Software Requirements Specification

for

Project Haptic Lightning

**Version 1.0 approved**

**Prepared by Colton Young, Florida Southern College**

**Created 4/25/2016**

**Table of Contents**

**Table of Contents**

**Revision History**

**1. Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Project Scope

1.5 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

**3. System Features**

3.1 System Feature 1

3.2 System Feature 2 (and so on)

**4. External Interface Requirements**

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

**5. Other Nonfunctional Requirements**

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

**6. Other Requirements**

**Appendix A: Glossary**

**Appendix B: Analysis Models**

**Appendix C: Issues List**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Colton Young | 1/25/16 | Filled out sections 1.1-2.4 | 0.01 |
| Colton Young | 2/5/16 | Filled out sections up to 4.1 | 0.05 |
| Colton Young | 2/22/16 | Added UML Diagrams to SRS | 0.1 |
| Colton Young | 3/19/16 | Revised features section to match what we will try to complete- updated user stories | 0.95 |
| Colton Young | 4/19/16 | Updated last few sections | 1.0 |

# Introduction

## Purpose

The purpose of this document is to enumerate the requirements needed for building a simple single-tap video game that can be run on Android devices and PCs. The document describes the requirements for all components of the game. The SRS document should directly lead to a feasible design for implementing the system.

## Document Conventions

The priorities for higher level requirements are inherited by the more specific requirements. For example, the priority overall will always be to have functional gameplay which carries over into each of the subsystems that make up the gameplay. Any elements highlighted in yellow in this document are considered priority requirements. Any elements highlighted in red are not completed.

## Intended Audience and Reading Suggestions

This document is intended for the developers, testers, evaluators of this classroom project, and documentation writers of Project Haptic Lightning. Section one and two should be read by all members of the intended audience. System designers should refer to sections three and four for a detailed description of the system’s functional requirements.

## Project Scope

The software to be designed is a video game played on a Windows PC or android device that challenges the user to click on different shapes on their device’s screen with increasing difficulty. There will be a timer constantly running down from a pre-specified amount of seconds to 0 and when it hits 0, the player loses. The shapes will appear randomly and shrink rapidly, meaning the user only has a small amount of time to react and tap the object before it shrinks to nothing. When the object shrinks to nothing, time will be subtracted off of the remaining time. If, however, the user manages to tap the shape before it shrinks away, time will be added to the timer. The longer you play, the more shapes will spawn and the more time will be subtracted for a miss- eventually, you will lose because you can’t keep up. The game is meant to entertain, but more specifically, it is designed in a way that should be attractive to users who want a simple or quick experience with relatively low-mental involvement. This game should be easy to pick up and play one round while waiting in a queue or for short, immediate entertainment.

## References

Minimum required specs: Android 2.3.1 Gingerbread or higher

# Overall Description

## Product Perspective

This product is intended to be a wholly self-reliant product that could feasibly be released to a mass market (i.e. as a downloadable game) and enjoyed by users. The product is part of an educational experience for the software engineers.

## Product Features

The product will be a compilation of multiple game systems such as reaction-based reward and negligence-based punishment and escalating difficulty. There will be only one game mode where the user attempts to keep the clock from reaching 0 by tapping as many shapes as possible. The shapes that will shrink and are the target of the user’s input will likely be instances that have some random or constant modifiers applied to them

## User Classes and Characteristics

Project Haptic Lightning will have one primary user class, the player. Members of this user class may be for ages 5+ and will have basic computer literacy skills (such as device navigation). There are no strict educational requirements for this user class.

## Operating Environment

The game will be used on a hardware device that runs a Windows OS and will run using the very flexible Unity Game Engine. We will use the 2D variant of this engine to create both the user interface for menus as well as for the game design itself. The game might draw on assets created from one of many asset design applications or from the Unity asset store itself. The product may use a database service to store high scores.

## Design and Implementation Constraints

Hardware limitations for Project Haptic Lightning will be minimal, requiring Android OS 2.3.1 Gingerbread or higher. The app must be able to write to a file to store necessary long-term data.

## Assumptions and Dependencies

With mobile games such as Project Haptic Lightning, a free-to-play model is usually used with ad space generating revenue for the product. Our assumption is that, given we are able to provide space for ads, there will be outside parties willing to use the ad-space.It is also assumed that the Unity game engine will be reliable and robust. We plan to use assets from the Unity Asset Store if we cannot generate them on our own. We will depend on the assets being fully functional.

# System Features

## Timer

3.1.1 Description and Priority

*High Priority- A working timer that is displayed on the screen for the user to see when playing the game. Timer will be displayed as a bar that grows for successful taps and shrinks for misses*

3.1.2 Stimulus/Response Sequences

*Users can affect the timer by adequately and efficiently clicking shapes to add time to the running clock. Users can also lose time on the clock by letting shapes shrink to nothing or tapping outside of a shape.*

3.1.3 Functional Requirements

*The clock will be displayed as a bar that grows and shrinks with successful or failed taps. The clock will not run past 0:00. The clock will not deduct time past 0:00.*

REQ-1: TIMER/TMR

User Story- Johnny Cats wants to play a game that pushes him to be quick to react. He wants to know just how quick he needs to be at all times, so something like a timer that shows how much longer he has left before he loses would make a game enjoyable for him.

## Randomly Appearing Shapes

3.2.1 Description and Priority

*High Priority- Random shapes will appear randomly within the game space.*

3.2.2 Stimulus/Response Sequences

*The User will have no control over the appearance of the shapes in normal mode. In normal mode, the shapes will appear increasingly quickly over time. In the second mode, the user will essentially have control over the shapes appearing because they appear immediately after the previous shape is pressed.*

3.2.3 Functional Requirements

*In the normal mode where shapes appear increasingly quickly, we will need some exponential function that will ramp up difficulty over time by increasing the speed at which shapes appear. For the second game mode, shapes will merely need to appear as soon as the previous one is gone.*

REQ-1: APPEAR/APPR

User Story-Sammy Sprinkles wants to play a game that’s a little bit different each time. He doesn’t like the traditional *level* structure of most games, but instead likes when a game keeps mechanics the same but varies the challenge by giving him a new scenario each time he plays.

## Shrinking Shapes

3.3.1 Description and Priority

*High Priority- From the moment the shape appears, it will shrink, slowly at first, but then rapidly after a half-second or so*

3.3.2 Stimulus/Response Sequences

*The User will have no control over the appearance of the shapes in normal mode. In normal mode, the shapes will appear increasingly quickly over time. In the second mode, the user will essentially have control over the shapes appearing because they appear immediately after the previous shape is pressed.*

3.3.3 Functional Requirements

*In the normal mode where shapes appear increasingly quickly, we will need some exponential function that will ramp up difficulty over time by increasing the speed at which shapes appear. For the second game mode, shapes will merely need to appear as soon as the previous one is gone.*

REQ-1: SHAPE/SHP

User Story-Mary Winters likes a game where that rewards quick reaction. If she’s not quick to react, she feels like the gameplay should become harder, but still not impossible.

## Deleting Shape Objects

3.3.1 Description and Priority

*High Priority- When a shape shrinks past a certain threshold, it will be removed as an object*

3.3.2 Stimulus/Response Sequences

*The user will have no control over this object.*

3.3.3 Functional Requirements

*Shapes will have a threshold attribute that will be the same across all shapes and when the scale of the shape is shrunk past this value, it will be removed*

REQ-1: DELETE/DLT

User Story-Sean Poole likes a game where there is consequence for his failure. He prefers games where, if his skill doesn’t meet the demands of the game, he is negatively impacted. He feels that this gives a game a sense of stakes and excitement.

## Touch Screen

3.1.1 Description and Priority

*Low Priority - The user will utilize the touchscreen to tap the buttons of the user interface as well as to tap the target shapes of the game.*

3.1.2 Stimulus/Response Sequences

*The user will be able to tap the screen in order to hit the target shapes of the*

*game and menu items with a high degree of accuracy*

3.1.3 Functional Requirements

*Touchscreen technology will be integrated through the use of Unity Software.*

REQ-1: TCHSCRN

User Story-Lawrence Sonny likes a game that he can play on his phone, with nothing other than his finger as an input device. He wants to be able to play the game entirely by tapping the screen.

## Creating Screen Transitions

3.1.1 Description and Priority

*High Priority - The user will be able to navigate the screens and menus of the game from one central hub page.*

*3.1.2* Stimulus/Response Sequences

*The user will not interact directly with the transitioning of screens.*

*3.1.3* Functional Requirements

*Screen transitions are created in Unity by using animation and State Machines to drive*

*and control each screen. We create a class called ScreenManager which will contain*

*an Animator Control with 2 states(Open and Closed) and boolean Parameter (open).*

REQ-1 SCRTX

User Story-Bruce Green likes a game that has an intuitive and simple menu so that he can navigate and find whatever setting or aspect of the game easily. He feels that it should only take one button navigation (one tap) to get to the piece of the game he wants to access

## Back Button

3.3.1 Description and Priority

*High Priority- The user will be able to navigate to the previous page or pause the in-progress game by pressing the back button*

3.3.2 Stimulus/Response Sequences

*The back button will be functional at all times and when pressed, will navigate one page back in the user interface or pause the game so the user can quit or change the settings.*

3.3.3 Functional Requirements

*The back button on the device must be usable at all times (i.e. never enters a sleep state) and will have functionality on all pages*

REQ-1: BACK/BCK

User Story-Adam Kove likes to be able to return to the previous screen that he accessed whenever he wants. He hates having to navigate through complicated menus or when a back/escape button takes him all the way back to the main menu. He wants to be able to navigate any app, just like he would using his browser’s back button.

## Achievement System

3.3.1 Description and Priority

*Medium Priority- The player is rewarded when certain game milestones are reached, such as a certain amount of time played, a certain high score reached, or for sharing with friends*

3.3.2 Stimulus/Response Sequences

*The user will be notified when they have completed the requirements for an achievement and will be able to view all possible achievements*

3.3.3 Functional Requirements

*The achievement system will be tied to the user’s ID in some way so progress over multiple game sessions is saved, therefore, some kind of database service may be necessary. The achievement system will accurately track progress*

REQ-1: ACHIEVE/ACHV

User Story-Jeff Ramsey likes a game that rewards him for his feats and the amount of time he spends playing. He feels that if there are no rewards or something showing his devotion to the game, it’s probably not worth playing. He’d like to be rewarded not only for his skill, but his time invested as well.

## Leaderboard—NOT COMPLETED

3.1.1 Description and Priority

*Medium Priority - The user will be able to view their rank against a global list of players as well as specifically a group of friends or choose to abstain from adding their high score*

3.1.2 Stimulus/Response Sequences

*The user will input a name which will be recorded at the end of gameplay and added to the list of players depending on their rank based on the length of time played for one game session, or choose to opt out*

2.1.3 Functional Requirements

*Authorization asked for before download to ensure that the user knows the app will be accessing their user information. A server will hold the list and update after each full play session.*

REQ-1: LEADERBOARD/LDRBRD

User Story-Elise Williams likes a game that she can tell her friends about and that they will want to play. She likes to compete with her friends for high scores and be able to show her skills off to others. She also likes to compare herself to everyone else playing the game to determine how good she is.

## Unity Game Over

3.1.1 Description and Priority

*Medium Priority - The user will play the game to the best of their ability. At the end of the game, the GameOver state will be initiated.*

3.1.2 Stimulus/Response Sequences

*The user will not interact directly with the GameOver sequence but can end it by tapping*

3.1.3 Functional Requirements

*At the end of a user’s game the GameOver class will be called which includes a*

*Gameover variable, restart delay, restart timer, and an Awake function.*

REQ-1: GMOVR

User Story-Matthew Peak likes a game that ends. Simply put, he hates games that continue indefinitely. He likes games that either end when certain goals are met, or if the game becomes to hard to continue. When the game ends, he likes to see relevant stats for his game session, such as how well he did.

## Advertisements—NOT FULLY FUNCTIONAL

3.3.1 Description and Priority

*Low Priority- The game will be monetized by providing ad space to clients who wish to advertise in our game*

3.3.2 Stimulus/Response Sequences

*The user will be directed to the advertisement’s website if pressed, but nothing is required*

3.3.3 Functional Requirements

*Advertisements will be banner ads on the top of the screen at the Game Over screen and the space will be sold by the game developers to clients*

REQ-1: ADVERTISE/ADS

User Story- Michael Burns likes a game that is free to play. He is completely fine with ads appearing in the gamespace, as long as he doesn’t have to pay to play the game.

## Social Media Integration- NOT FULLY FUNCTIONAL

3.1.1 Description and Priority

*Low Priority - The user will be able to log into Facebook to post the game score after a game.*

3.1.2 Stimulus/Response Sequences

*The user will log in to Facebook with their account information. The app will connect to the media site and will be able to upload a high score as a status update.*

3.1.3 Functional Requirements

*Authorization asked for before displaying the log in screen to ensure that the user knows the app will be accessing their social media data.*

REQ-1: SMI

User Story-Sanjay Patel likes games that he can show his friends and get his friends hooked on. He likes to be able to see how well they’ve doneand show them how well he’s done. He is Facebook friends with all of his closest buddies and always links his game accounts to Facebook so that he can easily post his high score.

## Online Co-op—NOT COMPLETED

3.1.1 Description and Priority

*Low Priority - The user will be able to host a game or connect to another person’s game and share the game space with them.*

3.1.2 Stimulus/Response Sequences

*The user will select an option to join or host game and, once connected, be able to participate in the game exactly as if playing a single player game.*

3.1.3 Functional Requirements

*The user should be able to find and host games from a single device and all experiences of the single-player game should carry over.*

REQ-1: SMI

User Story-Terry Richards likes to play games with his best friend where they can both work towards a shared goal in the same space.

# External Interface Requirements

## User Interfaces

*The User Interface will be a logical and simple menu system that gives the user access to the game, the options menu, the leaderboard, the rules, the credits, and finally, the ability to exit the game entirely. The user should never have to delve deeper than one button press to reach any option or game content that they wish to see. Mock-up at this link:*

*https://drive.google.com/file/d/0B-Dw9kbBjGlWYTVvQVp5R0sweG8/view?usp=sharing*

*The menu will allow you to reach any function that you might want to access with this game (sound options, change gamemode, etc.) in one tap.*

*The play space will have one button that allows you to begin the game The playspace will be a colorful background that is empty except for a time bar in the top center of the screen. The randomly appearing shapes will appear across this screen.*

*The instructions screen will be a text screen, detailing how to play the game.*

*The options screen will have two sliders- one for music volume and one for sound volume. These sliders will be able to be adjusted independently. The third option will be a theme button. The button will display, in text, the currently selected theme. When tapped, this button’s text will change to something else (e.g. “Kitten Theme” or “Monochrome Theme”) and the game assets will change to reflect this.*

*The leaderboard screen will be a local high score board showing the best scores achieved by players on that specific device.*

*The achievements screen will show the player’s progress towards certain achievements (e.g. time played, high score, etc.) and will show their progress numerically and as a progress bar.*

*The credits screen will show the names of the creators.*

*The exit button will close the game.*

*The Unity game engine provides tools for creating a game menu that can initiate different game states and that allow the user to modify some elements of their game (i.e. the options menu.)*

## Hardware Interfaces

*This app is supported on both Windows PCs and Android devices running Gingerbread 2.3.1 or higher. It is reliant on no other hardware.*

## Software Interfaces

*This product mostly uses the Unity Engine library to implement gameplay aspects using the pre-defined Start(), Update() and OnClick() methods. The C# Serialization library is also utilized to help write pertinent information to a file using a binary formatter.*

## Communications Interfaces

*This product can be run entirely locally, however, a network connection is required to utilize the Facebook implementation if it was completed and to populate new advertisements if that was implemented.*

# Other Nonfunctional Requirements

## Performance Requirements

### Game Board Update

From clicking the app icon, the main menu should load within 10 seconds and have minimal to no loading times.

The achievements page should track achievement progress with high precision, giving accurate data to the second.

## Security Requirements

Game data should not be accessible by the user so it is “encrypted” by formatting it and writing it to file. No sensitive user information is required in this game.

## Software Quality Attributes

### Portability

The game should be usable on-the-move, as a main audience for this product is people who are out-and-about waiting in a line or waiting on someone, so they can pull out the app for a quick match.

### Usability

The menu and game should be immediately understandable and the user should understand the gameplay aspects by spending only about 30 seconds on the instructions scene.

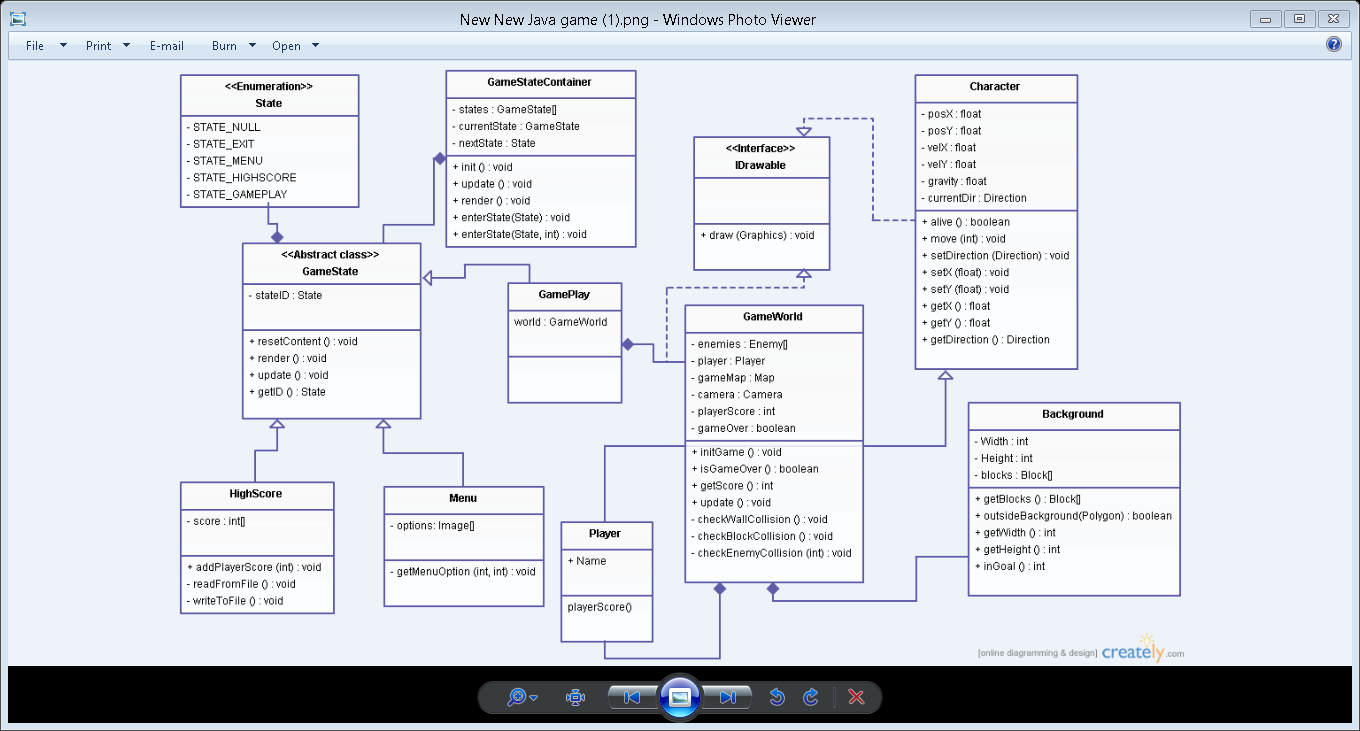
**Appendix A: Glossary**

**Tap** A tap occurs when the user has placed their finger (on an Android device) or clicked with their mouse cursor (on PC) in the “hitbox” of a shape

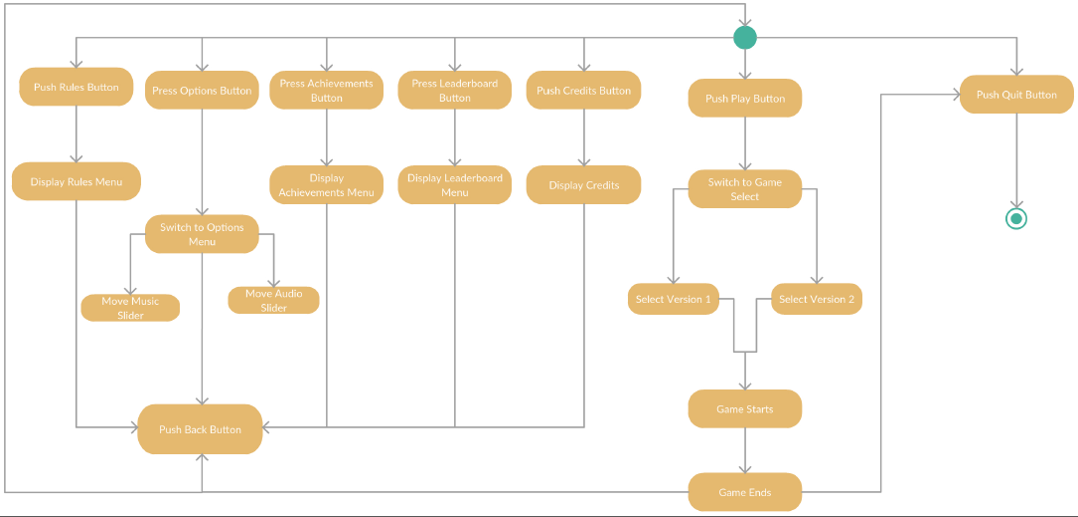
**Hitbox** The area encompassing the visual aspect of a shape that will recognize a tap and delete the object

**Appendix B: Analysis Models**

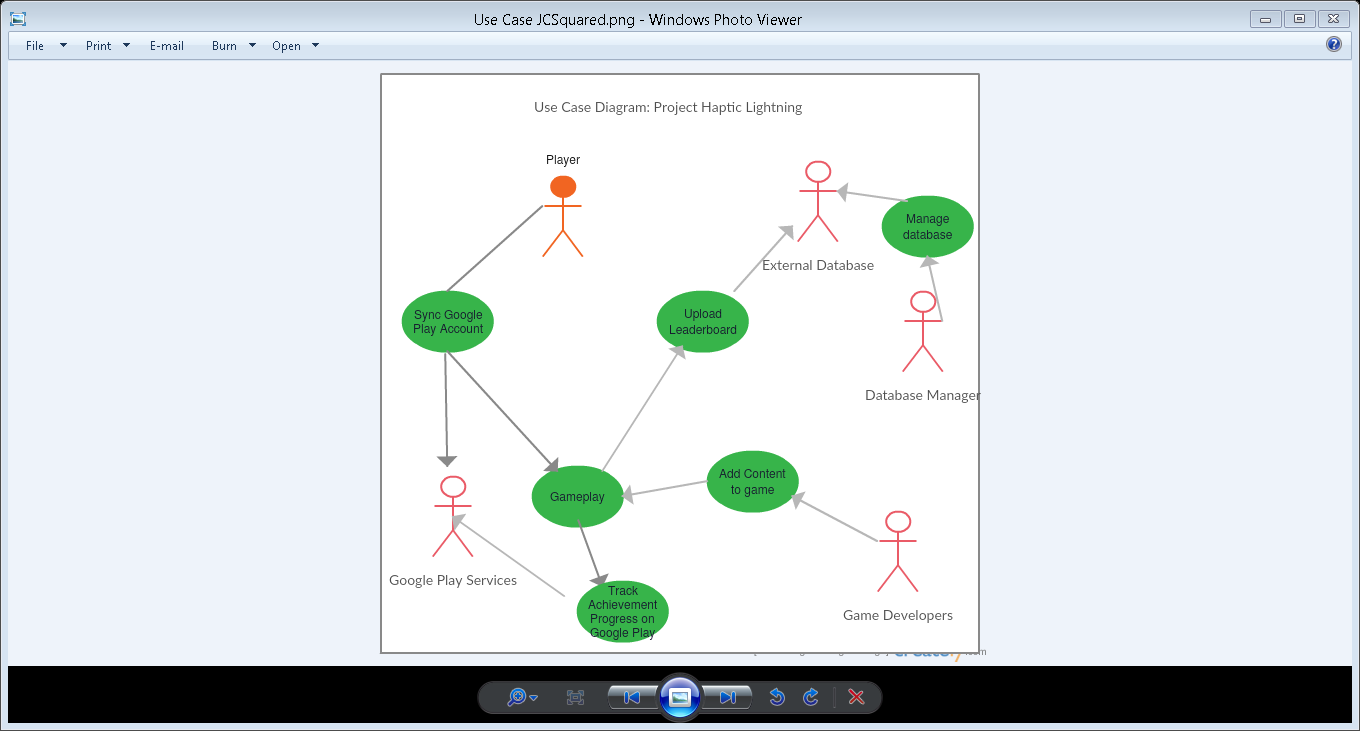
*Class Diagram*



Activity Diagram



Use Case Diagram



**Appendix C: Issues List**

The Facebook integration is not fully implemented. A login feature is available, but it is not usable and doesn’t lead to any further integration.

The online leaderboard was never fully implemented using an outside service- the achievements information however could converted into a form that tracks a local leaderboard.

The advertisements feature only works in test mode, meaning it never actually creates an advertisement. If run outside of test mode, the app will just show a blank black screen that the user must tap in the top right of the screen to get rid of.

Online multiplayer is not functional, however, there is a build where another user could watch someone play the game by joining into the game. However, they cannot interact with anything, nor can they see the timer.