

# Fruitful methods

Methods that return values

15 September 2006



CS111 Computer Programming

Department of Computer Science  
Wellesley College

## A new twist

The screenshot shows a Scratch script editor with a script for a 'Bugle' object. The script consists of the following blocks:

- new Bugle() - green hat
- forward() - grey block
- backward() - grey block
- left() - grey block
- right() - grey block
- getPosition() - grey block
- getHeading() - grey block
- getColor() - grey block
- isBrushDown() - grey block
- isFacingWall() - grey block
- isOverBagel() - grey block
- setPosition [ (new Point( 1 , 1 ))] - grey block
- setHeading [ ( Direction.EAST )] - grey block
- setColor [ ( Color.red )] - grey block
- setDelay [ ( 0 )] - grey block
- brushDown() - grey block
- brushUp() - grey block
- dropBagel() - grey block
- pickUpBagel() - grey block

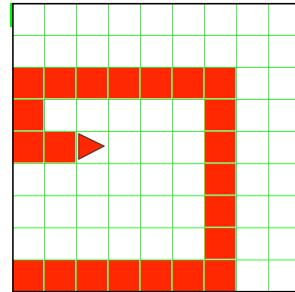
Below the script are four control buttons: Step, Run, Pause, and Reset.

Methods 4-2

## Boilerplate

```
public class SpiralWorld extends BoggleWorld
{
    public void run()
    {
        SpiralBoggle becky = new SpiralBoggle();
        becky.spiral();
        return;
    }
}

class SpiralBoggle extends Boggle
{
    public void spiral()
    {
        // our code goes here
    }
}
```

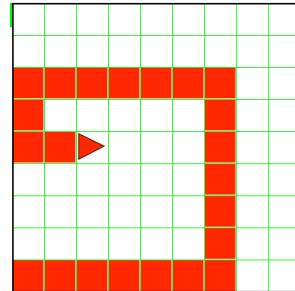


Methods 4-3

## Filling the void

```
public class SpiralWorld extends BoggleWorld
{
    public void run()
    {
        // as before
    }
}

class SpiralBoggle extends Boggle
{
    public void spiral()
    {
        this.forward(6);
        this.left();
        this.forward(6);
        this.left();
        this.forward(6);
        this.left();
        this.forward(2);
        this.left();
        this.forward(2);
        return;
    }
}
```

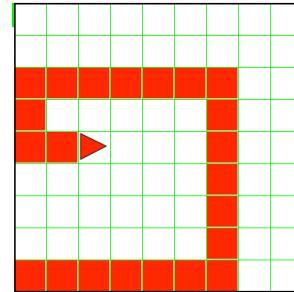


Methods 4-4

## A void method

```
/* tells the SpiralBuggle to draw a spiral */  
public void spiral()  
{  
    this.forward(6);  
    this.left();  
    this.forward(6);  
    this.left();  
    this.forward(6);  
    this.left();  
    this.forward(2);  
    this.left();  
    this.forward(2);  
    return;  
}
```

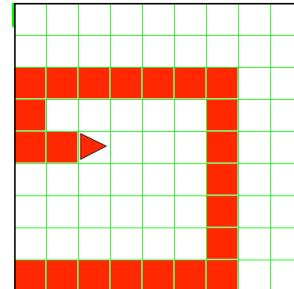
reserved word void  
says method does  
not return a value



Methods 4-5

## A fruitful method

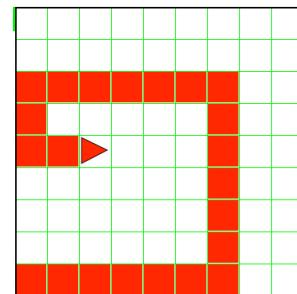
```
/* tells the SpiralBuggle to draw a spiral and  
   to return the SpiralBuggle's position after  
   completing the spiral.  
*/  
  
public _____ spiral()  
{  
    this.forward(6);  
    this.left();  
    this.forward(6);  
    this.left();  
    this.forward(6);  
    this.left();  
    this.forward(2);  
    this.left();  
    this.forward(2);  
    _____  
}
```



Methods 4-6

## A fruitful method

```
/* tells the SpiralBuggle to draw a spiral and
   to return the SpiralBuggles position after
   completing the spiral.
*/
public Point spiral()
{
    this.forward(6);
    this.left();
    this.forward(6);
    this.left();
    this.forward(6);
    this.left();
    this.forward(2);
    this.left();
    this.forward(2);
    return this.getPosition();
}
```



Methods 4-7

## A different twist

The image shows a Scratch script editor window. On the left, there is a list of blocks for a Buggie object. The blocks include: new Buggie(), forward(), backward(), left(), right(), getPosition(), getHeading(), getColor(), isBrushDown(), isFacingWall(), isOverBagel(), setPosition(new Point(1, 1)), setHeading(Direction.EAST), setColor(Color.red), setDelay(0), brushDown(), brushUp(), dropBagel(), pickUpBagel(). At the bottom of the editor are four buttons: Step, Run, Pause, and Reset. To the right of the script is a 10x10 grid where a Buggie has drawn a complex, non-spiral path. The path alternates between red and blue segments, forming a shape that looks like a stylized letter 'P' or a series of connected L-shapes.

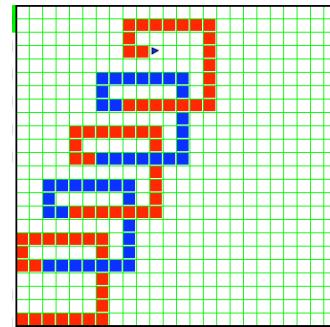
Methods 4-8

## More to the point

```
public class SpiralWorld extends BoggleWorld
{
    public void run()
    {
        SpiralBoggle becky = new SpiralBoggle();
        SpiralBoggle bobby = new SpiralBoggle();
        bobby.setColor(Color.blue);

        // our new code goes here
    }
}

class SpiralBoggle extends Boggle
{
    public Point spiral()
    {
        // code from previous page
    }
}
```

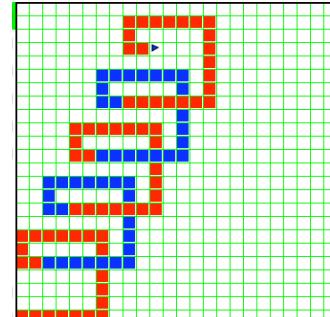


Methods 4-9

## Shall we dance

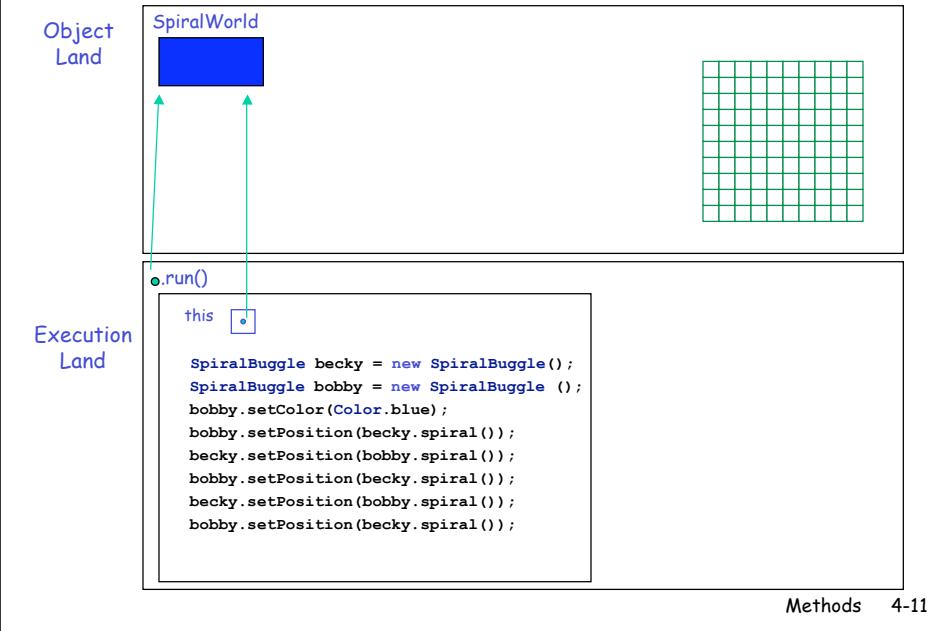
```
public class SpiralWorld extends BoggleWorld
{
    public void run()
    {
        SpiralBoggle becky = new SpiralBoggle();
        SpiralBoggle bobby = new SpiralBoggle();
        bobby.setColor(Color.blue);
        // spiral dance steps
        bobby.setPosition(becky.spiral());
        becky.setPosition(bobby.spiral());
        bobby.setPosition(becky.spiral());
        becky.setPosition(bobby.spiral());
        bobby.setPosition(becky.spiral());
    }
}

class SpiralBoggle extends Boggle
{
    public Point spiral()
    {
        // code from previous page
    }
}
```

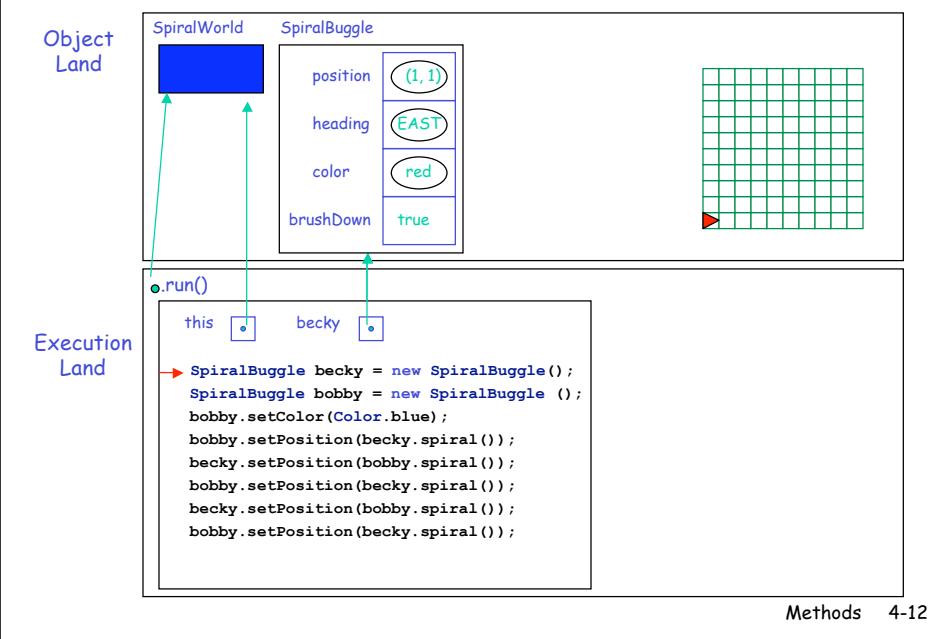


Methods 4-10

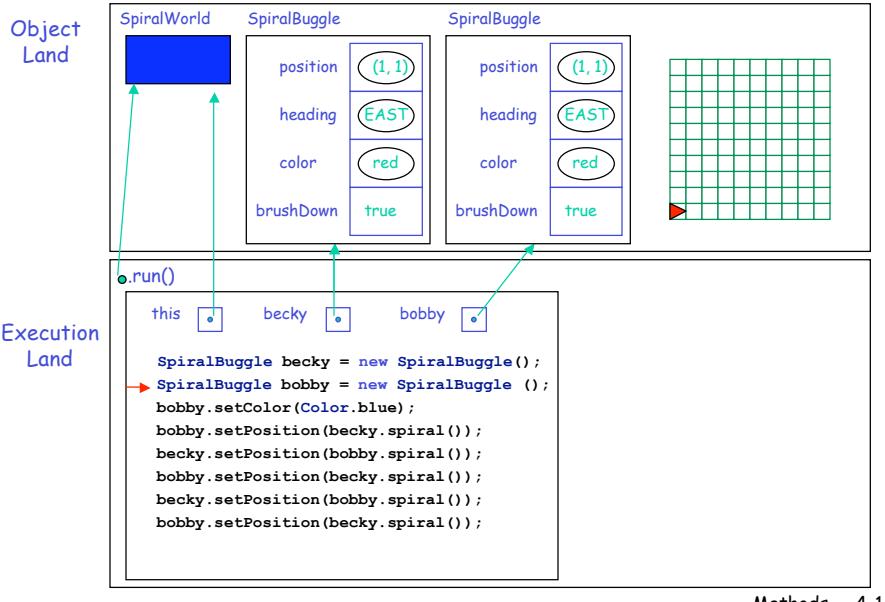
## JEM traces becky and bobby



## Enter becky

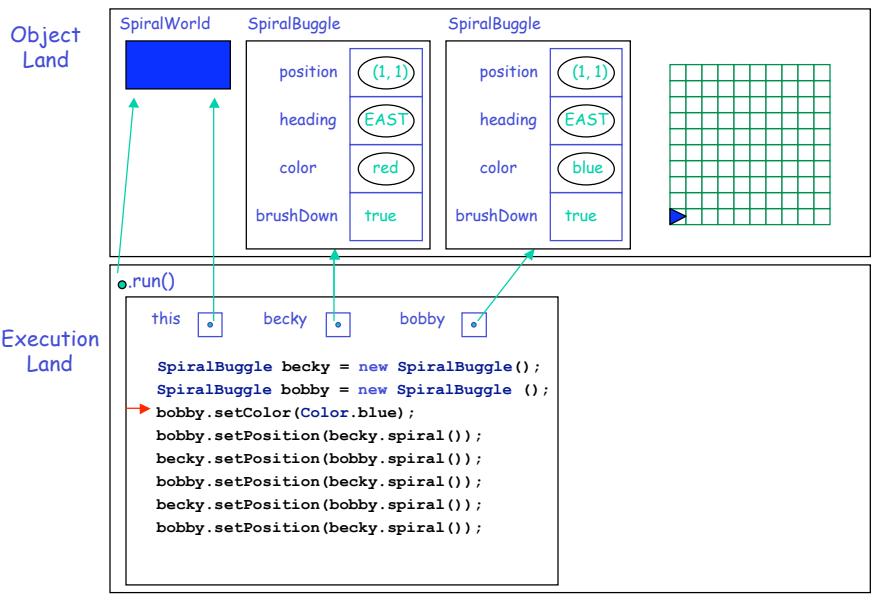


## bobby steps on becky



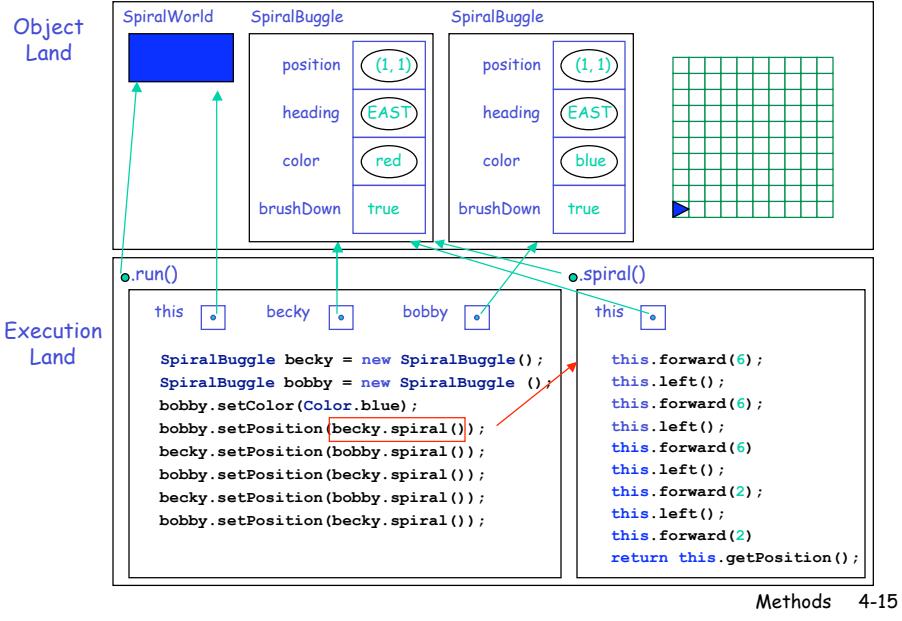
Methods 4-13

## bobby is blue

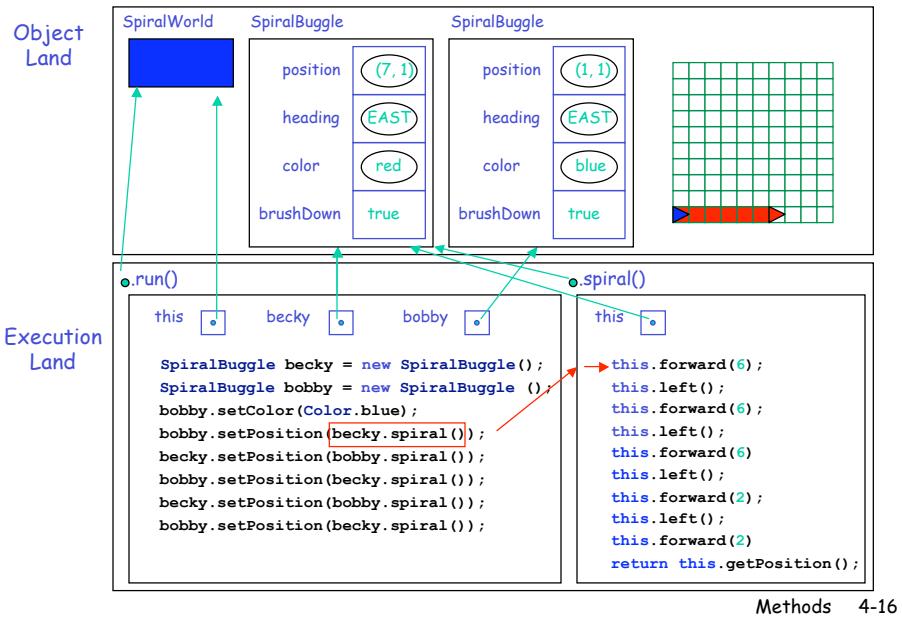


Methods 4-14

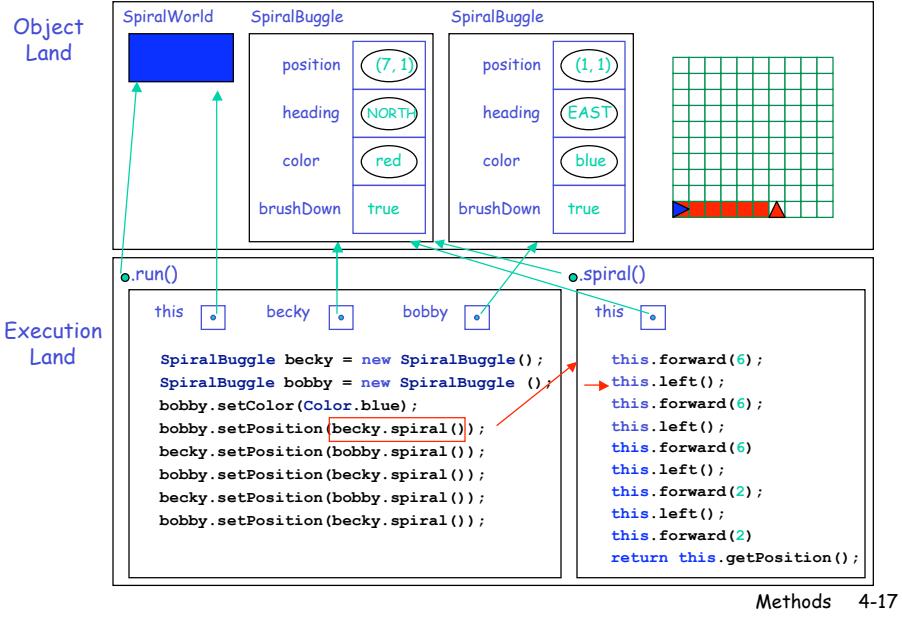
## becky spirals



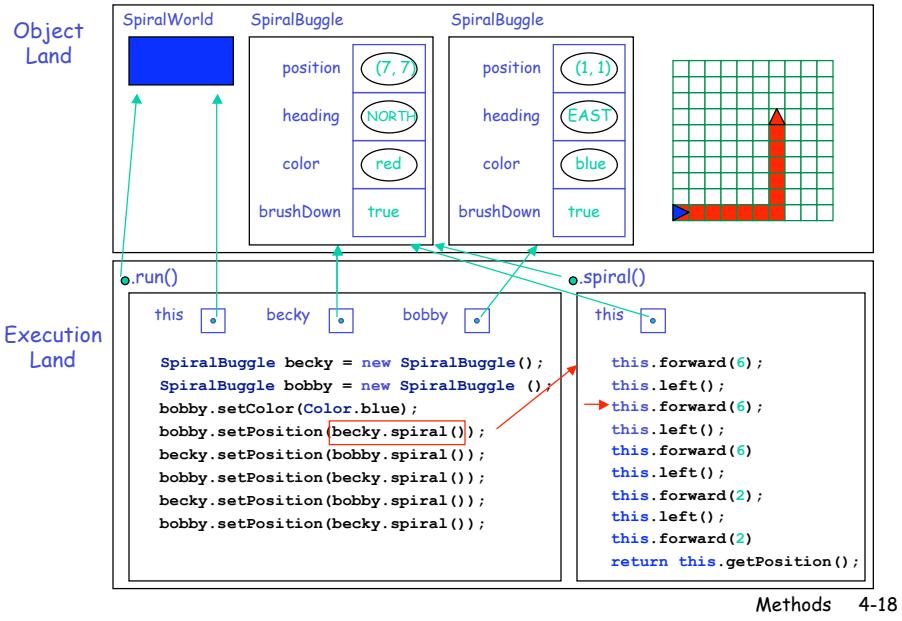
## You put your right foot in



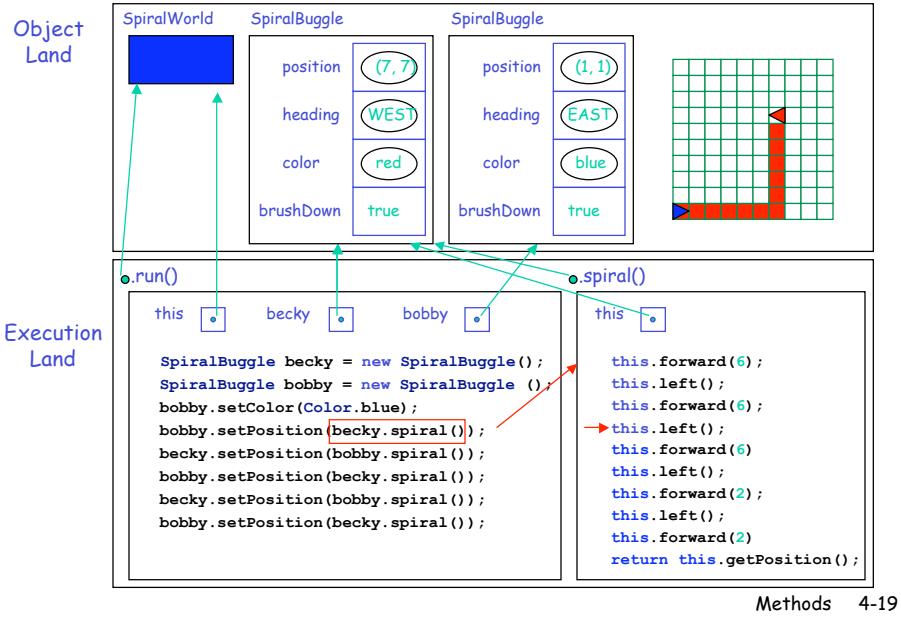
## You put your right foot out



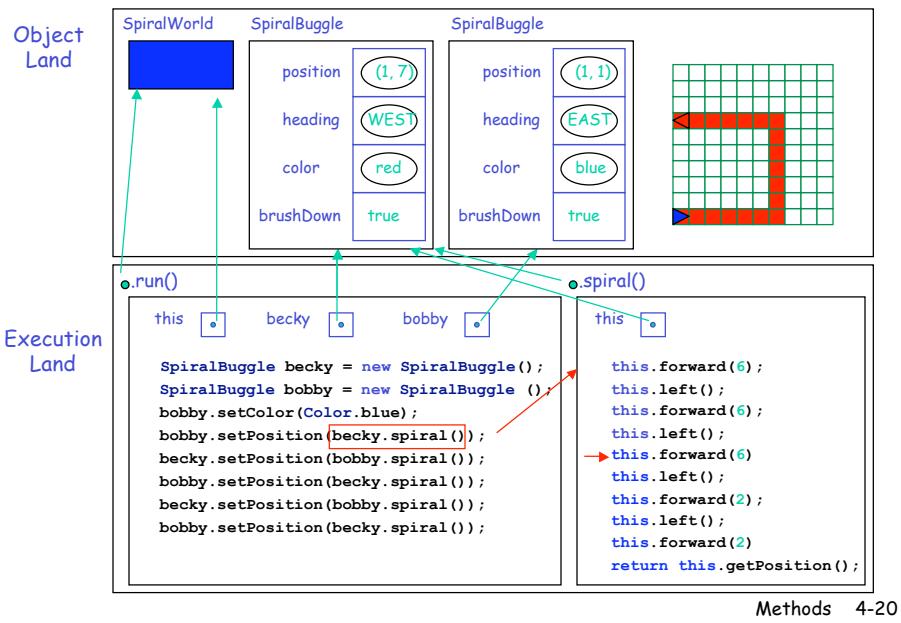
## You put your right foot in



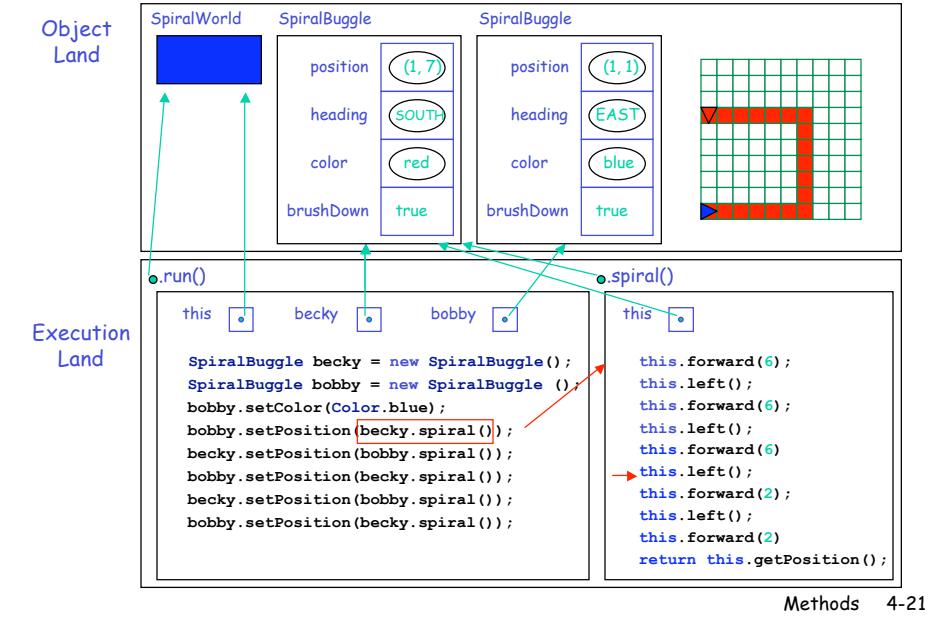
## And you shake it all about



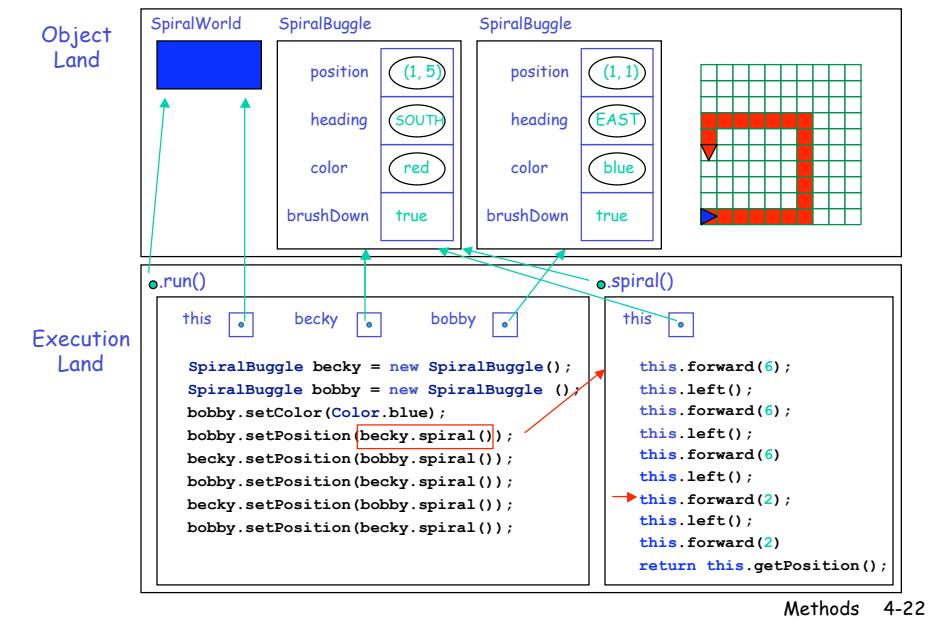
## You do the hokie pokie



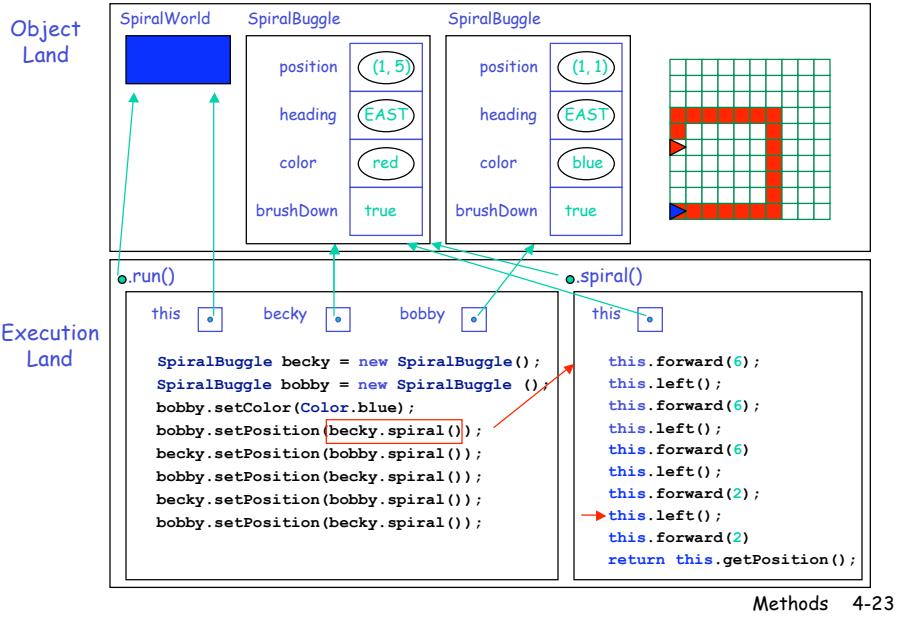
## And that's what it's all about



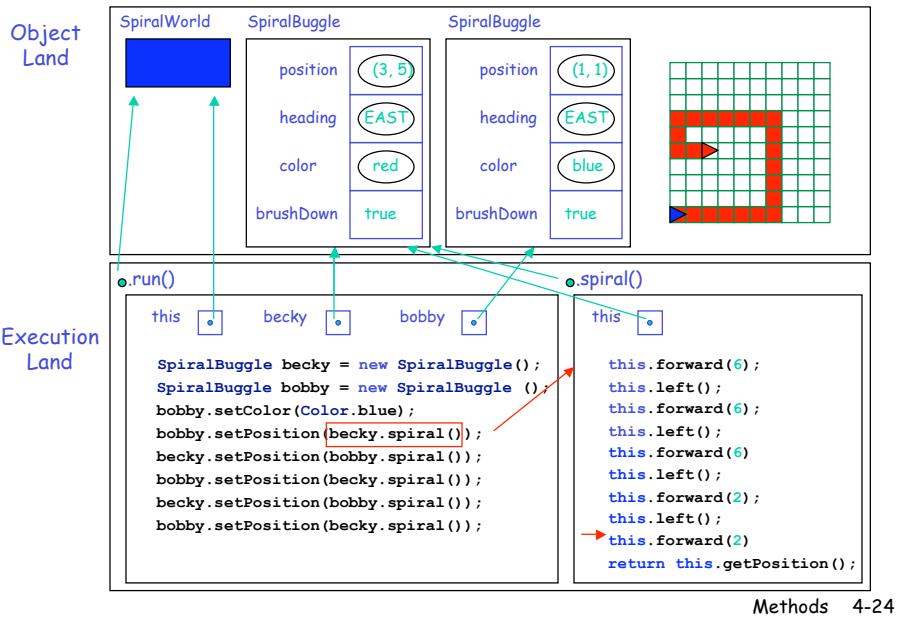
## You put your left foot in



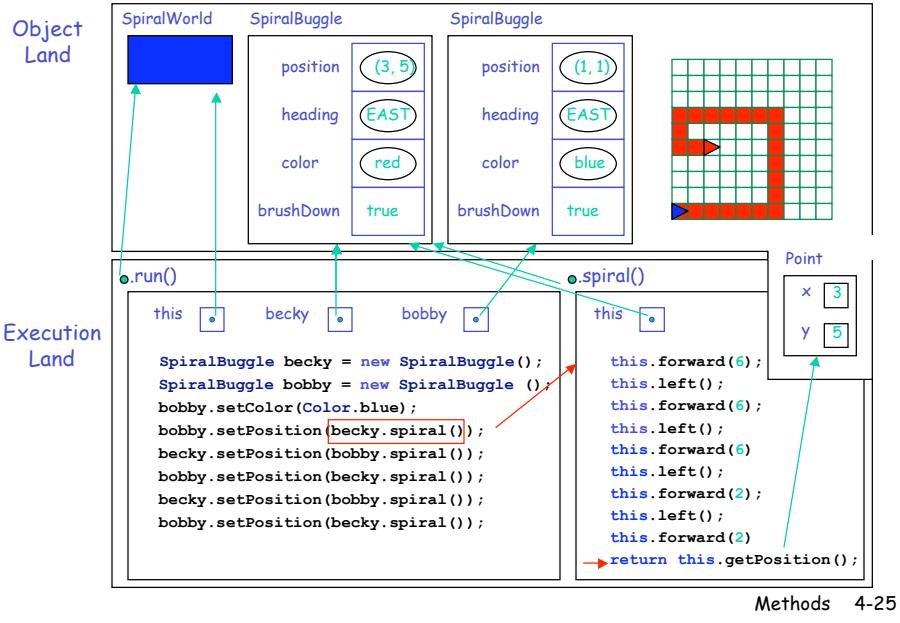
## You put your left foot out



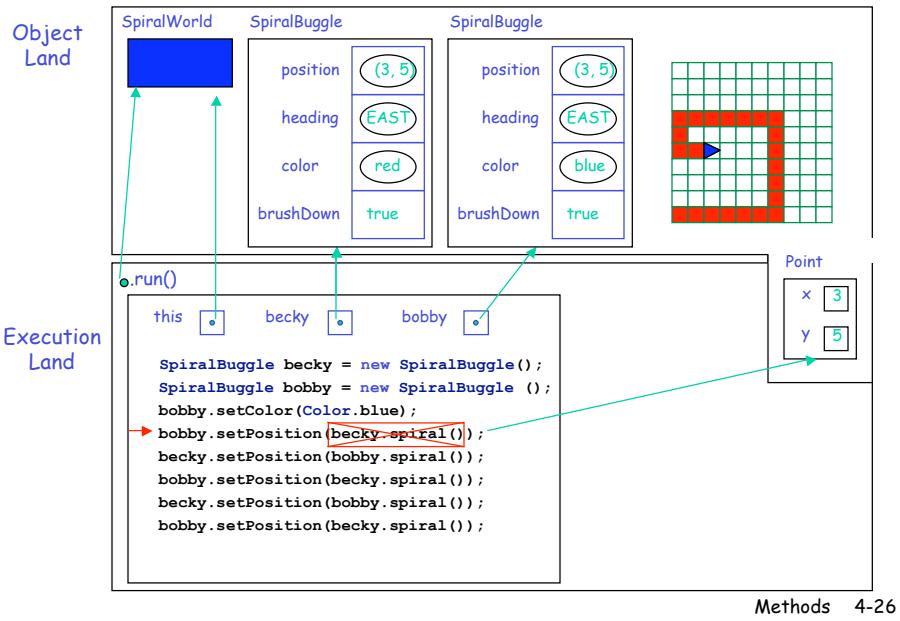
## You put your left foot in



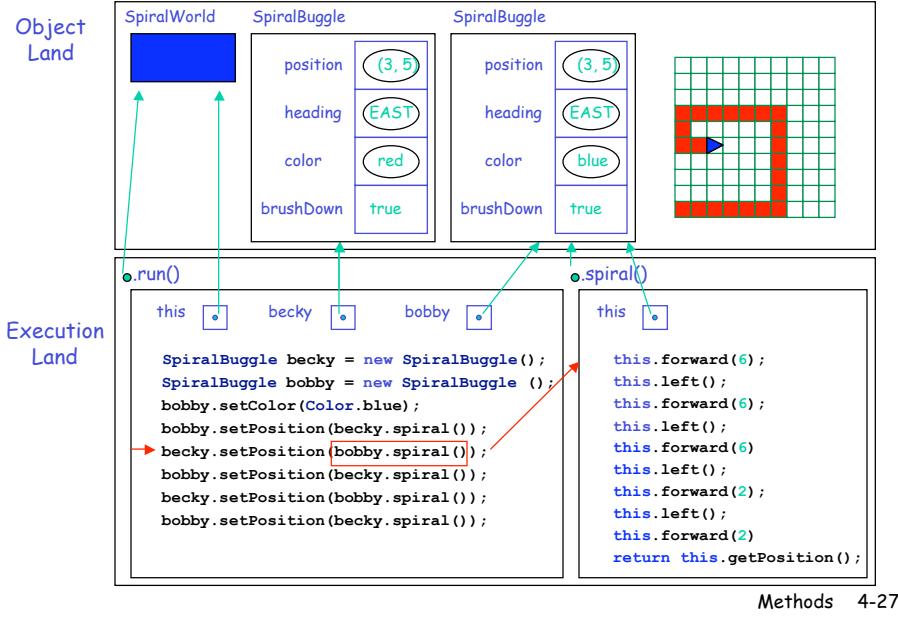
## And you shake it all about



## bobby steps on becky, again



### bobby turn to dance



### You put your left foot in

