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

February 28, 2020

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 User Input

1. Test-1

   Type: Functional, Dynamic, Manual

   Initial State: The difficulty is set to normal as default

   Input: The user does not select different difficulty and starts the game

   Output: The snake moves in the normal default speed when the game starts and the items are generated with normal frequency

   How test will be performed: The function that adjust the speed and item generation will run. We will check if the snake covers correct amount of blocks in a fixed time period, and also the new in-game items are randomly generated on time.

2. Test-2

   Type: Functional, Dynamic, Manual

Initial State: The difficulty is set to normal as default

Input: The user selects easy difficulty in the title screen and clicks start button

Output: The snake moves in the slow speed when the game starts and the items are generated with frequency for easy mode

How test will be performed: The function that adjust the speed and item generation will run. We will check if the snake covers correct amount of blocks in a fixed time period, and also the new in-game items providing positive effects are randomly generated on time, as well as the trap items removed.

3. Test-3

   Type: Functional, Dynamic, Manual

   Initial State: The difficulty is set to normal as default

   Input: The user selects hard difficulty in the title screen and clicks start button

   Output: The snake moves in the fast speed when the game starts and the items are generated with frequency for hard mode

   How test will be performed: The function that adjust the speed and item generation will run. We will check if the snake covers correct amount of blocks in a fixed time period, and also the amount of in-game items providing positive effects are decreased, as well as the trap items increased.

4. Test-4

   Type: Functional, Dynamic, Manual

   Initial State: The map is set to default

   Input: The user does not select any map in the title screen and clicks start button

   Output: The game will start after loading the default map with no barrier.

   How test will be performed: The function that loads the map will run. We will check if the default map is loaded when the game starts.

5. Test-5

   Type: Functional, Dynamic, Manual

   Initial State: The map is set to default

   Input: The user select a map other than the default one in the title screen and clicks start button

   Output: The game will start after loading the selected map with corresponding terrain and barriers

   How test will be performed: The function that loads the map will run. We will check if the correct terrain and barriers are loaded when the game starts

6. Test-6

   Type: Functional, Dynamic, Manual

   Initial State: Main screen waiting for start

   Input: The user clicks start button

   Output: The gaming screen will be shown and a snake with length 2 and default direction to right is generated on random location in the map

   How test will be performed: The function that loads the new snake will run. We will check if the snake with correct length and direction are generated when the game starts

7. Test-7

   Type: Functional, Dynamic, Manual

   Initial State: Start state of gaming screen

   Input: The user clicks space key on keyboard

   Output: The snake will start moving to the default direction

   How test will be performed: The function that moves the snake will run. We will check if the snake starts moving with correct direction and speed when space is pressed

8. Test-8

   Type: Functional, Dynamic, Manual

   Initial State: Start state of gaming screen

   Input: The user clicks space key on keyboard

   Output: The starting sound effect will be played

   How test will be performed: The function that plays sound effect will run. We will check if the sound file is output to the audio device successfully

9. Test-9

   Type: Functional, Dynamic, Manual

   Initial State: Start state of gaming screen

   Input: The user clicks space key on keyboard

   Output: A food item will be generated on the random empty location inside the map

   How test will be performed: The function that generates food item will run. We will check if the food item is shown on the random location in the valid area inside the map.

10. Test-10

    Type: Functional, Dynamic, Manual

    Initial State: Gaming state of gaming screen

    Input: The user clicks p key on keyboard

    Output: The game goes to pause state and the pause screen is shown

    How test will be performed: The function that pauses the game will run. We will check if the game is paused and the pause screen is shown correctly

11. Test-11

Type: Functional, Dynamic, Manual

Initial State: Pause state of gaming screen

Input: The user clicks p key on keyboard

Output: The game goes to gaming state again and the pause screen is hided

How test will be performed: The function that pauses the game will run. We will check if the game is running again and the pause screen is hided successfully

12. Test-12

    Type: Functional, Dynamic, Manual

    Initial State: Gaming state of gaming screen

    Input: The user clicks any direction key on keyboard

    Output: The direction of the snake is fixed to the corresponding direction of pressed key

    How test will be performed: The function that fix the direction will run. We will check if the direction of the snake is changed correctly

13. Test-13

    Type: Functional, Dynamic, Manual

    Initial State: The snake's head is at an empty location

    Input: The snake's head reaches a food item

    Output: The length of the snake is incremented by 1 at the tail

    How test will be performed: The function that grows the snake will run. We will check if the length of the snake is updated correctly when consuming the food

14. Test-14

    Type: Functional, Dynamic, Manual

    Initial State: The snake's head is at an empty location

Input: The snake's head reaches a food item

Output: The player's score will be incremented

How test will be performed: The function that increases the score will run. We will check if the score is updated when consuming the food

15. Test-15

    Type: Functional, Dynamic, Manual

    Initial State: The snake's head is at an empty location

    Input: The snake's head reaches a food item

    Output: The sound effect of scoring will be played

    How test will be performed: The function that plays the sound effect will run. We will check if the sound file is output to the audio device successfully

16. Test-16

    Type: Functional, Dynamic, Manual

    Initial State: The snake's head is at an empty location

    Input: The snake's head reaches a food item

    Output: The food item is removed and a new food item is generated on random empty location

    How test will be performed: The function that generates the food item will run. We will check if the new food item is generated on valid location

17. Test-17

    Type: Functional, Dynamic, Manual

    Initial State: Gaming state of gaming screen

    Input: Fixed time period passes

    Output: The effect item is generated on random empty location

How test will be performed: The function that generates the effect item will run. We will check if the new effect item is generated on valid location

18. Test-18

    Type: Functional, Dynamic, Manual

    Initial State: Gaming state of gaming screen

    Input: The snake's head reaches an effect item

    Output: The effect item is applied to the snake

    How test will be performed: The function that enables effect will run. We will check if the status of the snake is updated corresponding to the item consumed

19. Test-19

    Type: Functional, Dynamic, Manual

    Initial State: Gaming state of gaming screen

    Input: The snake's head reaches the border or barrier or snake body

    Output: The game over screen is shown

    How test will be performed: The function that shows game over screen will run. We will check if the game is stopped and the game over screen is shown correctly

20. Test-20

    Type: Functional, Dynamic, Manual

    Initial State: Game over screen

    Input: The player clicks restart button

    Output: The starting screen of a new game is shown

    How test will be performed: The function that restarts a game will run. We will check if the game is refreshed and a new game is initialized

21. Test-21

    Type: Functional, Dynamic, Manual

    Initial State: Game over screen

    Input: The player clicks back to title button

    Output: The title screen will be shown

    How test will be performed: The function that shows title screen will run. We will check if the game is stopped and the title screen is shown

22. Test-22

    Type: Functional, Dynamic, Manual

    Initial State: Title screen

    Input: The player clicks enable color blind option button

    Output: The color blind option is enabled

    How test will be performed: The function that enables color blind option will run. We will check if the colors used in the game are adjusted to visible to color-blind people

3.2 Tests for Nonfunctional Requirements

3.2.1 Usability

1. Test-23

    Type: Structural, Static, Manual

    Initial State: File explorer or command line terminal

    Input: Java command execution or jar file

    Output: The program is opened with Java virtual machine

    How test will be performed: The function that opens the program will be run using command or file on any Java virtual machine. We will check if the program can be successfully opened on Java environment in Linux and Windows system, and the main screen is completely shown.

3.2.2 Performance Requirements

1. Test-24

   Type: Structural, Dynamic, Manual

   Initial State: The program is launched and ready to be played

   Input: The player performs operations on the program and ends the game

   Output: The operations get respond from the program under processing time

   How test will be performed: A log of operations will be generated with a module in the program. System clock are used to calculate for the time period needed between receiving the user input and the response from the program.

3.3 Traceability Between Test Cases and Requirements

Traceability Between Test Cases and Requirements

Requirements	**Test cases**
FR3	Test 1-Test 3
FR4	Test 4-Test 5
FR6	Test 6
FR7	Test 7-Test 8
FR8	Test 9
FR9	Test 10
FR10	Test 11
FR12	Test 13-15
FR13	Test 16
FR15	Test 17
FR16	Test 18
FR17	Test 19
FR18	Test 20-21
FR19	Test 22
LF1	Test 23
PR1	Test 24

4 Tests for Proof of Concept

4.1 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:

4.2 Area of Testing2

...

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.