

Roshambo

Software Requirements Specification

1.0.2

2/23/2014

Joe A. DeLuca
Lead Software Engineer

Prepared for
MSIT 660—Software Development
Instructor: Renata Rand Mcfadden, Ph.D.
Winter 2014

Revision History

Date	Description	Author	Comments
01/26/14	1.0.0	Joe A. DeLuca	Initial Document
02/09/14	1.0.1	Joe A. DeLuca	Revision 1
02/23/14	1.0.2	Joe A. DeLuca	Revision 2

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

Table of Contents

REVISION HISTORY.....	II
DOCUMENT APPROVAL.....	II
1. INTRODUCTION.....	1
1.1 PURPOSE.....	1
1.2 SCOPE.....	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	2
1.4 OVERVIEW.....	2
2. GENERAL DESCRIPTION.....	2
2.1 PRODUCT PERSPECTIVE.....	2
2.2 PRODUCT FUNCTIONS.....	3
2.3 USER CHARACTERISTICS.....	4
2.4 GENERAL CONSTRAINTS.....	6
2.5 ASSUMPTIONS AND DEPENDENCIES.....	6
3. SPECIFIC REQUIREMENTS.....	6
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	6
3.1.1 <i>User Interfaces</i>	6
3.1.2 <i>Hardware Interfaces</i>	6
3.1.3 <i>Software Interfaces</i>	6
3.1.4 <i>Communications Interfaces</i>	6
3.2 FUNCTIONAL REQUIREMENTS.....	6
3.2.1 <i>Display option to play game</i>	6
3.2.2 <i>Choose to play game</i>	7
3.2.3 <i>Display Round</i>	7
3.2.4 <i>Display Gesture Selection</i>	7
3.2.5 <i>Choose Gesture – Player 1</i>	7
3.2.6 <i>Choose Gesture – Player 2</i>	7
3.2.7 <i>Display Countdown</i>	7
3.2.8 <i>Display Round Results</i>	7
3.2.9 <i>Display Ready Message</i>	7
3.2.10 <i>Indicate Ready</i>	8
3.2.11 <i>Display Game Results</i>	8
3.2.12 <i>Display Play Again</i>	8
3.2.13 <i>Display View History</i>	8
3.2.14 <i>Display Exit Game</i>	8
3.2.15 <i>Choose Play Again</i>	8
3.2.16 <i>Choose View History</i>	8
3.2.17 <i>Choose Exit Game</i>	8
3.2.18 <i>Choose Play Again</i>	8
3.3 NON-FUNCTIONAL REQUIREMENTS.....	9
3.4 DESIGN CONSTRAINTS.....	9
3.5 LOGICAL DATABASE REQUIREMENTS.....	9
4. CHANGE MANAGEMENT PROCESS.....	9

1. Introduction

Assignment #1 in MSIT 660 Software Development is to create a Software Requirement Specification (SRS) which will document all of the features of a “Rock, Paper, Scissors” game. The document will be used to complete assignment #2(Design) and #3(Implementation).

This document is the Software Requirement Specification (SRS) for the “Rock, Paper, Scissors” game called Roshambo. It will outline all of the features of the game.

1.1 Purpose

The purpose of the SRS is to provide a set of requirements for the “Rock, Paper, Scissors” game called Roshambo. This will be used in assignment #2 to design the game and in assignment #3 to implement the game using the python programming language.

1.2 Scope

The software to be created is called Roshambo which is a “Rock, Paper, Scissors” game.

1. The game will have 3 rounds per game and the winner of 2 out of 3 rounds wins the game.
2. Each round will allow the user to select one of three gestures: Rock, Paper, or Scissors.
3. When the player chooses the gesture, a count down will be displayed. The count down will start by displaying a 3. It will then display 2. Then 1. Then “Go”.
4. The system will then output the results on the screen.
 - Example: Round 1: You Win! Rock beats Scissors.
 - Example: Round 1: You Loose. Paper beats Rock
 - Example: Round 1: Tie. Scissors tie Scissors
5. The system will then present the player a prompt “Are you ready for Round #?”. The system only does this for Round 1 and Round 2. User input is required to proceed.
6. The system will then present the player with the gesture selection again for the next round. After round 3 the system will output the results for who won the game.
7. After round 3, the system will present the player with the option to play another game or exit main menu.
 - The game history only consists of previous games played during this session. The game history does not include other sessions.
 - Exit to main menu will present the user with play, view instructions, view history, and exit options.

1.3 Definitions, Acronyms, and Abbreviations

These are the terms used in this SRS relating to the game.

- Gesture – Rock, Paper, or Scissors
 - Rock – rock beats scissors
 - Paper – paper beats rock
 - Scissors – scissors beats paper
- Session – The game session begins when the application starts and ends when the application is exited or closed.
- Game – consists of 3 rounds. The best 2 out of 3 wins the game.
- System – the python application. Player 2.
- User – the person interacting with the system. Player 1.
- Player 1– same as the user.

1.4 Overview

This section is an overview of the whole system. This will describe the way user that will interact with the system and the functionality that the system will provide. It will also explain the constraints and assumptions for the system.

2. General Description

Starting the application results in a main menu where the user has the option to play the game, view instructions, view history, or exit the game.

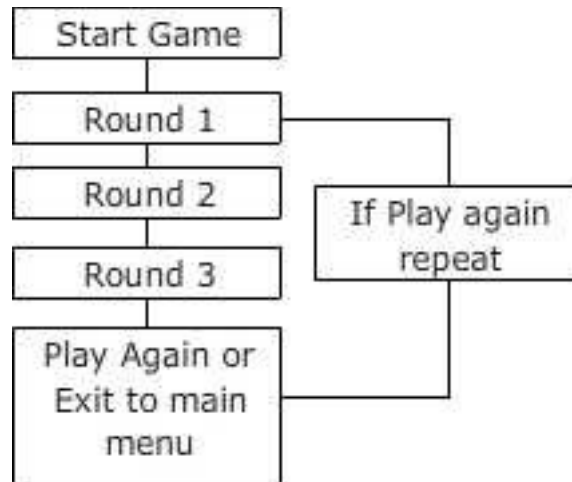
Roshambo is a game of “Rock, Paper, Scissors”. It allows the user to play against the computer. Each game consists of three rounds. The best 2 out of 3 rounds wins the game.

In each round, the player will select a gesture which starts a count down. The countdown begins at 3 and finishes with “Go”. Example: 3...2...1...Go. The countdown decrements every second. A second after “Go” is displayed, the results of the round is displayed.

On conclusion of round 1 and 2, the system will ask the user if they are ready to play the next round. When the user acknowledges with input, then the next round starts. On conclusion round 3, the system will display the results of winning the game and present the user with the option to play another game or exit to the main menu.

2.1 Product Perspective

This is a new self-contained product called Roshambo. The game of Roshambo consists of 3 rounds where player 1 plays against player 2. During each round, both make gesture selections and the system determines the winner based on the criteria outlined in this document. Round 1 leads to round 2. Round 2 leads to round 3. Round 3 is the final round and the player is given the option to play again, or exit to main menu. Playing again starts back at round 1.



2.2 Product Functions

With this game, the user will be able to play against the computer as many times as they would like. On the completion of each game, the user will be able to choose to play again or exit to the main menu.

When the game starts, the user will be able to choose their gesture. Choosing the gesture starts the count down beginning from 3. The count down is decremented each second by one. Instead of displaying 0, the system will display “Go”.

When the user makes his/her selection, the system will make its selection by choosing random number between 1 and 3. The number will represent a gesture.

A second after “Go”, the system will determine who won the round. Rock beats scissors. Scissors beats paper. Paper beats Rock.

After the results are shown in round 1 and 2, the system will display a message asking if the user is ready to start the next round. User input is required to continue past this message. Once the user has notified the system, then next round begins and the user can choose a gesture again.

When the user chooses to view history, the system will display a log of all the games and rounds played during this session.

When the user chooses to exit the game, the system will close or exit.

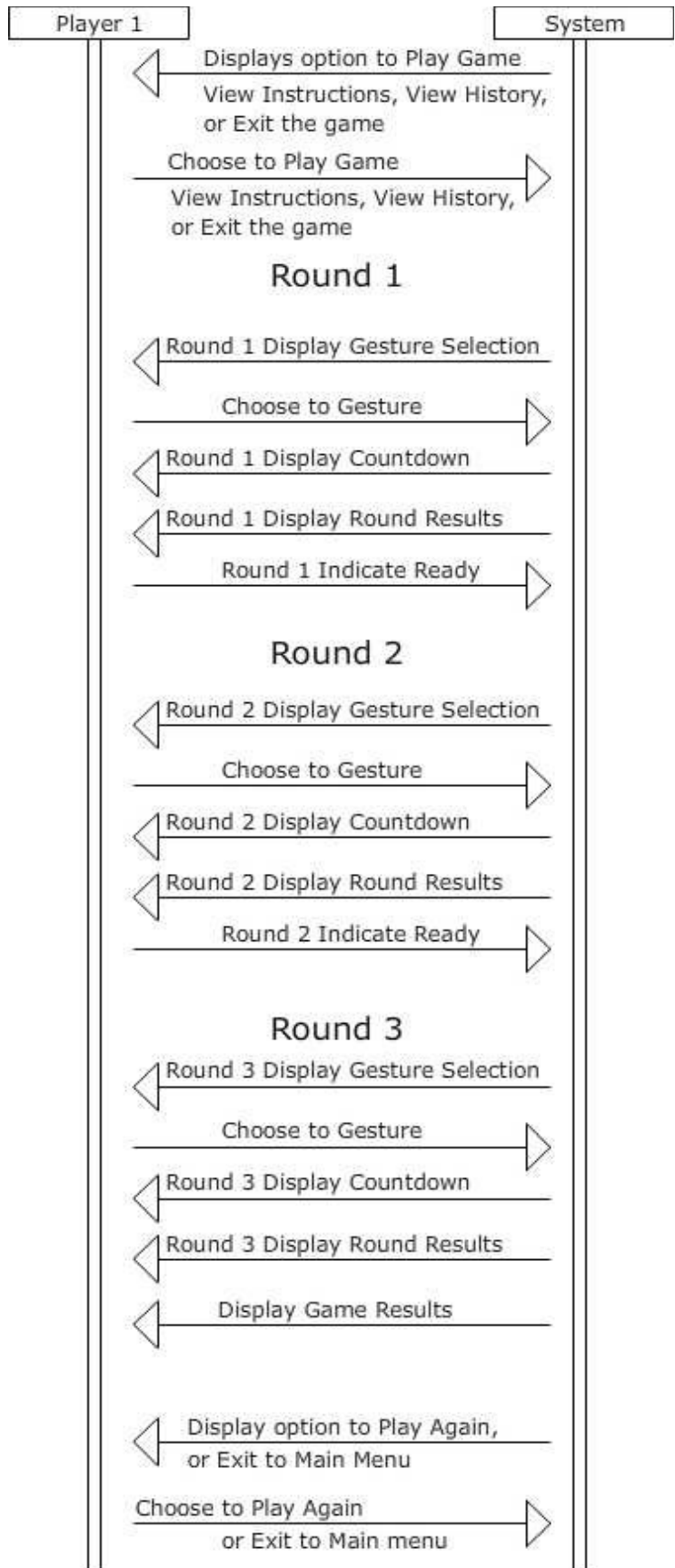
2.3 User Characteristics

There are two users. The user and the system. The user is referred to as user, player, or player 1. The system is referred to as system or player 2.

The user can choose to play game, choose a gesture, indicate they are ready to proceed to next round, choose to play game again, choose to view history, and choose to exit game. These are the inputs that will be received by the system from the user.

The system plays the role of player 2. The system also displays all output and choices that player1 can make. The system displays option to play game, displays gesture selection, displays countdown, displays round results, displays game results, displays option to play again, displays option to view history, display option to exit the game.

Below is a diagram of interaction between the user and the system during a game.



2.4 General Constraints

The only constraint is that python must be installed on the computer running Roshambo. Roshambo is to be programmed using the python programming language which requires the python interpreter in order to run the game.

2.5 Assumptions and Dependencies

One assumption about the product is that it will run on all systems that can install the python interpreter and have enough resources to execute with good performance. If the python interpreter can not be installed on the computer the user wants to play the game on, then the game will not run at all.

3. Specific Requirements

This section provides functional requirements of the system. It also gives a detailed description of all the systems features.

3.1 External Interface Requirements

This section provides details of all the inputs and outputs of the system. It gives basic prototypes of the user interface.

3.1.1 User Interfaces

This will be a text based game. All user input will be in the form of a character on from the keyboard. All messages from the system will be text output. Each message and option will appear on a new line leaving all previous lines of output visable.

3.1.2 Hardware Interfaces

The application does not have any designated hardware. It should run on any computer that can install the python interpreter. It does not require any other hardware to carry out any other function.

3.1.3 Software Interfaces

The game is self-contained and does not require integration with any other software.

3.1.4 Communications Interfaces

There is no communication with any other systems. It is self-contained.

3.2 Functional Requirements

This section includes the requirements that specify all the fundamental actions of the system.

3.2.1 Display option to play game

DESCRIPTION: This is a welcome screen where the user is presented with the option to start game

CHARACTER: System.

3.2.2 Choose to play game

DESCRIPTION: The user makes the choice to play the game and supplies user input.

CHARACTER: User.

3.2.3 Display Round

DESCRIPTION: The system displays what the current round is. There are 3 rounds to the game.

CHARACTER: System.

3.2.4 Display Gesture Selection

DESCRIPTION: The system gives the user the options to choose Rock, Paper, or Scissors.

CHARACTER: System.

3.2.5 Choose Gesture – Player 1

DESCRIPTION: Player 1 chooses Rock, Paper, or Scissors with user input. The user input will be in an integer between 1 and 3. 1 represents Rock. 2 represents Paper. 3 represents Scissors.

CHARACTER: User.

3.2.6 Choose Gesture – Player 2

DESCRIPTION: Player chooses Rock, Paper, or Scissors randomly. The system will randomly pick a number between 1 and 3. 1 represents Rock. 2 represents Paper. 3 represents Scissors.

CHARACTER: System.

3.2.7 Display Countdown

DESCRIPTION: The system will display a count down starting from 3. The count down is timed. It is decremented each second by one. Instead of displaying 0, the system will display “Go”.

CHARACTER: System.

3.2.8 Display Round Results

DESCRIPTION: At the conclusion of each round, the results of the round are displayed. The results will be determined by the criteria of rock beats scissors, paper beats rock, and scissors beats paper. If both the system and the user make the same choice there is a tie.

CHARACTER: System.

3.2.9 Display Ready Message

DESCRIPTION: At the conclusion of round 1 and round 2, the system will display a message asking player 1 if they are ready for next round. Round 3 does not have this message since it is the last round.

CHARACTER: System.

3.2.10 Indicate Ready

DESCRIPTION: Player 1 provides user input that they are ready to proceed to the next round.

Roshambo

CHARACTER: User.

3.2.11 Display Game Results

DESCRIPTION: The system will display the final results. The winner is the best 2 out of 3 rounds. In the event 2 rounds end in tie, the player that wins one round is the winner. It is unlikely that player 1 ties player 2 in all three rounds. Although unlikely, it is possible to end in a tie after all 3 rounds.

CHARACTER: System.

3.2.12 Display Play Again

DESCRIPTION: The system displays the option to play again.

CHARACTER: System.

3.2.13 Display View History

DESCRIPTION: The system displays the option to display history.

CHARACTER: System.

3.2.14 Display Exit Game

DESCRIPTION: The system displays the option to exit game.

CHARACTER: System.

3.2.15 Choose Play Again

DESCRIPTION: The user provides input to play again. When this option is selected, the game starts over at round 1.

CHARACTER: User.

3.2.16 Choose View History

DESCRIPTION: The user provides input to view history. When this option is selected, the system outputs results of each round of each game and the final results of each game. The history is only during this session and does not span game sessions.

CHARACTER: User.

3.2.17 Choose Exit Game

DESCRIPTION: The user provides input to end game. The application exits.

CHARACTER: User.

3.3 Non-Functional Requirements

To run Rosumbo, python needs to be installed on the users computer. The application should run on all computers that can run python interpreter. Rosumbo will only be tested using Microsoft Windows 8 operating system. Rosumbo will be written in the python programming language.

3.4 Design Constraints

There are no design constrains imposed by other standards, company policies, hardware limitations which will impact this application.

3.5 Logical Database Requirements

There are no database requirements. Data will only exist in memory during the execution of the program.

4. Change Management Process

If this SRS document requires an update, a new version of the SRS will be submitted by Joe DeLuca the lead applications developer on this project. It will be submitted to Dr. Renata Rand Mcfadden for approval through Blackboard Course Email with SRS attachment.