1 Executive Summary

Our team will be working to create an application that implements the board game Sorry! This document will specify certain guidelines and rules which will assist us in the process of achieving our goal of a fully functional system. The user will be able to play the game with the computer as the opponent. Standard game rules will apply, and the deck of cards will be identical to that used in the original board game. Each game will begin with a shuffled deck of cards, and ends when either the user or the computer wins the game. The game will be played using a web interface, and the graphical user interface will allow the user to see the game board, as well as other related information and functions. A general outline of our proposed GUI is included, as well as different tasks that must be completed in order to create the system.

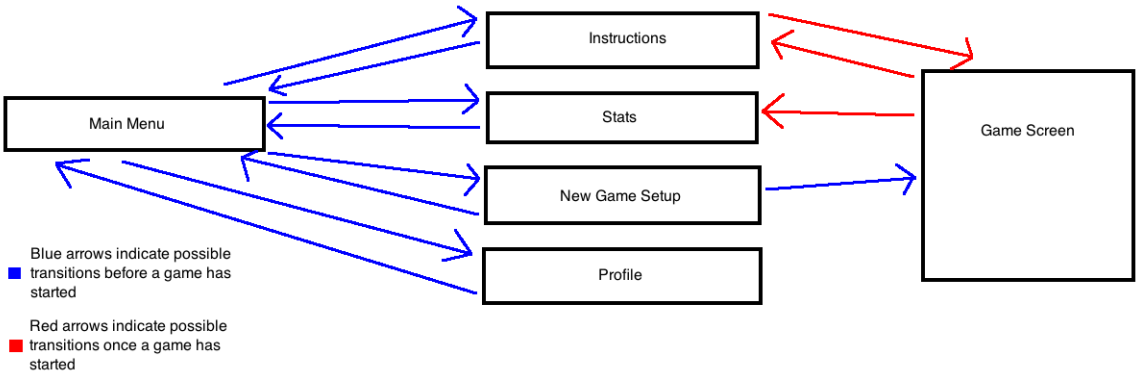

Figure 1: The overall structure of the program.

2 Description of Functionality

2.1 Overview

The basic functionality of the program is best described by a set of screens with associated states (see Figure 1). Each of these screens represents a user task and transitions between screens are tightly constrained to prevent inappropriate contexts from instantiating. For example, the core Game Screen shall only be accessed by first traversing the New Game Screen and may only be persistantly exited when the game ends or is aborted. Detailed descriptions of the functionality of each screen are found below.

Note that plans are to instantiate all parts of the program as Java web applet.

2.2 Main Menu Screen

Figure 2: Draft of Main Menu Screen with functionality provided by clickable buttons.

2.2.1 Summary

The Main Menu screen shall be used to control meta game operations i.e. user activities not actually related to playing the game.

2.2.2 Description

The Main Menu screen shall contain a list of clickable items which transition the display state to another screen. The options will include:

- Starting a new game (New Game Screen)
- Managing profiles (Profile Screen)
- Viewing the a compilation of play statistics (Statistics Screen)
- Viewing game instructions (Instructions Screen)
- Exiting the program (Terminates Program)

This screen will also be the logical return point for many other screens (see flowchart in Figure 1 for details).

2.3 Statistics Screen

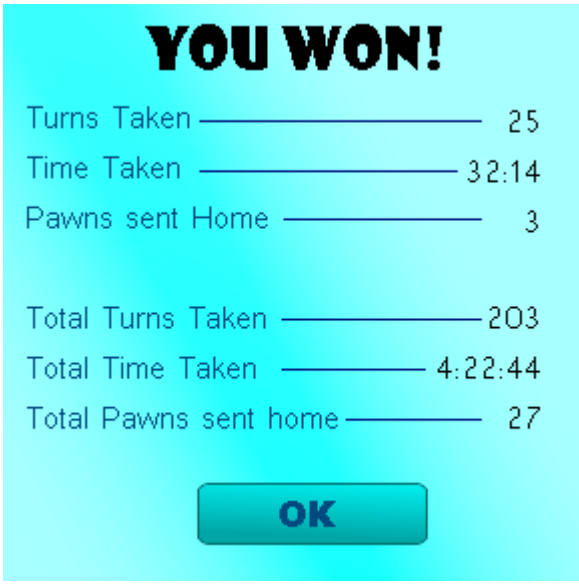

Figure 3: The statistics screen. Note that the "You Won" message may be "You Lost" and that the Win/Loss message and single game statistics will be absent if this screen was entered from Main Menu.

2.3.1 Summary

The statistics screen will display saved information about past games. The information displayed shall be context sensitive based on the previous screen and state.

2.3.2 Description

This screen will be reached via either the game screen (on victory) or the main menu screen. Its content shall be determined contextually based on the previous screen and internal state of the program. If this screen has been called from the game screen it displays the following information about the just completed game:

- Did the player win or lose? (The art department wants to put a splash image here.)
- Number of turns taken
- Time taken
- Pawns bumped, slid, and swapped

This screen will also display the following content for the selected player if a player profile has been loaded:

- Win/Loss ratio
- Average Play Time & Turns
- Total Time & Turns Played
- Total Pawns bumped, slid, and swapped

This screen will contain a single interactive object: a continue button which takes the player to the Main Menu Screen.

2.4 Instructions Screen

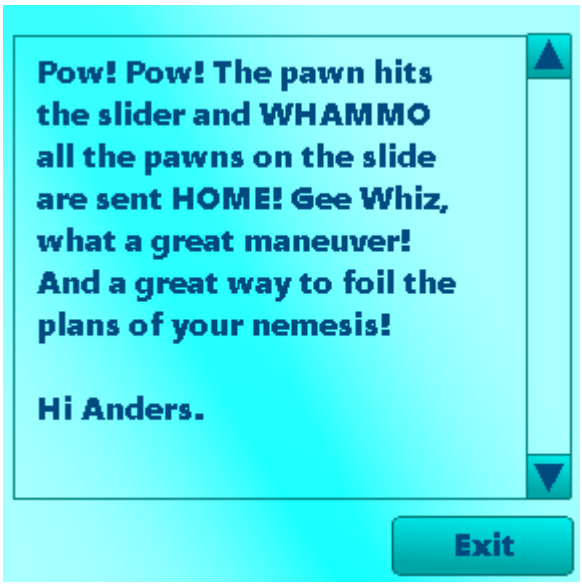

Figure 4: The instructions screen.

Caption: The instructions screen.

2.4.1 Summary

The instructions screen shall be a scrollable display of written instructions for using the program.

2.4.2 Description

This screen will contain provisions to read a set of static text instructions. This will involve either paging or scrolling through multiple screens of text. The screen shall also contain a button to exit the instructions. This exit button's functionality shall be context sensitive and will return the program to the prior display state (Either the Main Menu or Game Screen, see Figure 1.)

2.5 Profile Screen

Figure 5: The profile screen with two saved profiles ("Matt" and "Stephanie") and three empty profiles.

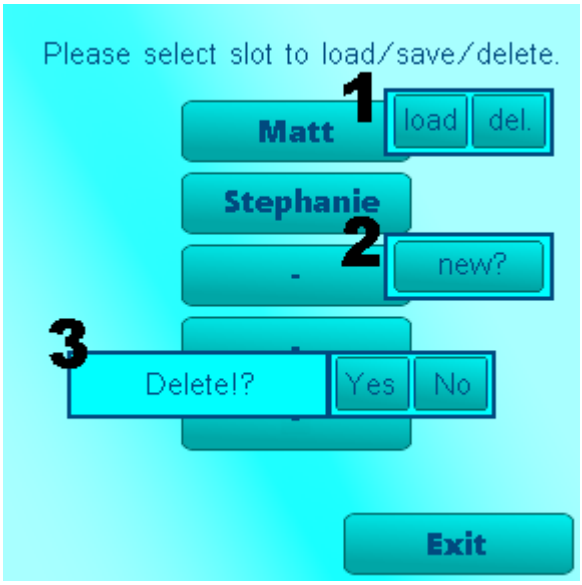

Figure 6: The profile screen with the load/delete prompt active (1), the create new profile prompt active (2), and the delete confirmation prompt active (3). During actual execution only one prompt may be open at a time.

2.5.1 Summary

The profile screen shall be used to create a profile, delete an existing profile, or load an existing profile.

2.5.2 Description

The profile screen will contain a finite number of clickable buttons. Each of these buttons will correspond to a single data file associated with a profile. Profiles will either be empty or active (as determined by their associated data file). The purpose of the data files shall be to provide persistent storage of all play statistics generated by a single user (see Statistics Screen section for examples of play statistics). The function of each button shall be context sensitive based on the current status of the associated data file:

- Data File Null: Prompts the user for a name to label the profile with and creates a new associated data file.
- Data File Not Null: Prompts the user for one of two options:
  - Load: Makes the data file associated with the selected profile active. This means that play data from any games played will be saved to this profile's file.
  - Delete: After an "Are you Sure?" prompt deletes the data file associated with this profile and makes it null.

This screen can only be reached via the Main Menu Screen and its back button returns the user to that screen.

Design Note: The current specification calls for local storage of save data as is customary with many in browser Flash games. In the event that this specification changes (for example to server side storage as with many Apps) the description of this screen may change substantially.

2.6 New Game Screen

Figure 7: The setup screen form.

2.6.1 Summary

The new game screen will prompt the user for several settings required to start a new game of "Sorry".

2.6.2 Description

This screen shall be laid out as a standard form. It will set the following variables required to start a new game:

- Primary player's color: The other colors and sequence of play will be determined algorithmically (i.e. clockwise).
- Number of human players: Primarily intended to permit all AI vs AI play for testing purposes.
- Total number of players

In the event that early testing proceeds well the design team had identified the following potential areas where functionality may be extended:

- Difficulty: Defined as AI competence. Would probably involve letting the AI cheat a little.
- Turn Limit: Would permit some points or position based victory determination at a fixed turn.
- Number of Pawns per Player
- Theme: A purely cosmetic setting that would change the look of the board and the pawns.

Note that all inputs shall be designed to make invalid options impossible to select. This shall be achieved via context sensitive radio buttons and sliders (max humans $\leq$ max players for example).

This screen may only be entered from the Main Screen. Its cancel button may be used to return to the Main Screen or its Continue button may be used to create a new game state in the Game Screen using this screen's form values.

2.7 Game Screen

2.7.1 Summary

The Game Screen is the primary screen and contains a large number of interactive areas. Please refer to Figure 8 for a visual reference.

2.7.2 UI Description

The Game Screen UI shall be composed of two main interactive areas: the deck and the board. The deck area will consist of a visual representation of a face up and a face down pile of cards. At the beginning of their turn a human player will click on the face down pile to model "drawing" a new card. Once the card has been drawn all possible, valid moves will be shown on the board. Valid moves shall be identified by highlighting all pawns on the board which may be moved and then once a pawn is selected by highlighting all spaces to which that pawn may be moved. In the event that no moves are possible the human player will be instructed to click anywhere on the board to forfeit their turn. Once the move is completed upkeep is performed (bumped pawns sent home, slid pawns advanced, etc.) and the turn is ended. Play then proceeds to the next player. The sequence of play is the same for AI players except that they interact with the deck and board via the back end instead of via the UI.

The Sequence of Play

```
1  while !gameover do
2      for eachplayer do
3          Draw Card
4          if move is possible then
5              Determine and Show moveable pawns;
6              Have player select moveable pawn;
7              Determine and Show moves for
               selected pawn;
8              Have player select move;
9          else
10             Forfeit Move;
11         end
12     end
13 end
```

In order to provide this functionality the Game Screen UI must possess the following features:

- An image of the game board with clickable hot spots capable of highlighting specific hot spots
- A face down deck of cards which may be clicked on to draw a card
- A face up pile of cards indicating the last card drawn
- Markers which indicate the position of each player's pawns and which may be highlighted to indicate valid movement candidates
- Spots on the board marking where players are in the safety zone, home, start, and on the sliding spaces
- An indicatation of whose turn it is
- A general purpose message box used to display textual feedback and prompts

We have also identified the following UI features which we believe will be useful to the player while not directly impacting the flow of play:

- A "help" button which displays the Instructions screen
- A "play for me" button which automatically plays the user's current turn for them using the standard AI algorithm used by the computer players.

2.7.3 Backend Description

In order to service the UI the Game Screen back end must model the game abstractly. To do this the back end will need to keep track of the following:

- About the Deck
  - Which card is face up
  - The contents of the deck
  - Each card has a certain set of options as to how to proceed with your turn.
  - When the deck runs out, it is randomly shuffled again
  - Two pawns cannot occupy the same space at the same time
- About the Board
  - Keep track of the spaces on the board and their positional relationships to eachother
  - Keep track of where the various pawns are on the board
  - When a player cannot make any valid moves after drawing a card, the system should recognize this
  - Recognize special board spaces like the safety zone, the slider, the start, and the home
  - Recognize when a player has successfully moved all of his/her pawns into home, winning the game
- About the State of Play
  - Who the players are
  - Whose turn it is
  - Given a card and board state identify movement candidates and valid moves for them

This bookeeping shall be done with the following data structures:

- Deck:
  - Card objects which record information of the types of moves they provide
  - Two mutable arrays of card objects capable of: being oriented top to bottom, having card objects exchanged between them, being merged, and being shuffled
- Board:
  - Space objects which contain information about special properties they have, which pawns are in them, and which board UI hotspots activate them
  - A multiply linked list modelling which Space objects are adjacent to each other in both the forward and reverse directions
- About the State of Play
  - Player objects which can be queried for the Pawn objects they own

- An array of Player objects which is used to quantify turn order
- A loop counter to keep track of turns
- A function which queries a given Player's pawns and returns an exhaustive list of legal moves for them given a specific Card's rules

2.7.4 Computer Players

Computer players behave in a manner similar to human players. When possible they must make two decisions each turn. Which pawn to move and where to move it. These decisions shall be made by appealing to the following sequence of preferences in some order of importance:

- Move pawns out of home vs move pawns around the board?
- Move pawns off of the spot in front of home to allow new pawns to enter the board?
- Move pawns out of danger (away from opponent pawns, off of slider spaces, move into safety zone)?

Note that all of these preferences are geared towards answering the question: "Should I help my own cause or hurt my opponent?". The basic AI shall have board conditions identified where each of the subsidiary questions are definitively answered "yes" or "no" thus choosing a specific, preferred move and an answer to our more general question.

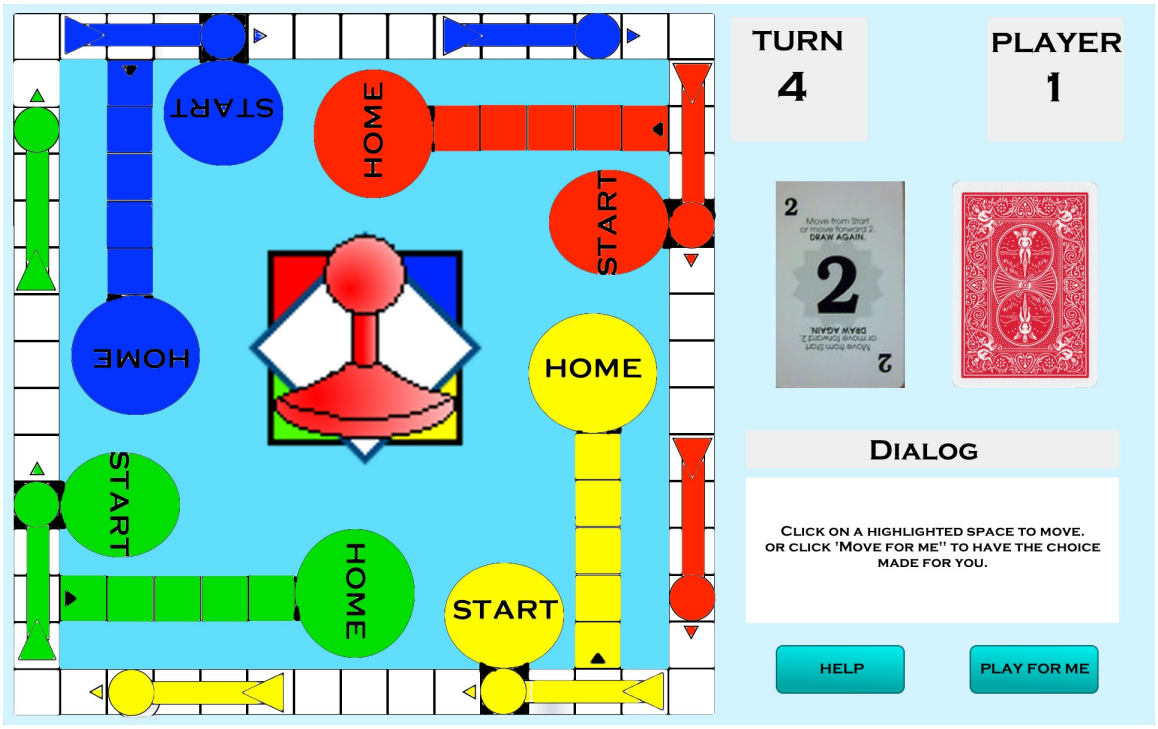

Figure 8: The game board screen.

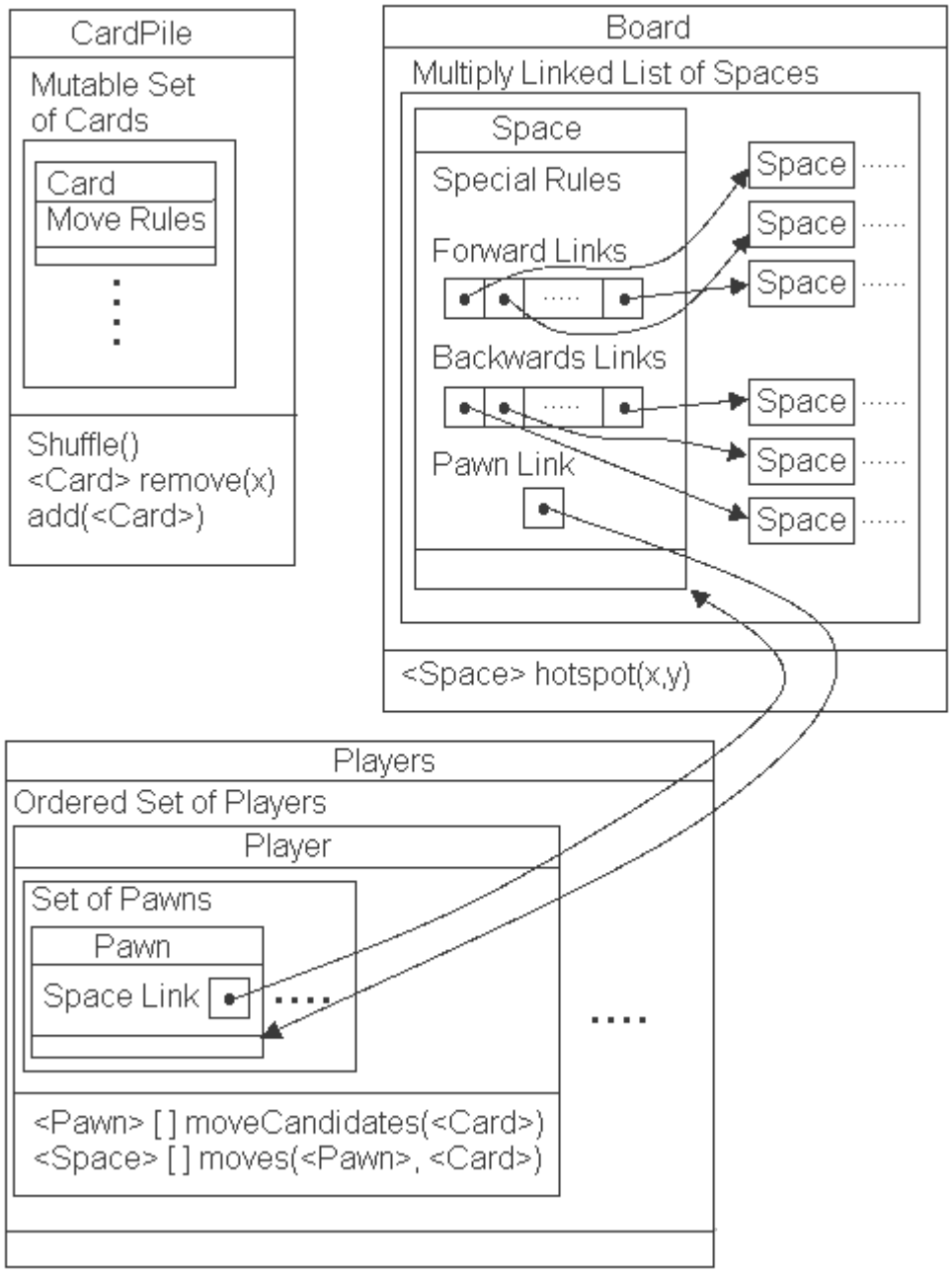

Figure 9: A graphical representation of several of the objects described in Section 2.7.3 and their relationships.