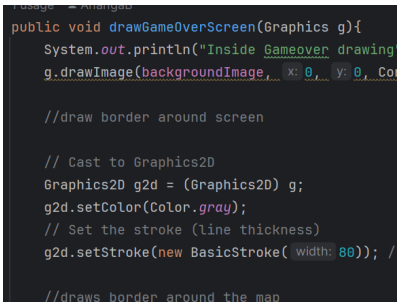
Assignment 4 - Code Review
By Mohamed Mustafa and Ananga

1. In the Game Panel, the game over screen is drawn directly inside an else statement. There are a large number of lines, which actually should be a method elsewhere. This is a problem of " poorly structured code".

Solution: Create a new method that does the job of drawing the game over screen.



I created a new method, in the Game Panel class, called drawGameOverScreen(Graphics g). Then the text blurb, I replaced with just a function call to method, on the left. The commit for this had SHA of a19dbe9be1908a0f19f35b6682b64616afc10b1f and was titled, "refactored GamePanel class, broke some blurbs of code into smaller fu…."

2. In the Game Panel, the start screen is drawn directly inside an else if statement. There are a large number of lines, which actually should be a method elsewhere. This is a problem of " poorly structured code".

Solution: Create a new method that does the job of drawing the Game Start Screen.

I created a new method, in the Game Panel class, called drawGameStartScreen(Graphics g). Then the text blurb above I replaced with just a function call to the newly created method. The commit for this had SHA of a19dbe9be1908a0f19f35b6682b64616afc10b1f and was titled, "refactored GamePanel class, broke some blurbs of code into smaller fu…."

3. In game class, there is another case of large blurb of code. This is shown on the left. This is a problem of " poorly structured code".

Solution: Implement a few other method calls to replace the blurb. Now we can also repeat these method calls potentially elsewhere. This was done in a commit, 3b66c6d68aaa52fec7033a654e0f6c9a14b0ad5d , titled "refactored game class, and created new methods to improve code readi…"

4. In Game class, the code that drew the regular game play screen, was all in a if statement, and made code readability hard. There was just a blurb of code for in the if statement that should have been taken out. This is a problem of " poorly structured code".

Solution: Create a new method that does the job of drawing the Game Play Screen.

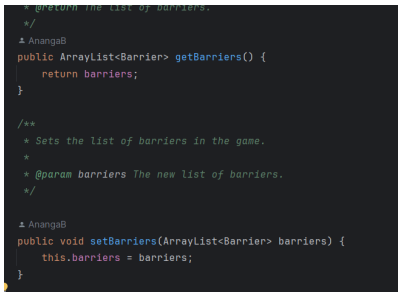
I created a new method, in the Game Panel class, called drawPlayScreen(Graphics g). Then the text blurb, I replaced with just a function call to the newly created method below. Clearly, the new modification makes it more clean:d

```
if(this.currentState == Constants.GAME_STATE.PLAY) {
    drawPlayScreen(g);
}
```

The commit for this had SHA of ab120c55440489d4edcf53d97949e8d0c472d2df and was titled, "refactored game class, and created new method: drawPlayScreen."

Assignment 4 - Code Review
By Mohamed Mustafa and Ananga

5. In RegularReward.java, there are redundant width and height variables. The width and height in the parameters of the class are used in the BufferedImage initialization, so they are unneeded. This is redundant and can lead to confusion. This problem is “Unused or useless variables”. **Solution: I removed the redundant variables to allow the width and height variables in the parameters of the class to be used.** Now the variables declared inside the class are relevant to the class itself. The commit for this had SHA of 2b0048b95df058752fbba7d165cc5ddaebedf6d3 and is titled “Refactored redundant variables”



6. In GamePanel.java, getBarriers() and setBarriers() methods are created but never used. This is redundant and the program works the same and uses less memory without the methods. This problem is “dead code”. **Solution: I removed the dead code methods.** The commit for this had SHA of 4c4ecb3a92b860e25d15c7ddf19f4937d5006fc3 and is titled “Removed dead code methods from GamePanel.java”

7. In isMovingEnemyContact, if contact is made with a movingEnemy, the main character’s score is decremented by damageamount. This damageamount belongs to the Punishment, and the same process of damaging in the Punishment is done in the MovingEnemy. As per the assignment instructions, the MovingEnemy ends the game upon contact with the main character and does not decrement the points. Therefore, the isMovingEnemyContact should not decrement the player’s points. This problem is “code duplication”. **Solution: Removed score decrementation from isMovingEnemyContact.** Now, according to the instructions, only the Punishment decrements the character’s score upon contact. The commit for this had SHA of d6832af570d2f5513fd77c3dabe6db58ad0fbe9d and is titled “Removed score decrementation from isMovingEnemyContact”.

8. In the MainCharacter class, the boolean hasCollectedAllKeys is from an earlier implementation of key collection and is no longer used, thus being redundant. This problem is “unused/useless variables”. **Solution: Removed the hasCollectedAllKeys variable.** This reduces code redundancy and reduces memory usage. The commit for this had SHA of 73873ed6ca7a8d1f5d15a7cc906ac4245b82d0bb and is titled “Removed hasCollectedAllKeys”