iSpaceship - Deliverable 1

Team
Pedram Yazdinia, yazdinip
Will Conry, conrywm
Torja Istiaque, istiaqum
Akila Kavisinghe, kavisina
Xiangxin Kong, kongx9

April 8, 2020

Contents

1 Introduction

This document is intended to provide a general overview of the requirements specifications for the video game *iSpaceship*. The document is organized with the following sections in this order: Introduction, Scope, Overall Description, Use Case Diagram, Functional Requirements, and Non-functional Requirements.

1.1 Purpose

The purpose of the software requirement specification is to provide a detailed overview of the software product being designed. The SRS will contain all measurable parameters and goals of the project. The document will also describe functional and non-functional requirements as well as the various stakeholders and users.

This document is intended for developers, testers, and project managers within the group including the TA's and the professor. Developers will refer to the document at each stage of the design process in order to meet the design requriments.

1.2 Scope

"iSpaceship" is a Rogue-like, turn-based, spaceship battle simulator, desktop video game. The software product will provide the user with an engaging gaming experience where they can build their own spaceship, face disasters, and battle other spaceships. The software product will be used for the enjoyment of the user and provide them with a sense of self-accomplishment. The objective is for users to have fun and feel rewarded when they put in the time and dedication to progress in the game. The application should hold interest of users over a long period of time.

1.3 Definitions, Acronyms, and Abbreviations

a) Rogue-Like - Video game genre where players restart completely upon permanent death of the player character.

b) Turn-Based - Strategy game where players take turns playing.

c) Gamer - A person who is enthusiastic about video games.

d) AWS - Amazon Web Services

e) XP - Points associated to the progress of the player in the game

f) Ability - Abilities are special powers that allows each spaceship to battle with other spaceships.

g) Components - Each component is used to improve the power that comes along with each ability.

1.4 References

1. SE3A04 tutorial slides2 and slides3 by Andrew LeClair
2. Unity documentation, available at: `https://docs.unity3d.com/`

1.5 Overview

This document encapsulates critical information which will be used as a foundation of the entire project. This document first discusses the general factors that affect the product and the requirements. This information enables the reader to have a better grasp and understanding on the requirements that come after. These factors include product perspective product functions, user characteristics and constraints. The reader is then presented with a Use Case diagrams which visually shows the interaction between different actors and tasks. The largest portion of the document is then dedicated to clearly stating and explaining the functional and non-functional requirements. Attached at the end is a spreadsheet that shows the division of labour within the team.

2 Overall Description

This section of the SRS will describe the general factors that affect the product and its requirements. It does not state specific requirements; it provides a background for those requirements and makes them easier to understand. It will also describes functionalities guaranteed for each stakeholder, in addition to constraints and assumptions that affect the overview of the system.

2.1 Product Perspective

Inspired by retro spaceship game Galaxia, which was published in 1970's by Bandai Namco, iSpaceship will be made up of an single player component as well as a multiplayer component. While the single player component is self contained as it only consists of the interactions between the player and the "AI" driven bots, the multiplayer component takes advantage of a server that is used to sync the game at all times. Doing so allows for both online and offline multiplayer as one of the desktops can also act as the server and the players would then play using LAN. An online server can also be used which gives the option to play with any one who has an internet connection, allowing players to play over any distance.

2.2 Product Functions

The system's functionality is centralized around the user upgrading their spaceship so that they are able to eventually complete the story mode. An experience level will be assigned to a user's spaceship based on their statistics, initially level 1. These statistics will improve as the user progresses thorough the story. Upgrades will be purchased with in-game currency and will come in the form of components. Purchasing of components will be made available through the shop. Components will augment the spaceships' abilities which are used in battle. In-game currency can be attained through winning battles or over time through the money generator. Battles involve the user's spaceship trying to eliminate an opponent's spaceship by using their abilities in a turn based fashion. Opponents will take the form of bots in the case of story mode, and other users in the case of multiplayer mode. Winning battles will result in in-game currency and XP being awarded to the user, losing 3 story mode battles will result in the user starting back to their base ship at level 1. Finally the user will be able to navigate through the shop, the story mode, and multiplayer mode.

2.3 User Characteristics

The game is intended for users of all ages and education levels. The game is run on a computer running Windows or Mac OS X, so basic technical knowledge of either operating system is required to install the game. Understanding written English is beneficial as well because the instructions and menu items are in English. The game is intended for players with any level of video game experience.

To facilitate this: The game must be appropriate for all ages. The installer will be simple enough for someone inexperienced with computers to use. The interface must be intuitive so that it can be quickly learned. To keep it interesting for players with varying levels of experience the game will progressively become more challenging.

2.4 Constraints

1. **Operating System**: The game is only intended for use on either Windows or Mac OS X.
2. **Game Engine**: The developers are constrained to using the Unity game engine.
3. **Input**: The only input methods allowed are with a cursor (mouse/trackpad) and a keyboard, touch-screen will not be supported.
4. **Networking**: Multiplayer functionality will use AWS for a web server to host the game.

2.5 Assumptions and Dependencies

1. **Hardware**: It is assumed that the target demographic has access to traditional computers with a mouse and keyboard.
2. **Game Engine**: It is assumed that the Unity game engine will be available for the developers.
3. **Operating System**: It is assumed that Windows and Mac OS X will be available for both the developers and users of the game.
4. **Web Server**: It is assumed that the AWS will be available to host the servers required for multiplayer.

2.6 Apportioning of Requirements

a) The game will only be made in English.

3 Use Case Diagram

Use Case	**Description**
Start Application	How the user initiates the system.
Enter Main Menu	How the user can load the game.
Load Profile	User loads existing game state.
Change Settings	User can change in-game settings
Create Profile	User can create a new game state.
Delete Profile	User can delete an existing game state.
Enter Hub	Main game screen where user can do various tasks.
Choose Mission	Where user selects their battles.
Play Previous	User can play a previously played mission.
Continue Story	User can continue the story where they left off.
Battle	User engages in a battle against AI spaceship.
Check Stats	User can check stats of enemy spaceship and their own.
Play Turn	User can make a move on their opponent on their turn.
Basic Attack	Users does flat damage attack.
Ability (Battle)	User uses an ability
Ability Cooldown	Ability cannot be used for several turns.
Skip turn	User can choose to let their opponent go.
Shop	Where user receives items for upgrades.
Buy	User can buy items.
Ability(Buy)	User buys new ability.
Generator Upgrade	User can upgrades their currency generators.
Ability Upgrade	User can upgrade power of existing ability.
Quit Game	User chooses to close application.
Auto Save	System prompts an autosave.
Select Overwrite	User selects an existing profile to overwrite save.

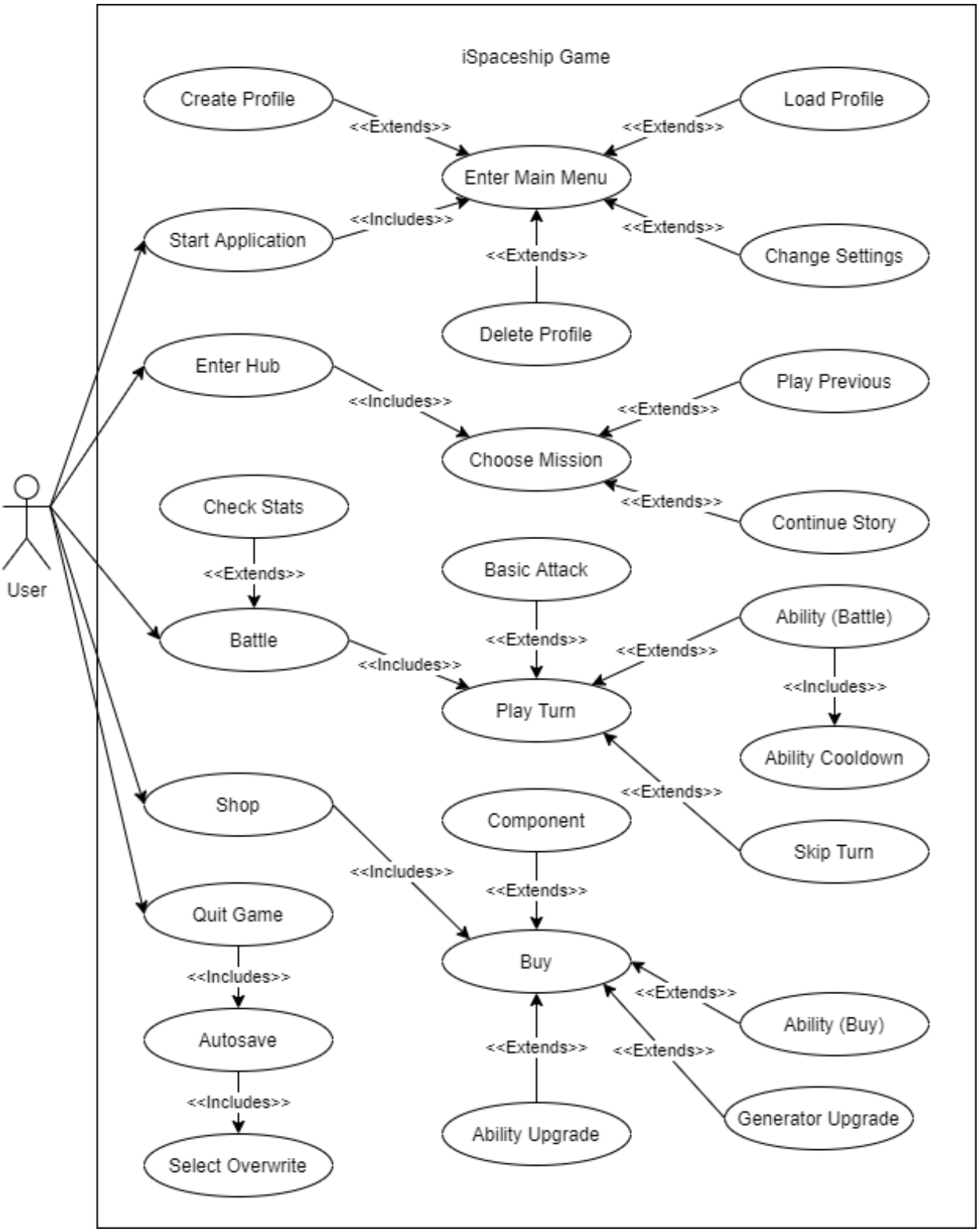

Figure 1: Use case diagram for the iSpaceship game.

4 Functional Requirements

VP1. User

BE1.1 The user starts the application.

i. The system shall allow the user to start a new game.
ii. The system shall allow the user to change the settings of the game.
iii. The system shall allow the user to load an existing profile.
iv. The system shall allow the user to play multiplayer through a connection.

BE1.2 The user wants to create a new profile.

i. The user is allowed to change the default name of the spaceship.
ii. The user is allowed to choose the colour of the spaceship.
iii. The system shall prompt the user for confirmation of profile details (color and username).
iv. The system must allow the user to cancel the process.
v. The system must present the new user with the backstory and then starts the first mission.

BE1.3 The user wants to buy a component/ability.

i. The system shall allow the user to buy components to upgrade abilities.
ii. The system shall update the stats of said ability, considering the component bought.
iii. The system shall allow the user to buy new abilities.
iv. The system shall update the deck of abilities available to the user.
v. The system shall unlock the components specific to the newly bought abilities.
vi. The system shall allow the user to buy power ups that can increase the money generation or XP gain for a period of time.

BE1.4 The user starts a mission.

i. The system shall present the user with the battle map depending on the mission.
ii. The system must allow the user to choose a difficulty.
iii. Upon victory, the system shall congratulate the user and unlock the next level.
iv. The system shall reward the user with XP and in-game currency.
v. Upon defeat, the system must take away one health slot from the user.
vi. The system must display the abilities of the spaceship to the user.
vii. The system shall display the user's and opponent's spaceship.
viii. The system must allow the user to choose a number of abilities available from the deck, only at the start of the mission.
ix. On the user's turn, the system must allow the user to choose an ability from the selected deck.
x. The system must control the opponent's spaceship, given it's the opponent's turn.
xi. Upon a mission completion, the system shall improve the opponent's spaceship

BE1.5 The user selects an ability in a mission.

i. The system shall render an animation that represents the ability used.
ii. The health of the player and opponent is modified accordingly in response to the ability.
iii. The system must deny any access to a used ability for a period of time.
iv. The system must calculate and show the effect of an ability used.

BE1.6 A discrete-time step has passed in the game.

i. The user receives a small amount of currency.
ii. The system shall display an updated version of the spaceship upon hitting a certain level.
iii. Upon each level increase, the system must increase the health and armor bar given to the user.

iv. The system must eliminate the spaceship if the user loses more than 3 battles.

v. Upon elimination, the system must restart the user's spaceship; returning the spaceship to level 0 and only having access to the first mission.

vi. The system shall automatically save the game.

BE1.7 The user wants to start a multiplayer game.

i. The system shall allow the user to create an online username.

ii. The system shall allow the user to establish a connection to a specific server.

iii. The system shall randomly select a player for the first turn.

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF1. The system shall use a minimalist design.

LF2. The spaceship shall be two-dimensional in design.

LF3. The game will be futuristic and rustic in design.

5.1.2 Style Requirements

LF1. The game will be two dimensional.

LF2. The game will follow standard C/C++ coding structure.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH1. The user shall only be prompted when decisions or confirmation are necessary.

UH2. The game state must be saved automatically upon the end of each battle.

5.2.2 Personalization and Internationalization Requirements

UH1. The game shall be available in several languages.

UH2. The hub can be fully customized.

5.2.3 Learning Requirements

UH1. Users can formulate their own unique play styles.

UH2. Users can formulate their own battle strategies.

UH3. Users will lay the foundation of a game meta.

5.2.4 Understandability and Politeness Requirements

UH1. The game instructions and given button shall be easy to understand and approachable.

UH2. The user with no gaming experience can learn battle mechanism after one round.

5.2.5 Accessibility Requirements

UH1. The game shall be accessible for deaf users.

UH2. This game shall be playable by individuals with at least one functioning hand.

UH3. The game will be playable by individuals with colour blindness.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR1. The system will allow at least 20 users play online simultaneously.

PR2. Any valid user input shall receive an immediate response by the game in a manner instantaneous to the user's perception (under 50ms).

PR3. Game initialization upon opening the application shall be less than 10 seconds.

5.3.2 Precision or Accuracy Requirements

PR1. All stats of spaceships must be an integer value.

5.3.3 Reliability and Availability Requirements

PR1. The software must be able to initialize and operate normally, not necessarily continuously, at any time.

PR2. The software must auto save after every important event.

5.3.4 Robustness or Fault-Tolerance Requirements

PR1. In the event of a crash, upon reboot, the last saved game state shall be loaded.

PR2. In the event of a corrupt save file, the user must restart from the beginning.

5.3.5 Capacity Requirements

PR1. The system shall store the players' profiles for up to 100 players.

PR2. The game will use a maximum of 20% CPU Power.

PR3. The game will require 500 mb of hardware space.

5.3.6 Scalability or Extensibility Requirements

PR1. The game shall have a developer API for game modifications.

PR2. The game will have shareable custom stories.

5.3.7 Longevity Requirements

PR1. The game will make periodic backups of player profiles.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE1. The system shall function on Windows or Mac OS devices.

OE2. The system shall function on devices with the Unity Engine.

5.4.2 Productization Requirements

OE1. The game shall have the option to minimize violence for younger markets.

OE2. The game shall be easy to install.

OE3. The game will follow industry file standards.

5.4.3 Release Requirements

OE1. Updates and maintenance releases must occur whenever bug-fixes or updates are made to the games software.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS1. The properties of game elements shall be updated appropriately to ensure game balance.

5.5.2 Supportability Requirements

MS1. The game will have a player report system to report player misconduct.

MS2. The game will have a feedback and support service.

5.5.3 Adaptability Requirements

MS1. The game shall be updated based on user feedback and feature suggestions.

5.6 Security Requirements

5.6.1 Access Requirements

SR1. Users shall be restricted from access to certain information such as feedback and game data.

5.6.2 Integrity Requirements

SR1. Users must not be able to modify data of local files.

SR2. The software shall be able to handle any malicious attack.

5.6.3 Privacy Requirements

SR1. The product shall not interact with any personal user data.

SR2. User data shall be kept private.

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

LR1. The product shall not contain any religious imagery or text.

LR2. The product shall not contain any references to any national disaster.

5.8 Legal Requirements

5.8.1 Compliance/Standards Requirements

LR1. This software shall comply with all national and federal software regulation laws.

LR2. This software shall comply with all relevant software standards.

LR3. This software shall comply with all relevant privacy acts.

LR4. Design shall follow Google C++ Style Guide

A Division of Labour

All members are responsible for 20% of the work for each milestone. The work for this document was divided equally amongst all group members.

Approved: _____
Group Member 1

Approved: _____
Group Member 2

Approved: _____
Group Member 3

Approved: _____
Group Member 4

Approved: _____
Group Member 5