The Answer of Part 2

Set 2

The source code for the BoxBug class can be found in the boxBug directory.

1. What is the role of the instance variable sideLength?

Ans: It decides the length of the side of the quare path of the bug.

2. What is the role of the instance variable steps?

Ans: It represents how many steps the bug have moved on the side.

3. Why is the turn method called twice when steps becomes equal to sideLength?

Ans: Because when bug turns once, the angle which it turns is 45 degree, so the angle is 90 degree for twice, which makes the path become a square.

4. Why can the move method be called in the BoxBug class when there is no move method in the BoxBug code?

Ans: Because the BoxBug is the class extended form then class Bug, so it can use the move method from class Bug, which is public.

5. After a BoxBug is constructed, will the size of its square pattern always be the same? Why or why not?

Ans: No, the BoxBug may be on the edge of the grid when it hasn't finish its path on the side of the sqaure.

6. Can the path a BoxBug travels ever change? Why or why not?

Ans:No, Because, the path of the BoxBug must be a fix rectangle, even the bug face some barriers.

7. When will the value of steps be zero?

Ans: When the sideLength is zero.

Exercises

In the following exercises, write a new class that extends the Bug class. Override the act method to define the new behavior.

1. Write a class CircleBug that is identical to BoxBug, except that in the act method the turn method is called once instead of twice. How is its behavior different from a BoxBug?

Ans: The angle the CircleBug turns is 45 degree, so finally the path will be like a hexagon, instead of

square.

2. Write a class SpiralBug that drops flowers in a spiral pattern. Hint: Imitate BoxBug, but adjust the side length when the bug turns. You may want to change the world to an UnboundedGrid to see the spiral pattern more clearly.

Answer: I add the side length when the bug turns, ,so the path will be like a sprial.

5. Study the code for the BoxBugRunner class. Summarize the steps you would use to add another BoxBug actor to the grid.

Answer: To add a new BoxBug, you should create the class extended from the class Bug and write the constructed function and override the act method. Then you should write the new BoxBugRunner class corresponded to your new BoxBug class.