**Iterative Testing**

Throughout the process of constructing the technical solution and the documented design, it was important for me to record errors when I was running the programs. Because the Maze Solving code was completed first and the two game modes are very similar in terms of actual coding, a lot of the issues found in the maze solving were not found in the maze creation code.

|  |  |  |
| --- | --- | --- |
| *Bug* | *Solution* | *Comments* |
| Found a bug where the user could not solve the maze even when on the red square | The coordinates changed into integers, the problem was that the coordinates were in decimals not integers, so a conversion was necessary. | This was also applied in the Maze Creation to prevent the same bug from occurring |
| Found a bug when restarting the game and choosing an option runs the maze but places the player in the last known location (at the bottom of the map) | The coordinates of the player had to be reset every-time the user runs the program | Also applied into Maze Creation to reset the start and end points |
| Found a bug where the ESC and R in the win Screen both return the user back to the solving menu | A while loop was added to the functions instead of when the function is called which meant the solving menu kept running again regardless of the key pressed | Also applied this to all functions to prevent the same bug from occurring. |
| Found a bug where pressing mouse down right when outside of the playable area causes the application to crash | Used a conditional statement to prevent the program from using the mouse coordinates outside of the playable area | - |
| Found a bug where the program runs scripts as a module not a python file. | Placed all the code into one large file instead. | This works better when all in one program, so this was a very beneficial bug |
| Found a bug where the pathfinding algorithm searched every node and recorded this as the path | Used an exit statement to crash the function at that point returning the path at that point. | To find the maximum number of nodes in the path |
| Found a bug where the program temporarily crashed when loading hard difficulty. | The current hardware I was using was not capable of computing these calculations quick enough causing a short buffer of the program. | I tried the same program on a better computer and the maze game runs much smoother and faster |
| Found a bug where the player could bypass the creation section | The player needs to place a minimum of 2 blocks before they can progress | - |
| Found a bug where the player could make a maze with no solution | The player can go to the creation menu if the user controls the solving, or the computer will automatically send the user back to the creation menu if the computer is chosen to solve. | - |

**Post-Development Testing**

The following testing uses a video (<https://youtu.be/wop84BuobI8>) which is a walkthrough of the game. The video is public on youtube.com which can be seen by anyone by using the link above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Requirements* | *Achieved?* | *Evidence (Screenshots)* | *Video Reference (minute: seconds)* | *Description* |
| The maze game contains two modes | ✔️ | A picture containing graphical user interface  Description automatically generated | Solve: 0.06  Create: 2.44 | The menu screen contains 2 playable options for the user. |
| The maze game uses a GUI (graphical user interface) for a human computer interaction | ✔️ | A picture containing timeline  Description automatically generated | 0.00 -> 8.00 | The game uses pygame’s inbuilt graphics to create the GUI. |
| The maze game uses menus to navigate around the game | ✔️ | Graphical user interface, text, application  Description automatically generated | Various Points between 0.00 🡪 8.00 | The reference shows one of the 3 menus in the game that the user can navigate to. |
| The menu (non-playable) screens don’t contain any interactive buttons. | ✔️ | A picture containing text, clipart  Description automatically generated | Various Points between 0.00 🡪 8.00 | The program accepts keyboard inputs to navigate menus instead of buttons. This is also seen at any menu screen (solve, create, main) |
| The game will use a default colour theme. | ✔️ | Graphical user interface, text  Description automatically generated with medium confidence | 0.00 🡪 8.00 | The program uses a black background with white text throughout, with win screens with green font. Games have white walls and blue and red icons means start and end points, respectively. |
| The Maze generation algorithm is dependent on the user input. | ✔️ | Text  Description automatically generated | Easy: 0.09  Medium: 0.32  Hard: 1.25 | The user chooses an option and then the maze is generated on a grid of a certain size and that the |
| An algorithm is needed to be added to verify the difficulty of the maze | ✔️ | See Technical Solution for Function Name: Difficulty Verification. Will return a value of True if the maze uses a minimum of 90% of the available nodes. | - | A Non-Functional requirement that works behind the scenes to create a maze of a set difficulty. Can verify this works as the program displays a maze instead of crashing/no maze, therefore, I can conclude this is a working requirement. |
| The user will be shown their time on the maze | ✔️ | Logo  Description automatically generated | 0.08, 0.31, 1.35, 7.35 | The timer is shown in the top right of the screen. This is shown on all solve games and if the user option is chosen in the create game mode. |
| The user interacts with the application to navigate the maze. | ✔️ | Text  Description automatically generated | Various Points between 0.00 🡪 8.00 | The user has to use the keyboard keys to interact with the software. The game contains in-game hints to ensure the user knows what keys to press to navigate the software. |
| The User is shown the controls and what each symbol means. | ✔️ | Graphical user interface, text, application  Description automatically generated | Screenshot at 2.31  Various Points (in-game) between 0.00 🡪 8.00 | The screenshot is an example of the in-game side tab which shows the user about the game. The colours of the icons were left out as these became obvious to the users after using the controls. The tab appears in any in-game screen (discludes menus and win-screens) |
| The user can exit the maze at any given time | ✔️ | Graphical user interface, text, application  Description automatically generated | Various Points (in-game) between 0.00 🡪 8.00 | In any given mode, the user can use ESC to return to that game mode’s menu screen. |
| The user can choose a grid size | ✔️ | Text  Description automatically generated | 1: 2.50  2: 3.50  3: 4.52 | The user uses the key presses to choose the size of the grid they wish to build their maze upon. |
| The Maze creation uses mouse clicks to interact with the maze. | ✔️ | A black background with white text  Description automatically generated with medium confidence | Shown for a period of time at 2.50, 3.50, 4.52 | An in-game hint for the user, this explains how the user can interact with the walls by clicking on them. |
| The user must confirm the completion of their maze. | ✔️ | Text  Description automatically generated | Shown for a period of time at 2.50, 3.50, 4.52 | The user has to press this button twice to progress to solving the maze, the first time to indicate the position of the walls, and secondly to show where the start and end point are located (blue and red, respectively) |
| The user can navigate the maze if this option is chosen | ✔️ | Graphical user interface, text, application  Description automatically generated | 3.33, 4.40, 7.29 | Gives the user an option whether the user wants the computer to solve the maze or another player to solve tha maze. |
| The shortest path algorithm is visualised. | ✔️ | A picture containing text, crossword puzzle, clipart  Description automatically generated | 3.36, 4.42 | The Algorithm is ran and then the output is represented on the screen. |
| After any mode has been completed, the user is presented with a win screen | ✔️ | Text  Description automatically generated | 0.22, 0.59, 2.37, 3.47, 4.48, 7.53 | The user is presented with a win screen that contains the time (if applicable) and the keys to return to the menu screens. |