PROJECT DELIVERABLE 3

BY :

SOHAIL AHMED ZAMAN

(ID 38879972)

EXERCISE 1 – TEST PLANNING

1) The objectives for my final function tests are

- Ensuring that the puzzle piece image is displayed properly on the JFrame.
- Checking if the puzzle piece moves when dragged.
- Checking if the incorrect puzzle is returned
- Checking if a new piece is generated after selecting the right position of the puzzle piece

2) The test will take place after the completion of the code

- The test will take place In parts, the first part will check if the image is displayed properly on the JFrame.
- The next test will be done to see if the puzzle piece moves correctly when dragged and return so for this I will move the pieces around and try to place them on existing pieces.
- The next test will be carried out to see if the puzzle pieces already on the board move when dragged for this I will drag the existing pieces on the board around.
- Finally the parts mentioned above will be thoroughly tested using JUnit testing individually

3) The tools and resources that will be for the testing involve Eclipse IDE and JUnit tests.

4) Testing criteria:

- The criteria for passing each of the tests will be checking if the part of the code being tested does what it is supposed to and does not cause any errors.

Exercise 2: Test Execution

1) For testing I used the JUnit test for each part. I declared each part separately into different files to run the test. During the test only one error was found which I investigated further and it lead to an anomalous behavior being noted

2) The anomalous behavior noted during the test execution was ArrayIndexOutofBoundsException. This was noted when testing if the puzzle pieces are returned when pieces are placed on an existing piece. When I tried to keep a new piece that was generated on top of an existing piece I realized that the pieces were the same and the new piece placed got stuck when kept on the existing piece and resulted in an error stating ArrayIndexOutofBoundsException. I tried running the program again this time the duplicate piece when I tried to move it around got stuck outside the puzzle board.

3) This anomalous behavior was caused when a new piece that was already on the board was generated. The piece when dragged and placed just outside the board or at the top of the existing piece gets stuck and results in an Array out of bounds exception. To fix this behavior I used a new temporary variable that stores the random number generated for the puzzle piece and a new do-while loop that checks if the randomly generated number already corresponds to an existing piece on the board. If the number belonged to a piece already printed then a new number is

generated again. The moving function was tweaked a bit to ensure that if the puzzle is not kept in the right position on the board it will return to its initial position immediately.

EXERCISE 3: TEST REPORTING

The entire testing plan and its execution were done regarding the way the code was written. So to test my program I divided my program into smaller parts, each with its use and implementations, and formulated the objectives of the test plan for these parts. The execution was carried out in the order in which the code was written in. It started by testing out if the image is displayed on Jframe correctly. After this, the movement mechanism and the solution of the puzzle game were evaluated and tested

During one of the test executions, an anmoumouls behavior was identified. It was an Arrayindexoutofbounds exception. The cause of this was traced down and found to be due to the duplication of the puzzle pieces which were already on the board to resolve this a new temporary value and do-while loop was introduced which holds the random number generated and checks if the number corresponds to an already printed piece if so then a new number is generated. After fixing the exception the entire program was tested using JUnit Test again for the last time and no bugs or errors were found. Thus completing the testing for the code