

Homework Turnin

Name: Xuqing Wu
Account: xw88 (xw88@uw.edu)
Student ID: 1933202
Section: AS
Course: CSE 142 19au
Assignment: a8
Receipt ID: d42e334aa3592b147c5277df6c22e0f1

Warning: Your turnin is 2 days late. Assignment a8 was due Tuesday, December 3, 2019, 11:59 PM.

Turnin Successful!

The following file(s) were received:

Bear.java (1301 bytes, sha256: 7c903e1c9c222aa233c13fce6bae329e)

```
1. // Xuqing Wu
2. // 12/5/2019
3. // CSE142
4. // TA: Ethan M Knutson
5. // Assignment #8
6. //
7. // This program will present a critter called bear which
8. // infect if an enemy is in front, otherwise hop if possible,
9. // otherwise turn left. Its color change between white and
10. // black. Its appearance chnage from / to \.
11.
12. import java.awt.*;
13.
14. public class Bear extends Critter{
15.     private int count;
16.     private boolean polar;
17.
18.     //the constructor
19.     //polar is to determine whether it's black or white
20.     public Bear(boolean polar){
21.         count = 0;
22.         this.polar = polar;
23.     }
24.
25.     //return the appearance of the bear(/ or \)
26.     public String toString(){
27.         if(count%2==0){
28.             return "/";
29.         }
30.         else{
31.             return "\\";
32.         }
33.     }
34.
35.     //return the color of bear(black or white)
36.     public Color getColor(){
```

```

37.     if(polar){
38.         return Color.WHITE;
39.     }
40.     return Color.BLACK;
41. }
42.
43. //return the movement of bear
44. //info is the information about the position situation of the bear
45. public Action getMove(CritterInfo info){
46.     count++;
47.     if(info.getFront() == Neighbor.OTHER){
48.         return Action.INFECT;
49.     }
50.     else if(info.getFront() == Neighbor.EMPTY){
51.         return Action.HOP;
52.     }
53.     else{
54.         return Action.LEFT;
55.     }
56. }
57. }

```

Giant.java (1206 bytes, sha256: 55e1506553f7cc3d33aeb80f225f502f)

```

1. // Xuqing Wu
2. // 12/5/2019
3. // CSE142
4. // TA: Ethan M Knutson
5. // Assignment #8
6. //
7. // This program will present a critter called giant which
8. //infect if an enemy is in front, otherwise hop if possible,
9. //otherwise turn right. Its appearance is "fee" for 6 moves,
10. //then "fie" for 6 moves, then "foe" for 6 moves, then "fum"
11. //for 6 moves, then repeat. Its color is gray.
12.
13. import java.awt.*;
14.
15. public class Giant extends Critter{
16.     private int step;
17.
18.     //the constructor
19.     public Giant(){
20.         step = 0;
21.     }
22.
23.     //return the color of giant(gray)
24.     public Color getColor(){
25.         return Color.GRAY;
26.     }
27.
28.     //return the appearance of giant
29.     //"fee", "fie", "foe", "fum" each for 6 moves
30.     public String toString(){
31.         String[] str = {"fee", "fie", "foe", "fum"};
32.         return str[step/6%4];
33.     }
34.
35.     //return the movement of giant
36.     //info is the information about the position situation of the lion
37.     public Action getMove(CritterInfo info){
38.         step++;
39.         if(info.getFront() == Neighbor.OTHER){
40.             return Action.INFECT;
41.         }
42.         else if(info.getFront() == Neighbor.EMPTY){
43.             return Action.HOP;
44.         }
45.         else{
46.             return Action.RIGHT;
47.         }
48.     }

```

```
49. }  
50.
```

Husky.java (774 bytes, sha256: 9ea1e93d257f20debfe0b7de862c463d)

```
1. // Xuqing Wu  
2. // 12/5/2019  
3. // CSE142  
4. // TA: Ethan M Knutson  
5. // Assignment #8  
6. //  
7.  
8. import java.awt.*;  
9. import java.util.*;  
10.  
11. public class Husky extends Critter{  
12.     private Random r;  
13.  
14.     public Husky(){  
15.         r = new Random();  
16.     }  
17.  
18.     public Color getColor(){  
19.         int choice = r.nextInt(2);  
20.         if(choice == 0){  
21.             return Color.YELLOW;  
22.         }  
23.         else{  
24.             return Color.BLUE;  
25.         }  
26.     }  
27.  
28.     public String toString(){  
29.         return "HUSKY";  
30.     }  
31.  
32.     public Action getMove(CritterInfo info){  
33.         if(info.getFront() == Neighbor.OTHER){  
34.             return Action.INFECT;  
35.         }  
36.         else if(info.getLeft() == Neighbor.EMPTY || info.getRight() == Neighbor.EMPTY){  
37.             return Action.HOP;  
38.         }  
39.         else{  
40.             return Action.RIGHT;  
41.         }  
42.     }  
43. }
```

Lion.java (1668 bytes, sha256: f7901df1c173802dcc8c9228cc41f1cb)

```
1. // Xuqing Wu  
2. // 12/5/2019  
3. // CSE142  
4. // TA: Ethan M Knutson  
5. // Assignment #8  
6. //  
7. // This program will present a critter called lion that Infect  
8. //if an enemy is in front, otherwise turn left if a wall is in  
9. //front or to the right, otherwise turn right if a fellow Lion  
10. //is in front, otherwise hop. Its appearance is L, and its color  
11. //is red.  
12.  
13. import java.awt.*;  
14. import java.util.*;  
15.  
16. public class Lion extends Critter{  
17.     private Random r;
```

```

18. private int time;
19. private Color color;
20.
21. //the constructor
22. public Lion(){
23.     r = new Random();
24.     time = 0;
25. }
26.
27. //return the appearance of lion(L)
28. public String toString(){
29.     return "L";
30. }
31.
32. //return the color of lion
33. //color changes every 3 steps randomly from red, green or blue
34. public Color getColor(){
35.     if(time%3 == 0){
36.         int choice = r.nextInt(3);
37.         if(choice == 0){
38.             color = Color.RED;
39.         }
40.         else if(choice == 1){
41.             color = Color.GREEN;
42.         }
43.         else{
44.             color = Color.BLUE;
45.         }
46.     }
47.     else{
48.         color = color;
49.     }
50.     return color;
51. }
52.
53. //return the movement of lion
54. //info is the infomation about the position situation of the lion
55. public Action getMove(CritterInfo info){
56.     time++;
57.     if(info.getFront() == Neighbor.OTHER){
58.         return Action.INFECT;
59.     }
60.     else if(info.getFront() == Neighbor.WALL || info.getRight() == Neighbor.WALL){
61.         return Action.LEFT;
62.     }
63.     else if(info.getFront() == Neighbor.SAME){
64.         return Action.RIGHT;
65.     }
66.     else{
67.         return Action.HOP;
68.     }
69. }
70. }

```