**Test Scenario Group: Main Menu**

**Test Case Description: Display Main Menu**

**Prerequisite:** N.A.

**Post-requisite:** N.A.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Passing/Failing Test Case** | **Action(s)** | **Input** | **Expected Output** | **Actual Output** | **Test Result** |
| 01 | Passing | Display Main Menu | NIL | System display main menu |  |  |
| 02 | Passing | User input option 1 | Select option: 1 | System display Read Maze page |  |  |
|  | Passing | User input option 2 | Select option: 2 | System display View Maze page |  |  |
|  | Passing | User input option 3 | Select option: 3 | System display Play Maze page |  |  |
|  | Passing | User input option 4 | Select option: 4 | System display configure maze page |  |  |
|  | Failing | Input other than numbers(Eg: Alphabet, symbols) | Select option: A | Error message: “Wrong input entered. Only numbers are allowed” |  |  |
|  | Failing | Input value not shown in main menu | Select option: 69 | Error message: “Wrong input entered. Enter the values from the menu” |  |  |
|  | Failing | Input blank value | Select option: | Error message: “No input is entered. Please input a valid option” |  |  |
| 03 | Passing | Exit the program | Select option: 0 | System exits program |  |  |

Analysis: This is one of the high priority features in the entire program as it allows users to select different choices of the maze program, which allows users to choose, play and edit different maze files based on their preference. This feature is tested first as it acts as a base which connects to different parts of the program. For the first run, the program did not request for the user input, thus causing it to crash. This is due to the QA receiving new “batch” of codes.

**Test Scenario Group: Read & Load maze**

**Test Description: User enter the file name and display the number of lines**

**Prerequisite:** N.A.

**Post-requisite:** N.A.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Passing/Failing case** | **Test Description** | **Input** | **Expected outcome** | **Actual Outcome** | **Test Result** |
| 01 | Passing | User input the filename | Enter filename of maze: maze.csv and enter | System display successful message | System display successful message | Passed |
| 02 | Failing | User enters wrong filename | Enter filename of maze: msze.csv | System will display error message | System display error message: “File not found | Passed |
| 03 | Failing | User inserts wrong file extension | Enter filename of maze: maze.docx | System display error message: “File not found” | System display error message: “File not found” | Passed |
| 04 | Failing | User does not enter filename | Enter filename of maze: | System will display an error message “File not selected” | This scenario did not pass as it was not implemented. | Failed |

Analysis: In this feature will allow the user to input the maze file to allow the user to perform the program. During the manual tests, no errors occurred, thus automated tests proceeded. The first scenario was to input the correct maze file which shows a successful message, where the second and third scenarios were meant to fail due to wrong file extensions and filename. The automated test returns a result of 3 passes with 1 warning, where the warning involves the reading of the csv file. It did not affect the testing, thus, the test is classified as a pass.

**Test Scenario Group: View Maze**

**Test Case Description: View Maze from selected file**

**Prerequisite:** File must be loaded before viewing

**Post-requisite:** User is able to view the selected maze

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Passing/Failing Test Case** | **Action(s)** | **Input** | **Expected Output** | **Actual Output** | **Test Result** |
| 01 | Passing | User views maze with file selected | Filename: maze.csv | User is able to view maze from selected file |  |  |
| 02 |  | User goes back to main menu | NIL |  |  |  |
| 03 | Failing | User views maze with wrong file type | Filename: Notmaze.docx | System display error message: “Incorrect file type inserted. Please insert correct file type.” |  |  |
| 04 | Failing | User views maze without maze inside | Filename: mazedraft.csv | System display error message: No maze found in file. Please load the correct maze file. |  |  |

Analysis: For the first run, there was an invalid syntax error during the manual test, which was faced again in the automated test. This shows that when a manual test fails, it is highly likely that automated test will fail too. This is because the system does a deeper verification of the test using pytest whereas the QA itself only sees the output of the program(ie: no error if output shows blank during manual test). Thus, the program was sent back to the developer to make the necessary changes

The program improved in the second attempt, without showing errors when the manual test was conducted. However, a syntax problem caused the automated test to fail when the user is trying to view the selected file.

**Test Scenario Group: Play Maze**

**Test Case Description: Play Maze from selected file**

**Prerequisite:** File must be read before playing

**Post-requisite:** Selected maze is shown for user to play

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Passing/Failing Test Case** | **Action(s)** | **Input** | **Expected Output** | **Actual Output** | **Test Result** |
| 01 | Passing | User proceeds to play maze with file loaded | Filename: maze.csv | User is able to see and play selected maze |  |  |
| 02 | Passing | User presses alphabetical keys(eg: ‘W’, ‘A’, ‘S’, ‘D’) to move | Key: ‘W’ | Character moves up. |  |  |
|  | Passing | User presses alphabetical keys(eg: ‘W’, ‘A’, ‘S’, ‘D’) to move | Key: ‘A’ | Character moves left. |  |  |
|  | Passing | User presses alphabetical keys(eg: ‘W’, ‘A’, ‘S’, ‘D’) to move | Key: ‘S’ | Character moves down. |  |  |
|  | Passing | User presses alphabetical keys(eg: ‘W’, ‘A’, ‘S’, ‘D’) to move | Key: ‘D’ | Character moves right. |  |  |
| 03 | Failing | User presses any other alphabetical keys, that is not ‘W’, ‘A’, ‘S’ or ‘D’. | Key: ‘Q’ | System displays error message: “Invalid key pressed. Use the movement keys to move your character” |  |  |
| 04 | Passing | User’s character hits a wall | Character hits ‘X’ | System displays error message: “Invalid movement. Please try again” |  |  |
| 05 | Passing | User reaches the End | Character reaches ‘B’ | System displays message: “Congratulations. You have solved the maze”. |  |  |
| 06 | Failing | User did not load file. | Filename: | System display error message: “File not selected. Please load a maze file” |  |  |
| 07 | Failing | User did not load correct filetype. | Filename: “Maze.docx” | System display error message: “Incorrect file type selected. Please load the correct file type.” |  |  |
| 08 |  | User exits the “Play maze” page. | NIL | System shows main menu. |  |  |

**Test Scenario Group: Configure Maze**

**Test Case Description: Configure Maze from selected file**

**Prerequisite:** File must be loaded before configuration

**Post-requisite:** Selected maze can be configured

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Passing/Failing Test Case** | **Action(s)** | **Input** | **Expected Output** | **Actual Output** | **Test Result** |
| 01 | Passing | User proceeds to configure maze with file loaded | Filename: maze.csv | System display menu for maze configuration with selected maze file. |  |  |
| 02 | Passing | User creates wall | Enter option: 1 | System display ‘X’ to allow user to place wall. |  |  |
|  | Passing | User moves wall with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘W’ | Wall moves up |  |  |
|  | Passing | User moves wall with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘A’ | Wall moves left. |  |  |
|  | Passing | User moves wall with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘S’ | Wall moves down. |  |  |
|  | Passing | User moves wall with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘D’ | Wall moves right. |  |  |
|  | Failing | User moves wall with invalid key | Key: ‘Q’ | System displays error message: “Invalid key pressed. Use the movement keys to move the wall” |  |  |
|  | Passing | User places wall | User press enter | System display message: “Wall has been placed” |  |  |
| 03 | Passing | User create passageway | Enter option: 2 | System display ‘O’ to allow user to place passageway. |  |  |
|  | Passing | User moves passageway with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘W’ | Passageway moves up |  |  |
|  |  | User moves passageway with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘A’ | Passageway moves left. |  |  |
|  |  | User moves passageway with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘S’ | Passageway moves down. |  |  |
|  |  | User moves passageway with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘D’ | Passageway moves right. |  |  |
|  | Failing | User moves passageway with invalid key | Key: ‘Q’ | System displays error message: “Invalid key pressed. Use the movement keys to move the passageway” |  |  |
|  | Passing | User places passageway | User press enter | System display message: “Passageway has been placed” |  |  |
| 04 | Passing | User create start point | Enter option: 3 | System display ‘A’ to allow user to place start point. |  |  |
|  | Passing | User moves start point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘W’ | Start point moves up |  |  |
|  | Passing | User moves start point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘A’ | Start point moves left. |  |  |
|  | Passing | User moves start point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘S’ | Start point moves down. |  |  |
|  | Passing | User moves start point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘D’ | Start point moves right. |  |  |
|  | Failing | User moves start point with invalid key | Key: ‘Q’ | System displays error message: “Invalid key pressed. Use the movement keys to move the start point” |  |  |
|  | Passing | User places start point | User press enter | System display message: “Start point has been placed” |  |  |
| 05 | Passing | User create end point | Enter option: 4 | System display ‘B’ to allow user to place endpoint. |  |  |
|  | Passing | User moves end point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘W’ | End point moves up |  |  |
|  | Passing | User moves end point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘A’ | End point moves left. |  |  |
|  | Passing | User moves end point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘S’ | End point moves down. |  |  |
|  | Passing | User moves end point with valid key (eg: ‘W’, ‘A’, ‘S’, ‘D’) | Key: ‘D’ | End point moves right. |  |  |
|  | Failing | User moves end point with invalid key | Key: ‘Q’ | System displays error message: “Invalid key pressed. Use the movement keys to move the end point” |  |  |
|  | Passing | User places end point | User press enter | System display message: “End point has been placed” |  |  |
| 06 | Passing | User exits configuration page with both start & end point placed. | Enter option: ‘0’ | System returns back to main menu with message: “Maze is saved” |  |  |
|  | Failing | User exits configuration page with start point placed without end point | Enter option: ‘0’ | System shows error message: “End point is not found. Please place an end point” |  |  |
|  | Failing | User exits configuration page with end point placed without start point | Enter option: ‘0’ | System shows error message: “Start point is not found. Please place an start point” |  |  |
|  | Failing | User exits configuration page without both end and start point | Enter option: ‘0’ | System shows error message: “Start and end point is not found. Please place start and end point” |  |  |