```java
package club.westcs.GridWorldBeckerbauer;

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

import info.gridworld.actor.Actor;
import info.gridworld.actor.ActorWorld;
import info.gridworld.actor.Bug;
import info.gridworld.actor.Critter;
import info.gridworld.actor.Flower;
import info.gridworld.actor.Rock;
import info.gridworld.grid.*;

public class WeatherCritterLevel3 extends Critter{
    private Random rand;
    private boolean rain, ice, highWinds, first;
    private int step1, ourStep, count;
    private Actor temp;
    ActorWorld world = new ActorWorld();

    public WeatherCritterLevel3() {
        rand = new Random();
        rain = false;
        ice = false;
        highWinds = false;
        ourStep = 0;
        first = true;
        count = 0;
        }

    public void run() {
        move1();
        ArrayList<Actor> neighbors = this.getGrid().getNeighbors(getLocation());
        for(Actor a: neighbors) {
            if(a instanceof Bug) {
                first = false;
                Weather();
```

```
39              }
40              else {
41                  move1();
42          }}
43          }
44
45      private void move1() {
46          count += 1;
47          first = false;
48              Grid<Actor> gr = getGrid();
49              if (gr == null)
50                  return;
51              Location loc = getLocation();
52              Location next = loc.getAdjacentLocation(getDirection());
53              if (gr.isValid(next)) {
54                  moveTo(next);
55                  step1 += 1;
56      }
57              ArrayList<Actor> neighbors = this.getGrid().getNeighbors(getLocation());
58              for(Actor a: neighbors) {
59               if(a instanceof Bug) {
60                  Weather();
61                  }}  }
62
63
64
65      public void Weather() {
66              while(rain == false || ice == false || highWinds == false) {
67          rain = rand.nextBoolean();
68          if(rain) {
69              for(int i = 0; i < 20; i++) {
70                  Flower rain1 = new Flower();
71                  rain1.setColor(Color.BLUE);
72                  rain1.putSelfInGrid(getGrid(), randomLocation());
73                  if(count == 3) {
74                      rain1.removeSelfFromGrid();
75                  }
76              }
```

```
77          }
78          else if(rain != true) {
79              ice = rand.nextBoolean();
80              if(ice) {
81                  for(int i = 0; i < 20; i++) {
82                      Rock ice1 = new Rock();
83                  ice1.setColor(Color.BLUE);
84                  ice1.putSelfInGrid(getGrid(), randomLocation());
85                  if(count == 3) {
86                      ice1.removeSelfFromGrid();
87                  }
88              }}
89              else if(ice != true) {
90                  highWinds = rand.nextBoolean();
91                  if(highWinds) {
92                      act();
93                  }
94              }
95          }}}
96
97      @Override
98
99      public void act() {
100         ArrayList<Location> locs = getGrid().getOccupiedLocations();
101         for(Location l: locs) {
102             temp = getGrid().get(l);
103             if(temp != null ) {
104             this.determineOppositeDirection(temp);
105             move();
106             }}
107         }
108
109     public Location randomLocation() {
110         Location loc = new Location(0,0);
111         do {
112             loc = new Location(rand.nextInt(this.getGrid().getNumRows()), rand.nextInt(this.getGrid().getNumCols()));
113         }
114         while(getGrid().get(loc) != null);
```

```
115         //System.out.println(loc);
116         return loc;
117     }
118 
119     public int determineOppositeDirection(Actor temp) {
120         if(this.getGrid() == null || temp.getGrid() == null) {
121             return 0;
122         }
123         if(this.getDirection() == Location.EAST) {
124             temp.setDirection(Location.WEST);
125         }
126         else if(this.getDirection() == Location.WEST) {
127             temp.setDirection(Location.EAST);
128         }
129         else if(this.getDirection() == Location.NORTH) {
130             temp.setDirection(Location.SOUTH);
131         }
132         else if(this.getDirection() == Location.SOUTH) {
133             temp.setDirection(Location.NORTH);
134         }
135         else if(this.getDirection() == Location.NORTHWEST) {
136             temp.setDirection(Location.SOUTHEAST);
137         }
138         else if(this.getDirection() == Location.NORTHEAST) {
139             temp.setDirection(Location.SOUTHWEST);
140         }
141         else if(this.getDirection() == Location.SOUTHWEST) {
142             temp.setDirection(Location.NORTHEAST);
143         }
144         else if(this.getDirection() == Location.SOUTHEAST) {
145             temp.setDirection(Location.NORTHWEST);
146         }
147         return 0;
148     }
149 
150     public void move() {
151         super.act();
152     }
153 
154 }
155 
```