

- Bloater: Long Parameter List in the spawnEnemies() method of the StartCombat class. This method has four parameters, with one of them (damage) not being necessary. This code smell exists because we had not yet decided that we wanted to make the damage always 1. Because of this, we just created a parameter for damage.

I fixed this code smell by removing the damage parameter from the method header. Additionally, I edited the spawnEnemies method by replacing any references to the old damage parameter with the constant value of 1. This allowed the spawnEnemies() method calls to be more concise and readable.

- Object Oriented Abuse: The place tower functionality was previously implemented with a switch statement within the MainGame class. This was because we were focused on making the visuals work rather than getting the logic right and putting everything in the MainGame class seemed most convenient. To fix this, I created a new placeTower() method in the abstract tower class that handles the logical part of placing towers while leaving the visuals to MainGame. Although the code still abuses a switch statement, it is an improvement over before.

Before:

```
switch (currentTowerIndicator) {
    case 0:
        Tower redTower = new RedTower(mouseEvent.getX(),
            mouseEvent.getY());
        ((Rectangle) redTower.getDisplay()).setX(mouseEvent.getX()
            - redTower.getWidth() / 2d);
        ((Rectangle) redTower.getDisplay()).setY(mouseEvent.getY()
            - redTower.getHeight() / 2d);
        center.getChildren().add(redTower.getDisplay());
        PlayerInfo.addTower(redTower);
        break;
    case 1:
        Tower greenTower = new GreenTower(mouseEvent.getX(),
            mouseEvent.getY());
        ((Rectangle) greenTower.getDisplay()).setX(mouseEvent.getX()
            - greenTower.getWidth() / 2d);
        ((Rectangle) greenTower.getDisplay()).setY(mouseEvent.getY()
            - greenTower.getHeight() / 2d);
        center.getChildren().add(greenTower.getDisplay());
        PlayerInfo.addTower(greenTower);
        break;
}
```

New Method:

```
public void place(double x, double y) {
    ((Rectangle) display).setX(x);
    ((Rectangle) display).setY(y);
    PlayerInfo.addTower(this);
}
```

- Mysterious Name: Within the Enemy class and child classes (i.e. PurpleEnemy, OrangeEnemy, YellowEnemy) there was a mysterious variable named "d". Since I was the one that wrote it I knew what it represented, but by leaving it in its current state my peers cannot comprehend what that variable represents. I left it there as a placeholder but forgot to change it in the later stages of my code. "d" is meant to represent damage, so to make the code more comprehensible I changed the variable `int d` to `int damage`. Therefore, now it is more clear what this variable's purpose is in our code. Therefore, I changed all instances of `d` in the Enemy class and child classes to `damage`.

- Duplicated Code: There were 3 sections of code that were duplicated in the StartCombat class. It was not only amateur, but it also hurt the flexibility of our code. We were not able to spawn enemies in any other order than the one set in the code block. To fix this, I moved removed the duplicates and placed the block of code further down. I made the differences in each of the sections decided earlier. Our code was much more flexible and no longer had multiple copies of that block of code.