Olde Worlde Phunne – Maze Game

Test Plan 29/07/2019

Version 1.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Software Application:**  Olde Worlde Phunne – Maze Game | | | **Name of Tester:**  Matthew Sawrey | **Date:**  01/08/2019 | |
| **No** | **Requirement** | **Test Case** | **Test Data** | **Expected Result** | **Actual Result** |
| 1 | A Maze shall consist of a set of any number of rooms. | On generation of a Maze, the game creates the number of rooms specified in the configuration file. | "NumberOfRooms": 6,  "NumberOfRooms": 80,  "NumberOfRooms": 750, | The number of rooms is equal to that specified in the configuration data. | ✔ |
| 2 | A Room shall have a number of connecting passages (between 1 and 4 on each room), with all exit passages connecting to another room, except for the final exit passage which shall exit the Maze. | On generation of a Maze, the game assigns each room a number of passages between 1 and 4. | - | On maze Generation, each rooms has between 1 and 4 passages. | ✔ |
| 3 | Each passage that is not the exit passage shall connect two Rooms, which might be the same Room. | Run multiple instances of the game, checking that all passages, apart from the exit passage, lead to another room. | - | All tested passages lead to a room. No exception is thrown from attempting to take a passage. | ✔ |
| 4 | A Passage shall be bi-directional, allowing the player to move back and forth between the same rooms. | Run multiple instances of the game, checking that a passage leading to one room can be used to return to originating room. | - | Returning through a passage will return the user to the originating room. | ✖ There is no way to demonstrate that passages are bi-directional as the game does not specify which passage the user entered a room from. |
| 5 | There can be only one exit passage in the Maze. | Run multiple instances of the game, checking that there is only ever 1 exit passage created on maze generation. | - | There is only 1 exit passage created on maze generation. | ✔ |
| 6 | A Room shall contain a number of interactable Items. | Run multiple instances of the game, checking that each Room contains a number of threats. | - | Each Room contains a number of threats. | ✔ |
| 7 | One room within the Maze shall contain the final exit point of the Maze, which is the end-goal of the game. | Run multiple instances of the game, checking that in each instance, one room contains the exit passage. | - | In each instance of the game, there is one room that contains an exit passage. | ✔ |
| 8 | An Item within a room shall be one of two types – Threat or Treasure. | Review code to determine if any other item types exist within the room object aside from the Treasures and Threats. | - | No other types that derive from the Item object are found within the room object. | ✔ |
| 9 | Each Treasure item shall have an integer property representing its value. | Review code to determine if the Treasure item has a “Value” property. | - | Treasure item has a “TreasureValue” integer property. | ✔ |
| 10 | The player’s total amount of collection Treasure shall tracked by the Player object. | Manually calculate the amount of treasure that the player should have at each stage of collecting and losing treasure and check these values in code using code-breaks in Visual Studio. | - | The player CollectedTreasure value corresponds correctly to gains and losses in treasure. | ✔ |
| 11 | The player shall be able to perform actions upon a Threat item in an attempt to remove the threat item from the game. Only one action shall be successful in removing a Threat. | Code review to check that the Threat item can only have one solution action and multiple instance runs of the game to ensure that the solution action in the Threats configuration file is what defeats the Threat. | - | Threat item only has a single “Solution” field.  Multiple instance runs demonstrate that entering the correct solution defeats a threat. | ✔ |
| 12 | Attempting unsuccessful actions on a Threat or attempting to leave a room before dealing with a Threat shall result in the Threat removing a value of Treasure from the Player’s total treasure count, if they have any Treasure. | Run multiple instances of the game, checking that in each instance, if a player is attacked by a threat, the amount of treasure removed from them is equal to the Threat’s TreasureTheftValue. | - | Multiple instance runs of the game show that in each instance, if a player is attacked by a threat, the amount of treasure removed from them is equal to the Threat’s TreasureTheftValue. | ✔ |
| 13 | The User shall be able to start the game executable file. | Run multiple instances of attempting to run the game .exe file. | - | Multiple instances of attempting to run the game .exe file are all successful. | ✔ |
| 14 | The User shall be able to enter their chosen player name. | Run multiple instances of the game, checking that the Player can enter their name and that the game remembers the Player’s name. | - | Multiple instances of the game that allow the Player to enter their name and this is remembered and used throughout the game. | ✔ |
| 15 | The program shall be capable or reading and parsing the contents of all configuration files. | Run an instance of the game with the appropriate configuration files and determine that the files are parsed correctly into C# objects. | - | The contents of the configuration files are correctly read and a maze correctly generated from them. | ✔ |
| 16 | The program shall alert the user of any errors encountered whilst reading and parsing the configuration file. | Deliberately cause errors with the configuration files. | Errors include:   * Removing the configuration file from the correct location. * Changing the format of the configuration file. * Change the structure of the content of the configuration file. | The correct error messages being displayed:  - ERROR: configuration file does not exist. Please ensure file exists.  - ERROR: configuration file could not be read. Please ensure that the format of the configuration file is correct.  - ERROR: Failed to deserialize json string. Please ensure that the format of the configuration file is correct. | ✔ |
| 17 | The program shall generate a Maze based on the values in the configuration files, including the Maze Seed value which will be used to seed the random number provider. | Run multiple game instances, changing the seed value for the Maze and determine that a different Maze is generated from the seed value. | - | Different game instances with different seed values, each of which generates a different maze structure and content. | ✔ |
| 18 | Once a new Maze has been generated, the User shall be able to start a new instance of a game. | Run multiple game instances in which Maze generation is successful and start the in-game loop. | - | Multiple game instances in which Maze generation is successful and the user is able to start the in-game loop using the generated mazes. | ✔ |
| 19 | The user shall be able to restart their game-instance at any point during the course of the In-Game loop. | Run multiple game instances in which the user uses the “restart” command to restart their in-game loop. | - | Multiple game instances in which the user is able to restart their in-game loop without error. | ✔ |
| 20 | On beginning the in-game loop, the game shall present the user with introductory text. | Run multiple game instances in which the in-game loop introductory text is observed. | - | Multiple game instances in which the user observes the in-game introductory text. | ✔ |
| 21 | The user shall begin the game from any one of the generated Maze Rooms. | Run multiple game instances and use Visual Studio breakpoints to ensure that the initial room index is randomised in every instance. | - | Multiple game instances in which the initial room index is observably randomised in every instance. | ✔ |
| 22 | The game shall present the user with a list of available user-input commands at all points the user is able to perform a command. | Run multiple game instances and observe that the user is presented with a list of available user-input commands at all points the user is able to perform a command. | - | Multiple game instances in which it is observed that the user is presented with a list of available user-input commands at all points the user is able to perform a command. | ✔ |
| 23 | If the user enters a command that is not recognised, the game shall inform the user that their command is not recognised, and present the user with the command list. | Run multiple game instances and observe that the game informs the user that their command is not recognised, and presents the user with the correct command list. | - | Multiple game instances in which it is observed that the game informs the user that their command is not recognised, and presents the user with the correct command list. | ✔ |
| 24 | On user entry of a successful command, the game shall perform the appropriate action and provide the user with feedback on how their scenario has changed. | Run multiple game instances and observe that on user entry of a successful command, the game performs the appropriate action and provides the user with feedback on how their scenario has changed. | - | Multiple game instances in which it is observed that on user entry of a successful command, the game performs the appropriate action and provides the user with feedback on how their scenario has changed. | ✔ |
| 25 | On entering a Maze Room, the game shall present the user with a description of the room they are currently in. | Run multiple game instances and observe that on entering a Maze Room, the game presents the user with a description of the room they are currently in. | - | Multiple game instances in which it is observed that on entering a Maze Room, the game presents the user with a description of the room they are currently in. | ✔ |
| 26 | The User shall be able to drop Treasure in their current room, as long as they have the available Treasure to do so, and view this treasure upon returning to the room | Run multiple game instances and observe that using the “dropcoin” command drops a coin in the room and adds it to the room’s treasures list. Also attempt to do the same when the user doesn’t have any collected Treasure and observe that a coin is not left in the room and a message to this effect is shown. | - | Multiple game instances in which it is observed that using the “dropcoin” command drops a coin in the room and adds it to the room’s treasures list. Also attempt to do the same when the user doesn’t have any collected Treasure and observe that a coin is not left in the room and a message to this effect is shown. | ✔ |
| 27 | The user shall be able to progress to different rooms through the use of passages. | Run multiple game instances and observe that the user is able to progress to different rooms through the use of passages, using the “takepassage {n, s, e, w}” command. | - | Multiple game instances in which it is observed that the user is able to progress to different rooms through the use of passages, using the “takepassage {n, s, e, w}” command. | ✔ |
| 28 | If the user does not clear all Threats from a room before attempting to leave, they shall be prevented from leaving the room. | Run multiple game instances and observe that, if the user does not clear all Threats from a room before attempting to leave, they shall be prevented from leaving the room. | Entering a “takepassage” command whilst threats are still present in the current room. | Multiple game instances in which it is observed that, if the user does not clear all Threats from a room before attempting to leave, they shall be prevented from leaving the room. | ✔ |
| 29 | The game shall track the total number of moves that the User has made during the course of the In-Game loop. | Run multiple game instances and observe that, the game tracks the total number of moves that the User has made during the course of the In-Game loop, and presents this number at the end of the game. | - | Multiple game instances in which it is observed that the game tracks the total number of moves that the User has made during the course of the In-Game loop, and presents this number at the end of the game. | ✔ |
| 30 | If the user reaches the exit point of the maze, the game shall present a summary of their progress. | Run multiple game instances and observe that, when the user reaches the exit point of the maze, the game presents a summary of their progress. | - | Multiple game instances in which it is observed that, when the user reaches the exit point of the maze, the game presents a summary of their progress. | ✔ |
| 31 | The Maze Game shall have a fail state, in which the user has died before reaching the end of the maze. | Run multiple game instances and observe that it is possible for the player to die before reaching the end of the maze. | - | Multiple game instances in which it is observed that it is possible for the player to die before reaching the end of the maze. | ✖ No health system has yet been built into the game and therefore there is no death fail-state. |