

System and Unit Test Template - CMPS 115 – Software Methodology

Amazing Maze

The Maze Runners (Kevin Andres, Jeffrey Deng, Samantha Soohoo, Xiaoli Tang)

System Test scenarios:

Sprint 1

- User Story 1 from Sprint 1: As a player of the game, I want to finish the maze.
- User Story 2 from Sprint 1: As a player I want a welcome menu that allows me to decide when to start playing the game.
- User Story 3 from Sprint 1: As an application developer, I want to player to just control their character with simple directional button.

Scenario

1. Start the “AmazingMaze” game
 - a. User should see a welcome menu
 - b. On the menu, select the option: PLAY.
 - c. On the level menu, select a level and start the game
2. Play the game
 - a. User will show up at the innermost layer and can navigate using in and out button
 - b. User can finish the game by getting to the most outer layer

Sprint 2

- User Story 1 from Sprint 2: As a player, I want to play a game with easy navigation and multiple levels of difficulties
- User Story 2 from Sprint 2: As a developer, I want my software to be free of faults so the user doesn't come across any failures.
- User Story 3 from Sprint 2: As a multi-tasking user, I want to be able to pause the game, start the game and end the game, so that I can multi-task.
- User Story 4 from Sprint 2: As a confused player solving the maze on my phone, I need to have a restart button because I don't want to give up the level nor use hints.
- User Story 5 from Sprint 2: As a user, I need an user documentation so the user can familiarize with the flow of the game

Scenario

3. During the game
 - a. User can select from multiple levels
 - b. All maze are randomly generated and are free of faults
4. Pause the game
 - a. On the upper right corner, the user can choose to pause the game
 - b. A new menu will show up to let the user choose to restart the game or choose a different level

5. User Documentation

- a. User can refer to the user documentation for the installation, navigation and other details of the game.

Sprint 3

- User Story 1 from Sprint 3: As a player I want a menu that has more functionality such as checking scores or changing difficulty level.
- User Story 2 from Sprint 3: As a player who finished a maze, I want to see my score/time and be able to play the game again

Scenario

6. A post menu will show the after the player finish the game
 - a. User can select “Retry” to restart the game
 - b. select “Change Level” to select a new level
 - c. select “Main Menu” to go back to the main page
7. A score/time will show up on the post menu

Unit tests:

Include a file/directory named ‘Testing’ in your Git Repository. There should be details (can be in a separate file in the directory) provided by each team member about the module and the functional testing they have done. Each team member picks a module or module and lists the equivalence classes and the test cases selected to cover all equivalence classes.

Since we are making a game, we can test the functionality by just playing the game.

Jeffrey Deng and Xiaoli Tang: Test the maze generation

Classes under Test:

- MazeDrawer.cpp: DrawMaze(), printall()
- MazeGenerator.cpp: MazeInitializer(), Door Generator(), AssignParenChild(), BlockGenrator(), setDifficulty()
- AmazingMaze.cpp: main()

Test cases:

- 1) Test the number of rings are generated within the range of its difficulty level
- 2) Test the door amount is correct, print the door amount based on the level and check to see if it is within the range
- 3) Test that the door and block is randomly placed on the ring and the block doesn’t come across the door

Kevin Andres: Test the navigation of the player

Class under test:

- PlayerMovement.cpp: moveUp(), moveDown(), PlayerMove(), DrawPlayer()

Test cases:

- testMoveUp(): Test that the player correctly moves up one layer of the maze in order to reach the doors that are accessible from the inside
- testMoveDown(): Test that the player correctly moves down one layer in order to eventually escape the maze
- testPlayerMove(): Test that the player can move in and out of the maze based on their desired input.

Samantha Soohoo: Test the menu items

Classes under test:

- menu.cpp: startM(), levelM(), InGameM(), PauseM(),

test cases:

- Test all menu button show up at the correct location
- Test to see all the buttons are clickable and lead to the correct menu