Assignment 3 - Report

Everyone helped to implement each package in the game, but we mostly focused on the “Characters” and the “RewardsAndPunishments”. So, we identified our code smells mostly in those packages and made the refactoring inside them. During this code review, we adjusted the code and tests related to these two packages, we found many problems, they can be divided into the following subtitles..

bad/confusing variable/method names

We found many confusing variable and method names which do not express their purposes well, such as the commit 71544c00, we changed the method’s name to “checkWinTheGame” so that others can obviously know what it is for.

As you can see in the commit 1e7719c6, we also fixed this problem by changing how some variables are used, which makes the improper name meaningful.

Also, we renamed some variables as well so they are more understandable for everyone as you can see in the commit 590d9c06.

methods that are too long and that could benefit from being refactored

After we went through the Characters package, we found that the 4 methods (moveUp(), moveDown(), moveLeft(), moveRight()) for character’s movement contain too many conditions and too many functions and the purpose of these 4 functions seemed similar. So we refactored these movement methods by extracting the function of judging whether the characters overlap and the function of repainting the pattern of the characters and created a new method to implement these 2 features instead in the commit 10d37414.

long list of method parameters

After we checked the constructors of all classes, we found that the speed at which the character moves in the x and y directions is determined by the update method in the GamePanel class, so we removed speedX and speedY for all characters and reduced the number of parameters in all constructors of the relevant class in the commit 4c8b8b2c.

We also found we can not have more than one player in our game, so we deleted the variable name in Player class which we originally wanted to use this variable to identify the player and removed a parameter from Player’s constructor in the commit 71544c00.

lack of documentation

When reviewing our code, we found that it is difficult to know what some classes can do and what their main purposes are from JAVADOC. So we add more explanations for these classes in the commit dc688036.

unused or useless codes

We removed some useless import lines we had from the code. They can be found in the commit 76f84e77.

There were also unused variables for Character, Player and Enemy classes to assign them speed parameters but there was no need so we removed them in the commit 4c8b8b2c.

We also had some methods that we never used for KeyHandler class, so we removed them as well in the commit 70455143.

unnecessary use of unsafe or unsound constructs

The array lists we used to store the Reward objects, the Punishment objects and the Enemy objects that we generate to be spawned at the game panel were static variables and because of that the tests we created were failing when they are being tested in a row due to not being able to create a new list when the constructor is called more than once, instead it was adding on to the list that created at first with the first constructor call, so we were ending up with not getting the exact amount of objects we want in the lists. Because of that, we changed the static list variables to non-static as you can see in the commits f859644d and 1e7719c6.

We also made the “location” and “value” variables “final” since they are only being initialized once, which can make our data more safe as you can see in the commit a9ef8113.

Also, the static members ‘MazeGame.GamePanel.level’ and ‘MazeGame.LevelGenerator.gameObjects’ were being accessed via instance reference, so we made them being accessed via classes ‘GamePanel’ and ‘LevelGenerator’ references as you can see in the commit 9d33977d.

unnecessary if/else or switch/case statements

We had if statements in the setup method as a part of @BeforeEach to make sure that the generator lists we will be using for the tests are null so we don’t get StackOverflow exception due to having generator list variables as static but since we refactored our static lists to non-static, we didn’t need of those if statements we had for the setup method in the test classes as you can see in the commit a71f2839.