Code Smell – Switch Statements – James Ball

```
        }
        int distance = 0;
        int length = 0;
        switch (ticksBeforeAttack) {
        case 0:
            distance = WIDTH * 9 / 4;
            length = WIDTH * 19 / 4;
            break;
        case 1:
            distance = WIDTH * 7 / 4;
            length = WIDTH * 15 / 4;
            break;
        case 2:
            distance = WIDTH * 5 / 4;
            length = WIDTH * 11 / 4;
            break;
        case 3:
            distance = WIDTH * 3 / 4;
            length = WIDTH * 7 / 4;
            break;
        default: //case 4
            distance = WIDTH / 4;
            length = WIDTH / 2;
            break;
        }
```

We were using a switch statement to calculate the length and distance of the “poison gas clouds”, the attack emitted from the BioTower. This occurred in the ‘draw()’ method of the BIOTower.java class. This only gave us a set of fixed cases for the animation, so if we wanted to change how the animation looked, we would either have to change every single value or add new cases to the switch statement.

```
        if (ticksBeforeAttack < 0 || ticksBeforeAttack > 4) {
            // no gas cloud to draw
            return;
        }
        int distance = setDistance(ticksBeforeAttack);
        int length = setLength(ticksBeforeAttack);

        gc.setFill(Color.DARKSEAGREEN);
```

```
    private int setDistance(int ticks) {
        return WIDTH * (9 - ticks * 2) / 4;
    }

    private int setLength(int ticks) {
        return WIDTH * (19 - ticks * 4) / 4;
    }
}
```

This switch statement was reduced simple methods that would automatically assign the value of both the length and the distance of the attacks. In the future if we want to change the animation, we will just have to tweak the values of each method's return statement instead of having to go through and change every case of the switch statement.