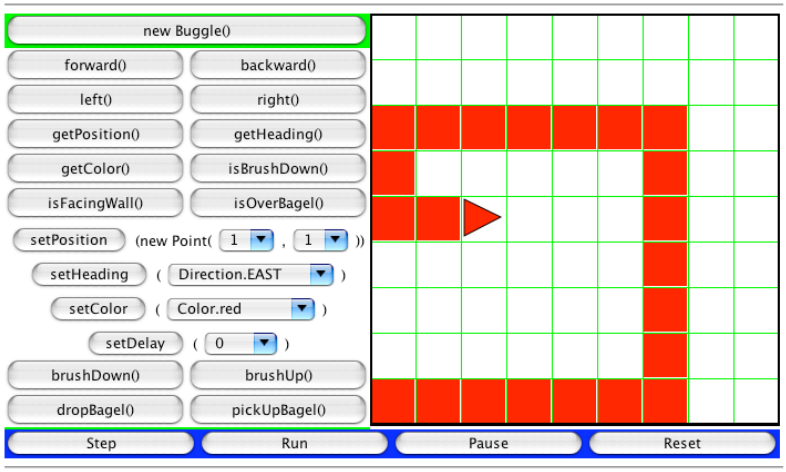
Fruitful methods

Methods that return values
15 September 2006

CS111 Computer Programming

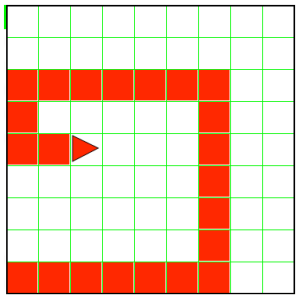
Department of Computer Science
Wellesley College

A new twist



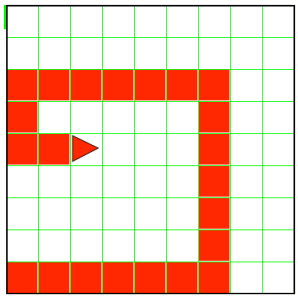
Boilerplate

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

class SpiralBuggle extends Buggle
{
     public void spiral()
     {
        // our code goes here

     }
}
```



Filling the void

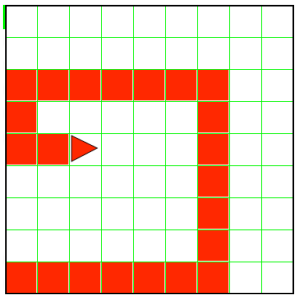
```
public class SpiralWorld extends BuggleWorld
{
     public void run()
     {
          // as before
     }
}

class SpiralBuggle extends Buggle
{
     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;
     }
}
```



A void method

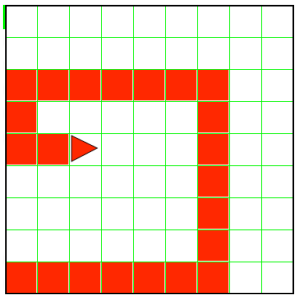
```
/* tells the SpiralBuggle to draw a spiral */

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



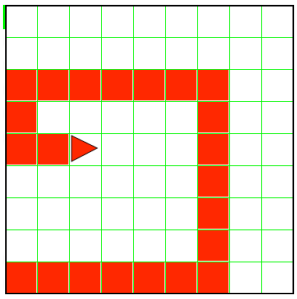
A fruitful method

```
/* 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);
      
}
```

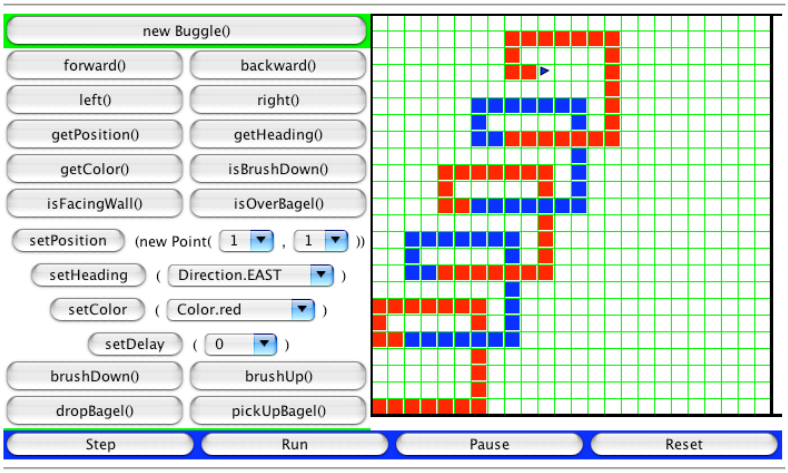


A fruitful method

```
/* 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();
}
```

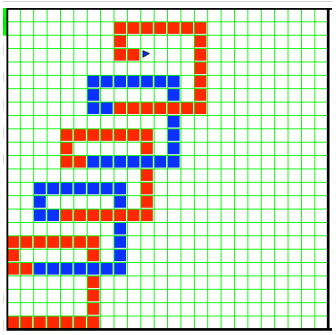


A different twist



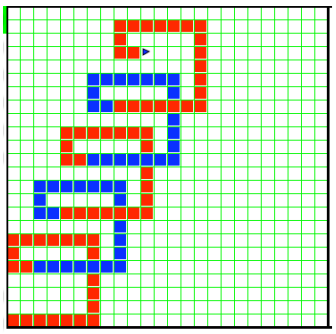
More to the point

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

          // our new code goes here

     }
}

class SpiralBuggle extends Buggle
{
     public Point spiral()
     {
          // code from previous page
     }
}
```

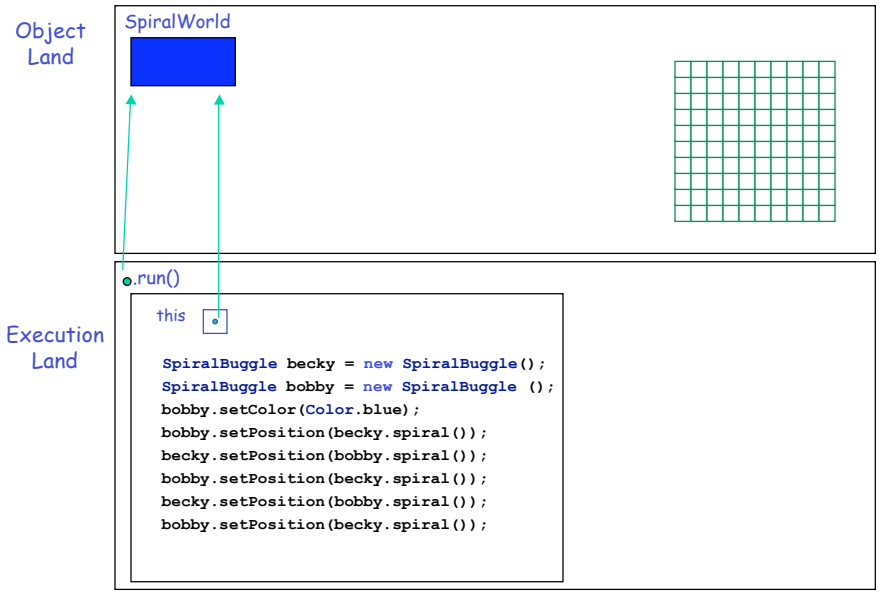


Shall we dance

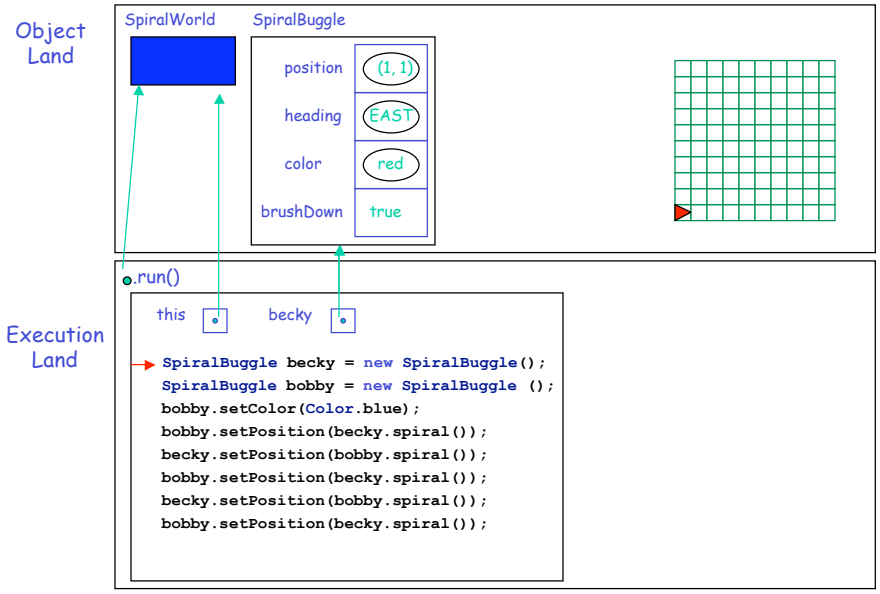
```
public class SpiralWorld extends BuggleWorld
{
     public void run()
     {
          SpiralBuggle becky = new SpiralBuggle();
          SpiralBuggle bobby = new SpiralBuggle();
          bobby.setColor(Color.blue);
          // spiral dance steps
          bobby.setPosition(becky.spiral());
  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

class SpiralBuggle extends Buggle
{
     public Point spiral()
     {
          // code from previous page
     }
}
```



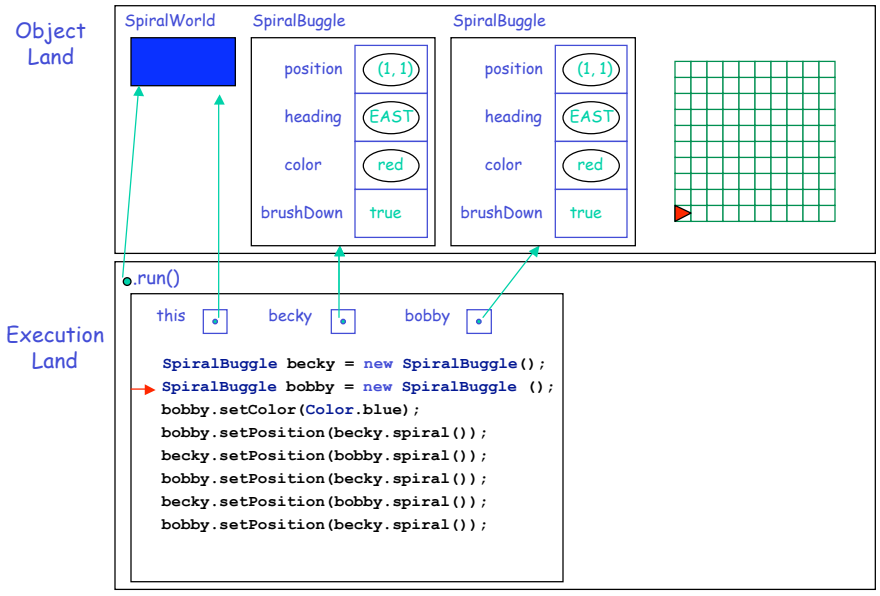
JEM traces becky and bobby



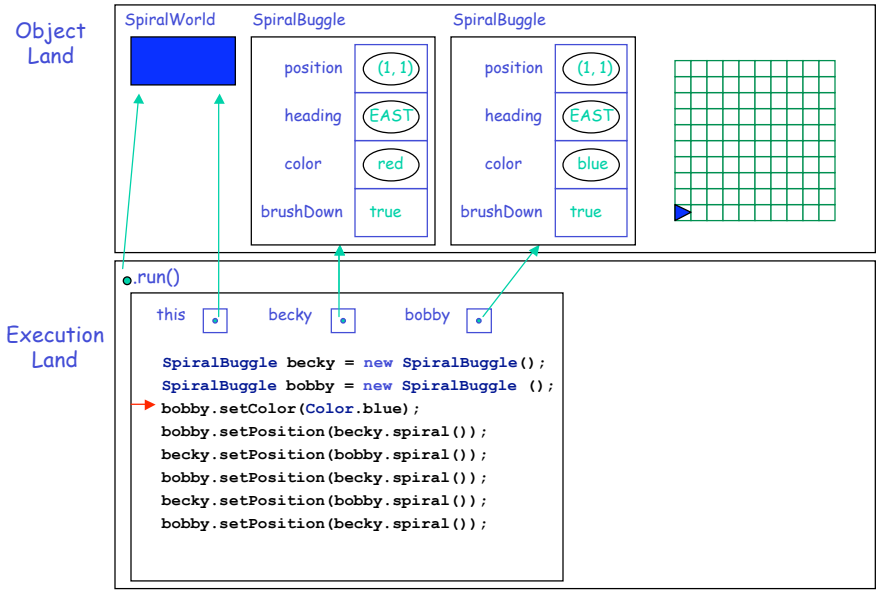
Enter becky



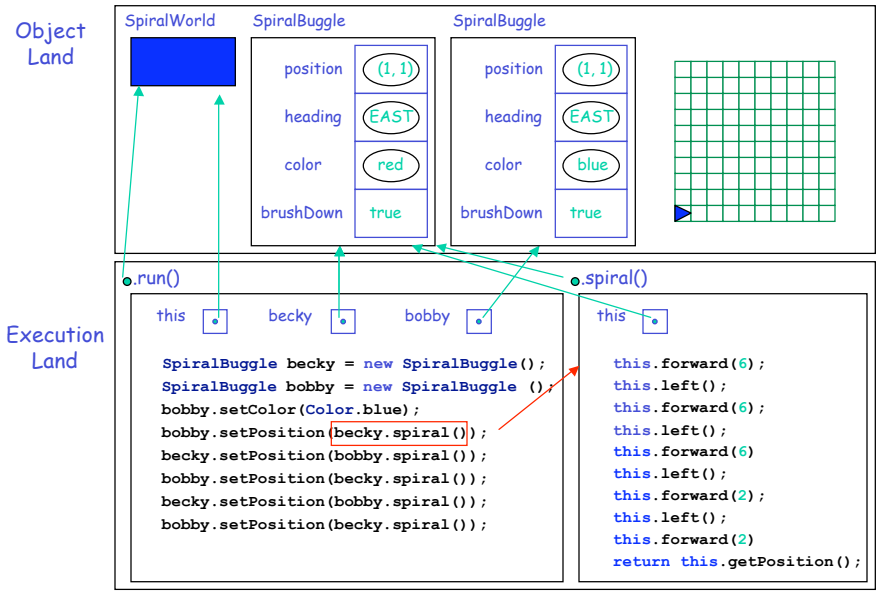
bobby steps on becky



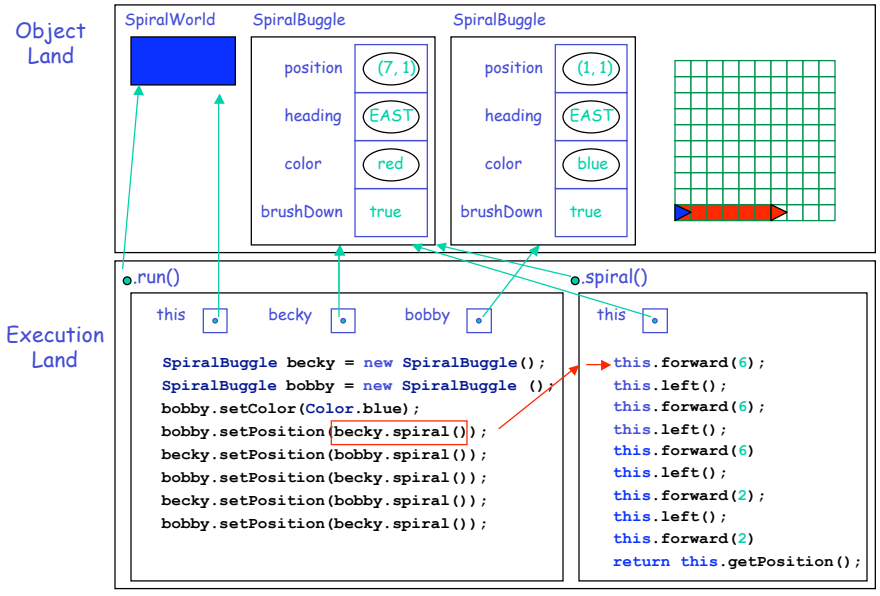
bobby is blue



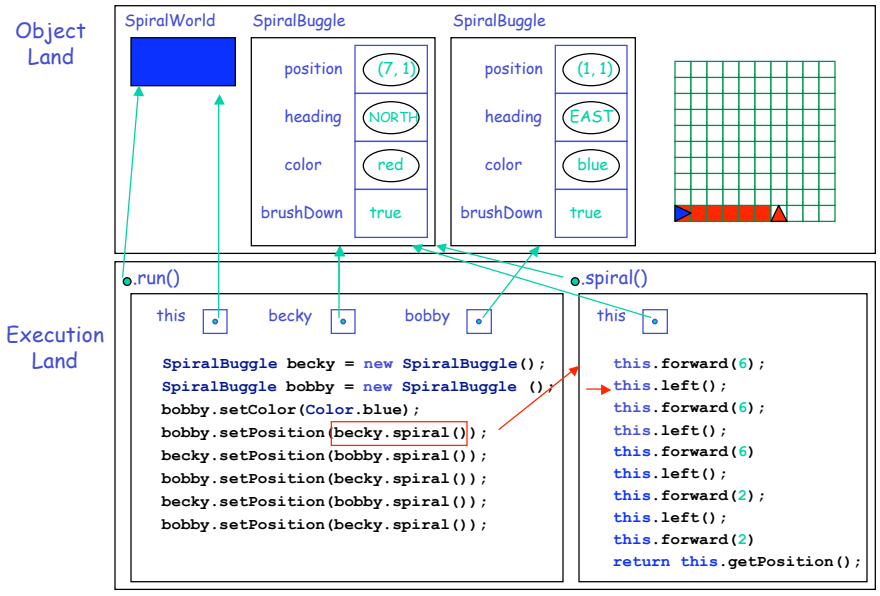
becky spirals



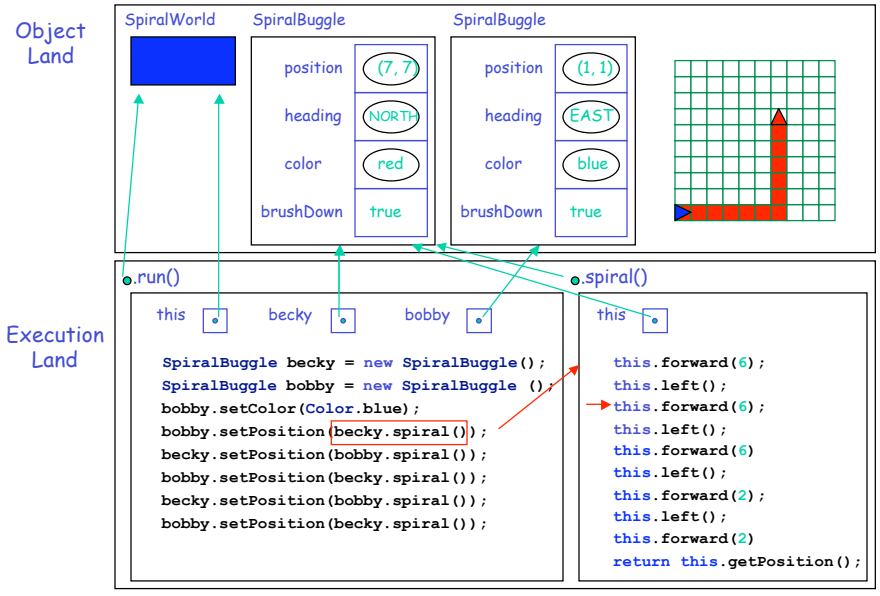
You put your right foot in



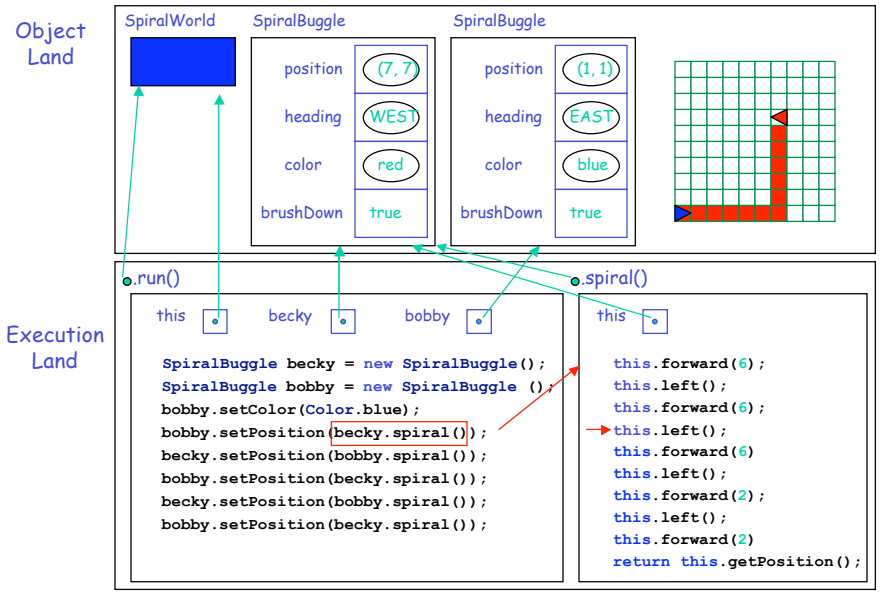
You put your right foot out



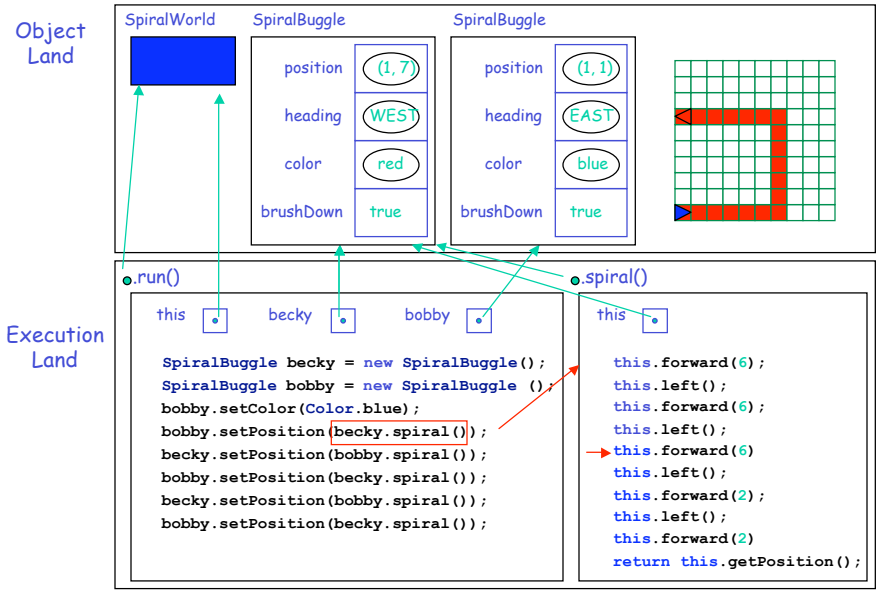
You put your right foot in



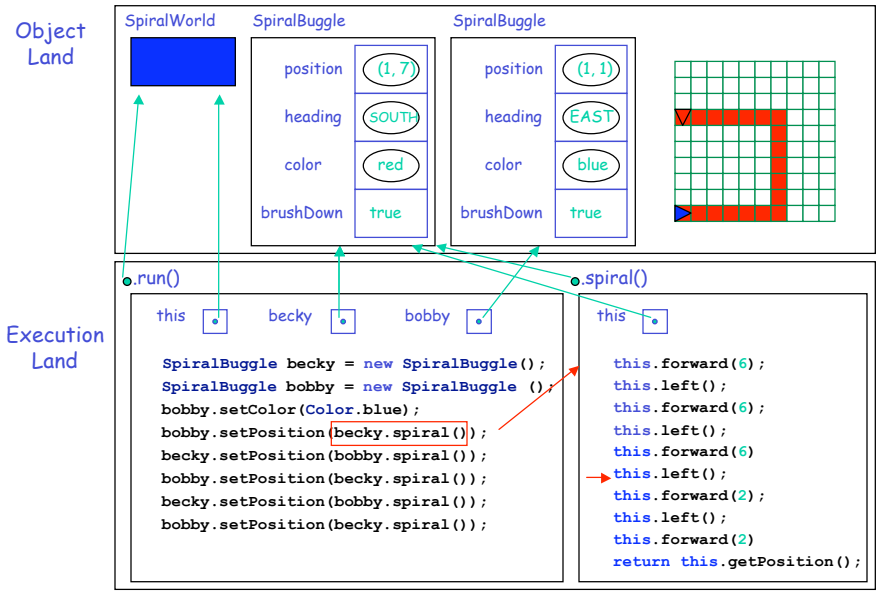
And you shake it all about



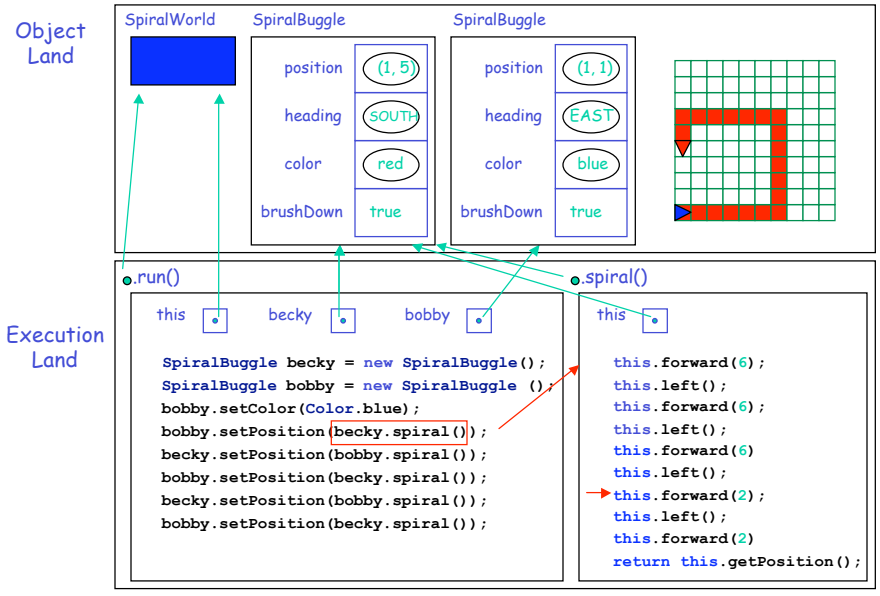
You do the the hokie pokie



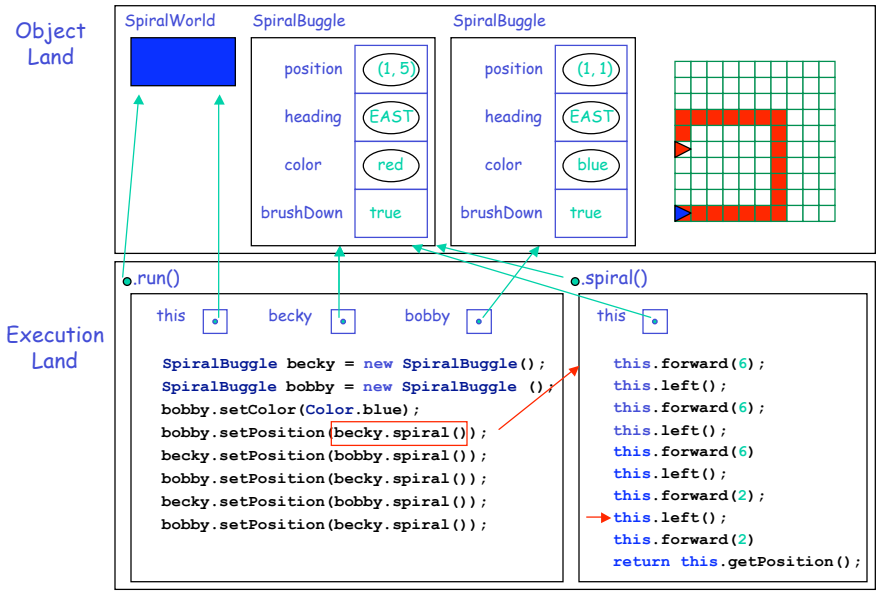
And that's what its all about



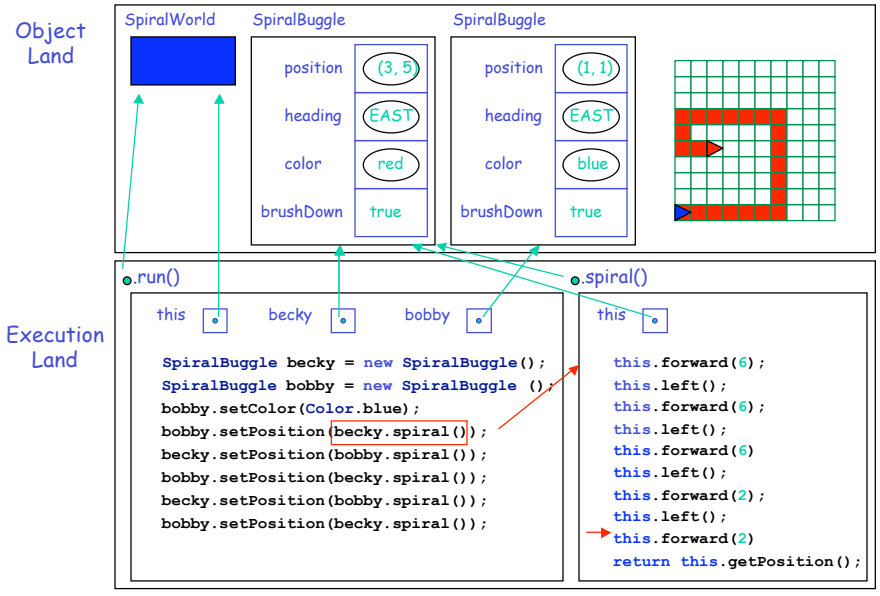
You put your left foot in



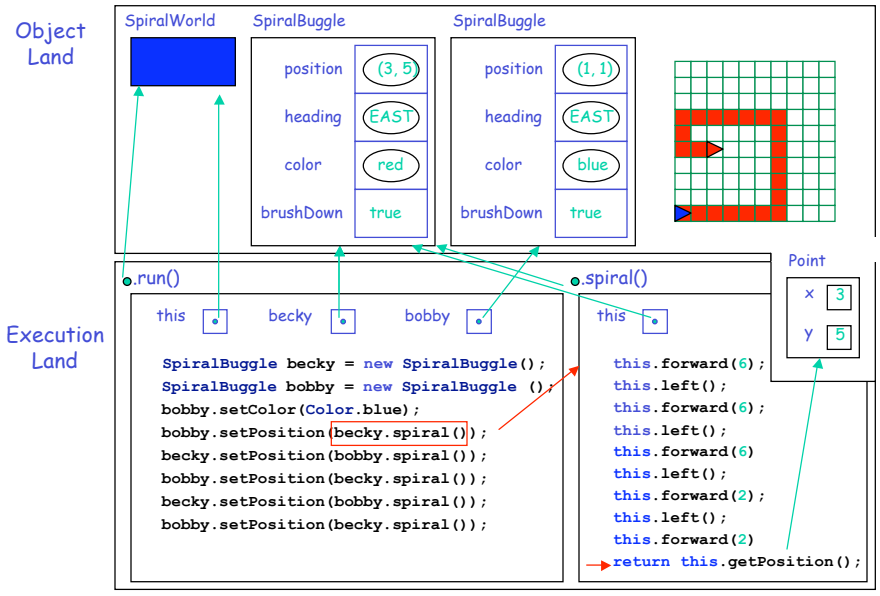
You put your left foot out



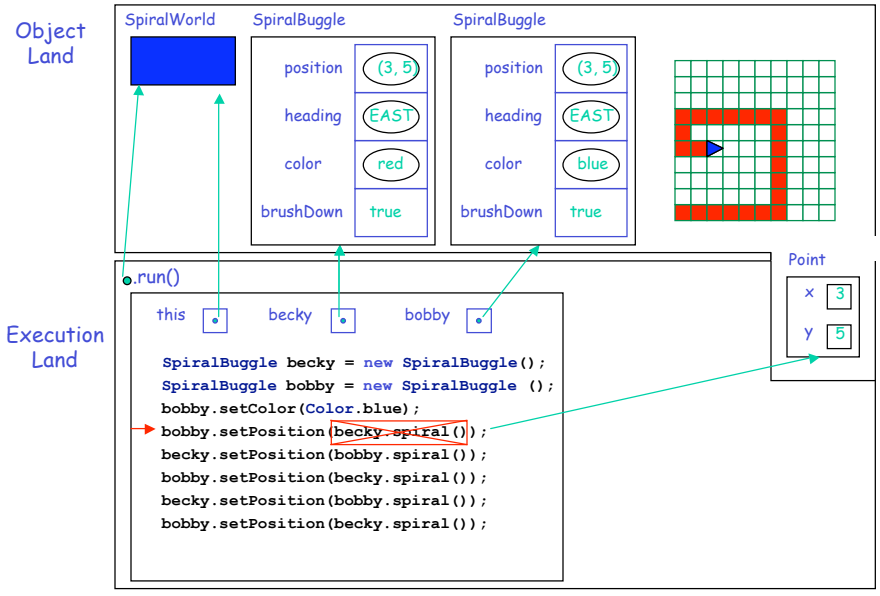
You put your left foot in



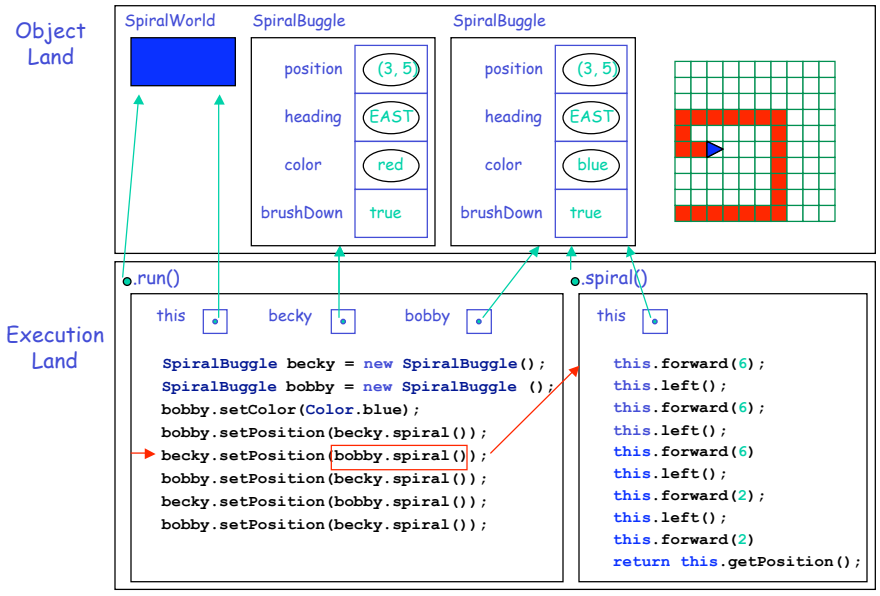
And you shake it all about



bobby steps on becky, again



bobby turn to dance



You put your left foot in