Test Document

We have created a JUnit test that tests the logic class of the text version of our game. More specifically, the class that is tested is called PacManMazeTextLogic.
In this document, we will outline what is tested in each of the methods in the test class and how each test was carried out.

1) Method name: test_allInstanceVariablesInitialized
Purpose of method: To test if all the instance variables in the text logic class is initialized when the initialize method is used.

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize() method was called to initialize the instance variables. Then we checked if each instance variable was set to null. If any of the instance variables were set to null, that means that the initialize() method does not properly initialize at least one of the instance variables, so the test is failed.

2) Method name: test_initializationLocation
Purpose of method: To test if the initial location of the avatar is set to (1, 1)

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize method was used to initialize the instance variables. Then using the getAvatar and getLocation methods, we obtained the initial location of the avatar and checked if the location is (1, 1). If the location is not at (1, 1), then this test is failed.

3) Method name: test_avatarMove
Purpose of method: To test if the avatar can successfully move left, right, up, down

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize method was used to initialize the instance variables. Then using the getInputAndMove method, we moved the avatar using “w” to go up, “s” to go down, “a” to go left, and “d” to go right. Each time the avatar was moved, we checked if it’s location was changed to the proper coordinates. If the location stayed the same or the new location of the avatar was different from the location we expected, the test is failed. Also,we made sure that each time the avatar was moved, there was no wall to obstruct its movement.

4) Method name: test_invalidInput

Purpose of method: To test that using invalid input for the getInputAndMove method does not cause the avatar to move.

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize method was used to initialize the instance variables. Then the getInputAndMove method was invoked 4 times with “m”, “8”, “@”, “wa” as parameters. Since the avatar will only move when “w”, “s”, “a”, “d” are used as parameters, the parameters mentioned above will not make the avatar move, those are invalid input. So each timer the getInputAndMove method is called with one of those invalid input, we check if the avatar location is the same as before. If the avatar location is not the same as before, then that means that the avatar has moved despite the invalid input, so the test is failed.

5) Method name: test_cannotMoveIntoWall
Purpose of method: To test if the avatar cannot move into walls (so the avatar location and wall location cannot overlap).

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize() method was used to initialize the instance variables. Initially, there is a wall directly to the right of the avatar. Using the getInputAndMove method, we input “a” to make the avatar move right. However since there is a wall to the right of the avatar, it should just stay in the same location. So using the getAvatar and getLocation methods, we check if the new location of the avatar is the same as the previous location. If the new location and previous location are not the same, that means that the avatar has moved into a wall and the test will fail.

6) Method Name: test_initialPointIsZero
Purpose of method: To test if the initial score of the avatar is set to zero.

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize method was used to initialize the instance variables. Then using the getAvatar and getScore methods, we obtained the initial score of the avatar. If the score is set to zero, then the test is passed. If the score is not set to zero, then the test is failed.

7) Method Name: test_PointCollection
Purpose of method: To test if the avatar’s score successfully increases when it collects pellets

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize method was used to initialize the instance variables. Right after initialization, the gameBoard instance variable should be filled with pellets that the avatar can collect and increase it's score. If we move the avatar right, it should have collected the pellet and gotten 10 points. If the avatar has no points, then the test is failed.
Then we move the avatar right again, and it should have collected another pellet and have 20 points, If the avatar does not have 20 points, then the test is failed.
Lastly, the avatar is moved left. Since the avatar has already collected the pellet from that location, it's score should still be 20. If the score is not 20, then the test is failed.

8) Method Name: test_GhostMove
Purpose of method: To test if the ghost moves only when avatar moves.

To carry out this test, an instance of PacManMazeTextLogic was created and the initialize method was used to initialize the instance variables. Then using the getInputAndMove method, the avatar was moved to the right. Then using the getGhost and getLocation methods, the initial location of the ghost and the location of the ghost after the avatar moved is compared. If the initial and current location of the ghost is the same, then that means that the ghost has not moved when it should have, and the test is failed.