

Activity:

A) Compile and run `BoxBug.java`. In the GUI, click "Run" to run the simulation and observe what happens.

```
cd GridWorldCode/projects/boxBug
javac -cp ../../gridworld.jar *.java
java -cp ../../gridworld.jar BoxBugRunner
```

Read: Pages 10 to 13 in the GridWorld Student Manual

Questions:

- 1) In the file `BoxBug.java`, what is the role of the instance variable `sideLength`?

- 2) What is the role of the instance variable `steps`?

- 3) Why is the `turn` method called twice when `steps` becomes equal to `sideLength`?

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

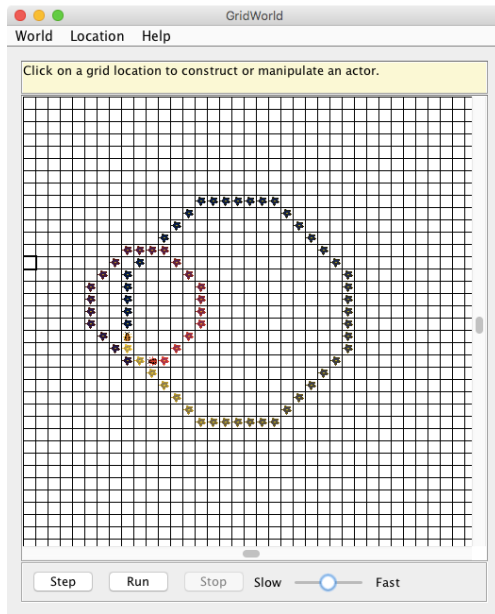
- 5) After a `BoxBug` is constructed, will the size of its square pattern always be the same? Why or why not?

- 6) Can the path a `BoxBug` travels ever change? Why or why not?

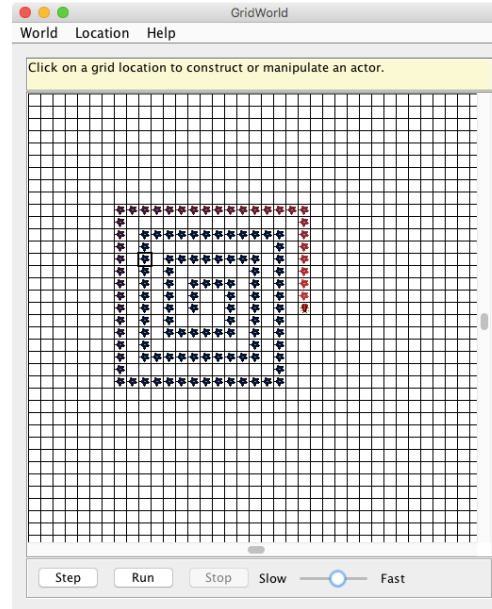
- 7) When will the value of `steps` be zero?

Activities:

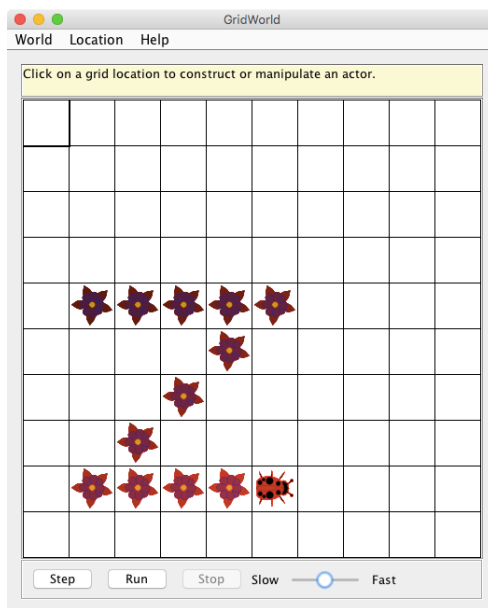
B) Write a class **CircleBug** that is identical to **BoxBug**, except that in the **act** method the **turn** method is called once instead of twice.



C) 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.



D) Write a class **ZBug** as described on page 14 of the GridWorld Student Manual.



E) Write a class **DancingBug** that “dances”. It is described on page 15 of the GridWorld Student Manual. Use the following array for your “dancing turns”.

```
private int[] turns =
    { 1, 0, 0, 0, 1, 0, 0, 3, 4,
      4, 0, 0, 1, 0, 3, 2, 0, 7,
      0, 0, 0, 3, 2, 1 };
```

