Software Requirements Specification Document

Project S

Kevin Choum, Eric Chu, David Cortez

Department of Computer Science, California State University, Fullerton

CPSC 362-07: Foundations of Software Engineering

Professor: Dr. Lidia Morrison

December 8, 2021

Table of Contents

1. Introduction

a. Description

Project S is a 2.5D platformer made in Unity and meant to run as a windows executable. In the game the user plays as an android who is trying to escape the facility they were made in while dodging the facility's many hazards and security. The player escapes by navigating through a series of room based levels before confronting a final boss at the end.

b. Intended Audience

The intended audience of this document is the professor of CPSC 362-07, Dr. Lidia Morrison as well as students enrolled in CPSC 362-07 which includes the members who created the application Project S. The intended target audience of Project S are teenagers and young adults who enjoy a movement-based platformer video game.

c. Operating Environment

Project S is a desktop application created using Unity, a game engine developed by Unity Technologies that allows for cross-platform development primarily using C#. The target operating system for this application is Windows 10 and above. However, Unity allows this application to be built for mobile devices, console devices, web applications, as well as linux operating systems.

d. Design and Implementation Constraints

Project S will only run on desktops with Windows 10 and above operating systems. Other operating systems such as MacOS or Linux are not compatible with this application. Project S is

built in a three-dimensional environment, but gameplay and user controls are constrained to a two-dimensional plane. Keyboard and mouse are the only input devices required to play and run Project S. Speakers are not necessary to play Project S; however, user experience may be worse.

2. Functional Requirements

a. Title Screen

- When the user clicks on the windows executable for this game, the system shall start the game and load the title screen.
- Title Interface
    - When the user clicks on the play button, the application shall load the first level of the game.
    - When the user clicks on the options button, the application shall display the options menu.
    - When the user clicks on the credits button, the application shall display the credits of the game.
    - When the user clicks on the exit button, the application shall close the game.
- Options Interface
    - When the user clicks on the return button, the application shall return to the title screen.
    - When the user clicks and moves the volume slider, the application shall adjust the volume level of the game according to where the user moves the volume slider.
    - When the user clicks on the options button in the title screen, the keybindings for the game shall appear.

b. In-Game

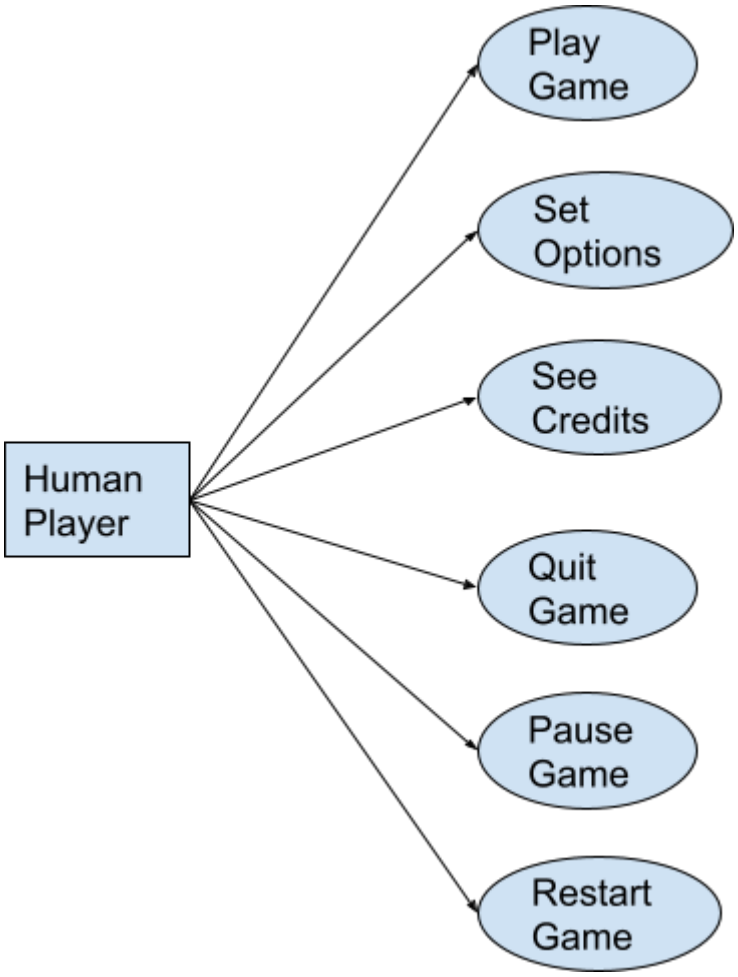
Note: the README.md file contains more detailed descriptions of the keybinds.

- Keybinds
    - When the user presses the “A” key on their keyboard, the application shall move the player left.
    - When the user presses the “D” key on their keyboard, the application shall move the player right.
    - When the user presses the spacebar on their keyboard, the application shall move the player up.
    - When the user presses the “J” key on their keyboard, the application shall fire a bullet towards the player’s current directional velocity.
    - When the user presses the “K” key on their keyboard, the application shall make the player perform the skill “TacticalRoll.”
    - When the user presses the “L” key on their keyboard, the application shall perform the skill “Dash.”
    - When the user presses the “U” key on their keyboard, the application shall make the player attack in front of them.
    - When the user presses the “I” key on their keyboard, the application shall make the player perform the skill “AirSlam.”
    - When the user presses the “O” key on their keyboard, the application shall make the player perform the skill “StyleSkill.”

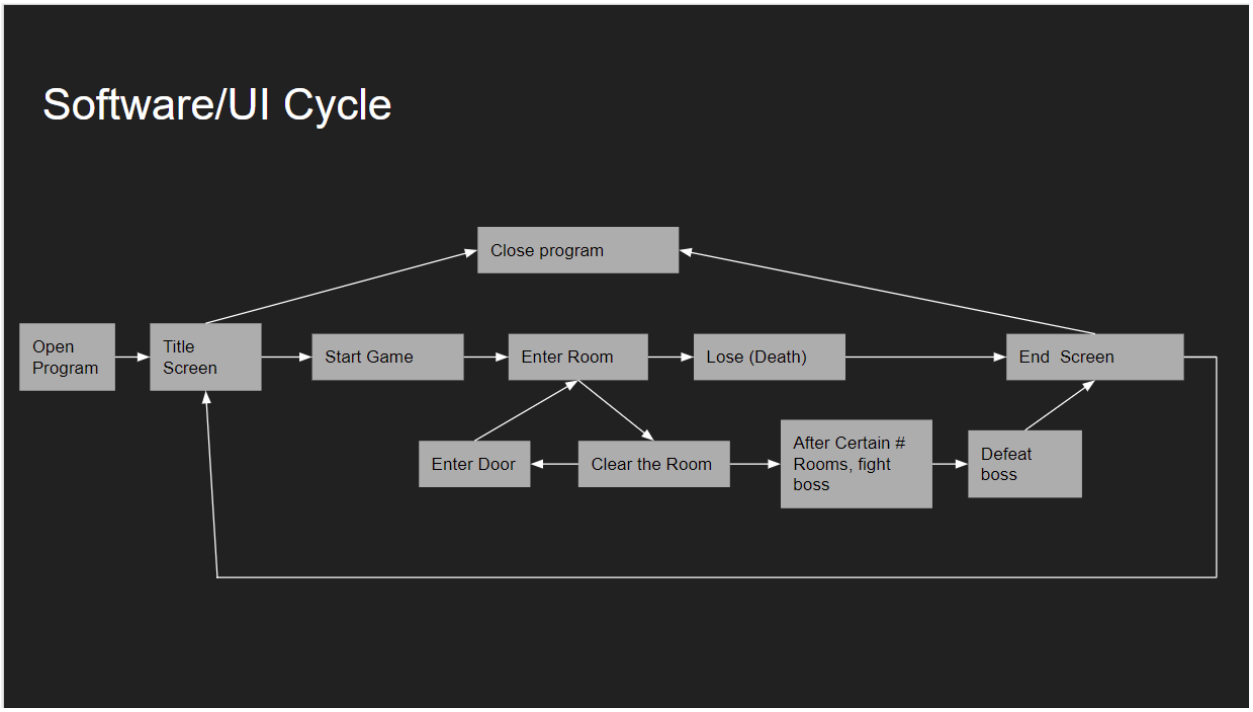
    - When the user presses the ESC key on their keyboard, the application shall pause the game and bring up the pause screen.
- Pause interface
    - When the user presses the ESC key on their keyboard while the pause screen is up, the application shall unpause the game and close the pause screen.
    - When the user clicks the title button, the application shall return to the title screen, which does not save the player's current progress in the game.
    - When the user clicks the options button, the application shall display the options menu.
    - When the user clicks the quit button, the application shall quit the game.
- Game-over Interface
    - When the user-controlled player reaches zero current heath points, the application shall display the game-over overlay.
    - When the user clicks the restart button, the application shall restart the current level.
    - When the user clicks the quit button, the application shall quit the game.
- Victory Interface
    - When the user presses the "G" key next to the teleporter, the application shall display the Victory overlay.
    - When the user clicks the proceed button, the application shall restart the current level.
    - When the user clicks the restart button, the application shall restart the current level.

- When the user clicks the quit button, the application shall quit the game.

3. Use-Case Diagram



4. Software/UI Cycle Graph



5. Sources

Dotween Library for very basic object animation/value tweening:

http://dotween.demigiant.com/documentation.php

ambientCG for PBR (physically based rendering) textures:

https://ambientcg.com/

A* Pathfinding Project Library for the turret section of the game:

https://arongranberg.com/astar/

UnityEditor to create the game:

https://docs.unity3d.com/Manual/index.html

Blender 3.0 to create the 3D models and animate them:

https://docs.blender.org/manual/en/latest/