Software Requirements Specification

for

Physics Based Pachinko

Prepared by Alex Flitter, Dan Cleaver, Christian Lostoski, Keenan Long

University of Maryland, Baltimore County

2/25/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This project will be a physics-based Pachinko game. There will be several "levels" with different art styles and pachinko courses. The ball will use physics simulations to determine its path as it collides with various obstacles. The player's objective is to catch said ball when it eventually reaches the bottom of the course. The player will be awarded points as they catch balls. This document will cover all requirements of the system, including but not limited to the system features, requirements, and impact.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document are both the investors and future developers of this game. This will serve as a guide to understanding the design decisions that were made. Future developers would find the sections for System Features(2), Performance Requirements(3.1), and Software Quality Attributes(3.4) most useful for their understanding of this project. Investors would find the sections for Product Scope (1.3), System Features(2), and Business Rules(3.5) most useful for their understanding of this project.

1.3 Product Scope

This program will be a multi-level Pachinko game. It will have various levels with increasing levels of difficulty. Each level will have a different art style attached to it as well. The "ball" that will be falling into the pachinko board will use a physics simulation to determine its path down the board. At the bottom of this board there will be a basket that will be controlled by user input. When the user successfully catches a ball in the basket, they will be awarded a point. The software being used for this project will include objects for both the ball and the basket as well as physics simulation software through a game engine.

2. System Features

2.1 Main Menu

2.1.1 Description and Priority - Priority 2

This menu has two buttons. The first button will be the PLAY button. Once pressed, the game will begin. When pressing the LEFT or RIGHT arrow keys, the game's level will change. The background of the main menu will change depending on what level is selected. The second button will be a cosmetics button. This will again open a submenu that will allow the user to alter several cosmetic options in the game. On the side of the main menu will be the leaderboard for the currently selected level. This will be the main controlling menu of the game, so this is a high priority, but not as high a priority as the actual levels themselves.

2.1.2 Stimulus/Response Sequences

If the user presses the PLAY button, the level will begin and the player can play the game. If the user presses LEFT or RIGHT the level will change. If the COSMETIC button is pressed, the cosmetics menu will appear.

2.1.3 Functional Requirements

REQ-1: Scoring System
A system of providing points to a user during a game. As the user catches balls, their score shall increase.

REQ-2: Writing File to Computer
The leaderboard information shall be stored in a separate file. As the user sets a score, it will be written to the file. On the main menu, said scores will be read from the file and displayed.

REQ-3: Levels to select from
The level menu needs levels to select from.

REQ-4: Objects for each cosmetic item
The cosmetics menu will need cosmetic objects and images.

2.2 Pachinko Game

2.2.1 Description and Priority - Priority 1

Our project will implement a physics-based Pachinko game with four levels of difficulty. The game's difficulty can be changed by the user within the main menu via a left or right button press. Each level of difficulty will have a unique animated background theme to further engage the user during the game. Users will be able to utilize power-ups like black holes to congregate the falling balls in an area, zero gravity to slow the falling speed of the balls, and increased basket size to aid in catching. The game will utilize a three-star scoring system with the requirement of two stars to unlock the next level of difficulty. A points range system will determine the number of stars a user earns during a round. To enable the points multiplier, the user will need to successfully capture five balls in a row to get two times points. Lastly, after completing a round, the user will be prompted to enter a username to be displayed on the leaderboard.

2.2.2 Stimulus/Response Sequences

Once the user has selected a level to play, the Pachinko balls will begin to enter the arena. As the player presses the LEFT and RIGHT arrow keys, the bucket at the bottom of the map will move in that respective direction. The objective is to catch as many balls in the basket as possible. Based on the percentage of balls caught in the basket at the end of the round, the player will be rewarded with either one, two, or three stars.

2.2.3 Functional Requirements

REQ-1: Ball Spawning

The rate at which the balls will spawn and where the spawn will need to be determined.

REQ-2: Physics Simulation Engine

Utilize the pygame physics engine to give the balls realistic falling speed and motion throughout the game board. Some environments that will be implemented will be low gravity and high gravity environments.

REQ-3: Bucket Movement

The bucket will need a function to move according to the user's commands.

REQ-4: Scoring System

A System of providing points to a user during a game. As the user catches balls, their score shall increase.

2.3 Leaderboard

2.4.1 Description and Priority - Priority 3

Our goal for the leaderboard is to keep track of the top ten scores for each level. When you are in the main menu, the leaderboard should correlate to whatever level is currently selected. The leaderboard will sort the score where the highest score will be at the top, and the lowest score will be at the bottom. Each score will have the username of the person who got it and when they got that score. When the user exits the program, the program will write a file to the computer that will be called the next time the program is executed. This file will keep track of the scores on each leaderboard.

2.4.2 Stimulus/Response Sequences

When the user gets a score higher than one of the leaderboard scores, it will congratulate the user and be shown the updated leaderboard. When the user returns to the main menu, the leaderboard will also be updated there.

2.4.3 Functional Requirements

REQ-1: Scoring System
A System of providing points to a user during a game. As the user catches balls, their score shall increase.

REQ-2: Writing File to Computer
The leaderboard information shall be stored in a separate file. As the user sets a score, it will be written to the file. On the main menu, said scores will be read from the file and display the scores.

2.4 Cosmetics

2.4.1 Description and Priority - Priority 4

This menu will have three selection types. The first selection type will be the background image. There will be at least four background images, one per level. As you beat each level, its respective background image is unlocked. The player can then select the image, and that will be displayed in the background of the game. The second selection will be basket look. The player will be controlling a basket to catch the balls. This basket will have several different appearances. As the player progresses in the game. More cosmetics unlock for the user to select. Lastly, there will be ball image selections. Just like the background image and basket, the ball images will unlock as the user progresses in the game.

2.4.2 Stimulus/Response Sequences

As the user's mouse hovers over a cosmetic to select. The specific image will be highlighted. If a user clicks on the cosmetic. It will be outlined in gold to denote that it is the currently selected cosmetic. From that point on, that cosmetic item will be used for its respective item.

2.4.3 Functional Requirements

REQ-1: Scoring System
A System of providing points to a user during a game. As the user catches balls, their score shall increase.

REQ-2: Objects for each cosmetic item
The cosmetics menu will need cosmetic objects and images.

REQ-3: Unlock Requirements
A statement that determines if a cosmetic has been unlocked yet.

3. Other Nonfunctional Requirements

3.1 Performance Requirements

This should be a relatively low-power program. This means that an average laptop should be able to run this program without the need for extreme hardware. Ideally, this program should run on a laptop with integrated graphics, 2 cores, and 8GB of ram. The program shall run at 30 frames per second, and will experience little to no lag. This game shall be able to run on both Windows and Linux operating systems.

3.2 Safety Requirements

This system shall not cause any form of harm or visual discomfort.

3.3 Security Requirements

The program shall not use the internet.

The program shall not access or create sensitive data.

No user authentication is necessary or will be implemented.

During normal operation, the program shall not crash the user’s computer or interfere with other running processes.

3.4 Software Quality Attributes

The program shall be distributed as an executable program, designed to be run on Windows and Linux. Running the program and playing the game should be easy to accomplish for the average user.

3.5 Business Rules

Our product shall be very consumer friendly, meet specified requirements, and satisfy desired attributes. Development of the product will follow the agile development process.

4. Other Requirements

Other requirements include how our file will write and read our leaderboard and cosmetic file. Each score in the file will have a string containing the username for the score, the score the user got, and the time they got the score. Each cosmetic will have a boolean value, false if the user has not unlocked the specific item and true if they have.