Software Requirements Specification

for

<Space Invaders>

Version 1.0 approved

Prepared by Katya Civil, Magnus Hsiung,

Yeuk Sing Chan, Andrew Alekseyenok

<3140 Group 2 Specs>

<02/25>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document's purpose is to provide specifications for our group project. Our goal is to create a playable version of the popular Space Invaders game. Ideally this game will consist of the player controlling a spaceship with limited movement, where the main goal is to survive against waves of AI-controlled aliens.

1.2 Document Conventions

For this document we'll be using an IEEE software requirements specification template along with Times New Roman font 12.

1.3 Intended Audience and Reading Suggestions

The intended audience is the members of Group 2 as well as the professor. The rest of this document will contain a set of descriptions and requirements which will be met as we continue to work on this project.

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

The software should represent our iteration of the popular game Space Invaders. We are making it to satisfy the group project portion of our class. Our goal is to have a version of the game ready before the semester is over.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Gameplay - https://www.youtube.com/watch?v=MU4psw3ccUI&ab_channel=GameArchive
Sprite Js - https://spritejs.readthedocs.io/en/latest/#:~:text=js%20documentation-,Sprite.,Sprite.
Html Game example - https://www.w3schools.com/graphics/game_intro.asp

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

This project is being done by Group 2 students in the 3140 web development for the professor who assigned it.

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

This product must at least be 1 minimum value project containing a playable wave of space invaders with scoring functionality and working title screen.

1. *Title Screen*
    a. *New Game*
        i. *Start a new game under a new User ID*
        ii. *Sets a new highscore when completed*
    b. *continue game*
        i. *Start a new game under a familiar User ID*
        ii. *logs highscore*
    c. *Scores*
        i. *shows leaderboard*
    d. *Settings*
        i. *allows change in difficulty*
        ii. *volume slider*

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

User Class

1. *allows user to create a UNIQUE user id limited to X characters *X because its undecided* and be a player*
2. *allows user to edit settings*
3. *allows user to look at leaderboards*

Player Class

1. *allows user to play a game*
2. *allows users scores to be recorded and put into leaderboard under its user ID*
3. *allows user to edit settings*
4. *allows user to look at leaderboards*

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

Time constraint: Due by the end of semester
Must be a subversion repository

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

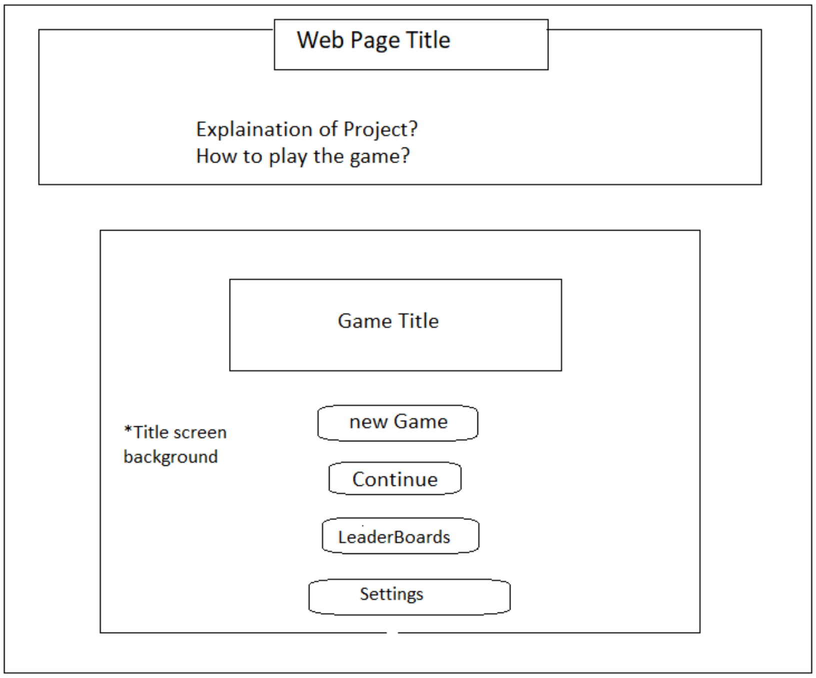
2.7 Assumptions and Dependencies

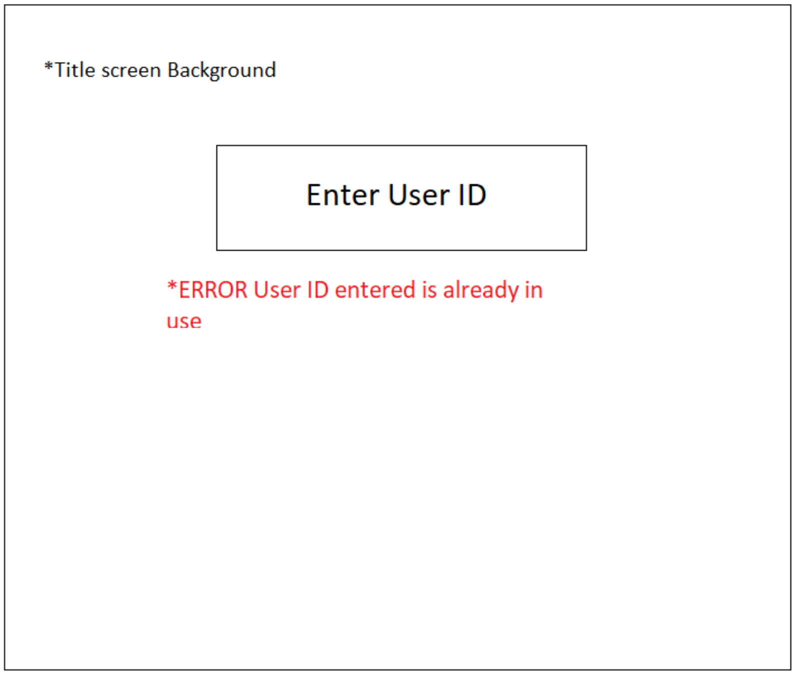
<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>



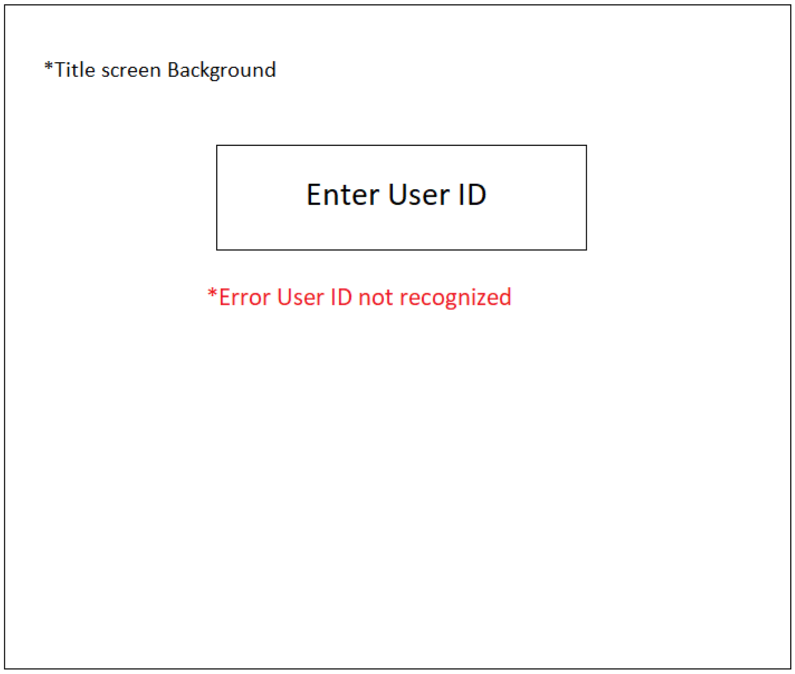


Continue Clicked

State 01: Title screen background with Textbox with placeholder requesting a User ID with a Char Limit of X Characters

State 02: User Id is unqiue creates a new user and starts a wave

State 03: User ID is one that is already found in database error text pops up

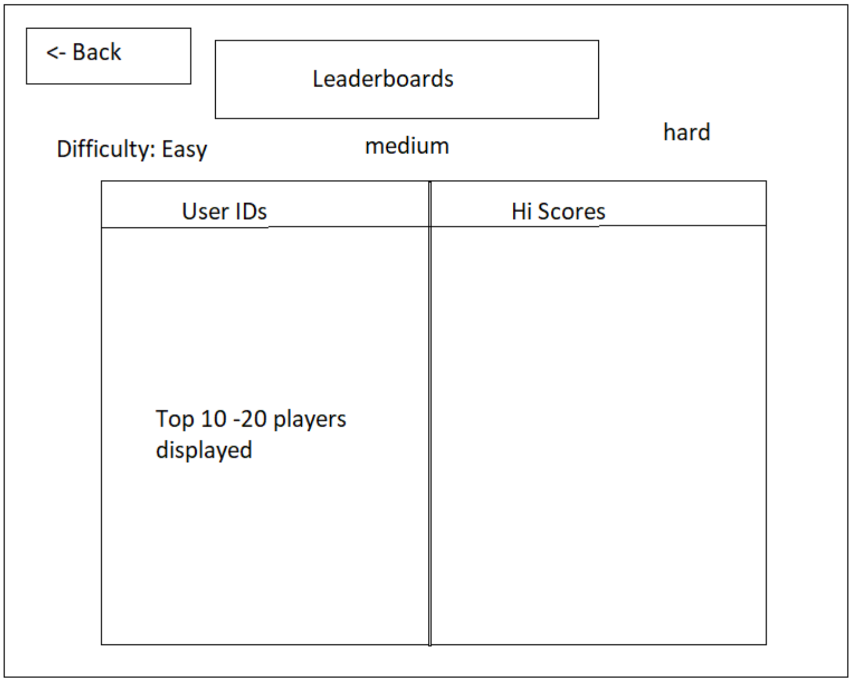


Continue Clicked

State 01: Title screen background with Textbox with placeholder requesting a User ID with a Char Limit of X Characters

State 02: User ID is in database starts wave under that User ID

State 03: User ID is not in database Error Text Appears



Leaderboards Clicked

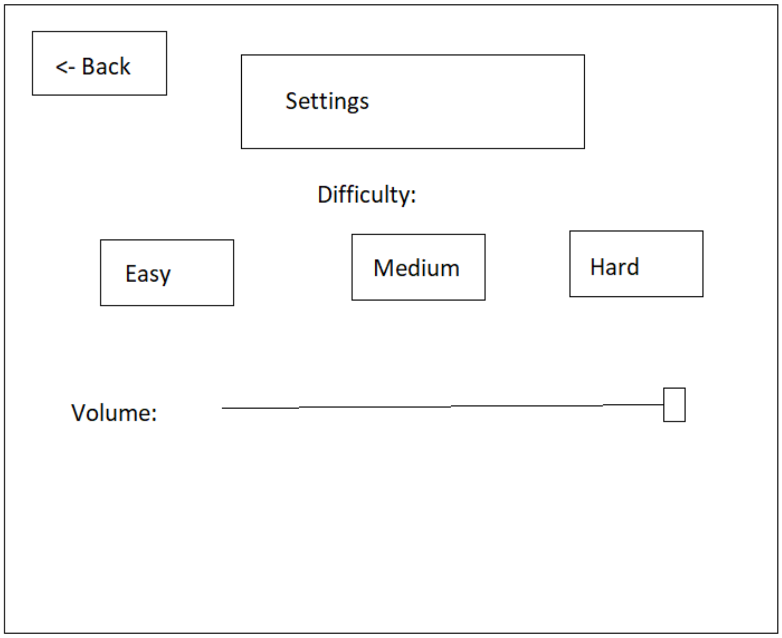
State 01: Loads up leaderboards for easy difficulty

State 02: medium is clicked and the Leaderboards for medium difficulty appear

State 03: hard is clicked and the Leaderboard for hard difficulty appears

State 04: Easy is clicked same as State 01

State 05: Back button pressed User sent back to main page



Settings Clicked

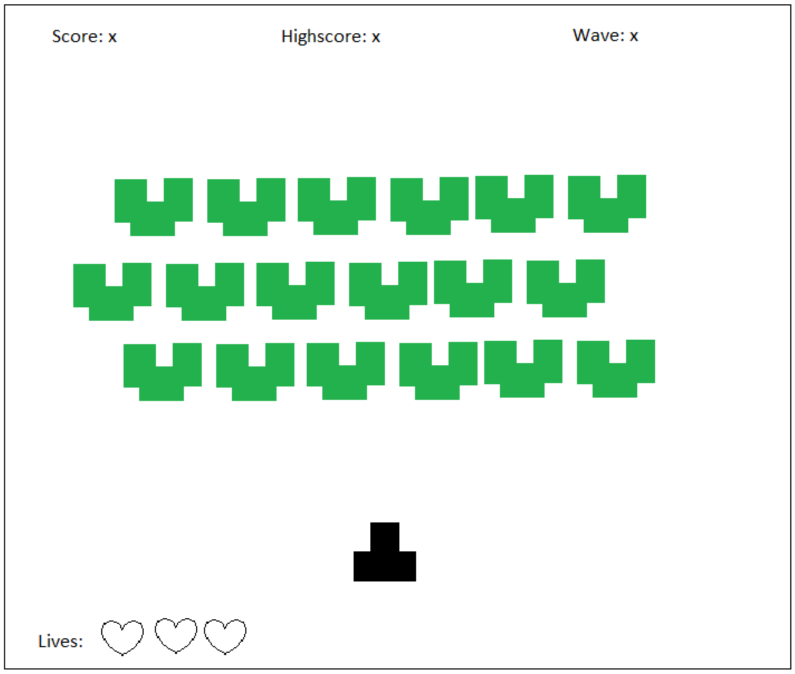
State 01: Medium difficulty selected by default and is highlighted volume defaulted to 100%

State 02: Easy selected Game difficulty set to "Easy"

State 03: Medium selected Game difficulty set to "Medium"

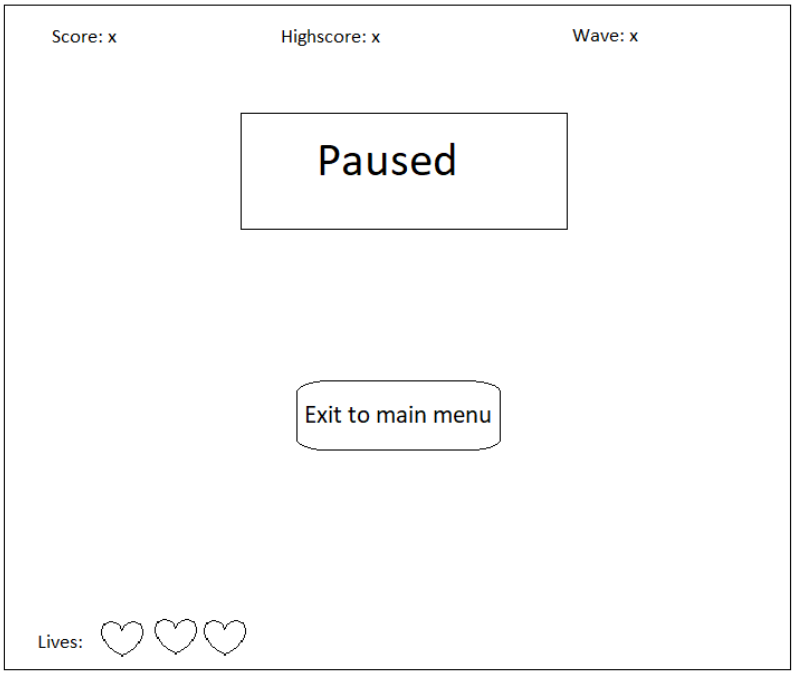
State 04: Hard selected Game difficulty set to "Hard"

State 05: Back button pressed returns user to "Main page"



Game Start

State 01: Game starts enemy wave of aliens approach. Player's Highscore, current score, # of lives and wave # (optional) are present on screen. Player controls ship.



Game Paused /ESC pressed

State 01: Game is paused. Game resumes when player presses esc again or whatever controls are set. This gives the option to exit the game back to main menu.

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

Game basic provides the core functionality of the game. --High priority.

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

Stimulus: Player requests to input their name
Response: System provides a box for a player to enter his/her name and display it as his/her username.
Stimulus: Player requests to start a game
Response: System starts a new game.
Stimulus: Player requests to pause a game
Response: System stop movement of all game related particles.
Stimulus: Player requests to resume a game

Response: System resume movement of all game related particles.
Stimulus: Play requests to restart the game.
Response: System reset all game related particles.
Stimulus: Play requests to exit the game.
Response: System close the game and show home screen
Stimulus: Play request to change the difficulty level

Response: System shows 3 options for players to pick.
Stimulus: Player requests to view leaderboard.
Response: System shows the top 10 highest score that the player had achieved life time
Stimulus: Player finishes a game with a new score. Check if the new score is higher than the top 10 scores. If the score is higher than top 10 scores, replace the old score with the new score.
Response: If a player got hit 3 times.
Stimulus: System shows the gameover screen and allow Player to restart the game.

4.1.3 Functional Requirements

Game basic

REQ-001: The system shall allow a user to be a player.

REQ-002: The system shall allow a user to input their name.

REQ-003: The system shall allow a user to start a game.

REQ-004: The system shall allow a user to pause a game.

REQ-005: The system shall allow a user to resume a game.

REQ-006: The system shall allow a user to restart the game in the middle of the game.

REQ-007: The system shall allow a user to exit a game.

REQ-008: The system shall allow a user to choose difficulty between 3 different levels, EASY, HARD, EXTREME, when creating a game.

Leaderboard

REQ-021: The system shall save the 10 best highest scores that a player achieved.

REQ-022: A high new score will be inserted into the leaderboard appropriately based on its score, and old scores will drop out of the leaderboard if they are no longer ranked in the top-10.

Game-Difficulty

REQ-041: EASY: all objects move at a speed of 1.0.

REQ-042: HARD: all objects move at a speed of 1.15.

REQ-043: EXTREME: all objects move at a speed of 1.75.

Game-over detection.

REQ-061: The system shall end the game if a player got hit 3 times in any difficulty.

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The program is to be without graphic, audio, or coding errors. The program is to interact and react to the users inputs in real time, without hiccups or any sort of unintended delay.

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

The program should not have any rapidly flashing or quickly alternating visual images such as flashing lights or patterns for those with epilepsy or a history of seizures. Game should have sound cues and audio at an acceptable level to those playing. Audio should not be introduced in any startling or potentially harmful manner.

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

No user identity authentication needed. Players should only be able to play the game in the manner intended and not access any developer or debugging mode intended for testing.

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

- Program should be designed with a level of quality that introduces no exploits to gain an undesired advantage over other players.
- Game should function in the manner the game was intended to be played. Game should have no visual errors that would affect the “fairness” of the game.

- ★ Objects such as sprites should have a collision region or hit detection region to match their size or current location.
- ★ Objects should not disappear at any point of time unless intended.
- ★ Game audio should be paired in synchronization with ingame events for clarity.
- ★ Game audio should be of clear quality and distinctive to match different events within the game.
- ★ Game should not have any errors that would cause an abrupt end when unintended.
- ★ Game should have a scoring system with consistent rewards in correspondence with the same action. (*Think: shoot 'X' type of enemy, get 100 points*)

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

Players should only be able to play the game in the manner intended and not access any developer or debugging mode intended for testing.

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>