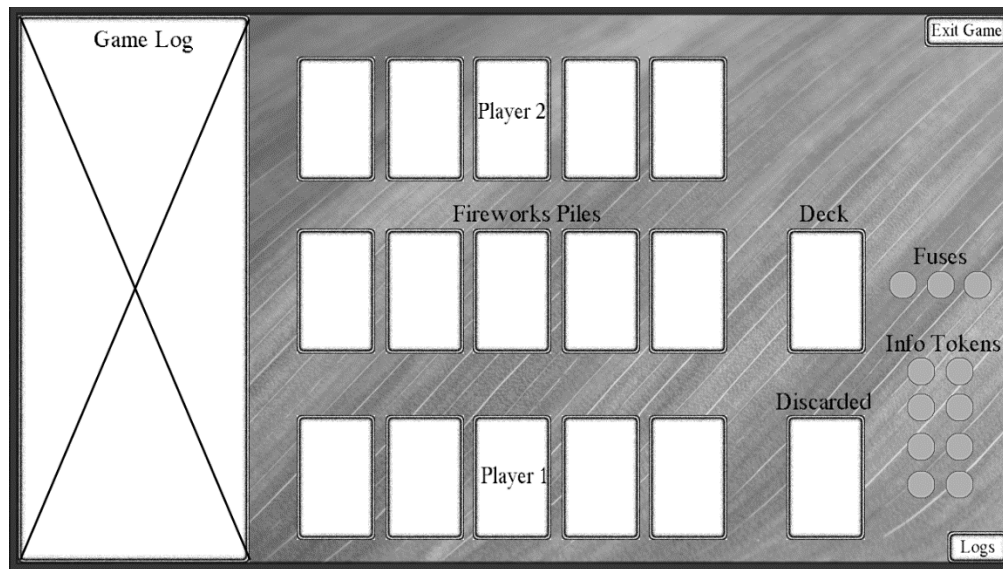


Fig. A.5: The Game screen when there are 2 Players. The other Player's cards are face up while their own cards are face down or partially revealed based on given information. The fireworks piles are shown face up in the center. The Fuse and Information tokens are displayed off to the side, as are face down Draw and Discard piles. The cards act as buttons for making moves (Play Card, Discard Card, Give Information). The "Exit Game" and "Log" buttons let Players leave the game and toggle the size of the Action Log so that game actions are or aren't displayed.

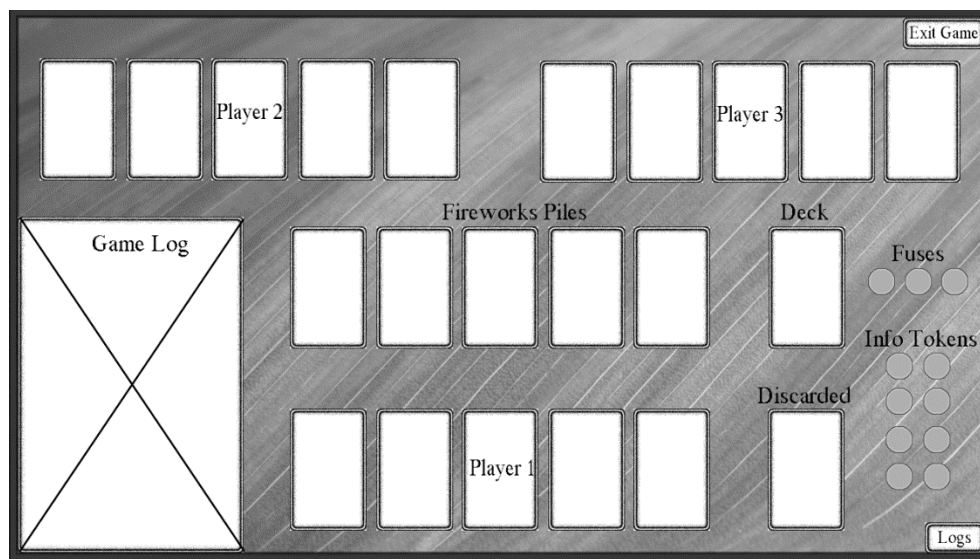


Fig. A.6: The Game screen modified for 3 Players.

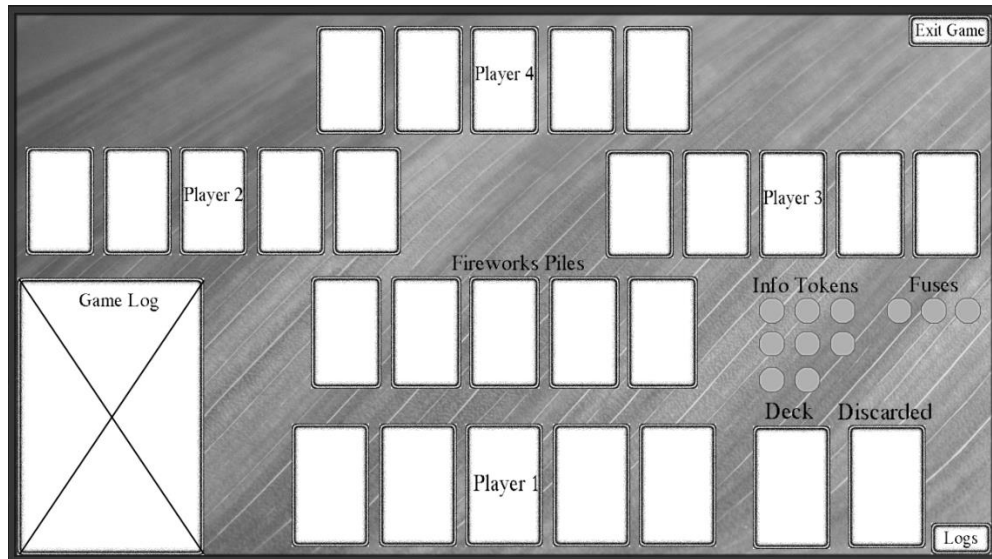


Fig. A.7: The Game screen modified for 4 Players.

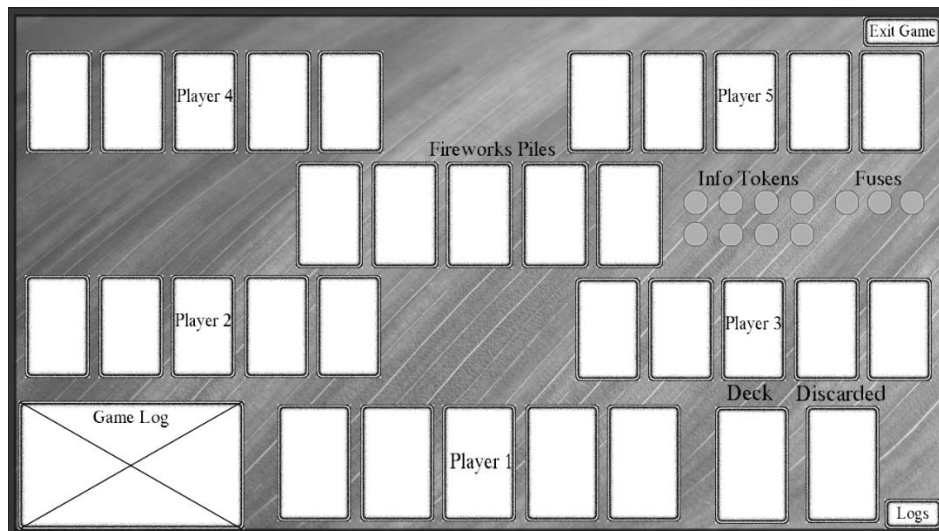


Fig. A.8: The Game screen modified for 5 Players.



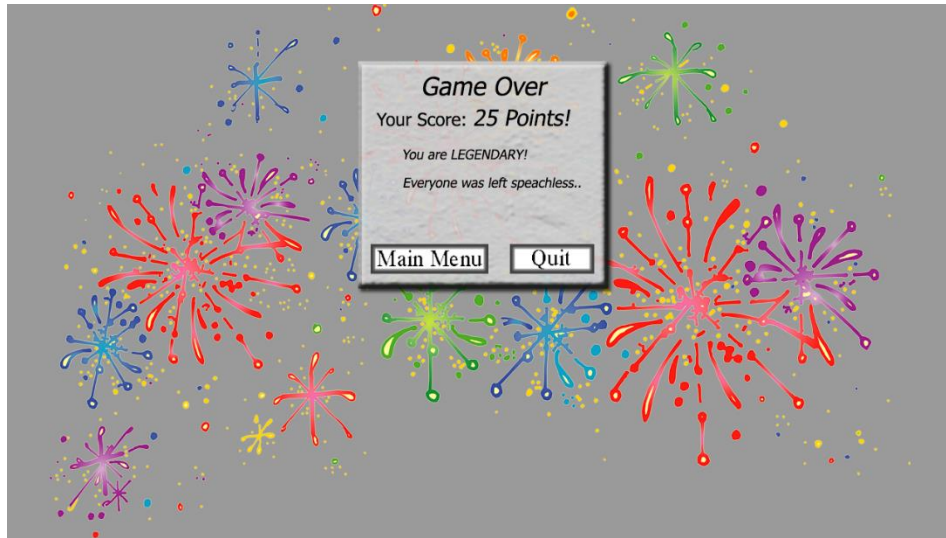


Fig. A.9: The Game Over screen, shown when the game ends due to natural causes or disconnections. A tallied score is shown, along with “Main Menu” and “Quit” buttons that let the Player disconnect and either go back to the Main Menu or close the Client.