Technical Design Document

|  |  |
| --- | --- |
| **Version** | **Description** |
| 1.0 | Initial Document Setup |
| 1.1 |  |

# Game Overview

## Game Summary

Simon is a game focused on pushing a player’s memory to his/her limits. The player must remember a series of lights and repeat the sequence exactly. If the player succeeds, the series of lights becomes slightly longer and more complex. The game ends when the player fails to select the correct sequence.

# Development Environment

## Development Team

* Richard Delamore: game design, programming and graphics.

## Programming Language and Graphical Framework

The game will be development using C++ and 2D Bootstrap library provided by AIE (Academy of Interactive Entertainment).

## IDE

## Source Control Procedures

Source control will be provided by GitHub.

## Third Party Libraries

2D Bootstrap.

cmath

## Project Management Tools

## Other Software

## Platform

PC only game.

Development Overview

# Timeline

Milestones

Alpha

Beta

Gold

Playtesting

# Game Overview

## Gameplay

The player must remember a series of lights and repeat the sequence exactly. If the player succeeds, the series of lights becomes slightly longer and more complex. The game ends when the player fails to select the correct sequence.

## Game Objects and Logic

**Controller:** This object dictates movement between the splash screens. It is also responsible for saving and opening the high score screen. The controller is responsible for instantiating **Player**.

**Player:** This object represents user input. It also controls the movement of the virtual hand. The Player instantiates **Simon**.

**Simon:** This object dictates the actions of **Simon**.

## Game Flow

**Start Screen:** Player is presented with two options. Start a new game or View High Scores.

**Game:** Player sees a three second count down. When the countdown hits zero the player, the game presents the first sequence to the user. The user then needs to enter the sequence they saw. If they got the sequence right, the high score count increases by one. If the player got the sequence wrong, the game ends and the player is presented with a screen asking if they want to enter their name into the high score table. If the player presses “Yes” the player needs to type their name in and then press “Enter” which will then lead them to the High Score screen. If the player presses “No” they will instantly be led to the high score screen.

**High Score Screen:** The high score screen presents a list of players sorted by highest score. The player can choose to return to the start screen.

Mechanics

Game Architecture

Core Data Structures

Classes

Algorithms

UI and HUD

Heads-Up Display

User-Interface