Functional Requirements Document

Group 15: Logo

Kohei Arai

Section 3, 3.1-3.4

Kevin Brown

Section 4.1.3-4.3

Christopher Buruchian

Section 4, 4.1.1-4.1.2

Sandra Hoopes

Section 1.5-1.6, 2

Jessica Moreno

Section 1, 1.1-1.4

Eric Smithson

Section 3.5

Version	Description of Change	Author	Date
0.1	Initial document	Sandra Hoopes	2/23/2017
0.2	Completed the introduction, purpose, scope, background, and references sections	Jessica Moreno	2/24/2017
0.3	Completed the user, software, and hardware interface sections	Chris Buruchian	2/25/2017
0.4	Completed UML tables and diagrams for Section 4.	Kohei Arai	2/27/2017
0.5	Completed functional requirements section.	Eric Smithson	2/28/2017
0.6	Completed the assumptions, constraints, and overview sections in section 1 and the methodology section. Made minor adjustments to previously completed sections, in light of new information.	Sandra Hoopes	3/2/2017
0.7	Completed computer interface, data conversion requirements, and hardware/software requirements. Made minor adjustments to previously completed sections.	Kevin Brown	3/5/2017
0.8	Completed user requirements and made adjustments to previously completed sections.	Sandra Hoopes and Kohei Arai	3/5/2017

Contents

1 INTRODUCTION

Since its introduction in 2009 by Drumond Park Games in the United Kingdom, the Logo Board Game has been a success. Its recognition draws from testing each player's consumer knowledge, particularly involving product and pop culture logos. Soon after, the toy company Spin Master helped distribute it internationally, where it remains a favorite among ages 12 and older.

1.1 Purpose

The purpose of this project is to create a digital representation of the Logo Board Game so that it can be easily accessible and usable to users via smart device, or computer, dependent upon the implementation chosen.

1.2 Scope

The scope of this functional requirement document is to describe the requirements that are needed to make a fully functioning Logo Board Game for smart devices. The game will be designed for a user to play against up to six players. These other players will either be provided to the user as an artificial player construct or through a multiplayer online server. The level of difficulty of the game can be selected by the user in regards to the artificial players. The interface will allow users (of ages 12 and older) to easily interact with the gameplay system and provide sufficient instructions on how to play the game.

1.3 Background

The team members of this group are classmates from the Spring 2017 semester of the course Human and Technology Interaction (CAP 4104). A team effort has been agreed upon to complete these requirements. Each member is responsible for providing specific parts of the requirements. However, if help is needed by any individual, the rest of the team will assist as necessary.

1.4 References

The development team will conduct meetings online via a communication medium, such as Slack or GroupMe. Discussions regarded each team member's responsibilities will be documented by the team lead or a designated individual.

The following links are references for the game which the development team intends to create:
https://en.wikipedia.org/wiki/Logo_Board_Game
http://spinmastergames.com/game-detail.php?pid=p10424
https://www.drumondpark.com/rules/logo
https://www.youtube.com/watch?v=CA_WamLCj1w

1.5 Assumptions and Constraints

- User has network capability for distribution of the software.
- User may want to play installed game while offline
- User has device capable of running software.
    - Device has sufficient memory allocation to download application.
    - Device has sufficient memory allocation to run single game.
- (optional) User must have valid email address and password.
- Patent or License must be obtained for Logo Board Game.

1.5.1 Assumptions

Depending on implementation chosen, the user must either have access to an internet-capable computer or smartphone to download the application. Software will be implemented as an electronic game being capable of running on a modern device. Any modern software or hardware platform can be chosen provided it is capable of implementing the design requirements. If the team of developers chooses to implement a login capability, the user must have a valid email address and password. If this capability is implemented, this should allow for user statistics to be saved in a database.

1.5.2 Constraints

The development team will follow procedures in accordance with this documentation. If the information within this documentation is not sufficient to complete the application development, then it will be left to the development team leader's discretion on how to proceed.

If there is a current patent owner of the Logo Board Game, then a license will have to be procured prior to release of software for the Logo Board Game application. Should there be no current patent owner of the Logo Board Game, the development team leader on behalf of the development team will obtain a patent for the Logo Board Game Application prior to the release of the software to the public. These procedures are in place to protect the development team from legal issues that may arise from not obtaining proper licensing.

1.6 Document Overview

This document contains the following sections:

- Section 1: This section includes the purpose, scope and background of the Logo Board Game application, along with the assumptions and constraints expected in its development.
- Section 2: This section explores the methodology intended to be used through the development process.
- Section 3: This section discusses the functional requirements and implementation, through demonstrative diagrams and charts.
- Section 4: This section provides other requirements necessary for the development of the Logo Board Game application software.

2 METHODOLOGY

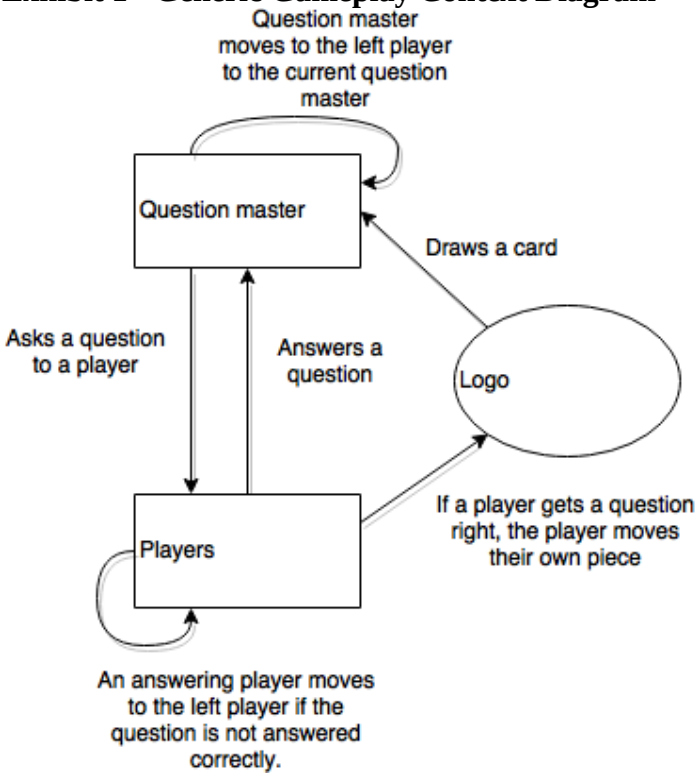
The software development methodology that will be used for this project is Agile with weekly sprints. Each week the development team will discuss the current state of the Logo Board Game application software and develop a list of assignments that will be necessary for the following week of the development process (i.e. documentation, programming goals, testing, etc.). The agile software development methodology should allow the team to adjust for any issues that may occur as they arise.

3 FUNCTIONAL REQUIREMENTS

3.1 Context

The user will request to join a game from the system, providing a Username and the game type (i.e. single player versus artificial constructs or multiplayer online). The Application will acknowledge and process the request. Following this initial start-up process. **Exhibit 1** demonstrates the gameplay dynamic. The "Question Master," or QM, will draw a card from the "Logo" game board. The QM will then ask the answering player the question from the card. If the answering player is the user, then the user will answer the question. If the question is answered correctly, the answering player's game piece will be moved forward on the gameboard. If the question is answered incorrectly, the QM and answering player will be rotated out of their current positions and the players to their respective lefts will take on their roles.

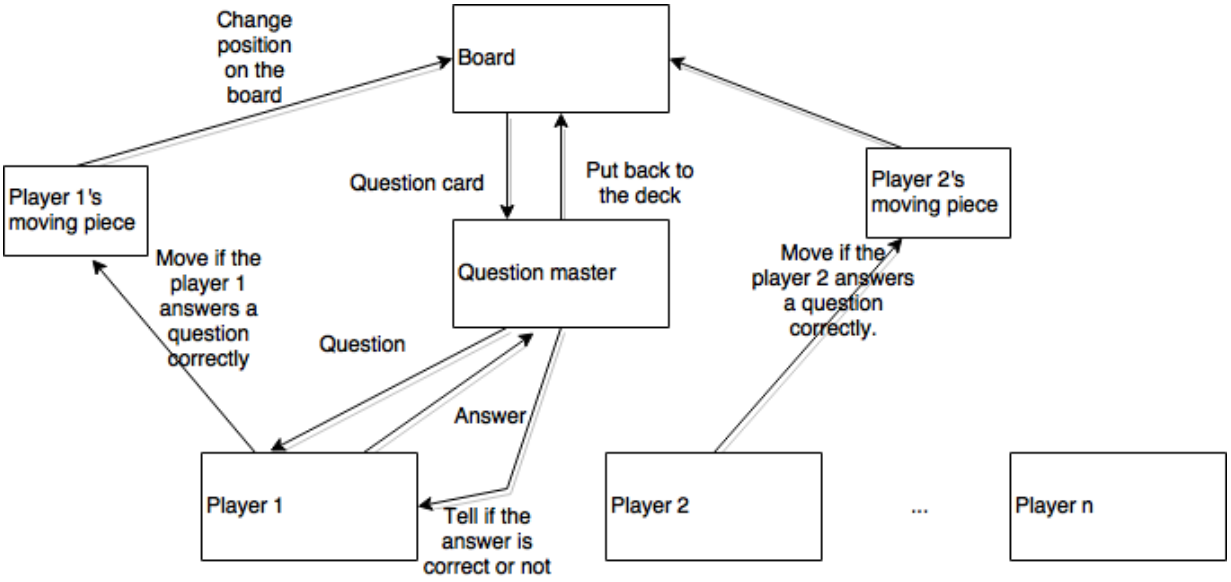
Exhibit 1 - Generic Gameplay Context Diagram



3.2 User Requirements

- Rule page - The user should be allowed access to the rules separate to, but not hindering, the gameplay interface. (i.e. pop-up object, redirect while maintaining active gameboard)
- Clearly defined buttons - The user will need clearly defined clickable objects to answer questions, or draw a card.
- Text Standards - The text size and font should be clearly legible to the user, with little variation.
- Display - Colors of display and game board should be consistent. A variety of colors should be used in different tones. Be aware of color choices to adjust for visual impairments.
- Zoom capability (card) - The user will require the ability to zoom in on the question being asked. This should help players with visual impairments to see the text more clearly.
- Zoom capability (gameboard) - The user should be allowed to view the entire gameboard, or zoom in on their respective piece.

3.3 Data Flow Diagrams



The diagram displays how each of the users or objects interact with each other. The question master draws a question card from the board and asks questions on the card to the player. The card contains four questions, and each question is assigned a color. The colors are purple, green, yellow, and red from the top of the card. The player who is asked the question provides their chosen answer. The question master acknowledges if the given answer is correct or not.

If the player gets the question right, he or she moves their piece to the next colored space matching the color of the question he or she answered. This process continues until the question master consumes all four questions on the card.

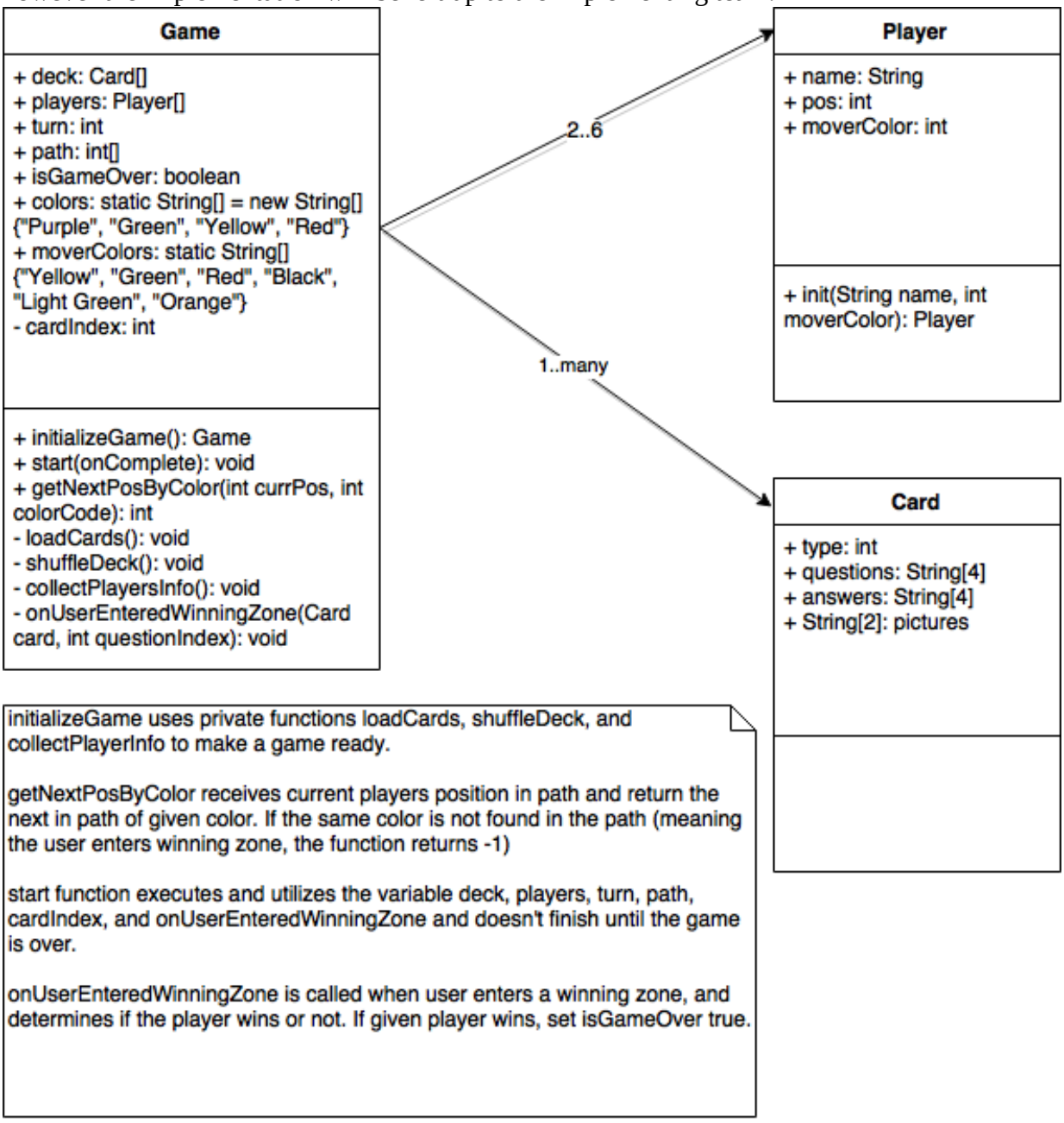
If a player answers the question incorrectly, the answering turn moves to the next player on the left of the current player, and the question is asked to the new answering player. If that player responds with the correct answer, then a new card is drawn and they begin answering questions from the top (purple) question.

The game proceeds until a player reaches the final spot on the gameboard. The process for the the player to successfully win the game is represented below.

- Once a player has moved onto the winning zone, the winning logic applies to determine if the player wins a game or not.
- The player who has moved into the winning zone has to answer correctly two questions in a row or one red question.
- If the player fails to answer the question correctly, the player to the left is given the opportunity to answer the question prior to drawing a new card to complete his or her turn.

3.4 Logical Data Model/Data Dictionary

The following class diagram provides an initial idea of how the program can be accomplished, however the implementation will be left up to the implementing team.



3.5 Functional Requirements

Section	Requirement Definition
FR1.0.	The system shall allow users to choose 2-6 players.

FR1.1	The system shall allow users to pick one colored game piece per player.
FR1.2	The system shall simulate a game board.
FR1.2.1	The board contains one long path of 50 squares that spirals inwards toward the center of the board.
FR1.2.1.1	These first four squares shall be, in order: “PURPLE, GREEN, YELLOW, and RED”
FR1.2.1.2	The remaining 46 squares shall be colored randomly picking from those four colors. The only limitation is that no two adjacent squares be the same color.
FR1.2.2	At the end of the path and in the center of the board there will be a WINNING ZONE
FR1.2.3	At the beginning of the path there shall be a START ZONE
FR1.3	The system shall simulate a stack of cards
FR1.3.1	The cards shall have a front and a back
FR1.3.2	There will be three types of cards
FR1.3.2.1	“Theme Cards” - there shall be a theme associated with these cards, also associated with this card is two logos.
FR1.3.2.2	“Picture Cards” - there shall be a picture associated with this card.
FR1.3.2.3	“Pot Luck Cards” - Questions from this card will not have a common theme or be associated with any one picture.
FR1.3.3	There shall be four questions associated with every card
FR1.3.3.1	Every question, in order, will be associated with these colors: “PURPLE, GREEN, YELLOW, and RED”
FR1.3.4	Every player will be assigned a uniquely colored “Mover”
FR2.0	The system shall simulate the game “Logo”
FR2.1	The system shall start by putting all the colored pieces up on the starting square, and then picking one player at random to receive the first question.
FR2.2	The game shall be split into “rounds”
FR2.2.1	Every round, the system selects a card randomly from the stack without replacement

FR2.2.1.2	If the card is a “Theme Card”, the system must display what the theme is, as well as the picture associated with the theme if there is one.
FR2.2.1.3	If the card is a “Picture Card”, the system must display the picture associated with the card.
FR2.2.1.3	If the card is a “Pot Luck Card”, the system does nothing special and continues displaying the questions.
FR2.2.2.1	The system will display the first question on the card to all the users, but only allow the user whose turn it is to answer.
FR2.2.2.2	If the player gets the answer correct, the system will ask that same player the next question.
FR2.2.2.3	If the player gets the question wrong, the system will ask the next player the next question.
FR2.2.2.4	If all players get the question wrong, the system will display the correct answer and then, if there are more questions left, ask the original player the next question.
FR2.2.2.5	The system must select the player to the left of the player who last correctly answered a question to ask the first question of the next round.
FR2.3.1	The system will advance a player’s mover every time they answer a question correctly
FR2.3.2	When a user correctly answers a question, they advance to the next color of the space which corresponds to the color of the question.
FR2.4	The system shall allow players to enter the “win zone” at the end of the board
FR2.4.1	The system will place a player in the “win zone” when they correctly answer the question, but there is no space of that color left on the path.
FR2.4.2	If a player enters the winning zone, it will still be their turn unless they entered with a red question.
FR2.4.3	If a player is in the winning zone and it is their turn to answer, they must answer two questions in a row or one red question to win the game.
FR2.4.4	If the player in the winning zone gets a question wrong before they win then the round continues as normal.
FR2.5	The system shall end the game when a player wins
FR2.5.1	The system will display the winner of the game on the screen to all the players before exiting.

4 OTHER REQUIREMENTS

If internet connectivity is lost during a multiplayer online game, then the player will be replaced by an artificial player construct. If points or wins are to be awarded, the player that lost connection will not receive those winning points.

4.1 Interface Requirements

Screens -

Start Screen - Initially upon first loading the application, the user will be brought to a start screen menu that will provide the following options:

- Start Game - Clicking this will navigate to the game screen.
- Rules - Clicking this will navigate to a screen that contains a summary of the game rules.

Game Screen - Once the game is started, the user will be prompted to enter the number of players (up to 6). Once the number of players are inputted, they will be prompted to select either online or offline play. Once completed, the appropriate number of game pieces will be shown and a random player will be chosen to go first.

Player Pieces - There should be multiple player pieces that are color-coded such that the player can easily identify his piece. The design on these can be flexible, but they should look something like this: http://qppcdn.appspot.com/mpc/images/site/qp/image012.jpg

Cards - The user interface must support the cards that are vital for player movements through the game.

- **Picture Card -** The picture card consists of pictures that the questions on the back relate to. See this image for an example: http://www.sahmreviews.com/wp-content/uploads/2015/05/Logo-Board-Game-Picture-Card.jpg
- **Potluck Card -** The potluck card consists of general questions based on the LOGO shown to the user. See this image for an example: http://www.sahmreviews.com/wp-content/uploads/2015/05/Logo-Board-Game-Pot-Luck-Card.jpg
- **Theme Card -** The theme card will show a theme that all of the answers have in common. See this image for an example: http://www.sahmreviews.com/wp-content/uploads/2015/05/Logo-Board-Game-Theme-Card.jpg

Board - The board should be a traditional logo board game board that has starting point that spirals into the center winning zone. The boards consists of colored positions. See this image for more details https://i.ytimg.com/vi/-1P0g-X2JFo/maxresdefault.jpg.

4.1.1 Hardware Interfaces

The developers are free to choose the platform that they are most comfortable with to develop this application. The game must be an electronic version that is able to be distributed to end users on modern devices. This is defined as devices manufactured within the last 5 years.

4.1.2 Software Interfaces

The game may support cloud login, however, this is not a strict requirement. The development team is free to choose any application interface in order to develop the game.

4.1.3 Communications Interfaces

The game should allow players to play games in a variety of different game modes. These should include:

- Single Player
- Multiplayer - Single Device
- Networked Multiplayer

4.1.3.1 Game Artificial Intelligence (AI) Players

The AI should be configured in a way such that difficulty of the game could be adjusted based on the player's preference. The technical details of implementation of the AI difficulty is a design choice that shall be left to the team implementing this feature.

The AI shall be able to do the following:

- Select the correct answer to a trivia questions
- Answer incorrectly some of the time to give human players a chance of winning. The percentage of correct to incorrect choices should be a component of the AI difficulty.
- Act as the question master when necessary for the player.

The AI should preferably be able to perform locally without any needing internet connectivity if the platform requirements for such an implementation are feasible. This would allow the end user to play the game in a setting in which internet connectivity is not available such as while traveling in places such as cruise ships, airplanes, or cars. While internet connectivity is more prevalent today than in years past it is still not universal, or in some cases exists but is quite expensive. A game such as logo lends itself well to being played casually in a travel environment so it should be expected to perform in this fashion.

In the event third party Application Programming Interfaces (APIs) such as the Azure Chatbot service are used the AI should still be able to function in an offline state using a fallback functionality. The fallback will likely have to be implemented as a database of all possible answers and behaviors stored locally on the device.

AI Players should be available in all game modes to take the place of a human player in the event someone needs to leave or come back into the game. It must be possible to dynamically replace

an AI player with a human player at any point during the game. It should also be possible to adjust AI difficulty during the game in the event the game is too hard or too easy.

The AI should be balanced and randomized to give a different play experience every time the game is played and should feel as close to a human player as possible. For instance if a player is getting ahead an AI should be in an anxious state, if it it is winning the AI should switch to an excited state. The states should cause changes in the type and tone of answers that are given. Time permitting animations and art for the AI characters should be present on the user so the state is visually present to the user.

Developing a set of AI characters that would each have it's own unique personality in answers given should be considered for the development team if possible within the project timeline. AI personalities could be more or less likely to go into each emotional state and change answer styles depending on the underlying emotion. They may also have unique or special answers that only exist on that character.

In giving incorrect answers the AI should emulate real answers from people and convey emotions in the answer. For instance if an AI doesn't know something the AI might switch to a nervous emotional context and give a funny answer.

For better replayability the inner workings of the AI should be completely transparent to the user in a way that no patterns or workings are discernable. The user must not be able to easily predict what the AI will do. Each game should provide a unique experience to the end user, allowing a long amount of replayability until the user gets bored of the game style and format.

To test replayability a sample of at least 10 games should be played and verified to make sure that the game feels unique on each iteration.

4.1.3.2 Single Player

The Single Player game mode is a key component of the final project deliverable. It is the section that is most likely to be demoed and is an essential feature for the game. The minimum deliverable for the single player game mode shall allow a player to play with between 1-5 AI characters.

The single player game mode shall be implemented to allow a player to play and shall consist of a way for the player to give answers. The game should be able to interpret answers in a way similar to real board game. In the event there are multiple possible meanings the game should be able to ask the user what they meant. If internet connectivity is required to access a natural language processing API then a multiple choice answer system could be substituted. This is a break from the way the game is played in real life however, so true Natural Language Processing (NLP) would be preferable.

A bonus deliverable would be for the user to deliver their answer verbally via text to speech APIs. The should have a fallback however in the event internet connectivity is not available.

4.1.3.3 Multiplayer on Single Device

Hot Swapping should be performed in a way to allow players to share a single computer or other display. This should be setup to support locations in which you might want to play the game connected to a projector, or when you only have one charged device to be shared among many people. This game mode should be the same as the single player game mode but should have different user states. For instance you should be able to pass the device between users when it is the other person's turn.

Due to the nature of this game mode chat and other remote multiplayer features are not necessary.

4.1.3.4 Networked Multiplayer

The game should be able to be played remotely over the internet between two people each in a different physical location, or connected locally with two people in the same room but on multiple devices.

The network multiplayer feature could be implemented in any way that the development team thinks is the best solution but shall include the minimum requirements:

- A feature allowing for multiple players to communicate with each other (via text and/or voice chat)
- A feature allowing players to ask the other player a question remotely
- A feature allowing players to join in and take the place of an AI player
- A feature allowing players to disconnect from a network interruption and reconnect to the same game
- A feature allowing players to pause a game to wait for someone to return
- A feature allowing players to remove a player from a game who is misbehaving
- A feature allowing players to see technical networking statistics such as ping

4.2 Data Conversion Requirements

The game should allow saving of game states to disk and should allow these games to be imported and resumed from the save game. It is not necessary for the game to retain backwards compatibility with game saves made with previous versions of the game.

4.3 Hardware/Software Requirements

The hardware requirements will vary depending on the technologies used, but the game should be playable on a resolution high enough to not require significant downgrading of the artwork and be playable at a minimum frame rate of 24 frames per second. The game should be able to play in networked mode with no major apparent jitters or lag.

The recommended hardware requirements shall allow a player to always play the game at a minimum of 60 frames per second or higher.

The software requirements are left to the team implementing the project requirements but should require software that is widely available to general consumers without excessive licensing fees.