**Tested before code development**

*Menu Unit Test Case*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Test Case Description** | **Value** | **Expected outcome** | **Actual Outcome** |
| 1 | Display menu invalid | To test if the python file would return an error if it fails to load the menu | Launch mazegame.py | Returns “invalid menu” |  |
| 2 | Display menu | Running the python file should display the main menu header, 5 options and prompt the user for his option | Launch mazegame.py | The python script displays 5 options as well as prompt the user for an option. |  |
| 3 | Invalid option | Selecting an option not available in the program. | value: 5 | Returns an error message: invalid option |  |
| 4 | Invalid option alphabets | To test if inputting alphabetical characters will return an error message. | value: abc | Returns an error message: invalid option |  |
| 5 | Invalid option symbols | To test if inputting symbols will return an error message. | value: @ | Returns an error message: invalid option |  |
| 6 | Invalid option with 2 or more numeric inputs | To test if inputting more than 1 character will return an error message | value: 92 | Returns an error message: invalid option |  |
| 7 | Option 1 | To test if inputting 1 will direct the user to the first option | value: 1 | Directs user to the first option |  |
| 8 | Option 2 | To test if inputting 2 will direct the user to the second option | value: 2 | Directs user to the second option |  |
| 9 | Option 3 | To test if inputting 3 will direct the user to the third option | value: 3 | Directs user to the third option |  |
| 10 | Option 4 | To test if inputting 4 will direct the user to the fourth option | value: 4 | Directs user to the fourth option |  |
| 11 | Option 0 | To test if inputting 0 will stop the python script | value: 0 | Returns a message that states |  |

*Option 1 (Read and Load maze from file) Unit Test Case :* ***Marcus Goh***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Test Case Description** | **Value** | **Expected outcome** | **Actual Outcome** |
| 1.1 | Test\_filenameinput | This is to test if the maze diagram can be read with the right file name input | maze.csv | The code will then run output the number of lines being read and load the maze.csv file. |  |
| 1.2 | test\_filenmae\_case\_sensitive | This is to test if the maze diagram can be read with the right file name input in capital letters | MAZE.CSV | The code will then run output the number of lines being read and load the maze.csv file. |  |
| 2.1 | Test\_filename\_invalid | This is to test if the maze diagram will not run if the filename the incorrect filename has been entered | Mazz.cev  Maz123  aofjb | The application will not be able to read and load any file and shows and error message. |  |

*Option 2 (View maze) Unit Test Case*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Test Case Description** | **Value** | **Expected outcome** | **Actual Outcome** |
| 1 | Unable to display list | To test if it will return an error if there is a list saved but it is unable to display | List stored | Returns an error message that |  |
| 2 | No list saved | To test if it will return an error message if there are no list saved | No list stored | Returns a message that states “No maze stored” |  |
| 3 | Loads the saved list | To test if the list will correctly print the saved list | List stored | Prints the maze |  |

*Option 3 (Play maze game) Unit Test Case*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Test Case Description** | **Value** | **Expected outcome** | **Actual Outcome** |
| 1 | Error input test | Entering input that is not requested | Test value1: “1”  Test value2: “z”  Test value3: “@”  Test value4: “12as%^” | Output 1: “Invalid input”  Output 2: “Invalid input”  Output 3: “Invalid input”  Output 4: “Invalid input” |  |
| 2 | Test upward movement | To test if the position can move accordingly | Test value: “W” | Output: “Successfully move upwards” |  |
| 3 | Test downward movement | To test if the position can move accordingly | Test value: “S” | Output: “Successfully move downwards” |  |
| 4 | Test left movement | To test if the position can move accordingly | Test value: “A” | Output: “Successfully move left” |  |
| 5 | Test right movement | To test if the position can move accordingly | Test value: “D” | Output: “Successfully move right” |  |
| 6 | Test wall blockage | To Test if the wall is able to restrict movement | Test value: “A” Change accordingly to start location | Output: “Invalid movement” |  |
| 7 | Test exit to menu | To test if user can return to menu page after playing | Test value: “M” | Output: Menu page |  |
| 8 | Test wall existing | To test if wall exist in moving direction | Test value: “X” | Output: “Wall” |  |
| 9 | Test free space | To test if wall exist in moving direction | Test value: “O” | Output: “Space” |  |
| 10 | Test finding current location | To test if it can locate current location | Test value: [X,O,O,A,O,O,X] | Output: Row1, Column3 |  |
| 11 | Test case sensitive | To test if correct input is case sensitive | Test Value: “a” | Output: “Moved upwards” |  |

*Option 4 (Configure current maze) Unit Test Case*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Test Case Description** | **Value** | **Expected outcome** | **Actual Outcome** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

*Option 0 (Exit Maze) Unit Test Case*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Test Case Name** | **Test Case Description** | **Value** | **Expected outcome** | **Actual Outcome** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

Screenshot of the pytest result in terminal: