Maze Game Testing Document

By Rory Gallie

# 1. Purpose

This document will outline the testing process. This will include a test plan and procedures.

With the purpose of detailing how the testing process has gone and if any bugs have been found.

# 2. Scope

The scope of this document is limited to an overview of the testing of a “maze game”, detailing test procedures, bugs found and test cases. The design will fall in line with the requirements given in the specification brief “*BCS Digital Industries Apprenticeship Software Development Technician Project B - Maze Game Version 1.2”* Create and execute a set of tests that demonstrate that the program meets its requirements.

• The tests may be manual or automated.

• The tests may be written before, after, or at the same time as the program code.

• For each test you should document its expected outcome and the actual result.

# 3. Goals and vision

The goal of this project is to have a working and tested maze game that is fully functional including some design given from the requirements brief.

The goal of this document is to outline the testing process of the “maze game” that is required for finding bugs within the application.

# 4. Test Plan

4.1 Objectives

The main objective of this test plan is to discover any issues or bugs within the newly created maze game.

This test plan supports the following objectives:

* To define the scope of what will be tested.
* Specify the approach taken to testing.
* What deliverables can be expected from test activities.

4.2 Scope

The scope of this test plan is the newly created maze game from unit testing perspective and manual testing within the User Interface.

Testing the Maze will be by the following means:

* System testing functionality.
* Testing the User Interface manually.

I would regard testing the user Interface tests as a type of use case for the system as well.

4.3 Testing strategy

This project follows a reactive testing strategy, meaning that test activities are started after design and coding are completed.

4.3.1 Unit testing

Unit testing will test the following within code:

* Test the Rooms total amount.
* Test the rooms have correct map dimensions from loaded file.
* Test the game regions are correct to the picture box.
* Test the player movement is working correctly.
* Test the players actions to attack or pickup coins works correctly
* Test the coin value is correct
* Test the files loaded from the levels (rooms) directory is correct.

4.3.2 User interface testing

The user interface testing will be to test the following during one run of the main application:

* Can the user move to a new room if a threat does not exist in current room.
* Can the user collect coins and kill threats
* Can the game be restarted at any point
* Can the user complete the game

# 5. Tests cases with results

5.1 Unit tests

|  |  |  |  |
| --- | --- | --- | --- |
| Test name | Test process | Expected results | Actual results |
| TestRoomsTotal | Run test in test explorer in visual studio | Test should check the rooms total amount of rooms | Test passed with correct values |
| TestRoomMapDimentions | Run test in test explorer in visual studio | Test should check the rooms map dimensions are the same as loaded map dimensions | Test passed with correct values |
| TestGameRegions | Run test in test explorer in visual studio | Test should check the game regions are correct to the Room map regions | Test passed with correct values |
| TestPlayerMove | Run test in test explorer in visual studio | Test should check that on a player move, location has changed | Test passed with correct values |
| TestPlayerAction | Run test in test explorer in visual studio | Test should check that on a player action, action is performed | Test passed with correct values |
| TestCoinValue | Run test in test explorer in visual studio | Test should check coin value exists from selected random seed. | Test passed with correct values |
| TestLoad | Run test in test explorer in visual studio | Test should check correct amount of files are loaded from folder. | Test passed with correct values |

5.2 Manual Tests

|  |  |  |  |
| --- | --- | --- | --- |
| Test name | Test Process | Expected results | Actual results |
| Can move new room | Start MazeGame.exe  Hit start on menu form  Use arrow keys to move to a passage | Player initially gets a popup window if threat exists if not can move to new room | Player does get popup and if threat is killed player can move to next room |
| Can kill threats an pickup coins | Start MazeGame.exe  Hit start on menu form  Use arrow keys to move to a threat  Press spacebar to kill threat | Threat popup appears saying the threat has been destroyed and how much life value has been lost. | Threat is destroyed from spacebar press and popup states that health has been lost, that health amount is also removed from health box. |
| Can be restarted | Start MazeGame.exe  Hit start on menu form  Click the button “start new game” | Button is clicked and new game is initiated | Button is clicked and game is reset back to base with all values being reset. |
| Can the user complete the game | Start MazeGame.exe  Hit start on menu form  Play the maze to room 7 | In room 7 if a threat exists player cannot use exit block however if threat is destroyed player can exit and complete the game | Player is able to complete the game with a popup displaying their current health and wealth. There is also an ok button to exit game. |

# 6.0 Bugs found and changes done.

A bug was found with the exit block code during manual tests, if a user was to destroy all threats in the room with an exit block any passage in the same location as an exit block on other maps would then become an exit block.

This was due to the code in GenerateGame.cs having

Rooms[7].NoThreats == true

in the if statement, this was then changed to

Rooms[CurrentRoom].Map == Rooms[7].Map && Rooms[7].NoThreats == true

To check if the user is also in the same room.

Another bug was found during the manual testing section that coin values where different to the displayed colour on screen and upon entering / exiting a room these would change every time allowing a player to always get a gold coin.

To solve this issue all information regarding coins was removed from the update function and added into the on load of a map function.

However this is still flawed and would need a change to some of the design to solve.