SE 3XA3: Test Plan
Title of Project

Team #, Team Name
Student 1 name and macid
Student 2 name and macid
Student 3 name and macid

October 31, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
Date 1	1.0	Notes
Date 2	1.1	Notes

This document ...

1 General Information

1.1 Purpose

1.2 Scope

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	**Definition**
Abbreviation1	Definition1
Abbreviation2	Definition2

Table 3: **Table of Definitions**

Term	**Definition**
Term1	Definition1
Term2	Definition2

1.4 Overview of Document

2 Plan

2.1 Software Description

2.2 Test Team

2.3 Automated Testing Approach

2.4 Testing Tools

2.5 Testing Schedule

See Gantt Chart at the following url ...

3 System Test Description

3.1 Tests for Functional Requirements

3.1.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

Output:

How test will be performed:

3.1.2 Area of Testing2

...

3.2 Tests for Nonfunctional Requirements

The different tests listed will demonstrate that the non-functional requirements in the software requirements specification are met.

3.2.1 Look and Feel Requirements

1. Game Environment

   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s): Development Team, Colleagues and/or testing group

   Description: The tester will see if the different environments are correct and meet the specifications

2. Game Hude/Interface

   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s): Development Team, Colleagues, and/or testing group

   Description: The tester will make sure that the score, time, lives, and amount of coins is not obstructive in the game?s view.

3.2.2 Usability and Humanity Requirements

1. Ease of Learning

   Type: Dynamic, Manual

Initial State: In-game state

Tester(s): Development Team, Colleagues, and/or testing group

Description: The tester will play through the game and will inform the development team of clarifications

2. Entertainment

   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s): Development Team, Colleagues, and/or testing group

   Description: The tester will make sure that the game is entertaining and follows similar principles to that of the original game

3.2.3 Performance Requirements

1. Controls/Commands

   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s): Development Team, Colleagues, and/or testing group

   Description: The tester will make sure that the game does not have any noticeable delays with controls, as well as any controls that seem odd/difficult to understand.

3.2.4 Operational and Environment Requirements

1. Operating System Support

   Type: Dynamic, Manual

   Initial State: Downloading/Installing

   Tester(s): Development Team, Colleagues, and/or testing group.

   Description: The tester will make sure that the game is able to run on Windows, MacOS, and Ubuntu.

3.2.5 Security Requirements

1. Altering Information
   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s): Development Team, Colleagues, and/or testing group

   Description: The tester will make sure that the game does not alter any files or processes that are not directly related to the game.

3.2.6 Cultural Requirements

1. Spelling and Grammar

   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s): Development Team, Colleagues, and/or testing group

   Description: The tester will make sure that the game has no spelling/grammar errors and that messages, menus, and overall interface is written in English.

2. Offensive Content
   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s): Development Team, Colleagues, and/or testing group

   Description: The tester will make sure that the game has no offensive content towards culture (religion, politics, ethnics, race, etc?).

3.2.7 Legal Requirements

1. License Adherence

   Type: Dynamic, Manual

   Tester(s): Development Team, and Colleagues

   Description: The tester will make sure that the game is not breaching the license that comes along with the game.

3.2.8 Health and Safety Requirements

1. Epileptic Prevention

   Type: Dynamic, Manual

   Initial State: In-game state

   Tester(s) ? Development Team, Colleagues, and/or testing group

   Description: The tester will make sure that the game does not trigger epileptic seizures as a result from playing.

4 Tests for Proof of Concept

A proof on concept test will be used to show that the development for Mari0 is feasible with the current skills and technology we have available to us. This section describes the proof of concept test and the details associated with it.

4.1 Demonstration Plan

For a proof of concept test we will create a small prototype that will be ran from Unity that can be used on Windows 10, MacOS, and Ubuntu. The prototype will be a small game demo demonstrating collision detection with different in game objects, the main gravity system, and the portal interactions. Many of these different game elements will be implemented using the Unity's collision and physics engines, as this will make our final goal easier to achieve. The main graphics that are used in the actual game will be used for this demo.

The prototype will be a floor that will be similar to the final game, which will be populated with the player character, six portals, two pipes, and platforms which the character can interact with. The player will be able to stand on the main floor and the platforms. There are also no walls to contain the character on either side.

The player (which will be represented by Mario) can be controlled in the following ways:

- The player moves left and right with the 'a' and 'd' keys respectively

- The player can jump by using the spacebar

The player will interact with the different objects in the following ways:

- The floor will be the main platform that the user will be able to stand on.
- The pipes will act like walls when approached from the side and not allow the user to pass through, and act like a floor when approached from the top.
- All 6 portals are paired in different ways, when entering a blue portal, the character will exit the orange portal, and vice versa.
- All physics will be maintained when entering through a portal, and portals can be on any surface that is at least two units.

Proof of Concept Test Many of the tests that are demonstrated in the proof of concept will be stated in the System Tests section of this document.

1. Proof of Concept

   Type: Manual

   Tester(s): Development Team and Colleagues

   Description: Tests whether significant risks to the completion of the project can be overcome.

5 Comparison to Existing Implementation

6 Unit Testing Plan

6.1 Unit testing of internal functions

6.2 Unit testing of output files

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

This is a section that would be appropriate for some teams.