Loop Testing

We use in total of nine loop in our project.

1. A while loop in getLevel method under StateManger class. The loop inside will always be executed 6 times since we have six levels made in total. Unless we make specified number of levels, we will not be able to follow the Loop Testing as described on the powerpoint guide lines. If we make our game to have only one level, we will be able to execute the loop only once. And so to other numbers. However, once we start our game, the getLevel method will automatically going through the loop six times and load the levels. Therefore, it is reasonable for us to do only one test rather than all of the test cases followed by Loop Testing guideline.

2. The second loop we have is under LoadTile method in Tile class. Same as the getLevel method, number of tiles are set in our game. If we change number of tiles, or the structure of the tiles, then the tiles will not be drawn correctly. Unless we modifies the structure or code for Tile class, we will not be able to test the LoadTile method loop. Therefore, it is impossible to do the loop testing as shown on the powerpoint guide.

3. The third loop that we have is the init method under the TileSet method. The number of loops this method went is always six because we need images for tiles, castle, dungeon, forest, light_dungeon, and winter in order to draw the game correctly. In another word, I will not be able to do the loop test for TileSet method unless I have specified number of images for drawing. In additional to that, we have this method to be private, therefore, it is impossible for me to test this one out.

4. The fourth loop that we have is the loadMap method under the Map class. Same as the reason for getLevel and LoadTile. We have to change the structure of Map in order to do loop tests followed by the powerpoint guide line. We tried to test it, but the cost will be breaking the whole project. In addition to that, we want to make this method to be private, therefore. It is impossible to test this case.

5. The fifth loop is the checkEntityCollision method under Projectile. This method is depending on the size of entity. If the size of Entity is 2, then this loop will be looping two times. If checkEntityCollision does not depends on Entity size, then I will be able to test this method. In additional to that Projectile class that contains checkEntityCollision method requires abstract class Entity to be part of the constructor. There is no way to instantiate the Entity object, therefore, we did not test this loop method.

6. Sixth loop that we have is the render method under Map class. It requires a Graphics2D in order to do the tests. Unless I have a map that are drawn already. we will not be able to test them ourselves.

7. The seven loop that we have is the constructor method under SpriteSheet class. This method loads the image from specified path and create ArrayList of images. Unless we change the assets or the structure SpriteSheet, then we will not be able to test it out. Unfortunately, we do not want our game to be buggy, therefore not doing the test cases is more reasonable.

8. The eighth method that contains loops is the update method under Level. Unfortunately, the the loop is not depending on the variables in the Level class, but the Entity class. Therefore, number of loops it goes through can not be assigned. Besides that Level constructor requires a StateManager instance. The Level will cause errors if the game has not started. Therefore, it is kind impossible to do loop tests for this method.

9. The ninth method that contains loops is the render method under Level. Unfortunately, same as the update method shown above, the number of loops depends on the size of Entity class which are affected by other classes, therefore it is nearly impossible to do loop test. Besides that, there is no way of constructing a Graphics2D object unless we are playing the game.