**Software Requirements Document for Cyfighter**

Author: Group 18

Edwin Benggawan

Zachariah Brase

Brendan McCluer

Yangxiao Wang

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change** |
| 0.1 | 02/16/16 | Grp18 | Initial Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

[Introduction](#h.gjdgxs)

[Purpose](#h.30j0zll)

[Scope](#h.1fob9te)

[Definitions, acronymns, abbreviations](#h.3znysh7)

[References](#h.2et92p0)

[Overview](#h.tyjcwt)

[Overall Description](#h.3dy6vkm)

[Product Perspective](#h.1t3h5sf)

[Concept of Operations](#h.jrvh94owp80f)

[Major User Interfaces](#h.u83cfdslcf2f)

[Example Screenshot and description](#h.dja73y5n2ipb)

[Hardware Interfaces](#h.cfxf17asonaz)

[Software Interfaces](#h.oxm69a45x8l4)

[Communication Interfaces](#h.9xbdvp5a345f)

[Memory Constraints](#h.3vvlkhu94zoo)

[Operations](#h.prdwb6a2h1v2)

[Site Adaptation Requirements](#h.u0rff6e5zas)

[Product functions](#h.4d34og8)

[Example use case](#h.hkf7mc90owpa)

[User characteristics](#h.2s8eyo1)

[Constraints](#h.17dp8vu)

[Assumptions and Dependencies](#h.3rdcrjn)

[Specific Requirements](#h.26in1rg)

[External Interface Requirements](#h.lkjpwcbpaem5)

[User Interfaces](#h.lnxbz9)

[Hardware Interfaces](#h.ex4dxzx5y9gz)

[Software Interfaces](#h.riy9sowxauvx)

[Communications Interfaces](#h.w98htw75or0c)

[Features](#h.2aipcz56vhkb)

[FEATURE-1 ….](#h.vnyih1ybdz2z)

[Performance requirements](#h.ny9nu2ty0afa)

[Design Constraints](#h.44sinio)

[Software System Attributes](#h.2jxsxqh)

[Reliability](#h.53fqsxhjiylk)

[Availability](#h.w4ssq4okrnpu)

[Security](#h.3yfqhzotav38)

[Maintainability](#h.i5rltg8bw5wh)

[Portability](#h.cxn3tpgqlz9z)

[Other Requirements](#h.z337ya)

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the Cyfighter mobile video game on the Android OS. It will explain the features and variables of the system, the interfaces and screens of the system, the use cases and actors of the system, and the constraints under which the system must operate. This documented is intended for both stakeholders and developers of the system.

## Scope

The system will be a mobile video game for all users of android devices. This system is designed to allow users to play 10 different levels in a single player mode or co-op mode. The system will calculate and save the highest score for a particular level, as well as post the score to a global leaderboard. Using coins and Cys, the system will allow the user to purchase upgraded planes.

## Definitions, acronymns, abbreviations

// alphabetical list of terms and their descriptions

// This is part of analysis and you must make sure you describe terms used in this document

|  |  |
| --- | --- |
| Term | Description |
| Actor | Person who interacts with the system. |
| Administrator | User who has the privileges of a debugger, as well as changing prices, disabling and enabling features. |
| Android Studio | The IDE used for development of the game. |
| Appwarp | The Library used for multiplayer sessions. |
| Coin | A reward earned during levels of the game. The coins can be used to purchase upgrades in the game. |
| Co-op Mode | State in which the system is connected to **Appwarp**, sending and receiving messages to/from users**.** |
| Cy | A reward earned during levels of the game. The Cys can be used to purchase upgrades in the game. |
| Debugger | User who is granted special privileges, i.e. access to all features, levels, and logs. |
| Host Player | A **Player** user who will act as the “host” in a multiplayer session (Player 1). In **Co-op Mode**, this user has higher privileges than the **Joining Player,** such as choosing the level to play. |
| Joining Player | A **Player** user who will join the **Host Player** in a multiplayer session (Player 2). In **Co-op Mode**, this user has lower privileges than the **Host Player**. |
| LibGDX | The platform/library used to develop the specific gaming tasks, i.e. Camera, Sprites, Game States, Assets, Animation, handling input, updating, and rendering. |
| MySQL | Collection of information monitored by the system. |
| Leaderboard | The collection of high scores. |
| Linux Server | The server. |
| Player | A user with the lowest privileges, i.e. the consumer. |
| Room | A session created by Appwarp to hold multiple users on multiple systems, with whom will send and receive messages between them. |
| Server Database | The database used by the serverto hold leaderboard information. |
| Software Requirements Specification | A document describing all the functions of a system, as well as the constraints under which it must operate, i.e. this document. |
| Stakeholder | Person with an interest in the project who is not a developer. |
| System | The android program running the software. |

## References

// list of references for the reader of this document (if any)

## Overview

[NONE]

# Overall Description

// brief overview of the application (what does it do)

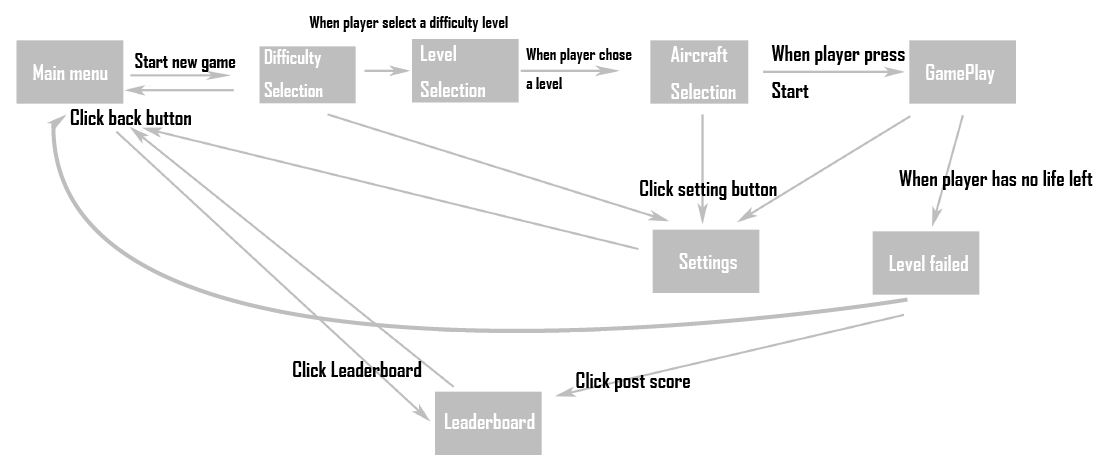
## Product Perspective

The Google Play store has multiple mobile games that are similar to the system described in this document, i.e “STRIKERS 1999”, “Raiden war 2015”, “STRIKERS 1945-2”, “raiden 2048 HD”. These games are similar in that they are vertical scrolling games in which the user navigates the screen and eliminates enemies to complete the levels. There are multiple planes that can be purchased.

### Concept of Operations

// brief TWO paragraphs on **how** the application will operate <do write about the parts of the system (eg ATM, Terminal etc>. << IMPORTANT SECTION: You can also draw a simple block diagram to clarify/explain>>

### Major User Interfaces



#### Example Screenshot and description

// <give a reference to your screens – i.e. to the APPENDIX>

### Hardware Interfaces

The hardware required for this system is a physical device capable of running the Android OS with a minimum of Android SDK 8.

### Software Interfaces

[NONE]

### Communication Interfaces

[NONE]

### Memory Constraints

[NONE]

### Operations

[NONE]

### Site Adaptation Requirements

[NONE]

## Product functions

// First draw the overall use-case diagram and give a brief description

// then – describe the major use-case

### 2.2.1 “Start Co-op Session” –**UC 1** (Start a level in co-op multiplayer mode, **Host Player**/**Joining Player**, Brendan McCluer)

In co-op mode, two aircrafts are controlled side by side with two players. Each player has their own copy of the game, selects their aircraft and weapons, controls their own aircraft, and posts their initials onto the leaderboard. This use case defines the steps to connect the two players and begin a session.

Main Scenario

1. **Host** **Player** selects “Co-op” from the Start screen.

2. **Host Player** selects “Host Session”.

3. **Host Player’s system** creates a new **room**.

4. **Host Player’s System** displays the **room ID.**

5. **Joining Player** selects “Co-op” from the Start screen.

6. **Joining Player** selects “Join Session” and enters the **room ID.**

7. **System** enters the **Joining Player** into the **room**.

Extensions

1a. **Host Player** is an **Administrator** or **Debugger.**

1a1. System gives the **Host Player Administrator** or **Debugger** privileges (ability to select any aircraft, weapon, and level).

3a. Failure in creating a new **room**

3a1. System displays an error message to the **Host Player** and exits.

5a. **Joining Player** is an **Administrator or Debugger**

5a1. System gives the **Joining Player Administrator** or **Debugger** privileges (ability to select any aircraft and weapon)

7a. Failure in joining a **room**

7a1. System displays an error message to the **Joining Player** and exits.

### 2.2.2 “Post High Score” –**UC 2** (Upload the score after game over, **Player**, Brendan McCluer)

Upon reaching the Game Over screen, the user may view the Leaderboard Screen. From here the user can post his/her score to the **server** where it may be viewed by other players.

Main Scenario

1. **Player** selects emblem and types initials.

2. **Player** submits to the **leaderboard**.  
 3. **System** sends the emblem and initials to the **server**.

4. **System** displays success message to **player**.

Extensions

3a. **Server** is offline/fails

3a1. **System** displays “server unavailable” message to **player**.

2.2.3 “Select Difficulty” **–UC 3**

(Choose a difficulty level, **Player**, Edwin Benggawan)

The Difficulty Selection screen appears after the user started a new game. There are three levels of difficulty: easy, normal, and hard. The **system** will set the multiplier for obstacles corresponding to the chosen difficulty.

*Main scenario:*

1. **Player** selects one of the three difficulty levels (easy, medium, or hard).
2. **System** sets the difficulty multiplier.
3. Screen is displayed with UI to show the available levels based on the selected difficulty.

*Extensions:*

2.2.4 “View Leaderboard” **–UC 4**

(View scores from previous games, **Player**, Edwin Benggawan)

The user can view worldwide single scores and co-op scores as well as scores for different difficulty levels using tabs. This screen appears by choosing “Leaderboard” from the main menu. The user can also view the Leaderboard screen after a game ends.

*Main scenario:*

1. **Player** selects “Leaderboard” from the Start screen.
2. **System** retrieves data from **Server Database**.
3. Screen displays the scores along with emblems and initials for single player and easy level by default.

*Extensions:*

2a. **Server** is offline/fails:  
 2a1. **System** displays “server unavailable” message to **player**.

3a. **Player** selects other difficulty levels tab and/or co-op:   
 3a1. Screen displays the scores with emblems and initials for the selected difficulty level and/or co-op.

## User characteristics

// typical user characteristics, frequency of usage et**P**c

## Constraints

// all conditions that may limit design options (INCLUDE NON FUNCTION CONSTRAINTS)

## Assumptions and Dependencies

// hardware and software assumptions and dependencies

# Specific Requirements

// Here you need to put in details (if any). Mark items [None] if you do not have any information.

## External Interface Requirements

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communications Interfaces

## Features

### FEATURE-1 ….

## Performance requirements

## Design Constraints

## Software System Attributes

### Reliability

### Availability

### Security

### Maintainability

### Portability

## Other Requirements

// ADD Appendices (if any)

// Regenerate Table of Contents