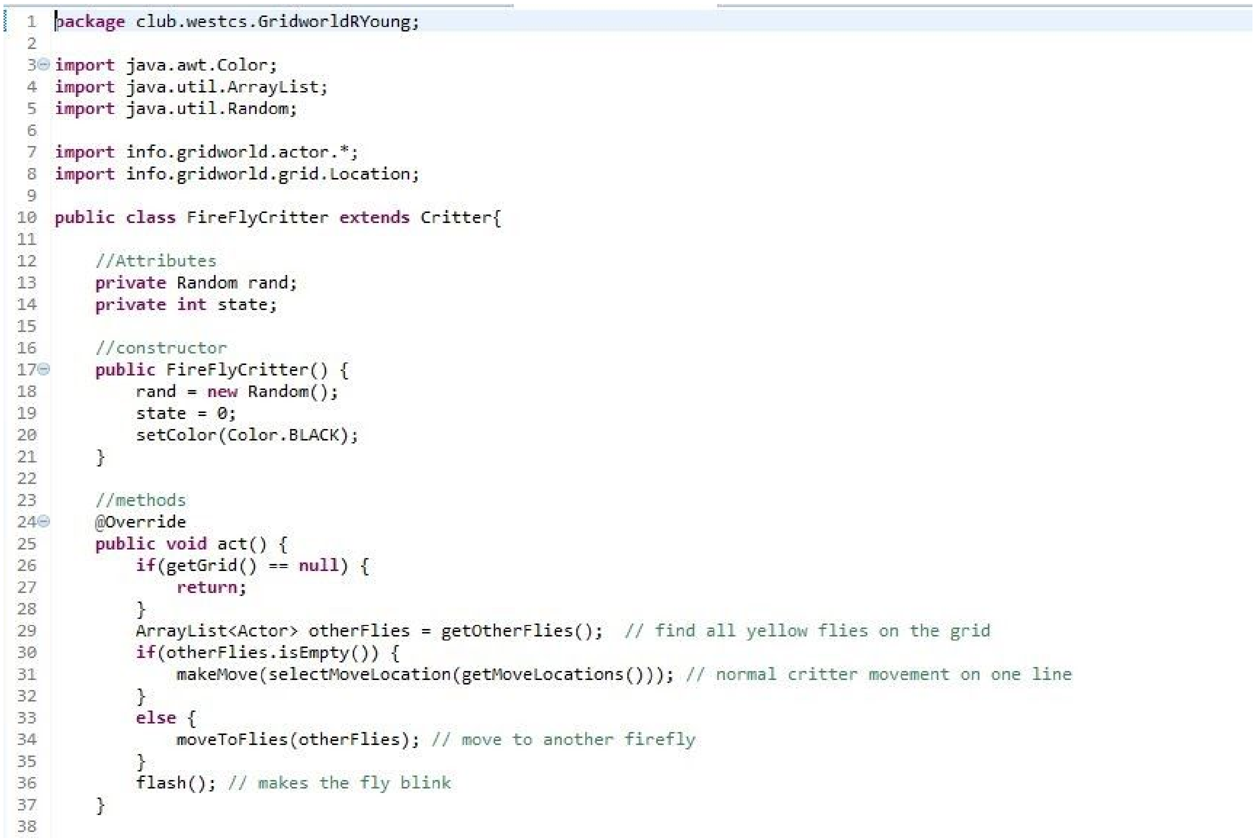
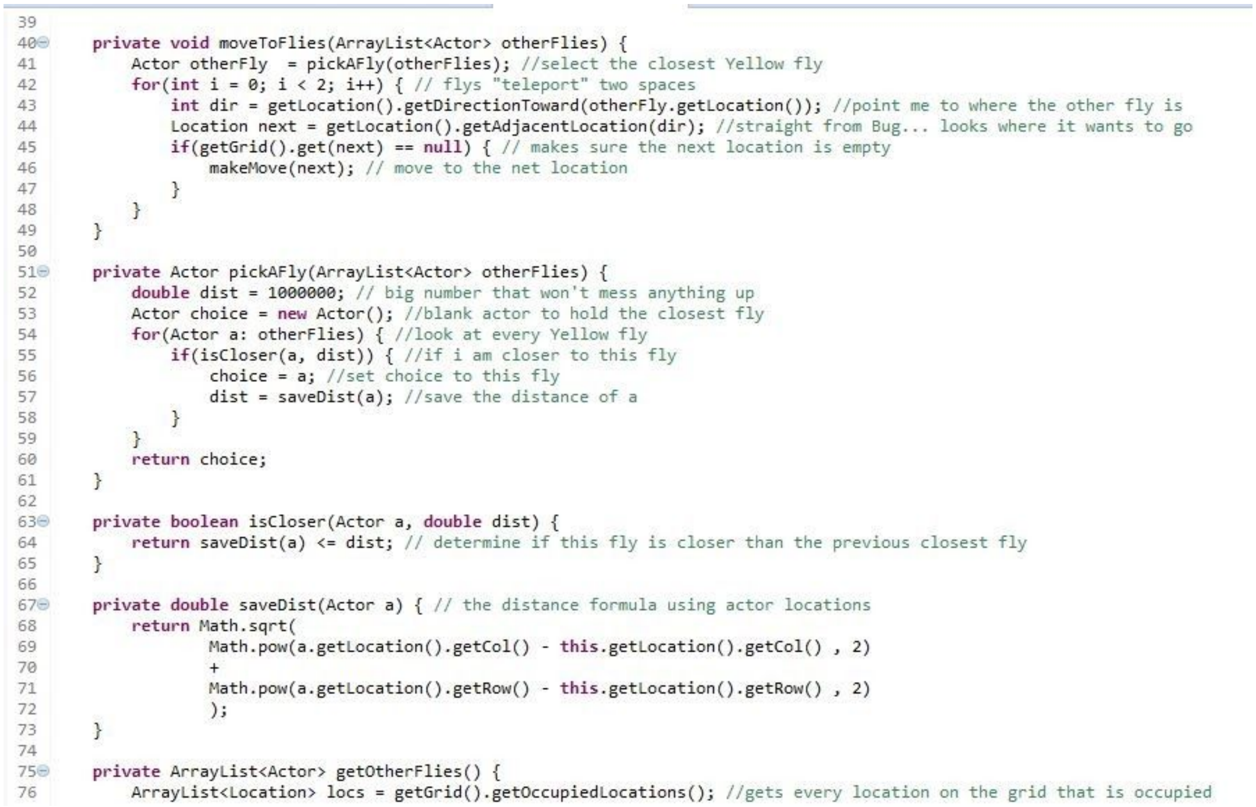
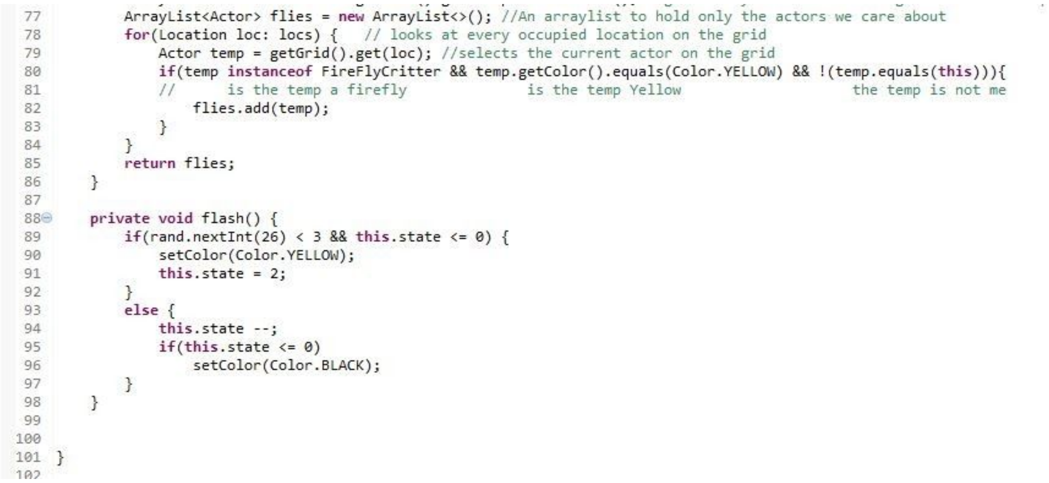


```java
package club.westcs.GridworldRYoung;

import java.awt.Color;
import java.util.ArrayList;
import java.util.Random;

import info.gridworld.actor.*;
import info.gridworld.grid.Location;

public class FireFlyCritter extends Critter{

    //Attributes
    private Random rand;
    private int state;

    //constructor
    public FireFlyCritter() {
        rand = new Random();
        state = 0;
        setColor(Color.BLACK);
    }

    //methods
    @Override
    public void act() {
        if(getGrid() == null) {
            return;
        }
        ArrayList<Actor> otherFlies = getOtherFlies();  // find all yellow flies on the grid
        if(otherFlies.isEmpty()) {
            makeMove(selectMoveLocation(getMoveLocations())); // normal critter movement on one line
        }
        else {
            moveToFlies(otherFlies); // move to another firefly
        }
        flash(); // makes the fly blink
    }

```



```java

    private void moveToFlies(ArrayList<Actor> otherFlies) {
        Actor otherFly  = pickAFly(otherFlies); //select the closest Yellow fly
        for(int i = 0; i < 2; i++) { // flys "teleport" two spaces
            int dir = getLocation().getDirectionToward(otherFly.getLocation()); //point me to where the other fly is
            Location next = getLocation().getAdjacentLocation(dir); //straight from Bug... looks where it wants to go
            if(getGrid().get(next) == null) { // makes sure the next location is empty
                makeMove(next); // move to the net location
            }
        }
    }

    private Actor pickAFly(ArrayList<Actor> otherFlies) {
        double dist = 1000000; // big number that won't mess anything up
        Actor choice = new Actor(); //blank actor to hold the closest fly
        for(Actor a: otherFlies) { //look at every Yellow fly
            if(isCloser(a, dist)) { //if i am closer to this fly
                choice = a; //set choice to this fly
                dist = saveDist(a); //save the distance of a
            }
        }
        return choice;
    }

    private boolean isCloser(Actor a, double dist) {
        return saveDist(a) <= dist; // determine if this fly is closer than the previous closest fly
    }

    private double saveDist(Actor a) { // the distance formula using actor locations
        return Math.sqrt(
                Math.pow(a.getLocation().getCol() - this.getLocation().getCol() , 2)
                +
                Math.pow(a.getLocation().getRow() - this.getLocation().getRow() , 2)
                );
    }

    private ArrayList<Actor> getOtherFlies() {
        ArrayList<Location> locs = getGrid().getOccupiedLocations(); //gets every location on the grid that is occupied
```



```java
        ArrayList<Actor> flies = new ArrayList<>(); //An arraylist to hold only the actors we care about
        for(Location loc: locs) {   // looks at every occupied location on the grid
            Actor temp = getGrid().get(loc); //selects the current actor on the grid
            if(temp instanceof FireFlyCritter && temp.getColor().equals(Color.YELLOW) && !(temp.equals(this))){
            //      is the temp a firefly              is the temp Yellow                     the temp is not me
                flies.add(temp);
            }
        }
        return flies;
    }

    private void flash() {
        if(rand.nextInt(26) < 3 && this.state <= 0) {
            setColor(Color.YELLOW);
            this.state = 2;
        }
        else {
            this.state --;
            if(this.state <= 0)
                setColor(Color.BLACK);
        }
    }


}
```