Software Requirements Specification

for

Space Adventures

Version 1.0 approved

Prepared by Raechel Griffin, Kiana Ross, David Perrone, Lars Riewe, Jason Yang, Ajibola Olatunji

CSC 305: Team H Studios

March 8th, 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Project Setup	2/15/2021	Project Proposal 1	1.0
System Features and Naming Project	03/01/2021	Project name was changed and added System Features in section 3.	1.1

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers.>

1.2 Document Conventions

<Describe any standards or typographical conventions used, including the meaning of specific text styles, highlighting, or notations. If you are manually labeling unique requirement identifiers, you might specify the format here for anyone who needs to add one later.>

1.3 Project Scope

The software allows the user to pilot a spaceship. The user must defeat enemies and evade asteroids in order to beat the high score. This software entertains the user.

1.4 References

<List any documents or other resources to which this SRS refers. Include hyperlinks to them if they are in a persistent location. These might include user interface style guides, contracts, standards, system requirements specifications, interface specifications, or the SRS for a related product. Provide enough information so that the reader can access each reference, including its title, author, version number, date, and source, storage location, or URL.>

2. Overall Description

A 2d game where the user controls a spaceship and shoots at aliens (that can shoot back), asteroids, and other objects. Players will earn more points based on what targets they shoot and how long they last without dying.

2.1 Product Perspective

The idea for this game came from combining some aspects of a few different arcade games into one.

2.2 User Classes and Characteristics

The audience for this software is anyone that enjoys playing games and beating high scores.
It could be younger people looking for another game to play or older generations looking to relive their gaming days.

It could be younger people looking for another game to play or older generations looking to relive their gaming days. Kids or teenagers may have characteristics that make them more inclined to beat high scores, play harder levels, and look for more options.

Operating Environment
The game will be designed to operate on Windows computers. It will be downloaded as a zip file and extracted into an executable file.
Design and Implementation Constraints

2.3 Operating Environment

Implementing different platforms will be a goal of this project. Cross platform utilization will be important with the varying kinds of operating systems like Windows versus Mac. Java will handle hardware limitations like timing or memory requirements.

2.4 Design and Implementation Constraints

Certain libraries are needed when programming in Java. Also, the program will be developed among the same IDE (IntelliJ).

<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, stimulus, response, or combinations of these, whatever makes the most logical sense for your product.>

2. System Features

2.5 Playable character

2.5.1 Description

As a user I need a spaceship I can control so that I can play the game. This is a high priority feature because it is the only means for the user to interact with the game.

2.5.2 Stimulus/Response Sequences

Stimulus: The user presses the arrow keys. **Response**: The spaceship moves on the screen to the corresponding direction.

2.5.3 Functional Requirements

<Itemize the specific functional requirements associated with this feature. These are the software capabilities that must be implemented for the user to carry out the feature's services or to perform a use case. Describe how the product should respond to anticipated error conditions. Use "TBD" as a placeholder to indicate when necessary, information is not yet available.>

2.6 Enemies

Description

As a user I need enemies to attack me so that I have a challenge in the game. This is a medium priority feature because it is important for gameplay, but it is not necessary.

3.2.2 Stimulus/Response Sequences

Stimulus: Enemies have a scripted behavior. They move across the screen. **Response**: Enemies move as scripted.

3.2.3 Functional Requirements

3.3 Projectiles

3.3.1 Description

As a user, I need projectiles that can be fired from my ship to enemies or from enemies to my ship so that I can fight the enemies. This is a medium priority feature because it is important for gameplay but not necessary.

3.3.2 Stimulus/Response Sequences

Stimulus: The user clicks the left mouse button. **Response:** The spaceship fires projectiles in the direction the ship is facing.
Stimulus: The enemies have their scripted behavior. **Response**: The enemies shoot as scripted.

3.3.3 Functional Requirements:

3.3.3.1 Must have a function that creates the projectile.
3.3.3.2 Must have a speed/direction to control where the projectile goes.
3.3.3.3 Must have collision detection to tell if projectile collided with player or enemy.

3.4 Leaderboard

3.4.1 Description

As a user, I need a leaderboard to record my high scores so that I have the incentive to get. better at the game to beat my records. This is a low priority feature because it is not important for game play.

3.4.2 Stimulus/Response Sequences

Stimulus: The player kills an enemy. **Response:** The player's score is increased
Stimulus: The player dies. **Response:** The player's score is saved on a csv file and sorted
Stimulus: The user beats the old high score. **Response:** A message tells the user they have a new high score

3.4.3 Functional Requirements:

3.5 Asteroids

3.5.1 Description

As a user, I need asteroids to fly on the screen so that I must move my ship to avoid them. This is a high priority feature because is necessary for gameplay to provide a challenge to the player.

3.5.2 Stimulus/Response Sequences

Stimulus: A timer spawns asteroids at a certain rate. **Response:** The asteroid flies onto the screen

3.5.3 Functional Requirements

3.5.3.1 Must have a timer that counts either frames or seconds.
3.5.3.2 Must have a function that spawns asteroids.
3.5.3.3 Must have a set direction/speed of asteroids.

3.6 Power-ups

3.6.1 Description

As a user, I need power-up items in difficult locations to collect/shoot at, so that I feel accomplished after taking a risk. This is a medium priority feature; the game of obstacles and shooting them can still exist without it.

3.6.2 Stimulus/Response Sequences

Stimulus: An enemy dies **Response:** An item is spawned
Stimulus: The player shoots the item. **Response:** The player receives some power-up

3.6.3 Functional Requirements

3.6.3.1 Must have collision detection with the user or projectiles in order to collect it.
3.6.3.2 Must have a random spawn rate, so that it doesn't appear after every enemy is destroyed.
3.6.3.3 Must provide the user with a different type of power-up every time, for variety.

2.7 System Feature 2 (and so on)

3. Data Requirements

Logical Data Model

3.1 Logical Data Model

Data Dictionary

3.2 Data Dictionary

Reportsprojects.>

3.3 Reports

Data Acquisition, Integrity, Retention, and Disposalring the detailed report layout to the design stage.>

3.4 Data Acquisition, Integrity, Retention, and Disposal

<If relevant, describe how data is acquired and maintained. State any requirements regarding the need to protect the integrity of the system's data. Identify any specific techniques that are necessary, such as backups, checkpointing, mirroring, or data accuracy verification. State policies the system must enforce for either retaining or disposing of data, including temporary data, metadata, residual data (such as deleted records), cached data, local copies, archives, and interim backups.>

4. External Interface Requirements

User Interfaces

4.1 User Interfaces

Software Interfaces

4.2 Software Interfaces

Hardware Interfaceswebsites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the intercomponent communications. Identify data that will be exchanged between or shared across software components. Specify nonfunctional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.>

4.3 Hardware Interfaces

Communications Interfaces.>

4.4 Communications Interfaces

<State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.>

5. Usability

5.1 Usability

Performance; error avoidance, handling, and recovery; efficiency of interactions; accessibility; and ergonomics. Sometimes these can conflict with each other, as with ease of use over ease of

learning. Indicate any user interface design standards or guidelines to which the application must conform.>

5.2 Performance

Security

5.3 Security

Safetyphysical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.>

5.4 Safety

[Others as relevant]

5.5 [Others as relevant]

<Create a separate section in the SRS for each additional product quality attribute to describe characteristics that will be important to either customers or developers. Possibilities include availability, efficiency, installability, integrity, interoperability, modifiability, portability, reliability, reusability, robustness, scalability, and verifiability. Write these to be specific, quantitative, and verifiable. Clarify the relative priorities for various attributes, such as security over performance.>

6. Internationalization and Localization Requirements

<Internationalization and localization requirements ensure that the product will be suitable for use in nations, cultures, and geographic locations other than those in which it was created. Such requirements might address differences in: currency; formatting of dates, numbers, addresses, and telephone numbers; language, including national spelling conventions within the same language (such as American versus British English), symbols used, and character sets; given name and family name order; time zones; international regulations and laws; cultural and political issues; paper sizes used; weights and measures; electrical voltages and plug shapes; and many others.>

7. Other Requirements

<Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections. >

Appendix A: Glossary

<Define any specialized terms that a reader needs to know to understand the SRS, including acronyms and abbreviations. Spell out each acronym and provide its definition. Consider building a

reusable enterprise-level glossary that spans multiple projects and incorporating by reference any terms that pertain to this project.>

Appendix B: Analysis Models

<This optional section includes or points to pertinent analysis models such as data flow diagrams, feature trees, state-transition diagrams, or entity-relationship diagrams. You might prefer to insert certain models into the relevant sections of the specification instead of collecting them at the end.>