Java Game Suite Test Plan & IWD

Revision 8

CMSC 495 6382

August 30, 2021

Group Charlie

Sherry Funches, Oyewole Sanusi, Janee’ Jones, Wayne Mack, Jeffrey McGlinn

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Date** | **Description** | **Name** |
| 1 | 8/30 | Creation of Doc and Outline | Oyewole |
| 2 | 9/4 | Adding Context Diagram and Citations | Janee’ |
| 3 | 9/4 | Add test plan for Snakes | Oyewole |
| 4 | 9/5 | Add test plan for Slider Puzzle and minor edits | Jeff |
| 5 | 9/5 | Add test plan for word search | Sherry |
| 6 | 9/5 | Add test plan for Sudoku | Janee’ |
| 7 | 9/5 | Add test plan for Maze | Wayne |
| 8 | 9/5 | Enter and format all test plans into unified document | Sherry |

**I. Test Plan**

A. Main Menu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Requirement Number(s)** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 |  | Startup of application | GUI in ready state with no unexpected hardware issues or system failure. |  |  |
|  |  |  |  |  |  |
| 2 |  | User initiated termination of application | Program terminated and all items are deconstructed. |  |  |
|  |  |  |  |  |  |
| 3 |  | Testing of initial Map functionality in Map Game | Map can be generated using methods and player can move. | Map can be generated using methods and character moves with functional collision detection |  |
|  |  |  |  |  |  |

B. Maze

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Requirement Number(s)** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 |  | Startup of Program | Program Starts from main menu | Program executes from main menu | Pass |
|  |  |  |  |  |  |
| 2 |  | Program Controls with keyboard arrows | Game responds with keyboard arrows |  | Pass |
|  |  |  |  |  |  |
| 3 |  | Alternate Control Scheme | Program Controls with W,A,S,D Characters | Game Responds with W,A,S,D characters | Pass |
|  |  |  |  |  |  |
| 4 |  | Enemy AI | Enemy responds to AI |  | Pass |
|  |  |  |  |  |  |
| 5 |  | Screen Refresh | Screen Refreshes seamlessly | Screen is sluggish | Pass |

C. Snake

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Requirement Number(s)** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 |  | Startup of application | GUI in ready state with no unexpected hardware issues or system failure. | Game starts immediately | Pass |
|  |  |  |  |  |  |
| 2 |  | User initiated termination of application | Program terminated and all items are deconstructed. | The APP ends by closing window | Pass |
|  |  |  |  |  |  |
| 3 |  | User maneuvers the snake using keys on their keyboard (←, ↑, →, ↓). | Snake moves up using the ↑ arrow, down using the ↓, left using the ←, and right using the → arrow on the keyboard. | Snake moves up, down, left, and right using the following keys on the keyboard. ((←, ↑, →, ↓) | Pass |
|  |  |  |  |  |  |
| 4 |  | Apple is consumed and the snake length increase by one dot. | Apple is consumed and the snake length increase by one dot. | Apple is consumed and the snake length increase by one dot and new apple is generated in a different location. | Pass |
|  |  |  |  |  |  |
| 5 |  | Troubleshooting: triggering a “Game Over” with snake consuming itself | “Game Over” is triggered when snake consumes itself | “Game Over” appears and the game will need to be relaunched to play again. | Pass |

D. Sudoku

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Requirement Number(s)** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 |  | Startup of application | GUI in ready state with no unexpected hardware issues or system failure. | Game begins | Pass |
|  |  |  |  |  |  |
| 2 |  | User initiated termination of application | Program terminated and all items are deconstructed. | Game ended by clicking exit or by closing the window | Pass |
|  |  |  |  |  |  |
| 3 |  | User starts the game by choosing a number from the list and input where appropriate | Number user chooses is inputted into the appropriate box | Number user chooses is inputted into the box of user choice | Pass |
|  |  |  |  |  |  |
| 4 |  | User enters number into box of choice where number belongs | Blue hint box should disappear once user enters number | User sees blue hint box disappear where number was entered | Pass |
|  |  |  |  |  |  |
| 5 |  | User can check progress by clicking the check button | User clicks check button, and it will display either green for correct or red for incorrect | User is displayed either green or red for the box where number was entered | Pass |
|  |  |  |  |  |  |
| 6 |  | User completes game by entering in all the numbers in the blank boxes | User clicks check button, and it will display all numbers in green that are correct and all numbers in red that are incorrect | All green numbers user has won game, or mixture of numbers user has not won game and must go back to correct red numbers | Pass |

E. Word Search

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Requirement Number(s)** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 |  | Start up of application | Display GUI. |  |  |
|  |  |  |  |  |  |
| 2 |  | Drop down list “Select Puzzle” contains list of all puzzles. | Exact list of puzzles to be determined. |  |  |
|  |  |  |  |  |  |
| 3 |  | Drop down list has default puzzle selected so that null values cannot be loaded. | Default puzzle is selected. |  |  |
|  |  |  |  |  |  |
| 4 |  | Program generates and displays each puzzle when “Load Puzzle” button is clicked. | TO DO: Include image of expected result |  |  |
|  |  |  |  |  |  |
| 5 |  | All words hidden in the puzzle displayed on GUI’s word list panel | TO DO: Include list of words from Farm 1 |  |  |
|  |  |  |  |  |  |
| 6 |  | All words in the list are in the correct position on the search grid | TO DO: Include image of expected result |  |  |
|  |  |  |  |  |  |
| 7 |  | When the user clicks on a word in the list, the word is crossed out. | Line appears through word when clicked |  |  |
|  |  |  |  |  |  |
| 8 |  | When the user clicks on a crossed out word on the list the line is removed | Line is removed from word when clicked |  |  |
|  |  |  |  |  |  |
| 9 |  | The user can drag their mouse over letters in the grid to highlight a word | Letters are highlighted when mouse dragged over them. |  |  |
|  |  |  |  |  |  |
| 10 |  | Clicking “Submit” button produces a message regarding the number of correctly identified words | Dialog box appears with message “X words out of N words correctly identified.” (TO DO fill in x and n) |  |  |
|  |  |  |  |  |  |
| 11 |  | Clicking “Submit” causes all unfound words to appear in red font on the grid. | Puzzle words that are not highlighted when “Submit” clicked turn red. |  |  |
|  |  |  |  |  |  |
| 12 |  | Program rejects highlighted words when the correct letters plus additional letters are highlighted | To Do |  |  |
|  |  |  |  |  |  |
| 13 |  | Program rejects words when all the letters are not highlighted | To Do |  |  |
|  |  |  |  |  |  |
| 14 |  | Program rejects words when the highlight does not pass through the center of each letter in a continuous stroke | TO DO: |  |  |
|  |  |  |  |  |  |
| 15 |  | Clicking the “Clear” button removes all highlights from the grid and cross lines from words in the word list. | All highlights removed from the grid when “Clear” button clicked. |  |  |
|  |  |  |  |  |  |
| 16 |  | Clicking the “Undo” button removes the last highlight from the grid. | The last highlight is removed from board when “Undo” clicked. |  |  |
|  |  |  |  |  |  |
| 17 |  | Exiting the window returns to main menu | The main menu screen appears when window exited. |  |  |

F. Slider Puzzle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Requirement Number(s)** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 |  | Startup of application | Display GUI. | GUI is displayed without errors. | Pass |
|  |  |  |  |  |  |
| 2 |  | Select a new puzzle. | Display a new puzzle. | New puzzle displayed without errors. | Pass |
|  |  |  |  |  |  |
| 3 |  | Select the image on the puzzle that has the power to swap with adjacent tile until the puzzle is complete. | Once selected, the image will shift to the adjacent tile. Repeat until puzzle is complete. | Image shifts to the adjacent tile until the puzzle is complete. | Pass |
|  |  |  |  |  |  |
| 4 |  | Display a message once the user completes the puzzle. | Message displayed once the user has completed the puzzle. | The message is displayed successfully. | Pass |
|  |  |  |  |  |  |
| 5 |  | Select a “puzzle hint” to display the puzzle’s final image. | Once a puzzle hint is selected, display the final image of the puzzle. | The final image of the puzzle is displayed. | Pass |
|  |  |  |  |  |  |
| 6 |  | Select a new puzzle. | Display a new puzzle for the user to complete. | A new puzzle is successfully displayed. | Pass |
|  |  |  |  |  |  |
| 7 |  | Exit slider puzzle window to return to main menu. | Once exited, the user is returned to the main menu. | The user is successfully returned to the main menu. | Pass |

**II. Inner Working Details (IWD)**

**Between User and GameSuiteGUI**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **From** | **To** | **Name** | **Type** | **Value Range** | **Default Value** |
| User | GameSuiteGUI |  | String/ float? |  |  |
| User | GameSuiteGUI |  | String/ float? |  |  |
| User | GameSuiteGUI |  | Float/ String? |  |  |
| GameSuiteGUI | User |  | Float/ String? |  |  |

**Between GameSuiteGUI and Maze:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **From** | **To** | **Name** | **Type** | **Value Range** | **Default Value** |
| GameSuiteGUI | Maze |  | String/ float? |  |  |
| GameSuiteGUI | Maze |  | String/ float? |  |  |
| GameSuiteGUI | Maze |  | Float/ String? |  |  |
| Maze | GameSuiteGUI |  | Float/ String? |  |  |

**Between GameSuiteGUI and Sudoku:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **From** | **To** | **Name** | **Type** | **Value Range** | **Default Value** |
| GameSuiteGUI | Sudoku |  | String/ float? |  |  |
| GameSuiteGUI | Sudoku |  | String/ float? |  |  |
| GameSuiteGUI | Sudoku |  | String/ float? |  |  |
| Sudoku | GameSuiteGUI |  | Float/ String? |  |  |

**III. Context Diagram**

Diagram

Description automatically generated

**IV. Citations**

*Context diagrams*. Cs.uct.ac.za. (2021). Retrieved 5 September 2021, from https://www.cs.uct.ac.za/mit\_notes/software/htmls/ch06s06.html.