Wheel of Fortune
Software Requirements Specification

Author: Kavan Wadhwa
Emerald Games Enterprises
November 24th, 2017
Version 1.0

Abstract: This document describes the program requirements for the classic game show, **Wheel of Fortune**. The program interface and functionality as well as the overall goals of the project will be described.

Based on IEEE Std 830™-1998 (R2009) document format

1 Introduction

Millions of people have watched the American game show, Wheel of Fortune, created by Merv Griffin. Wheel of Fortune is an extended version of the classic Hangman game. An empty phrase is placed on a board and the category name is written below the phrase. Contestants take turn spinning a wheel with numerous wedges that contain a price, a prize, or both and receive a certain amount of money for guessing a correct letter. Contestants keep spinning until they guess an incorrect letter. Contestants aim to collect the most money possible and guess correct letters and phrases over the course of a three round game. The contestant with the most money advances to the bonus round to compete for a larger prize, often a substantial dollar amount or a car.

We aim to implement a basic version of Wheel of Fortune as a computer game that allows individuals to take the driver's seat from their homes.

1.1 Purpose

The purpose of this document is to specify the appearance and functionality of the ***Wheel of Fortune*** game. The intended audience for this document is a single individual because the project is developed by a single member. There are no communication issues. Upon completing the reading of this document, one should be able to visualize how the game application will look and operate as well as understand the game rules.

1.2 Scope

Here at Emerald Games Enterprises, we look forward to serving as a company that continues to develop popular game shows. Other game shows that we would like to develop include *Chain Reaction* and *Lingo.* Since we're lining up a whole series of computerized game shows, it is probably helpful to implemetn good design decisions that result in reusable components. As such, a framework (or set of frameworks), should be designed and constructed to be used to make ***Wheel of Fortune*** with the potential for reusing parts for other word games or perhaps different implementations of the original ***Wheel of Fortune*** game. Note that the point of ***Wheel of Fortune*** is simply to entertain.

1.3 Definitions, acronyms, and abbreviations

Word-based Game Show – Game shows that deal with manipulating or guessing words or phrases. Such shows include *Wheel of Fortune, Lingo,* and *Chain Reaction.*

IEEE – Institute of Electrical and Electronics Engineers, the "world's largest professional association for the advancement of technology".

Framework – In an object-oriented language, a collection of classes and interfaces that collectively provide a service for building applications or additional frameworks all with a common need.

GUI – Graphical User Interface, visual controls like buttons inside a window in a software application that collectively allow the user to operate the program.

Wheel of Fortune – The SONY Pictures Television game show that will be replicated by our software. The game is founded on Hangman. Contestants spin a wheel and guess consonants or buy vowels and attempt to solve the random phrase displayed on the board. The objective is to guess correct letters and phrases and earn money from doing so. The winner proceeds to a bonus round with a chance for winning a large cash prize or car.

Game Setup Screen – A game screen that allows the player to select the number of players and their names

UML – Unified Modeling Language, a standard set of document formats for designing software graphically.

Use Case Diagram – A UML document format that specifies how a user will interact with a system. Note that these diagrams do not include technical details. Instead, they are fed as input into the design stage (stage after this one) where the appropriate software designs are constructed based in part on the Use Cases specified in the SRS.

Wheel – A circular object that contains wedges with varying text. In Wheel of Fortune this wheel is spun by a contestant before he/she guesses a letter.

"Buying a Vowel" – In Wheel of Fortune, consonants are free but vowels cost $250 each.

Category – Wheel of Fortune has various categories from which phrases are selected. Such categories include places, things, food and beverage etc.

1.4 References

IEEE Std 830™-1998 (R2009) – IEEE Recommended Practice for Software Requirements Specification

1.5 Overview

This SRS will clearly define how the Wheel of Fortune game should look and operate as well as how to actually play the game. Note that this is not a software design document. This document does not specify how to build the appropriate technologies, it is simply an agreement concerning what to build, not how to build it. Section 2 of this document will provide the context for the project and specify all the conceptual design, including the game rules and parameters. Section 3 will present how the game

interface should be laid out and all program functionality. Section 4 provides a Table of Contents, an Index, and References.

2 Overall description

Wheel of Fortune is word puzzle game rooted in the classic Hangman. It is traditionally played by three contestants and the game show includes:
- 1 large board on which the phrase is displayed,
- 1 wheel containing several wedges with different prices and prizes,
- 3 rounds,
- 1 bonus round

2.1 Game Description

The contestants take turns spinning a wheel to guess letters in an empty phrase put on the board. Contestants receive prizes and dollar amounts for guessing correct letters but are also at risk of losing earnings if they land on the Bankrupt wedge. The idea is to use skill and careful consideration to choose letters and guess phrases in order to accumulate the most earnings of the three contestants and advance to the bonus round.

Preparation:
- The user selects to play a new game and will be prompted to select the names of each player. For now, we will arrange the game to be played by 2 individuals.
- The player to go first will be randomly selected
- Players will be reminded of the basic rules of the game

Play
- Randomly selected player begins round
- At each turn each player can choose to spin the wheel or guess the phrase
- If the player guesses the phrase correctly, the round ends
- If the player spins the wheel and does not land on Bankrupt or Lose a Turn, the player guesses a letter or buys a vowel for $250.
- The cash amount landed on indicates the cash this player earns for each correct letter in the puzzle/phrase. If the wedge landed on is a specialty wedge, follow the information provided in the appendix.
- If the player guesses a correct letter (either by guessing a consonant or buying a vowel), he/she has the option to spin again, buy a vowel, or guess the phrase
- If the player guesses an incorrect letter, his/her turn ends and the next player spins
- This continues until the phrase is completed by guessing letters or until a contestant correctly guesses the phrase

Lose a Turn or Bankrupt:
- A player's turn is over.
- If Lose a Turn, the player loses his/her turn.
- If Bankrupt, the player loses his/her earnings for the current round.

Free Play:
- A player gets a free guess; that is, the player can guess a consonant or vowel without penalty

.

End of the Game
The player with the most earnings at the end of the three rounds will advance to the bonus round to compete for a larger prize. We will not implement the bonus round until the next version, but the bonus round is played as followed:
- The winner spins the wheel of prizes
- A phrase is put on the board
- The player is given the letters R, S, T, L, N, and E.
- The player then selects three more consonants and one more vowel.
- The board is updated to reflect the player's selections
- The contestant then has 10 seconds to guess the correct phrase (he/she has infinite attempts)
- If the contestant guesses correctly he/she wins the prize landed on in addition to the earnings from regular game play.
- If the contestant fails to guess the phrase, he/she keeps his/her earnings from regular game play but does not win the bonus prize.

***Potential added features:**
Letter Probability
- If user clicks on an empty space on the board a new window will open up separate from the game.
- The window will show the probability that the letter at the clicked index is either a, b, c, d, e…, m, … x, y, or z.
Spin Power Meter
-User can adjust power meter from default and the wheel will be spun accordingly

2.1.1 System Interfaces

Because the Wheel of Fortune game does not entail saving games, the game will not need to provide a means to access saved games. Note that ***Wheel of Fortune*** will ultimately exist as a standalone application, though hopefully one that will be played on a number of different platforms.

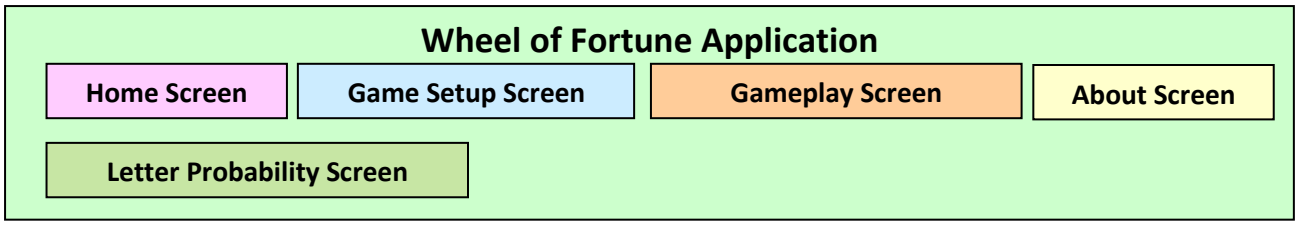

Figure 2.1: The various screen contexts for Wheel of Fortune.

2.1.2 User Interfaces

Due to the fact that we want to port this to a mobile interface in the future and we wish to keep game flow simple so that users can pick up the game easily, it is important to keep the interface controls simple. We need to make sure players will use the application in limited ways. For example, we will limit player interactions to the mouse for selecting a play option and guessing letters, and to the keys for guessing and phrases. Figure 2.2 below summarizes the ways with which the user will interact with our ***Wheel of Fortune*** application, which will be furthered detailed using a UML Use Case diagram for each. Note that "Clicks" always refers to a left-mouse-button click. Here is the full list of UML Use-Case Diagrams:

Figure 2.2: Overview of Use-Case Diagrams

Figure	Screen	Use Case
2.3	Home Screen	Play Game
2.5	Home, Gameplay Screens	Open About Screen
2.6	Splash, Gameplay Screens	Quit Game
2.7	Game Setup Screen	Select Players and Give Names
2.9	Game Setup Screen	Start Game
2.17	About Screen	Return from About Screen
2.10	Gameplay Screen	Spin
2.11	Gameplay Screen	Guess Letter
2.12	Gameplay Screen	Buy a Vowel
2.13	Gameplay Screen	Guess Phrase
2.14	Gameplay Screen	Click on empty letter
2.15	Letter Probability Screen	Return from probability screen
2.11	Gameplay Screen	Close Win Dialog

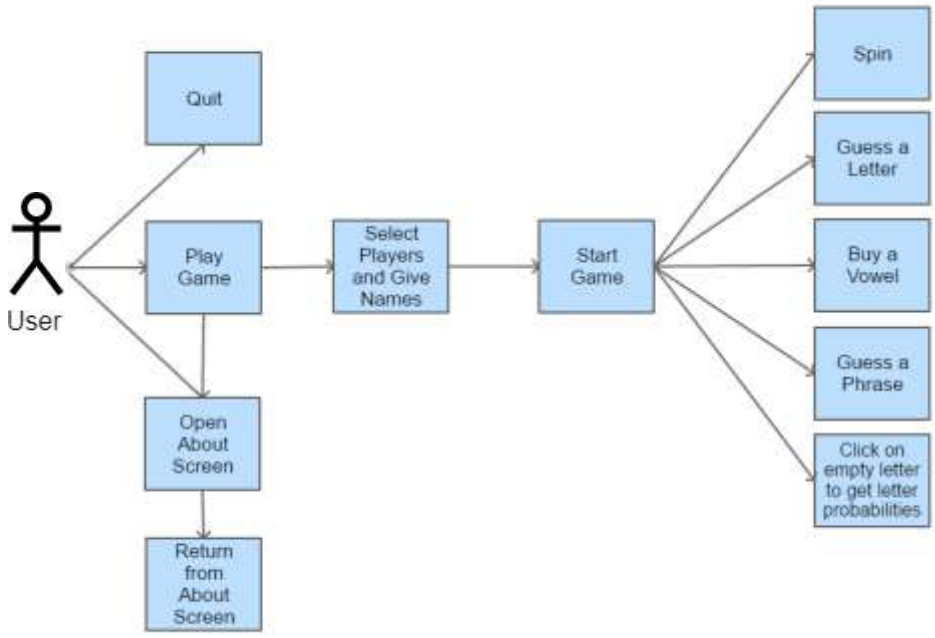

Figure 2.3: Play Game Use Case

Use-Case:	Play Game
Primary Actor:	Player(s)
Goal in Context:	Start a brand new game
Preconditions:	The game application has been started
Trigger:	The player clicks on the "Play" button
Scenario:	1. Player starts the application, which loads game 2. Player views the Home Screen, which has two options, one to play a new game and one to see the about screen 3. Player wants to play so presses "Play" button 4. Application loads Game Setup Screen
Exceptions:	This button should always be enabled from the Home Screen.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Used every time the player starts the application
Open Issues:	Size, location, and style of button should be finalized by UI designer. Should be emphasized as primary choice.

Figure 2.4: Open About Screen Use Case

Use-Case:	Open About Screen
Primary Actor:	Player
Trigger:	The user clicks on the "About" button either in the Home or the Gameplay screens.
Scenario:	1. A player clicks on About button. 2. A page is displayed with background information on the Wheel of Fortune game show as well as rules that will help contestants play the game. The screen will also contain information on Emerald Games Entertainment as

	well as a Return button.
Priority:	Essential, must be implemented (rules is the essential part)
When available:	Second Benchmark (assume basic knowledge of game, thus, not high priority)
Frequency of use:	Many times for every game
Open Issues:	Size, location, and style of page should be finalized by UI designer. Should be minimized in gameplay screens as to not distract user from gameplay but should be evident enough that players know where to look for help. On the home screen, the about button should be smaller than the Play button as to make the playing option more appealing.

Figure 2.5: Home Screen

Figure 2.7: Return from About Screen Use Case

Use-Case:	Return from About Screen
Primary Actor:	Player
Trigger:	The user clicks on the "Close" button in the About Screen.
Scenario:	1. A player clicks on Close button. 2. The UI is returned to the previous screen (Gameplay or Home).
Priority:	Essential, must be implemented
When available:	Second Benchmark
Frequency of use:	Many times for every game
Open Issues:	Size, location, and style of page and close button should be finalized by UI designer.

Figure 2.8: Select Players Use Case

Use-Case:	Select Players
Primary Actor:	Player
Goal in Context:	Select the names of players
Preconditions:	The game has started and the user selects the number of players.
Scenario:	1. Player is viewing the Game Setup Screen. 2. The player enters the names of the contestants that will be participating in the game.
Exceptions:	None.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Once per session for every game
Open Issues:	Size, location, and style of controls should be finalized by UI designer.

Figure 2.9: Start Game Use Case

Use-Case:	Start Game
Primary Actor:	Player
Goal in Context:	User starts game from the Game Setup Screen.
Preconditions:	The app has started and the player is currently viewing the Game Setup Screen.
Trigger:	The player clicks on the “Let’s play!” button.
Scenario:	1. Player is viewing the Game Setup Screen. 2. Player clicks on “Let’s play!” button. 3. Gameplay Screen displayed and the game **Preparation** is done. The board is set up with a random phrase and the wheel, leaderboard, player card, and letter board are shown.
Exceptions:	None.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Once per game session.
Open Issues:	Size, location, and style of button should be finalized by UI designer..

Figure 2.10: Gameplay Screen

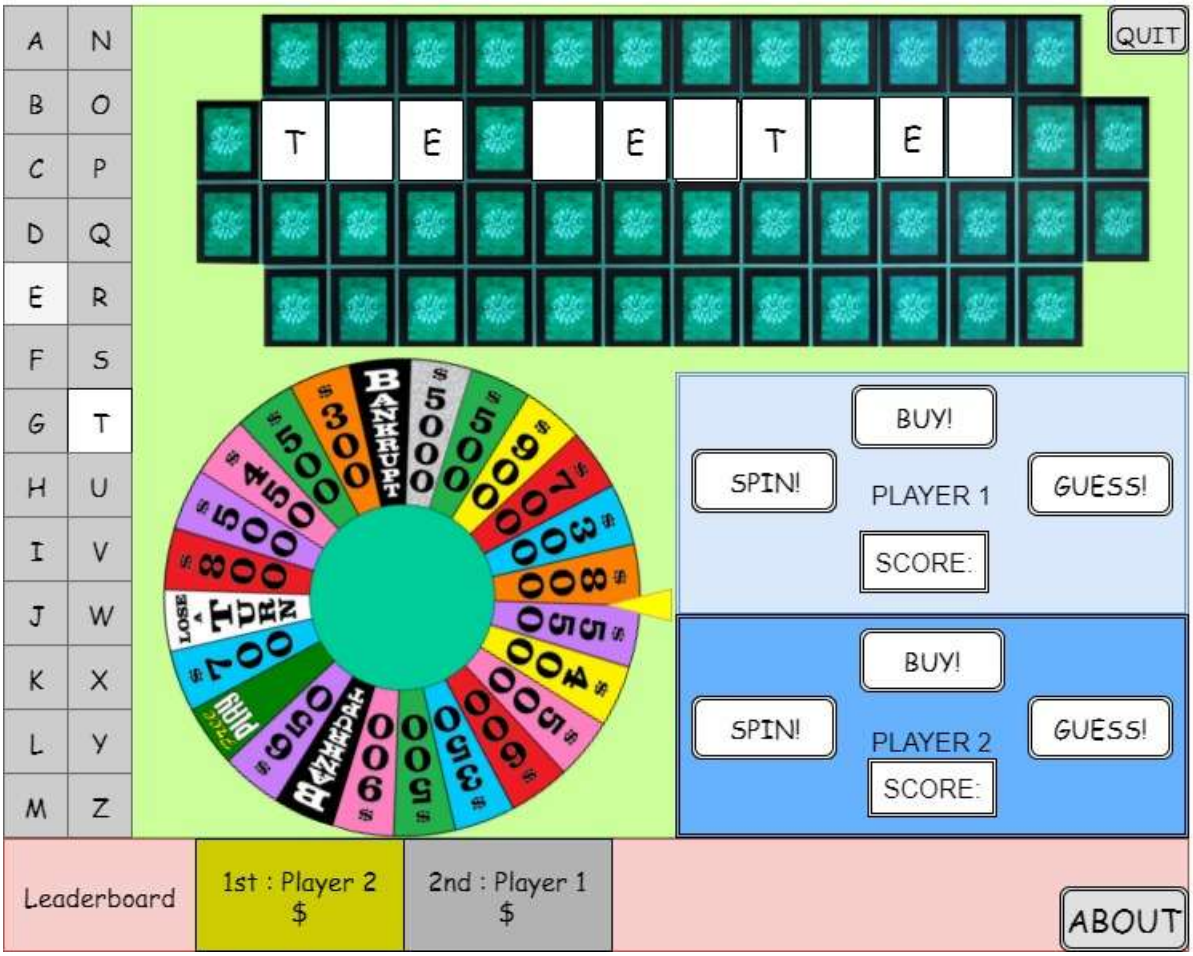

Figure 2.11: Quit Game Use Case

Use-Case:	Quit Game
Primary Actor:	Player
Goal in Context:	Quit and close the game application when the player requests it.
Preconditions:	The application has started and is in the Gameplay Screen mode.
Trigger:	The player clicks on a “Quit” button.
Scenario:	1. Player is viewing the Gameplay Screens. 2. Player clicks on the “Quit” button. 3. Application closes.
Exceptions:	None.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Once per session.
Open Issues:	Size, location, and style of button should be finalized by UI designer. Should be almost hidden as to discourage user from quitting the game because the game will not be saved.

Figure 2.12: Spin Use Case

Use-Case:	Spin the Wheel
Primary Actor:	Player
Goal in Context:	Spin the Wheel and show the label of the wedge that has been landed on
Preconditions:	The application has started and is in the Gameplay Screen mode.
Trigger:	The player clicks on the "Spin" button when it is his/her turn
Scenario:	1. Player is viewing the Gameplay Screens. 2. Player clicks on the "Spin" button. 3. Animation of wheel spinning and the text of the wedge landed on is shown
Exceptions:	None.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Numerous times per session.
Open Issues:	Size, location, and style of button should be finalized by UI designer. Should be shown big and clear, preferably near the wheel. A power meter will also be shown next to the wheel and can be adjusted by the user.

Figure 2.13: Guess Letter Use Case

Use-Case:	Guess a Letter
Primary Actor:	Player
Goal in Context:	Update board and gameplay screen based on letter guessed
Preconditions:	The application has started and is in the Gameplay Screen mode.
Trigger:	The player clicks on a letter when it is his/her turn after having spun the wheel
Scenario:	1. Player is viewing the Gameplay Screen. 2. Player clicks on a letter in the board of letters on the screen 3. Board state and scores are updated accordingly if the letter is in the phrase 4. If not the game turns to the next contestant
Exceptions:	None.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Numerous times per session.
Open Issues:	Size, location, and style of letter board should be finalized by UI designer. Letters should be shown in a sizeable font either to the left or bottom of the screen.

Figure 2.14: Buy a Vowel Case

Use-Case:	Buy a Vowel
Primary Actor:	Player
Goal in Context:	Allow a player to guess a vowel by buying one for $250
Preconditions:	The application has started and is in the Gameplay Screen mode.
Trigger:	The player clicks on the "Buy a Vowel" button when it is his/her turn
Scenario:	1. Player is viewing the Gameplay Screens. 2. Player clicks on the "Buy a Vowel" button. 3. Player is prompted to enter a vowel or select from the board of letters (vowels and consonants will be disabled and enabled accordingly)

Exceptions:	None.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Numerous times per session.
Open Issues:	Size, location, and style of button should be finalized by UI designer. Should be sizeable so the user knows this option exists.

Figure 2.15: Guess Phrase Case

Use-Case:	Guess the Phrase
Primary Actor:	Player
Goal in Context:	Allow player to guess the phrase and check if it is correct
Preconditions:	The application has started and is in the Gameplay Screen mode and the player has just guessed a correct letter.
Trigger:	The player clicks on the "Solve" button when it is his/her turn after having guessed a correct letter
Scenario:	1. Player is viewing the Gameplay Screen. 2. Player clicks on the "Solve" button. 3. Player is prompted to enter his/her guess 4. The guess is checked 5. If correct the player wins the round and is presented with a win dialog
Exceptions:	None.
Priority:	Essential, must be implemented
When available:	First Benchmark
Frequency of use:	Numerous times per session.
Open Issues:	Size, location, and style of button should be finalized by UI designer. Should be sizeable so the user knows this option exists.

Figure 2.16: Click on empty letter Case

Use-Case:	Click on empty letter in phrase board
Primary Actor:	Player
Goal in Context:	Display the probabilities for each letter for a given spot
Preconditions:	The application has started and is in the Gameplay Screen mode.
Trigger:	The player clicks on an empty index in the board when it is his/her turn
Scenario:	1. Player is viewing the Gameplay Screens. 2. Player clicks on an empty spot in the board 3. A new alert message appears displaying the probabilities of each letter
Exceptions:	None.
Priority:	Non-essential, additional feature
When available:	Third Benchmark
Frequency of use:	Many times for every game if implemented
Open Issues:	User must be made aware that this tool exists. Only way to access tool is by clicking on one of the empty letters.

Figure 2.17: Return from Letter Probability Screen Use Case

Use-Case:	Return from Letter Probability Screen
Primary Actor:	Player
Trigger:	The user clicks on the “Close” button in the Letter Probability Screen.
Scenario:	1. A player clicks on Close button. 2. The UI is returned to the previous screen (Gameplay).
Priority:	Non-essential, additional feature
When available:	Third Benchmark
Frequency of use:	Many times for every game if implemented
Open Issues:	Size, location, and style of page and close button should be finalized by UI designer.

Figure 2.18: Close Win Dialog Use Case

Use-Case:	Close Win Dialog
Primary Actor:	Player
Goal in Context:	One player is victorious at the end of each round
Preconditions:	One player has won the round or the entire game (has the most earnings)
Trigger:	Any player solves a puzzle correctly.
Scenario:	1. Players are working on solving a puzzle 2. A player solves the puzzle correctly 3. A “Player X won the round!” message appears and the player confirms by clicking on an OK button. 4. The game ends
Priority:	Essential, must be implemented
When available:	Second Benchmark
Frequency of use:	Once every round played
Open Issues:	Size, location, and style of message and close button should be finalized by UI designer.

Hardware Interfaces

The game should ideally be designed and constructed such that it may be easily ported and published on multiple platforms. Target platforms should include PC and Android, so implementation of the game should be done in Java. Mobile platforms are a natural match for this game because it solely involves clicking and typing. In addition, the essentially single game play screen makes the transition to a smaller mobile screen easier.

2.1.3 Software Interfaces

Wheel of Fortune will be developed using the Java language and the JavaFX library. We plan for this to be the first game show among many, but it is not reasonable to implement a separate Game Framework hand in hand with this application because there is not enough overlap between game shows to do so. The game will also be developed using Java's JavaFX framework for building user interfaces and rendering 2D graphics.

Java version 8.0 will work on all desired platforms with the notable exception of the iPhone. The game would have to be ported to Objective C to operate there. For that reason, development will proceed first as a Java game, and porting will proceed after completion of the Java game. Note again that due to the variations in screen resolutions, careful planning should be done to account for platform independence in all game data management.

2.1.4 Communications Interfaces

Note that this game will operate solely as a local game. There will be no networking requirements. At some point in the future the app may be integrated with Facebook accounts to allow friends to play one another in *Wheel of Fortune*, but for now it operates as a standalone application.

2.1.5 Memory Constraints

Since this game will be ported to mobile devices, memory will ultimately become an important issue. All images should be provided in the precise sizes they will be rendered. Images should never be scaled up or down. Note that the memory requirements for each platform will be different, but that memory budgets should be defined carefully before each port. Luckily, it does not seem that this game, at least in its basic version will require substantial memory.

2.1.6 Operations

It is the goal of the player to complete one *Wheel of Fortune* game, but an individual / many individuals can play the game over and over again.

2.1.7 Site Adaptation Requirements

N/A

2.2 Product functions

Wheel of Fortune aims to entertain and allow users to experience what may be the next closest thing to a game show experience. The game assumes that individuals will play the game in one sitting. The game does not need to save multiple accounts data or player settings, and is not intended to save the progress of the current game. The game assumes multiple users will take turns and go only when it is their respective turns.

2.3 User characteristics

Though the *Wheel of Fortune* game show appeals to people of all ages, the necessary knowledge of the English language suggests that the game appeals to individuals aged 13+ specifically. As such, our computerized version should attract a similar audience. Individuals should find the GUI controls easy to pickup and use because of the limited controls required.

2.4 Constraints

Mobile screen resolutions, however, present a porting challenge. Different models support different resolutions and all mobile platforms fall short of PC/Mac capabilities. Note that the key hardware to plan for are the mouse (or for mobile versions, touch), keyboard and the screen. Another constraint that exists for the standalone Java computer application is that players will have to take turns and constant switch control over mouse and keyboard. Perhaps creating keyboard shortcuts in the future can resolve this issue.

2.5 Assumptions and dependencies

The main assumption for the non-mobile version of this game is that users will respect each other and switch control of the mouse and keyboard when it is each of their turns respectively. We also assume that the game will be started and finished in one sitting and will be played by solely two players.

2.6 Apportioning of the Requirements

It will be determined at a later date precisely how this *Wheel of Fortune* application will be ported to additional platforms. The program should be constructed such that it may easily be modified.

3 Specific requirements

Wheel of Fortune will use a simple interface with few controls. Note that the GUI controls should always be neatly drawn, carefully spaced, and properly aligned.

3.1 External interfaces

The player will use mouse input to start and run the game, so the GUI will be straightforward. The gameplay screen has been previously provided.

3.2 Functions

One of the important things to consider in our game application is providing the appropriate feedback to the user while the game is happening. Players need feedback to enjoy their game experience. This is typically done with visual cues or sound. We will not be employing sound in this project, but we will certainly need to use visual cues.

3.2.1 Animation Cues

The player will need animation cues for a number of scenarios. We'll need the following animations:

- **Spinning wheel** – wheel is spun and stops spinning smoothly, gradually, not suddenly.
- **Board update (optional)** – after guessing a correct letter the appropriate indices in the board phrase should be highlighted before the letters are displayed (as in the game show). In case of an incorrect letter a red X should appear on the screen. Again this is not as important because a player will know when his/her turn is over and can very easily tell whether they guessed a correct letter (either by looking at the board or by understanding that if their turn is not over they must have guessed a correct letter).
- **Whose turn**? – the screen should make evident who's turn it is by highlighting the appropriate player or displaying the player's name and score on the screen

3.2.2 Basic Features

- **Check letter** – need to check if a letter is in a phrase and appropriately update the game screen as well the score of the contestant currently playing their turn
- **Check guess** – need to check if a player's guess is correct and update the game screen accordingly by moving to the next round or declaring a winner for the game

3.2.3 Potential Additional Features

- **Power Meter** – a power meter will be added next to the wheel; users can adjust the meter beyond a minimum as they please
- **Letter Probability** – probability of letters at different indices will need to be calculated in order to provide players with consistent and accurate statistics

3.3 Performance requirements

The primary performance concerns will be with rendering, since it is a real-time graphical application. Note that rendering performance testing should be an important component of the development process. It is important that buttons are disabled and enabled at appropriate times and that scores and board states are appropriately updated.

Additionally, options should be easy to find for users. Such options include the letter board from which a contestant will guess letters and the spin, solve, and buy a vowel buttons. Additionally, the window that will display letter probabilities for a given space should appear outside but directly next to the main screen as to not interfere with main gameplay.

3.4 Logical database requirements

N/A

3.5 Design constraints

As mentioned, a primary concern should be efficient rendering. When available, the program should be able to take advantage of hardware (GPU) acceleration for rendering. However, in order to appeal to the maximum number of users, it should also work well when that is not available. This is a reasonable constraint, since this is a 2D game with little artwork to render.

3.6 Software system attributes

We are dedicated to producing robust software that exceeds the expectations of our customers. In order to achieve this level of quality, we should build a product with the following properties in mind:

3.6.1 Reliability – The program should be carefully planned, constructed and tested such that it behaves flawlessly for the end user. Bugs, including rendering problems, and disabling of buttons problem are unacceptable. In order to minimize these problems, all software will be carefully designed using UML diagrams and a Design to Test approach should be used for the Implementation Stage. Additionally, text files containing words and phrases ought to be appropriately formatted such that there are no issues in choosing a random phrase or parsing words/phrases.

3.6.2 Availability –Customers may download and install the game application for free, but will be restricted to playing 10 games a day.

3.6.3 Security – All security mechanisms will be addressed by future revisions. Security will not pose a pressing issue until social media accounts are incorporated.

3.6.4 Extensibility – It is important that more phrases and perhaps even different languages can be added to the game, so file formats for words and phrases of different categories should be

carefully considered such that the "word bank" of this *Wheel of Fortune* game can be easily extended.

3.6.5 Portability – To start with, the game will target all platforms that support Java. An Objective C port should follow thereafter to make the game playable on the iPhone.

3.6.6 Maintainability – Update mechanisms will be addressed by future revisions.

3.7 Organizing the specific requirements

Note that the game is simple enough that we need not worry about using an alternative arrangement of the content of this document. The specific requirements for this application already fit neatly into the sections listed in the IEEE's recommended SRS format.

3.8 Additional comments

It is important to keep in mind that the primary result of using this document should incorporate the basic features of the *Wheel of Fortune* game show. As such, the additional features are, additional, and should therefore be implemented after the initial components have been successfully implemented, even if this means waiting until version 2.0 to release these features.

4 Supporting Information

Note that this document should serve as a reference for the designers and coders in the future stages of the development process (in this case a single member), so we'll provide a table of contents to help quickly find important sections.

4.1 Table of contents

4.2 Appendixes

4.2.1 Information on price and specialty wedges

Wedges – Parts of the wheel. Most wedges contain price values, which are indicative of the money that will be added to a contestant's total based on the letter he/she guesses and the number of this letter present in the given phrase. Other wedges are "specialty" wedges; instead of a price they present a special opportunity; the "specialty" wedges are defined below.

"Bankrupt" Wedge – Causes a contestant to lose all of his/her money earned during that round and "specialty" tags earned during the entire game.

"Free Play" Wedge – Permits a contestant to guess the phrase or a letter with no penalty.

"Lose a Turn" Wedge – Contestant loses a turn.