

```

@Override
public List<Enemy> attack(List<Enemy> enemyArray) {
    for (int i = 0; i < enemyArray.size(); i++) {
        if (Math.abs(enemyArray.get(i).getXLoc() - xLoc - 37.5) < (150*range) &&
            Math.abs(enemyArray.get(i).getYLoc() - yLoc - 37.5) < (150*range) &&
            enemyArray.get(i).getHealth() != 0) {
            enemyArray.get(i).setHealth(enemyArray.get(i).getHealth()-damage);
            break;
        }
    }
    return enemyArray;
}

```

A team member used interface for towers and declared the method attack() in that interface class. This meant that we have to implement and override that method for each tower. However, Tower1 and Tower2 have same method convention. The above picture is from Tower1.java and the picture below is from Tower2.java.

```

@Override
public List<Enemy> attack(List<Enemy> enemyArray) {
    for (int i = 0; i < enemyArray.size(); i++) {
        if (Math.abs(enemyArray.get(i).getXLoc() - xLoc - 37.5) < (150*range) &&
            Math.abs(enemyArray.get(i).getYLoc() - yLoc - 37.5) < (150*range) &&
            enemyArray.get(i).getHealth() != 0) {
            enemyArray.get(i).setHealth(enemyArray.get(i).getHealth()-damage);
            break;
        }
    }
    return enemyArray;
}

```

This can be fixed by making the TowerInterface.java have the same body to define the default method call. This also requires to declare two variables damage and range and make that override to ensure this method works with new damage and range. This change is made since I thought initially the TowerInterface file was a lazy class.

```

int damage = 0;
int range = 1;

default List<Enemy> attack(List<Enemy> enemyArray) {
    for (int i = 0; i < enemyArray.size(); i++) {
        if (Math.abs(enemyArray.get(i).getXLoc() - getXLoc() - 37.5) < (150*range) &&
            Math.abs(enemyArray.get(i).getYLoc() - getYLoc() - 37.5) < (150*range) &&
            enemyArray.get(i).getHealth() != 0) {
            enemyArray.get(i).setHealth(enemyArray.get(i).getHealth()-damage);
            break;
        }
    }
    return enemyArray;
}

```