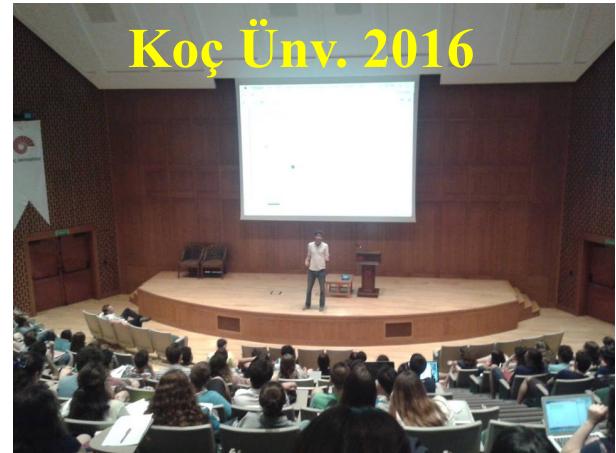




CS-Bridge

Darüşşafaka
Lisesi. 2014





Asena



Bryce



Julia



Nick



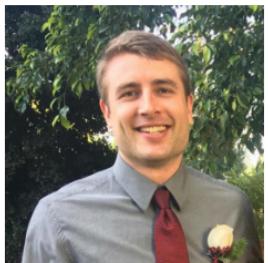
Chris



Lisa

Great team!

Teachers



Tyler



Greg



Jan



Jaroslav



Julia



Marek



Radek



Sam



Travis



Vaclav



Zoe



Bryce



Asena



Nick



Josef

Logistics

Date	July 10rd	July 11th	July 12th	July 13th
Day Num	0	1	2	3
Weekday	Tuesday	Wednesday	Thursday	Friday
9:30 am	Welcome Lecture	Internet *Nick	Graphics	Parameters and Arguments
9:45 am				
10:00 am				
10:15 am	Survey			
10:30 am	Collect Newspaper	Variables	Programming is Awesome	Mad Max
10:45 am				
11:00 am	Karluv Most (Section)			
11:15 am	*Remember survey too	Guess + Medicine (Section)	Mystery Square + String Art (Section)	Target (Section)
11:30 am				
11:45 am				
12:00 pm	Lunch	Lunch	Lunch	Lunch
12:15 pm				
12:30 pm				
12:45 pm				
1:00 pm	Control Structures	Variables	Animation	Review (Methods/ Variables)
1:15 pm				
1:30 pm				
1:45 pm				
2:00 pm	Build Karluv Most	Sandcastles/ Some Sum	Random Circles	Optical Illusion
2:15 pm				
2:30 pm				
2:45 pm				
3:00 pm	Break	Break	Break	Break
3:15 pm				
3:30 pm	Mountain Karel			
3:45 pm				
4:00 pm				
4:15 pm				
4:30 pm				
4:45 pm		Game of Nimm	Bouncing Ball	Sunset Movie
5:00 pm	Random Painter			
5:15 pm				
5:30 pm				
5:45 pm				
6:00 pm	Dinner	Dinner	Dinner	Dinner
6:15 pm				
6:30 pm				
6:45 pm				

Stanford?



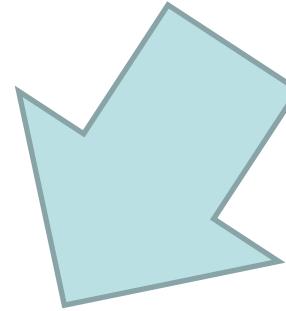
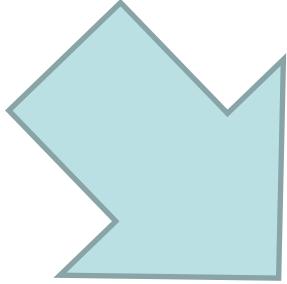
Stanford



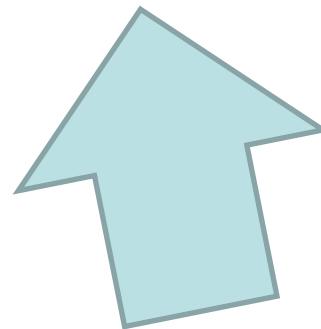
Prerequisites



Course Website

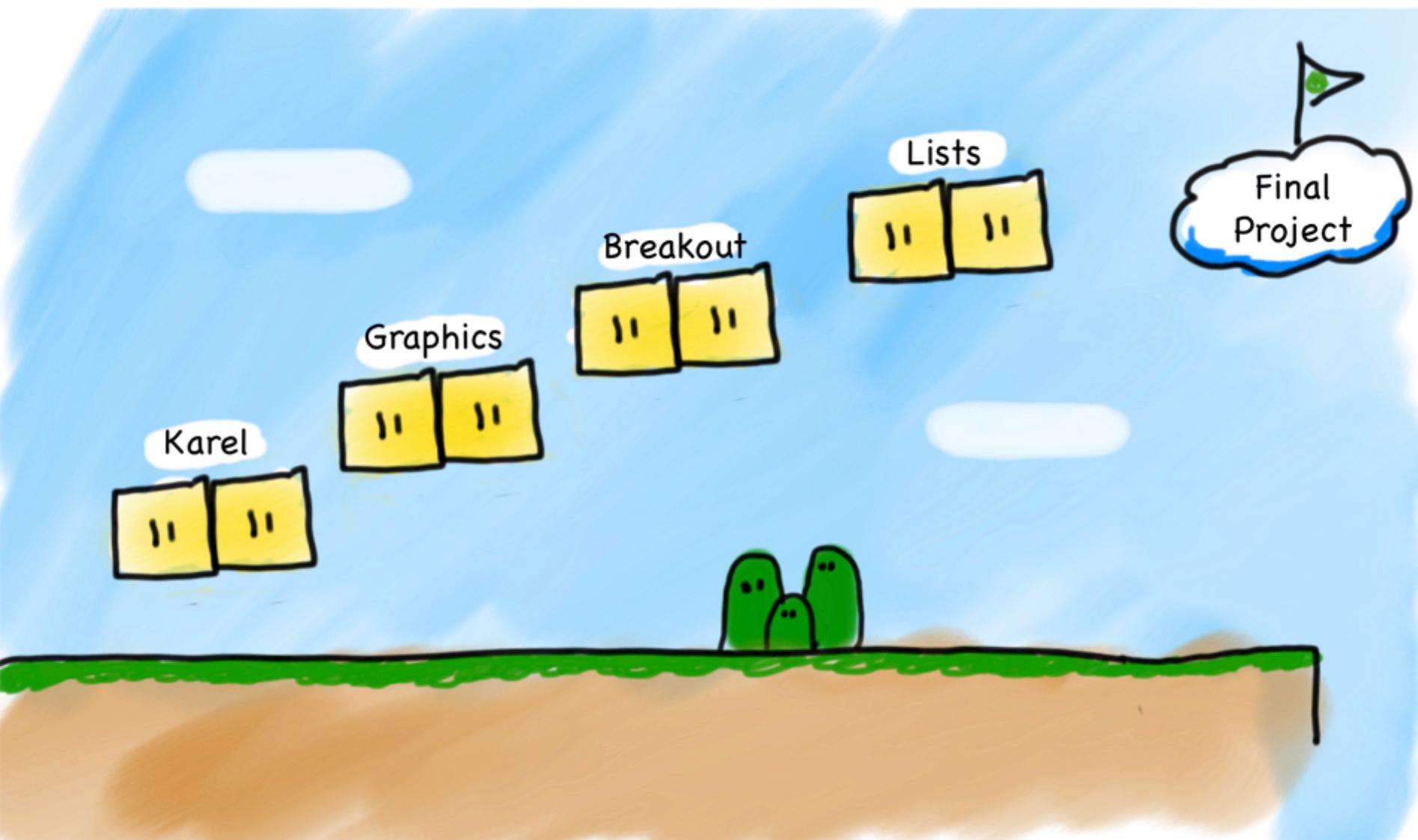


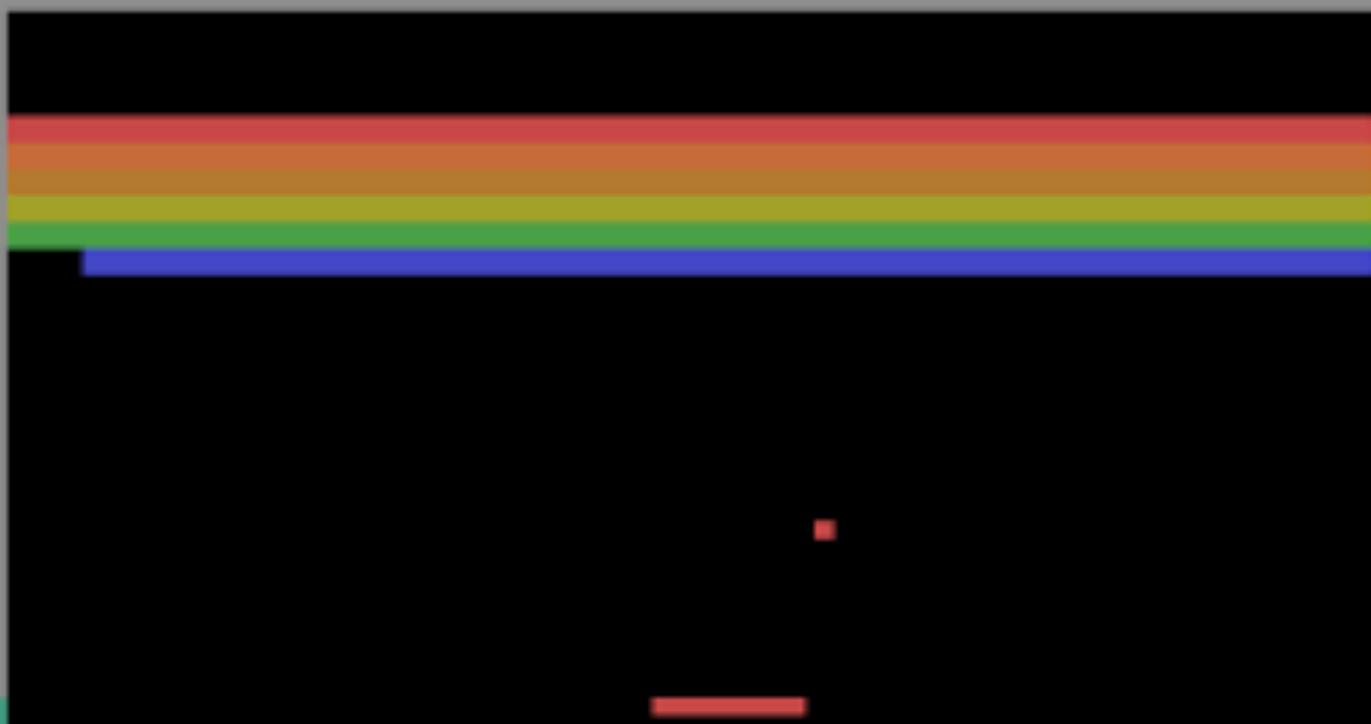
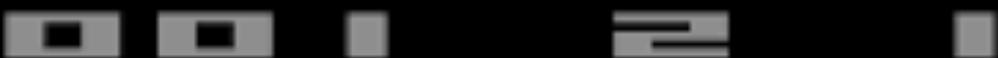
<http://ctu.csbridge.org>



*note that its **org** not **com**

Very High Level





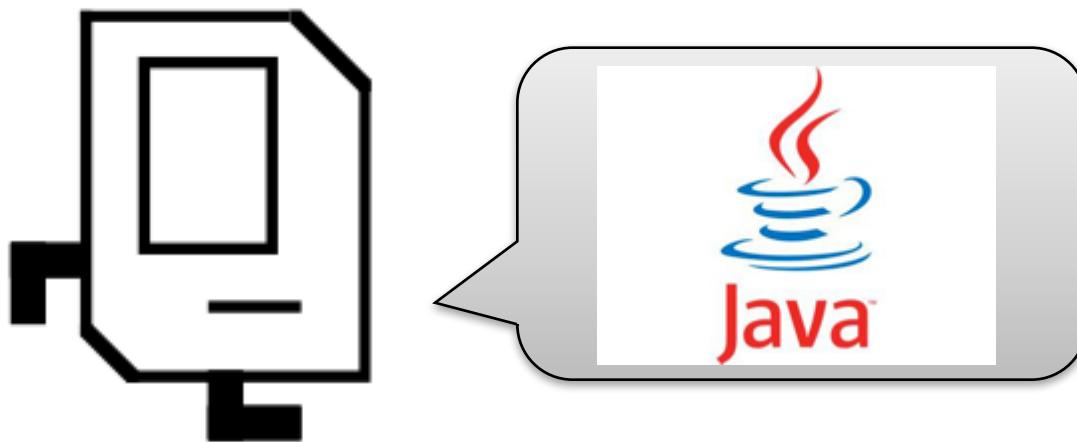
Breakout

What if I fall behind?

Share Ideas Not Code

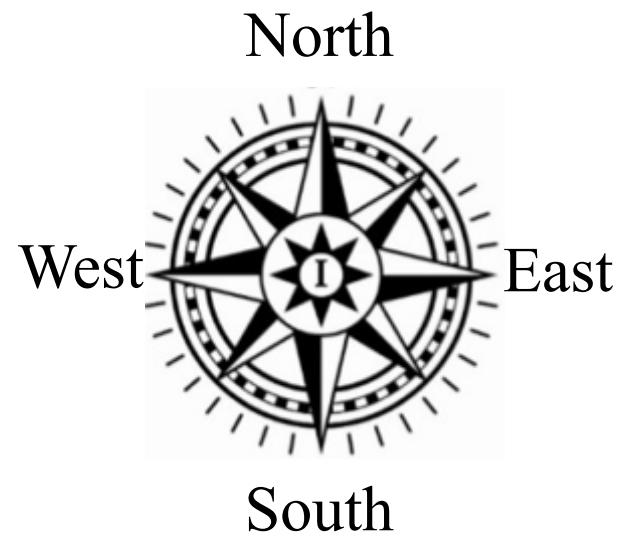


Karel Speaks Java

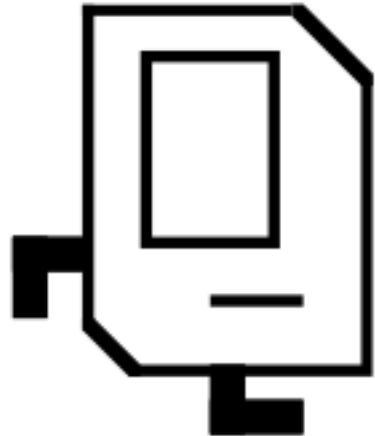


Karel's World

3	+	+	+	+	
2	+	+	+	+	
1		+	+	+	
	1	2	3	4	5



Knows Four Commands



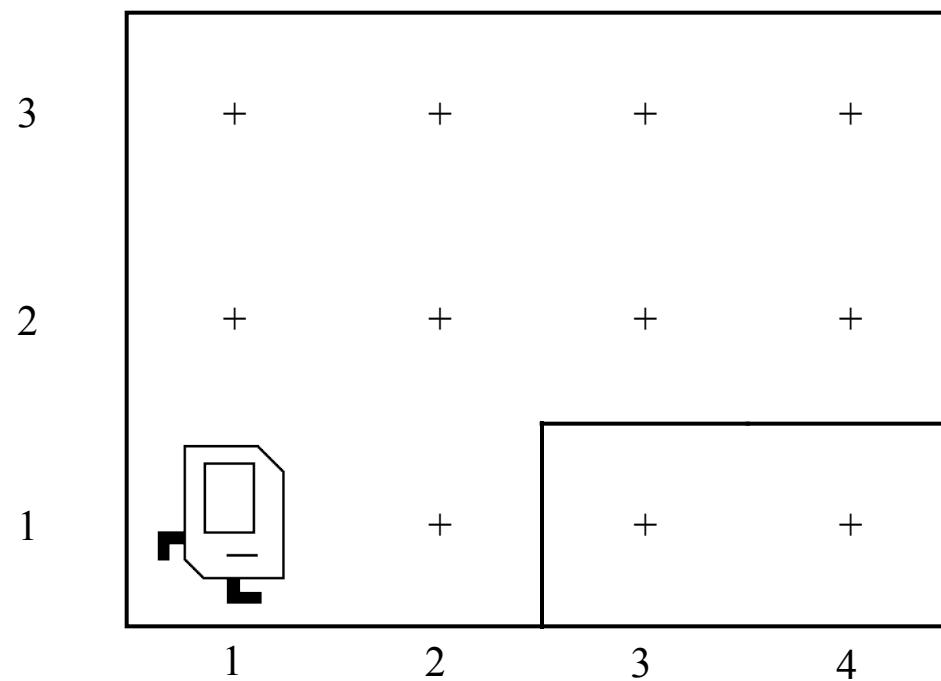
`move();`

`turnLeft();`

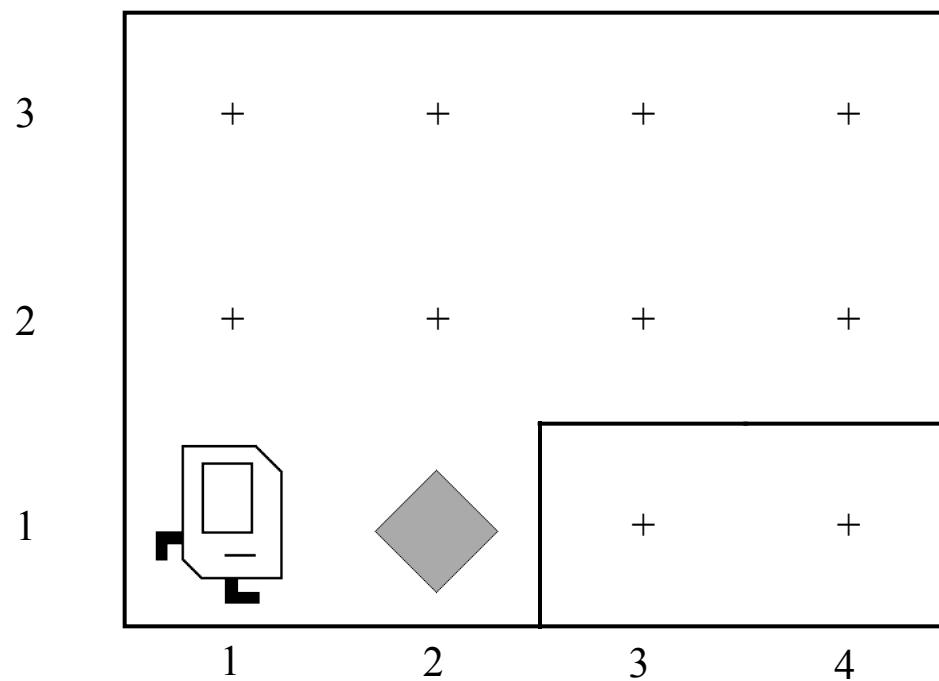
`putBeeper();`

`pickBeeper();`

Walls

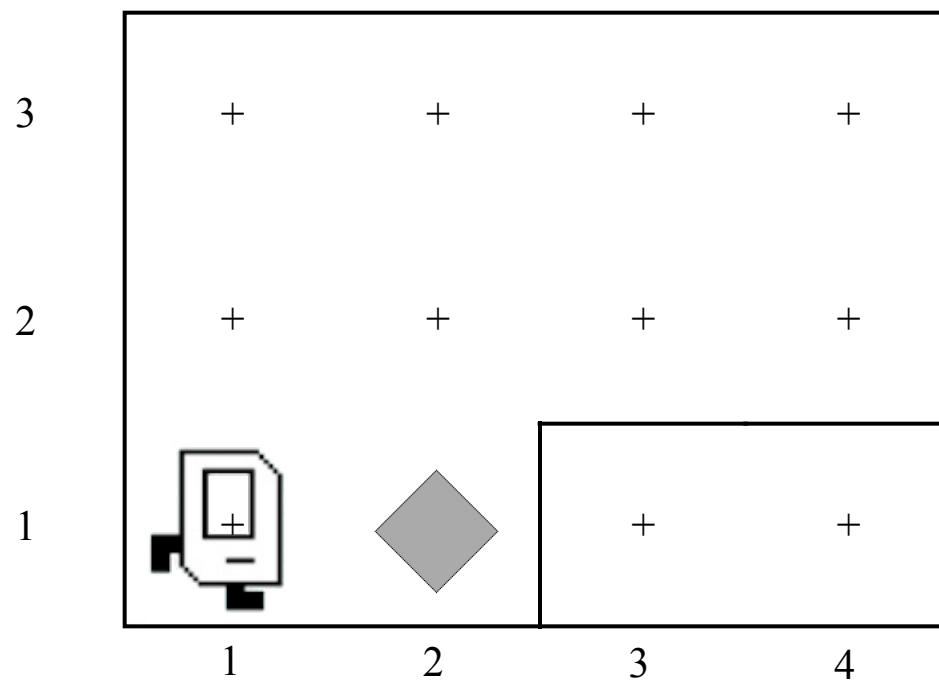


Beepers

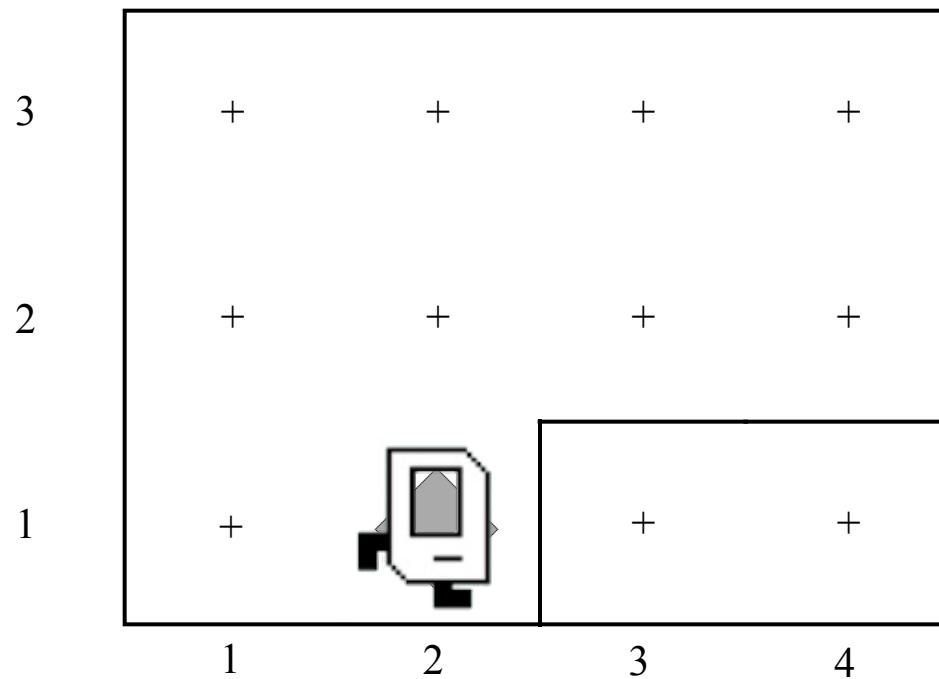


move();

move();

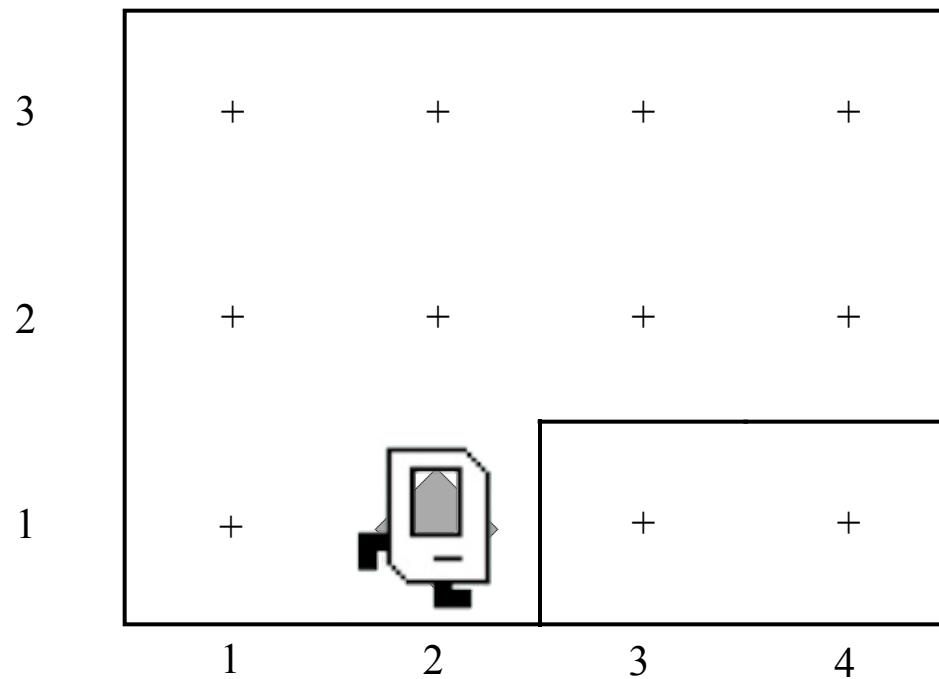


move();

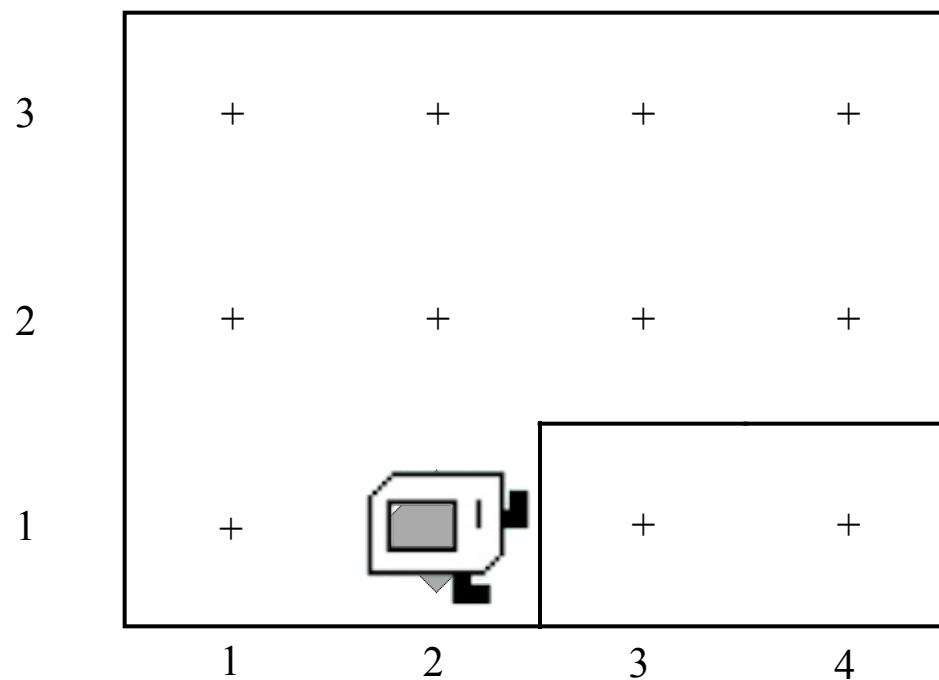


turnLeft();

turnLeft();

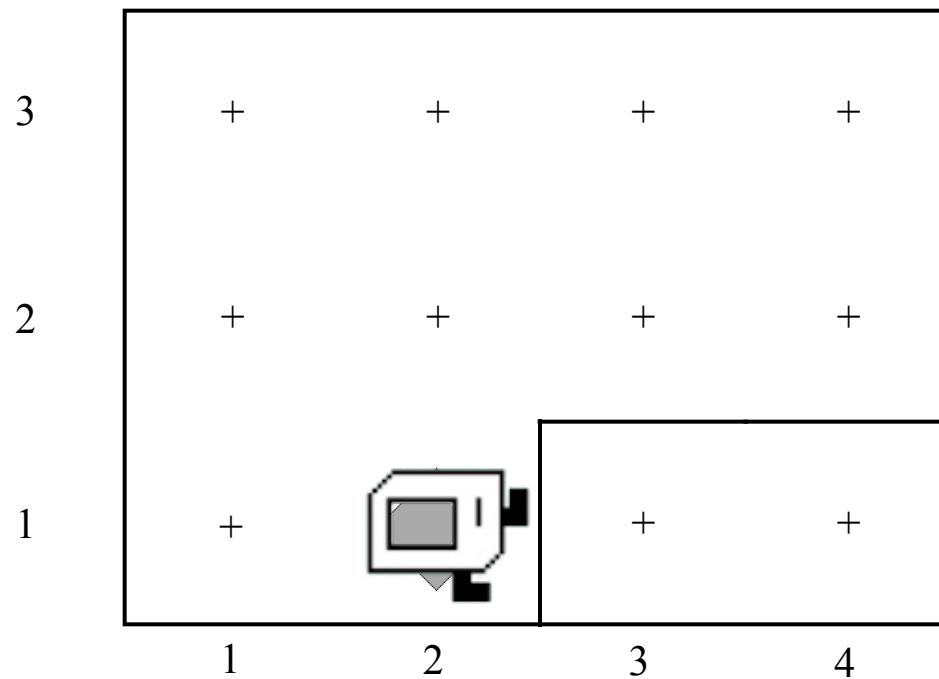


turnLeft();

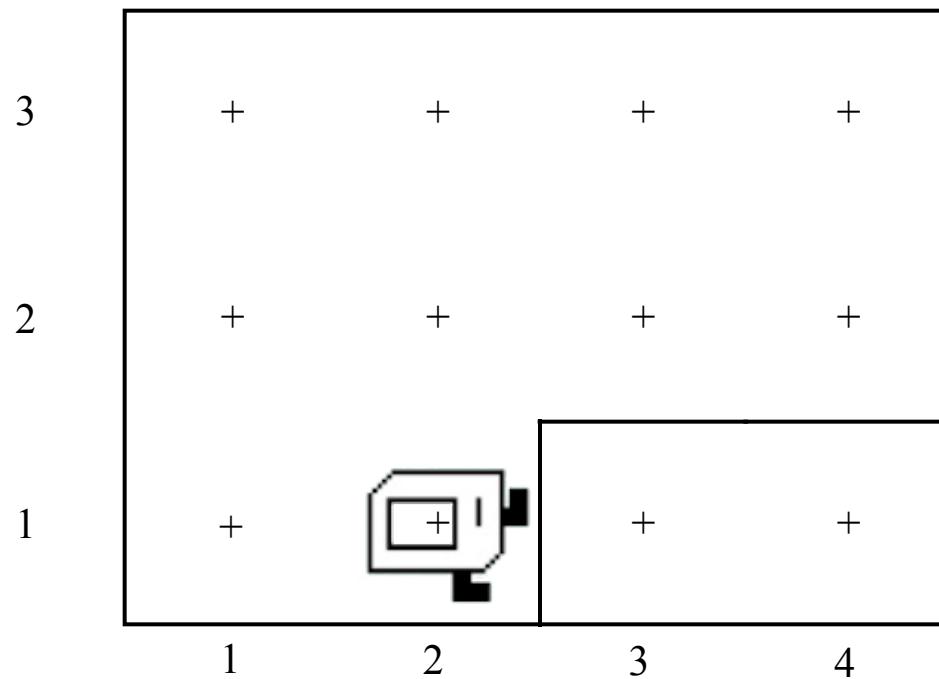


pickBeeper();

pickBeeper();

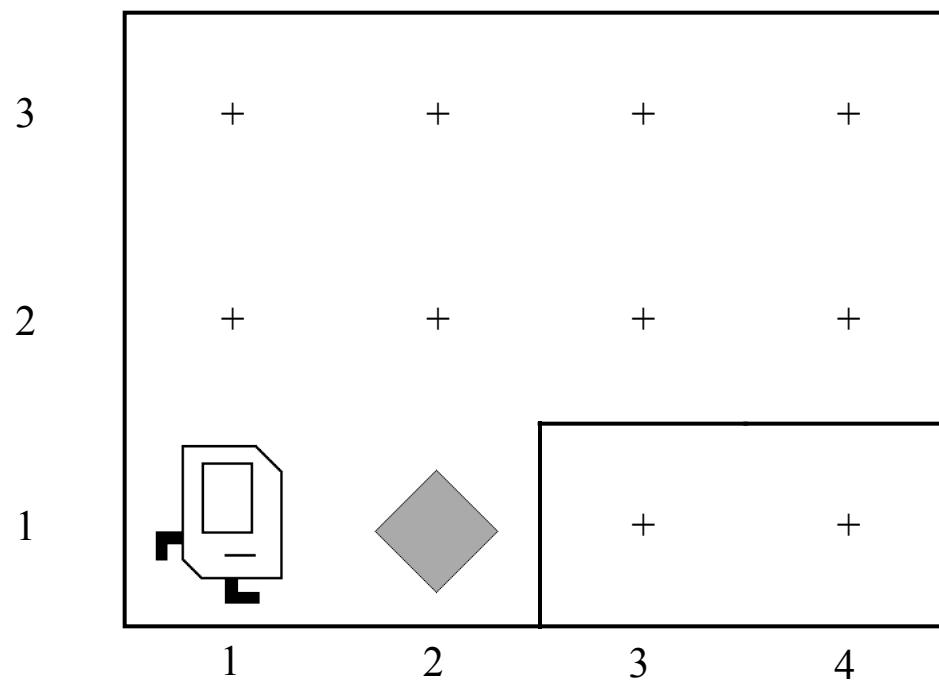


pickBeeper();

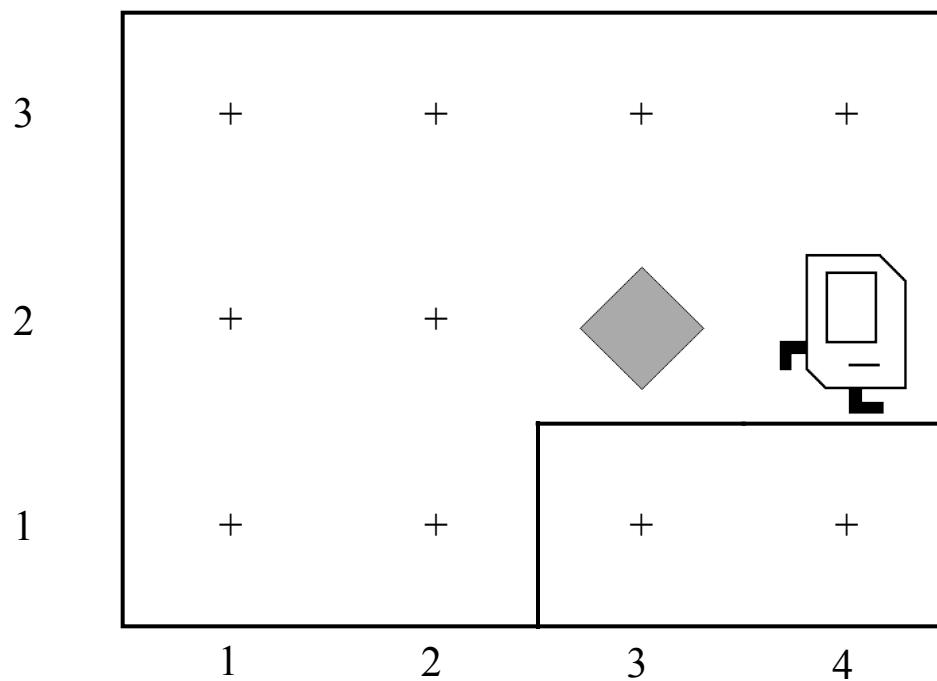


Questions?

First Challenge



First Challenge



Need a Volunteer



Lets Try It

The Eclipse logo, consisting of the word "eclipse" in a white, sans-serif font. The letter "e" is positioned to the left of a large, dark blue circle. The circle has several horizontal blue lines passing through it, creating a sense of depth or reflection. The background features concentric, blurred circles in shades of purple and pink, radiating from behind the central circle.

eclipse

Program Style

```
public void run() {  
    move();  
    pickBeeper();  
    turnLeft();  
    move();  
    turnLeft();  
    turnLeft();  
    turnLeft();  
    move();  
    putBeeper();  
    move();  
}
```

```
public void run() {  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
    turnLeft();  
    move();  
    turnLeft();  
    turnLeft();  
    turnLeft();  
    move();  
    move();  
    pickBeeper();  
    turnLeft();  
    turnLeft();  
    move();  
    move();  
    turnLeft();  
    turnLeft();  
    turnLeft();  
    move();  
    turnLeft();  
    turnLeft();  
    turnLeft();  
    move();  
    turnLeft();  
    move();  
    putBeeper();  
    turnLeft();  
    move();  
}
```

Questions?

Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {
    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```

Anatomy of a Program

```
import stanford.karel.*;  
  
public class OurKarelProgram extends Karel {  
    public void run() {  
        move();  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
  
    private void turnRight() {  
        turnLeft();  
        turnLeft();  
        turnLeft();  
    }  
}
```

This is the program's
source code

Anatomy of a Program

```
import stanford.karel.*;  
  
public class OurKarelProgram extends Karel {  
    public void run() {  
        move();  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
  
    private void turnRight() {  
        turnLeft();  
        turnLeft();  
        turnLeft();  
    }  
}
```

This piece of the program's **source code** is called a **method**.

Anatomy of a Program

```
import stanford.karel.*;  
  
public class OurKarelProgram extends Karel {  
    public void run() {  
        move();  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
}
```

```
private void turnRight() {  
    turnLeft();  
    turnLeft();  
    turnLeft();  
}
```

This line of code gives the
name of the method
(here, run)

Anatomy of a Program

```
import stanford.karel.*;  
  
public class OurKarelProgram extends Karel {  
    public void run() {  
        move();  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
  
    private void turnRight() {  
        turnLeft();  
        turnLeft();  
        turnLeft();  
    }  
}
```

This line of code gives the
name of the method
(here, turnRight)

Anatomy of a Program

```
import stanford.karel.*;  
  
public class OurKarelProgram extends Karel {  
    public void run() {  
        move();  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
  
    private void turnRight() {  
        turnLeft();  
        turnLeft();  
        turnLeft();  
    }  
}
```

This is called an ***import statement***. It tells Java what Karel is.

Anatomy of a Program

```
import stanford.karel.*;

public class OurKarelProgram extends Karel {
    public void run() {
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
    }

    private void turnRight() {
        turnLeft();
        turnLeft();
        turnLeft();
    }
}
```

This is called a
code block

Method Definition

```
private void name() {  
    statements in the method body  
}
```

This adds a new
command to Karel's
vocabulary

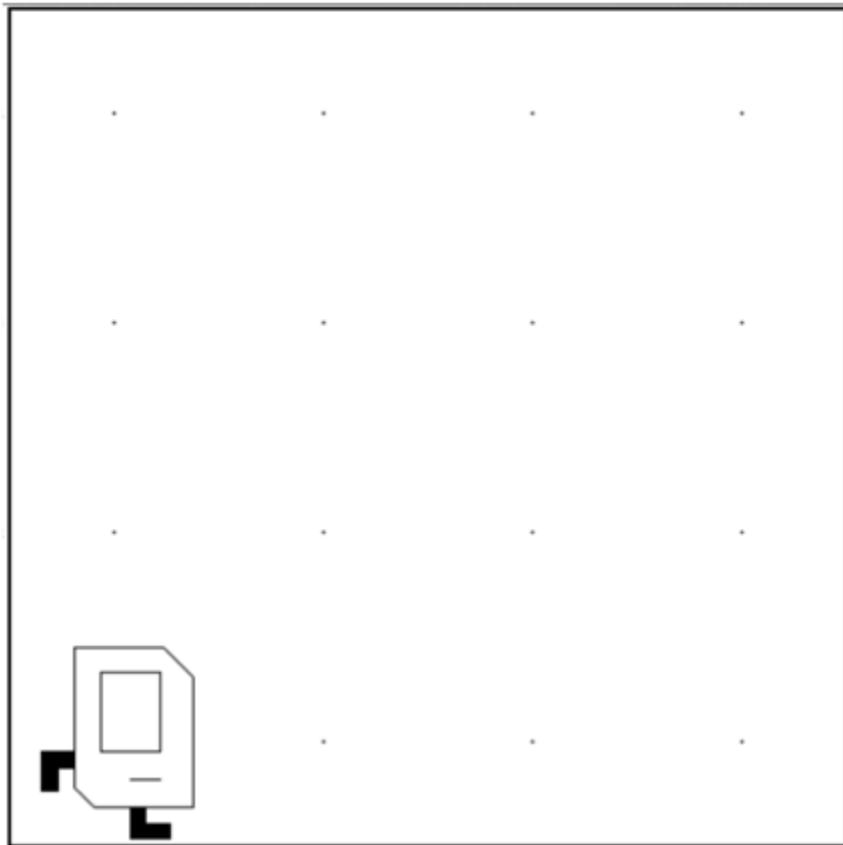
Program Style

```
import stanford.karel.*;
```

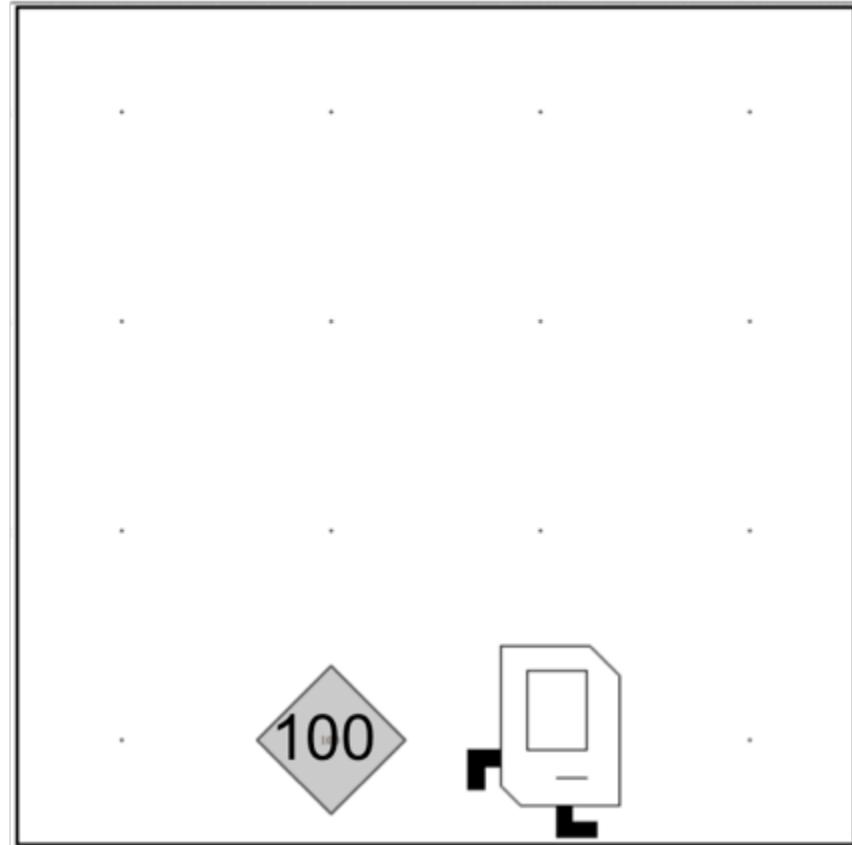
```
public class OurKarelProgram extends Karel {  
    public void run() {  
        move();  
  
        pickBeeper();  
        move();  
        turnLeft();  
        move();  
        turnRight();  
        move();  
        putBeeper();  
        move();  
    }  
  
    private void turnRight() {  
        turnLeft();  
        turnLeft();  
    }  
}
```

Place 100 beeper?

Before



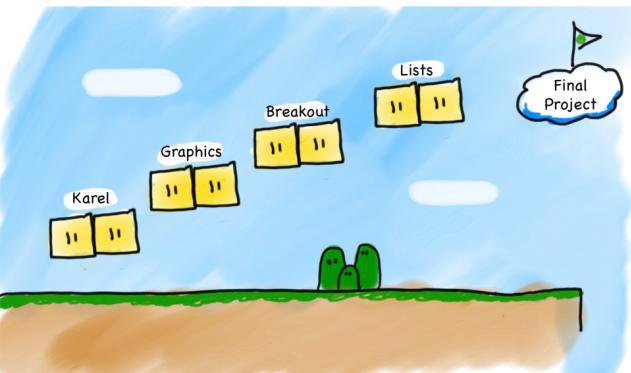
After



For Loop

```
for(int i = 0; i < N; i++) {  
    // to repeat N times  
}
```

Review



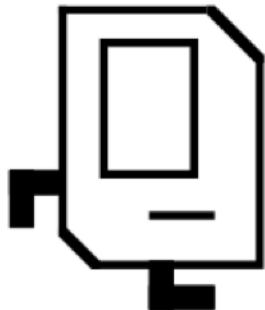
```
for(int i = 0; i < N; i++) {  
    // to repeat N times  
}
```

move();

turnLeft();

putBeeper();

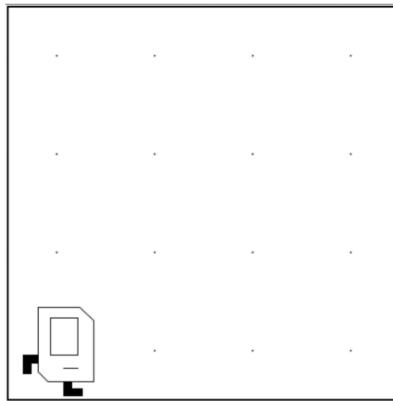
pickBeeper();



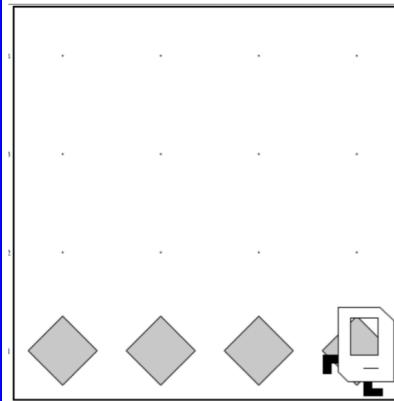
```
private void name() {  
    statements in the method body  
}
```

Work in Any World

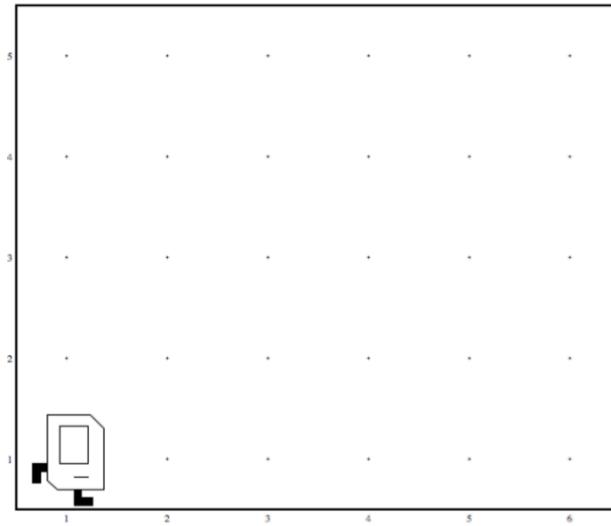
Before



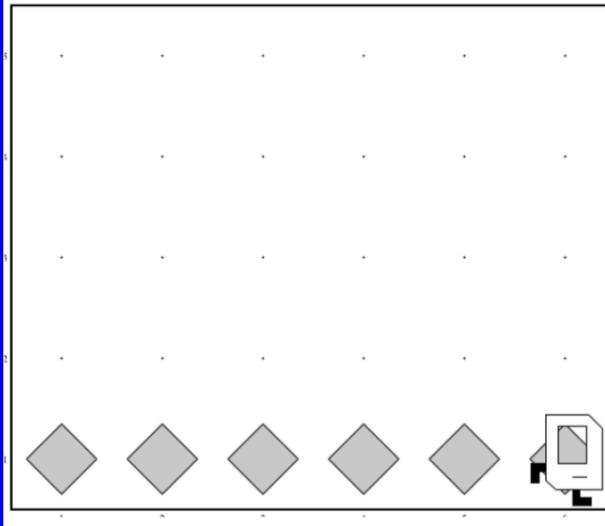
After



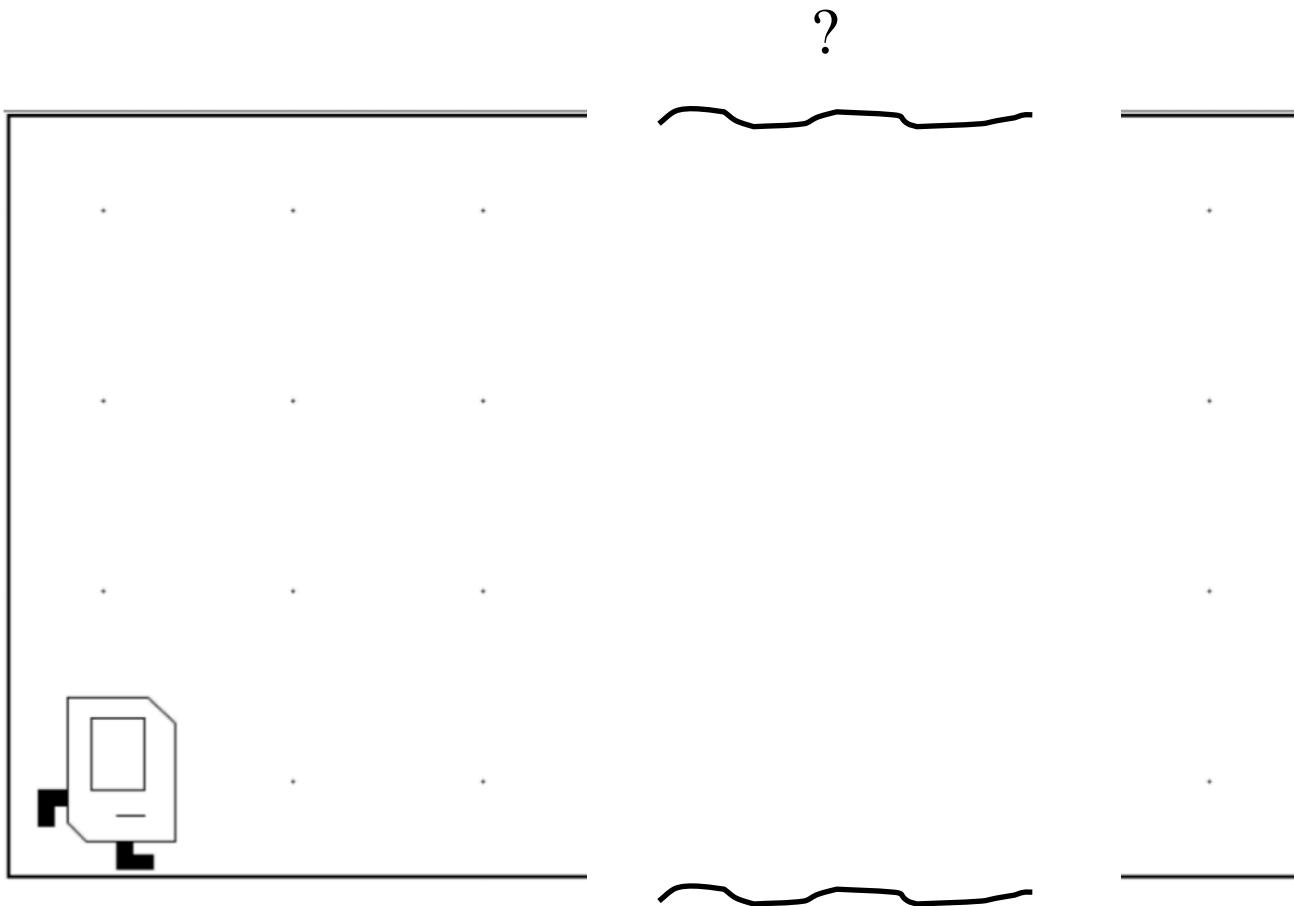
Before



After



Don't Know World Size



Fireside Chat
Tonight 19:00 – 20:00
Here

Labs and Sections!