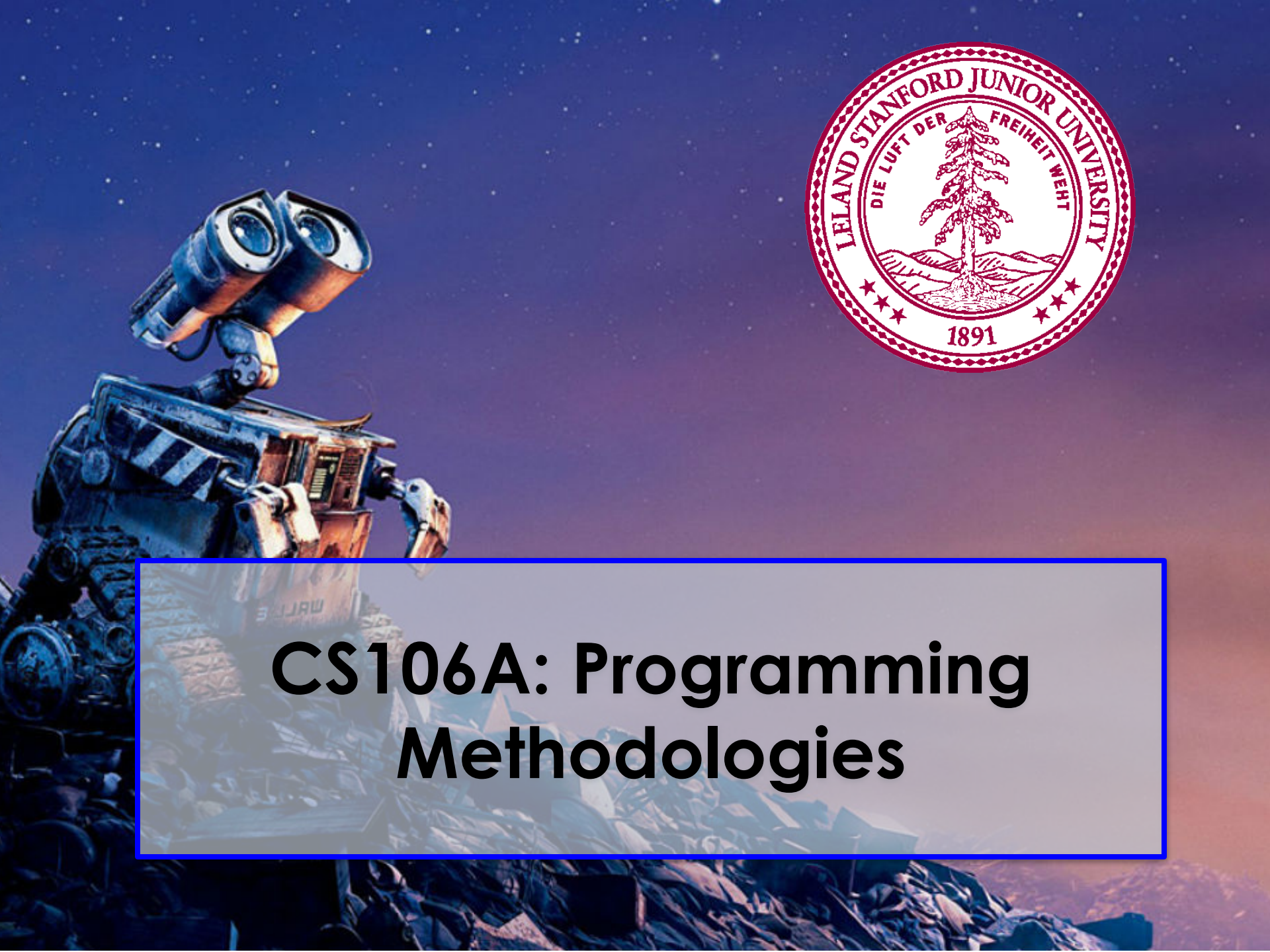


A background image of the WALL-E robot from the Pixar movie "WALL-E". The robot is a small, boxy, blue and orange machine with large, expressive eyes. It is standing on a pile of dark, jagged rocks. The background is a deep blue sky with many small white stars.

CS106A: Programming Methodologies

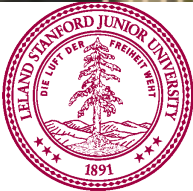
Chris Piech

My parents are interesting folks

I originally concentrated in graphics and worked at Pixar

- Childhood: Nairobi, Kenya
- High School: Kuala Lumpur, Malaysia
- Stanford University Ph.D. in Neural Networks
- Research lab on AI for Social Good

The problem I really want to solve is to make high quality more education accessible



Chris Piech



11 years ago to this day, I was sitting in your seats

Piech, CS106A, Stanford University



Head TA: Nick Troccoli



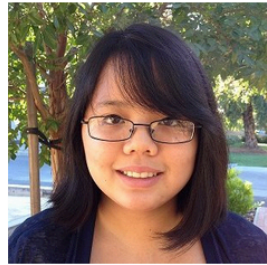
Piech, CS106A, Stanford University



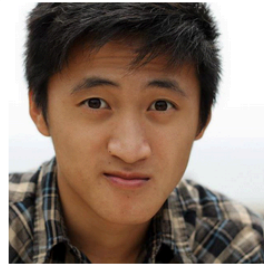
Section Leaders



Alisha



Amy



Brian



Bryce



Çağatay



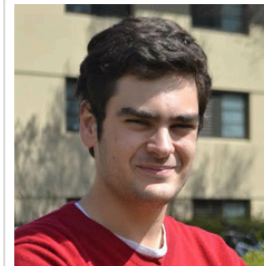
Dilara



Göktuğ



Julia



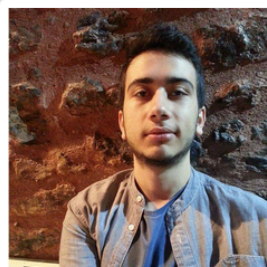
Kerem



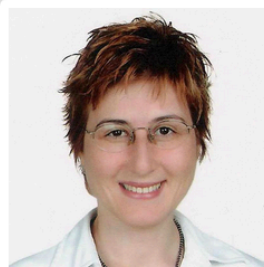
Kevin



Pınar Ö



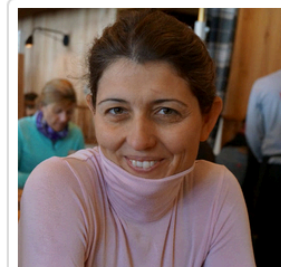
Şahin



Pınar Y



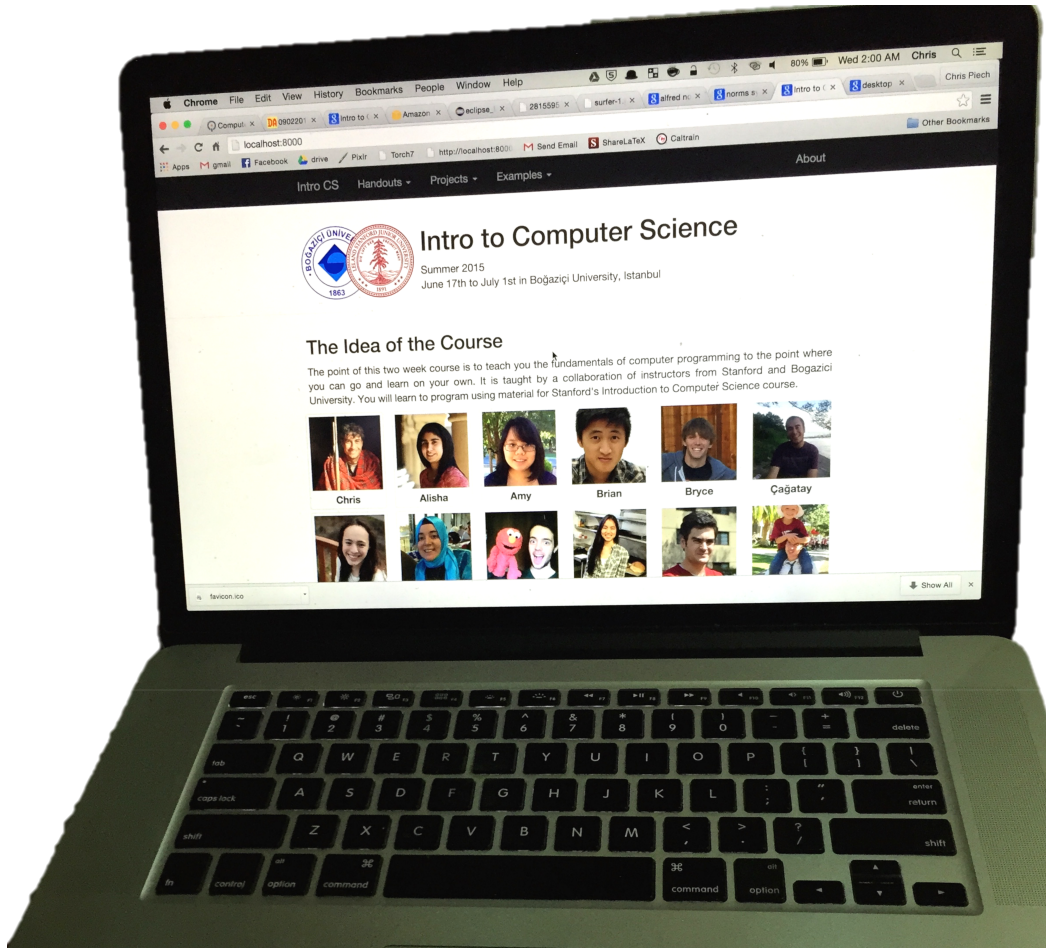
Nick



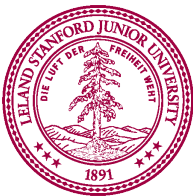
Asena

Who are you?

Prerequisite Test



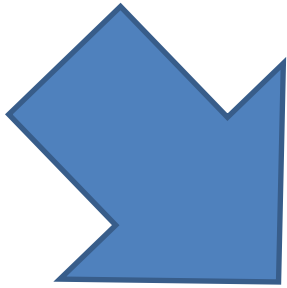
Piech, CS106A, Stanford University



Course mechanics

(this is a light version. Please read the handout for details).

Course Website



<http://cs106a.stanford.edu>



Lectures and Sections

- Weekly 50-min section led by awesome section leaders (the backbone of the class!)
- Signups begin on Thursday at 5:00pm and close Sunday at 5:00pm



Piech, CS106A, Stanford University

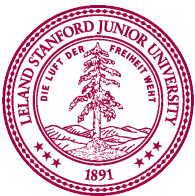


Office Hours



LaIR: 6pm until midnight Sunday through Thursday
(starting next Sunday)

Piech, CS106A, Stanford University



Grading Scale

Functionality and style grades for the assignments use the following scale:

++

A submission so good it “makes you weep.”

+

Exceeds requirements.

✓+

Satisfies all requirements of the assignment.

✓

Meets most requirements, but with some problems.

✓-

Has more serious problems.

-

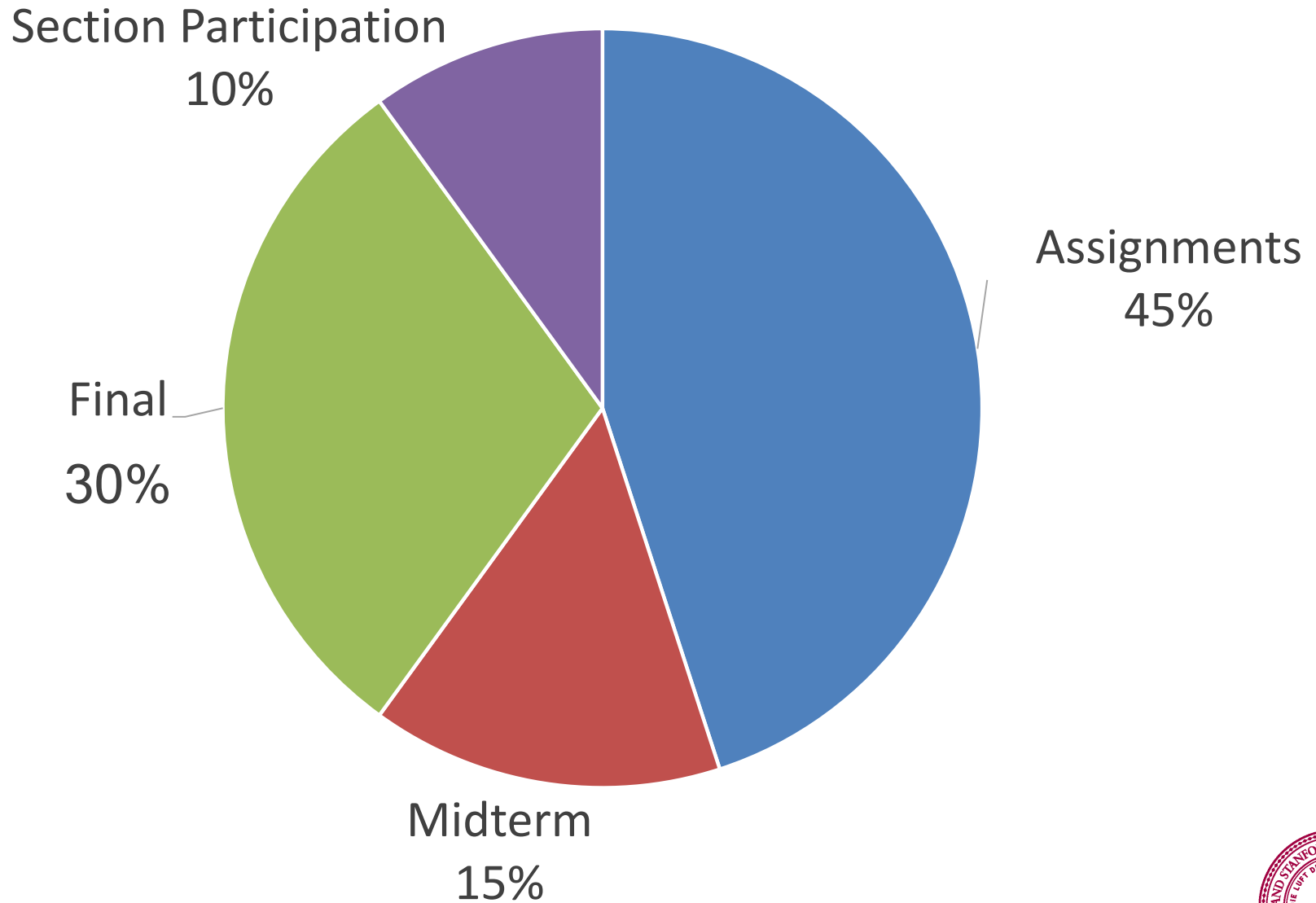
Is even worse than that.

--

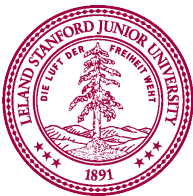
Better than nothing.



What we will ask you to do



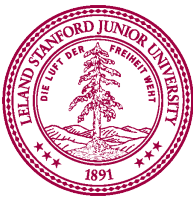
Piech, CS106A, Sta *Two free late days



Optional Contest



Piech, CS106A, Stanford University



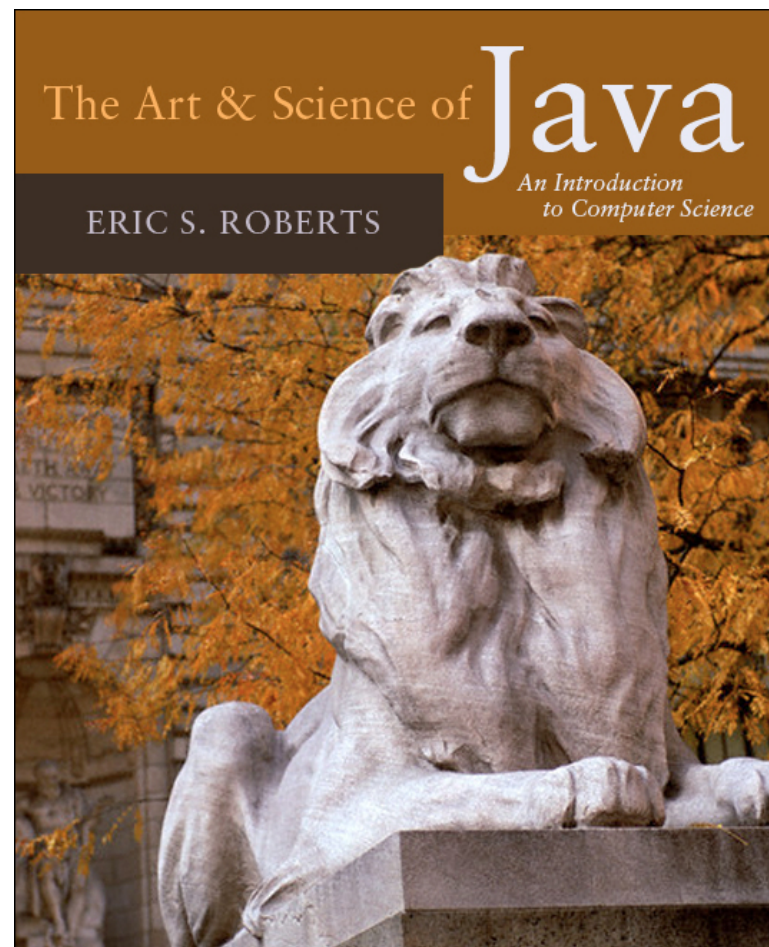
Textbook

The Art & Science of Java by Eric Roberts

- written here at Stanford
- tailored to this course
- a valuable reference
- usable on open-book exams

Karel the Robot Learns Java

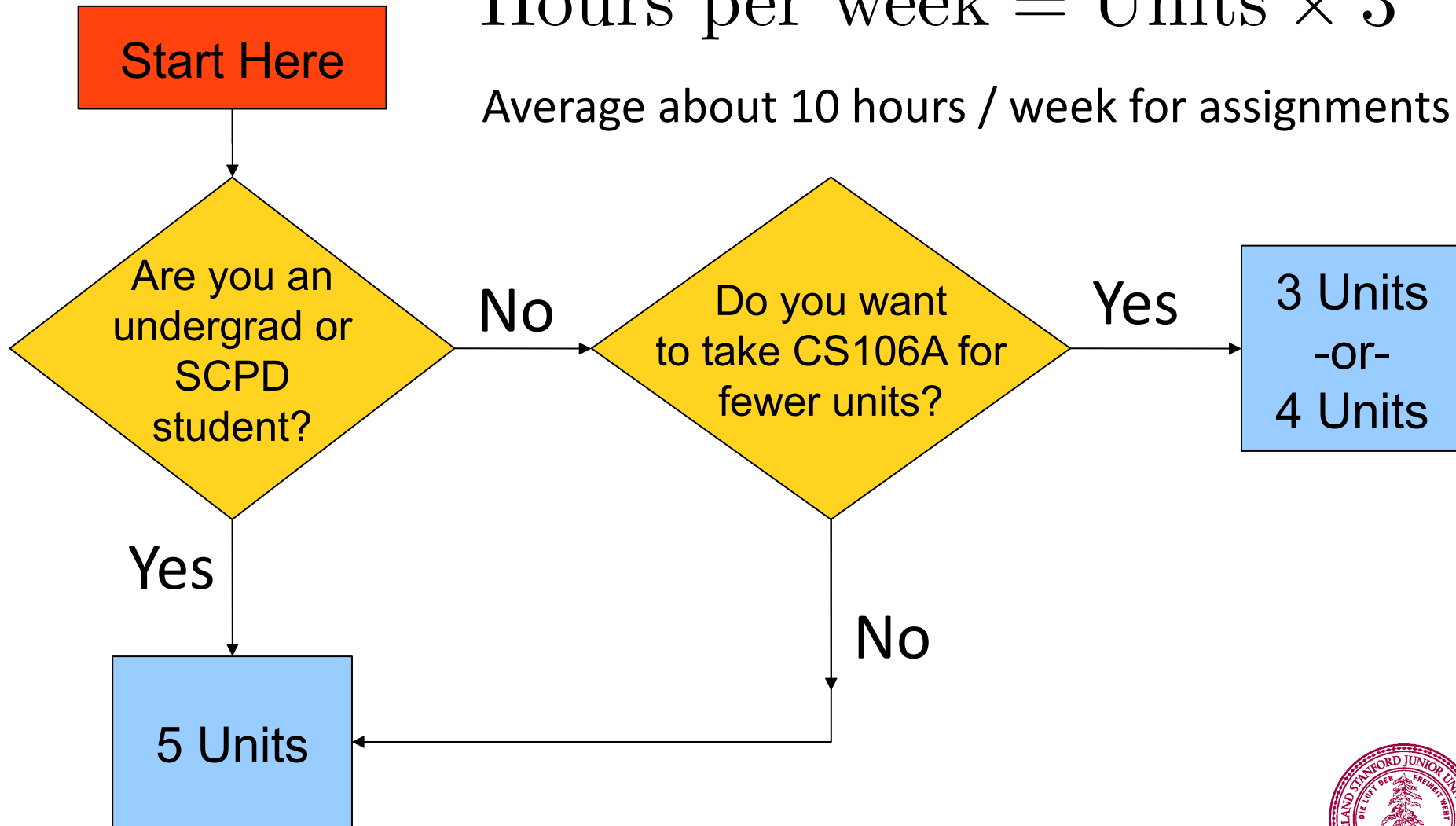
- First week of material



CS106A Units

Hours per week = Units $\times$ 3

Average about 10 hours / week for assignments



Lectures are Online

Stanford Center for Professional Development | Graduate Courses | Seminars | Previews | Tech Support

Chris James Piech (cpiech) | Log Out

Stanford | Center for Professional Development

ClassX *engineered for engineers* [Learn more >>](#)

Programming Methodology

CS106A



Watch Now: Wednesday, December 7, 2016

Course Details

Course Description [View License](#)

This introduction to the engineering of computer applications emphasizes modern software

[classx.stanford.edu/Home?Graduate=1](#) r "D's" of development: Design,

Course Sessions (29):

[Show All](#)

Week 11

Day/Date:	Watch Online:	Duration:
Wednesday December 7, 2016	▶ WATCH NOW	00:48:08
Monday December 5, 2016	▶ WATCH NOW	00:48:19

Week 10

[+](#)

Week 9

[+](#)

Week 8

[+](#)



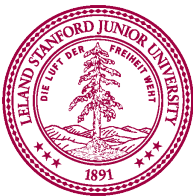
Are you in the right place?

What is CS106A?

Computer Science

Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do.

- Michael Fellows and Ian Parberry



Learning Goals

- *Learn how to harness computing power to solve problems.*
- To that end:
 - Explore fundamental techniques in computer programming.
 - Develop good software engineering style.
 - Gain familiarity with the Java programming language.



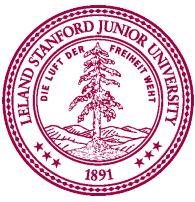
There are a lot of cool
programs you may one day
write

Computer Graphics



Pat Hanrahan, one of the founders of Pixar is a professor here

Piech, CS106A, Stanford University



Consumer Applications



Autonomous Surgery



Piech, CS106A, Stanford University

(c) 2012 Intuitive Surgical, Inc.

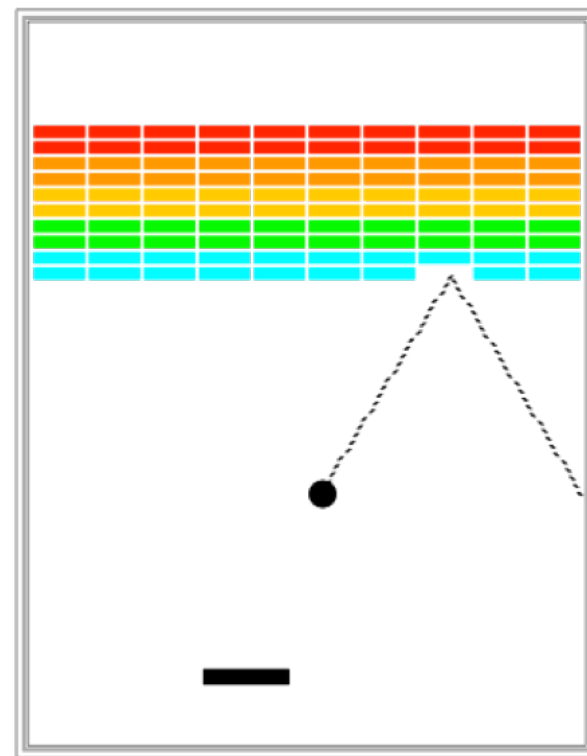
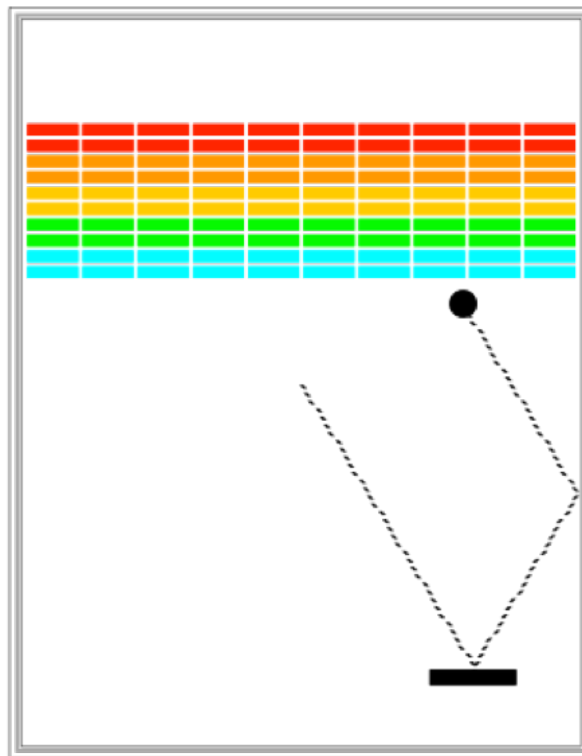
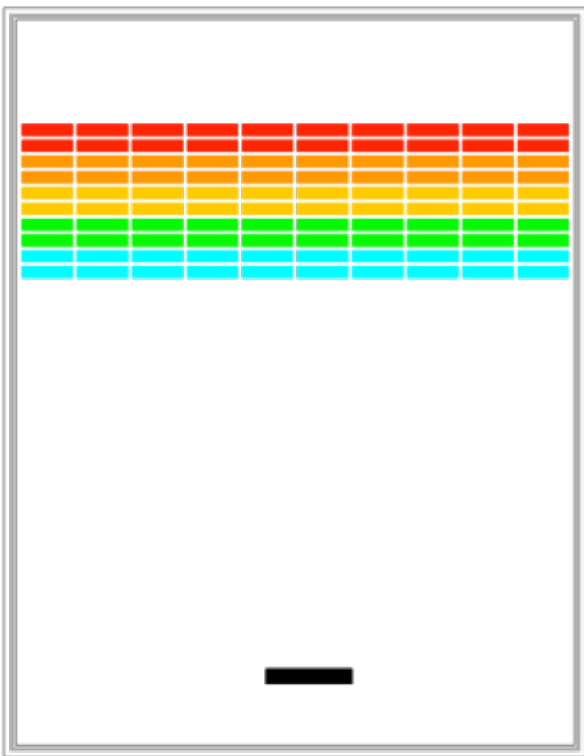


Self Driving Car

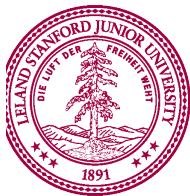
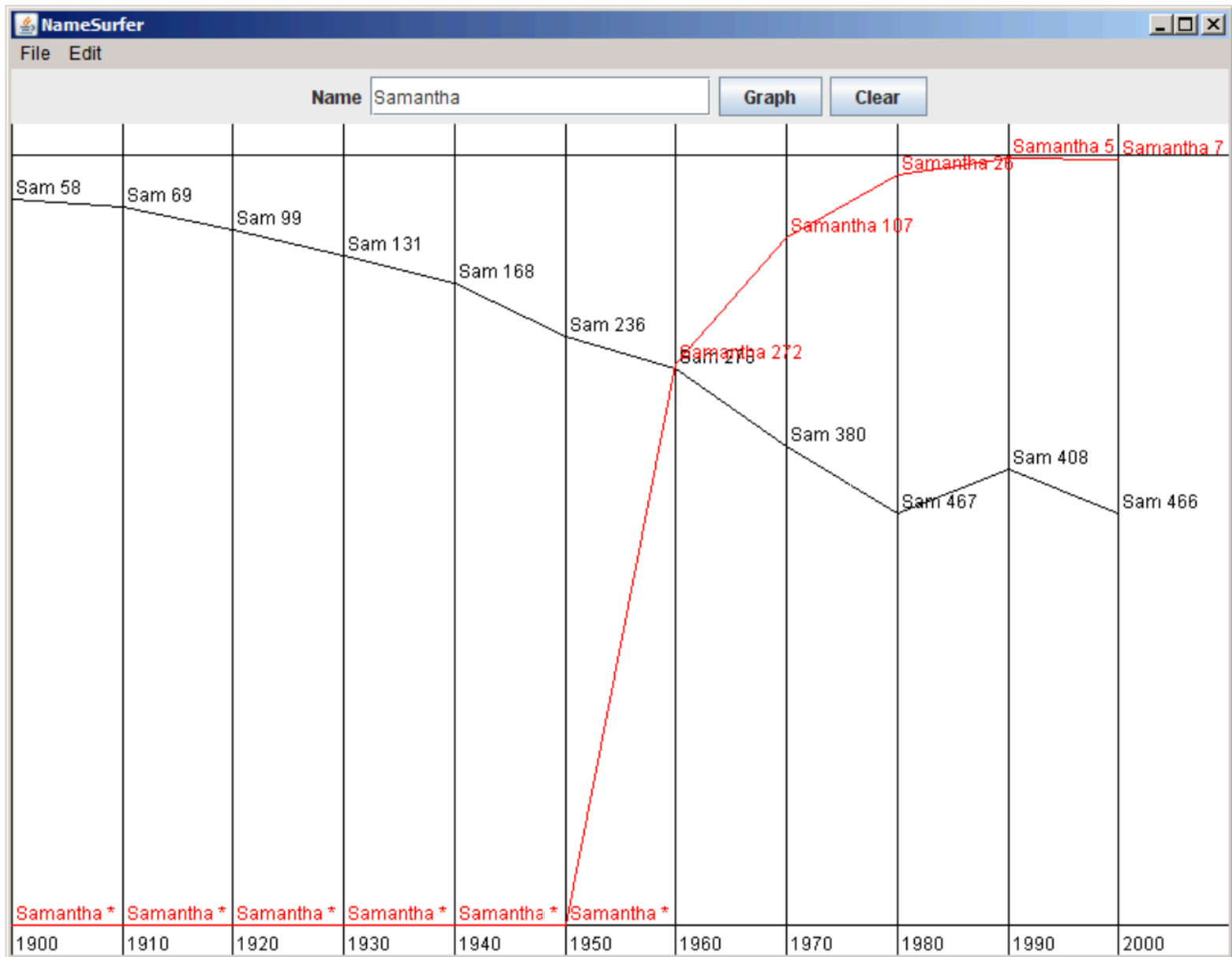


If only we could program self
driving cars...

Graphical Games



Data Visualization



Mini Facebook

FacePamphlet

File Edit

Name

Mehran Sahami

Friends:



Mehran Sahami is coding like a fiend

Displaying Mehran Sahami



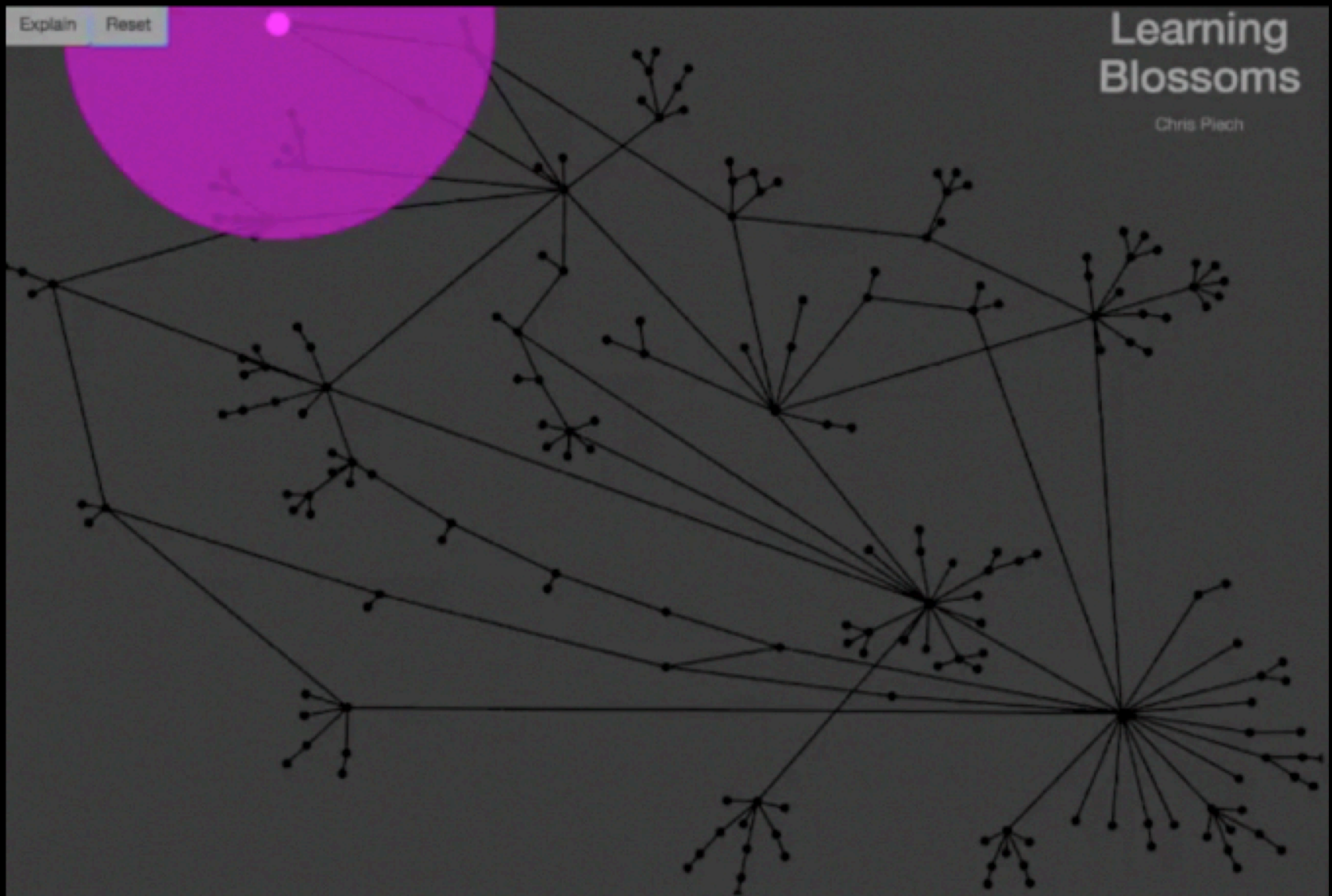
Strive for Everyone to Succeed



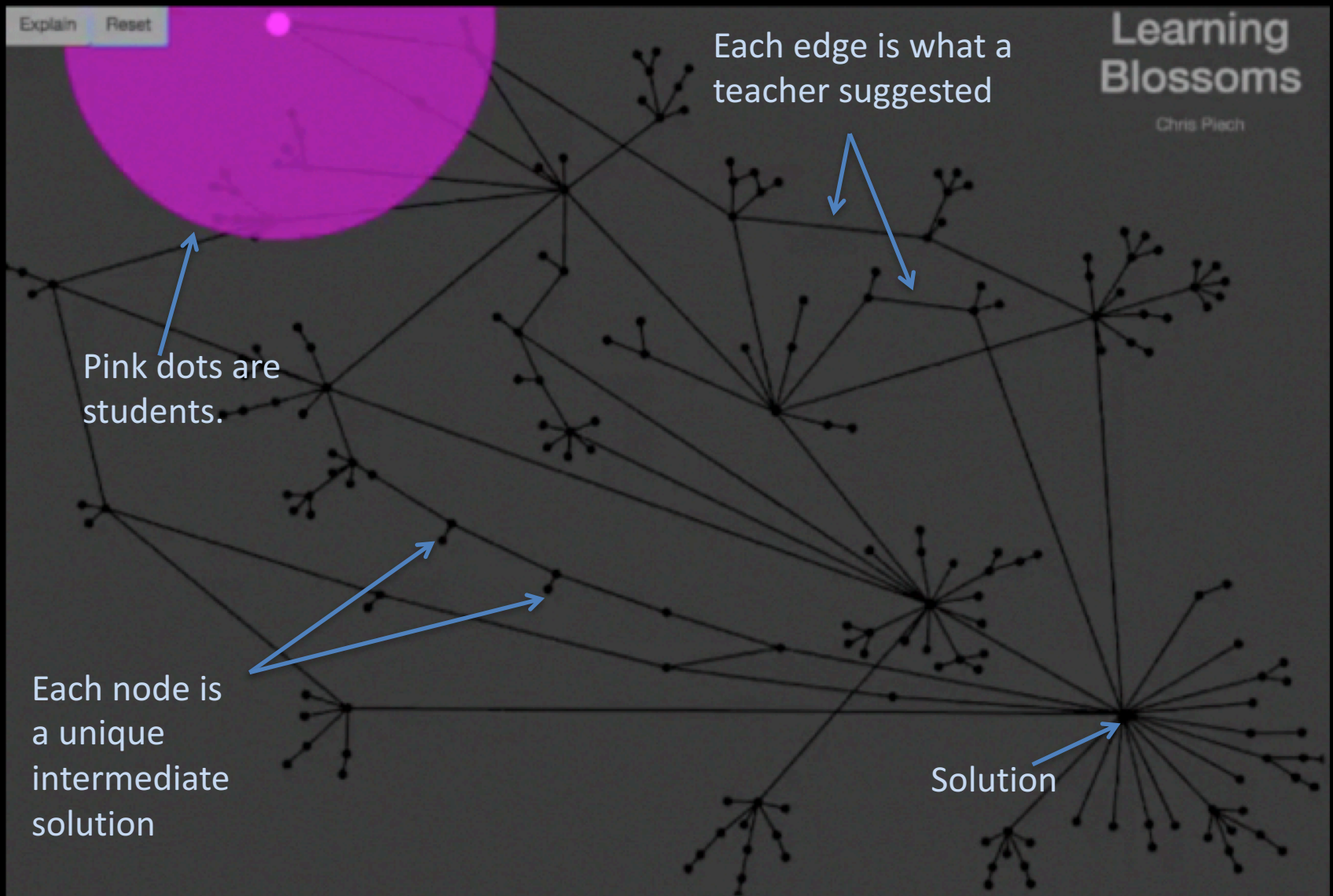
Art of Computer *Science*



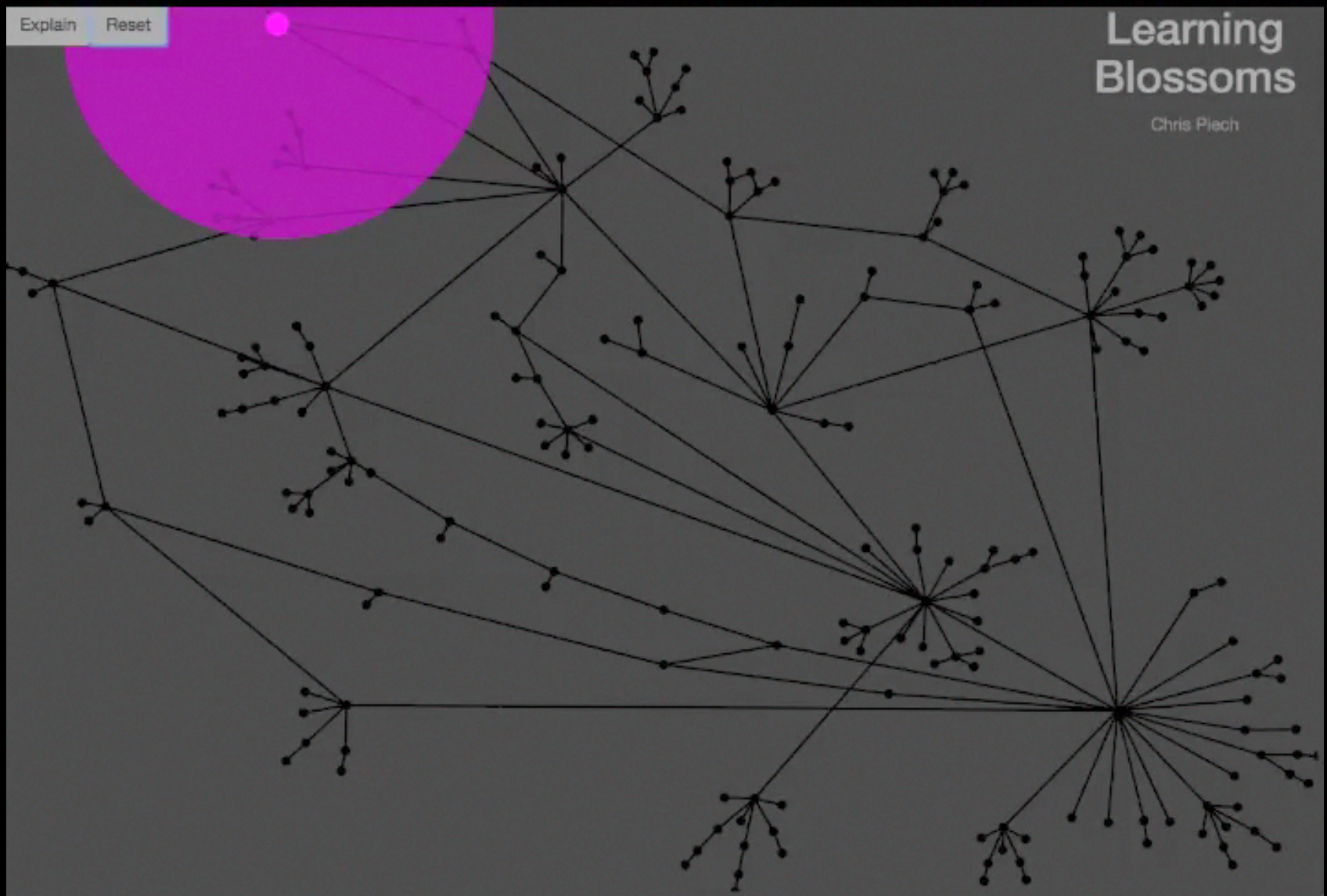
Something cool



Autonomously Generating Hints by Inferring Problem Solving Policies
Chris Piech, Mehran Sahami, Jonathan Huang, Leo Guibas

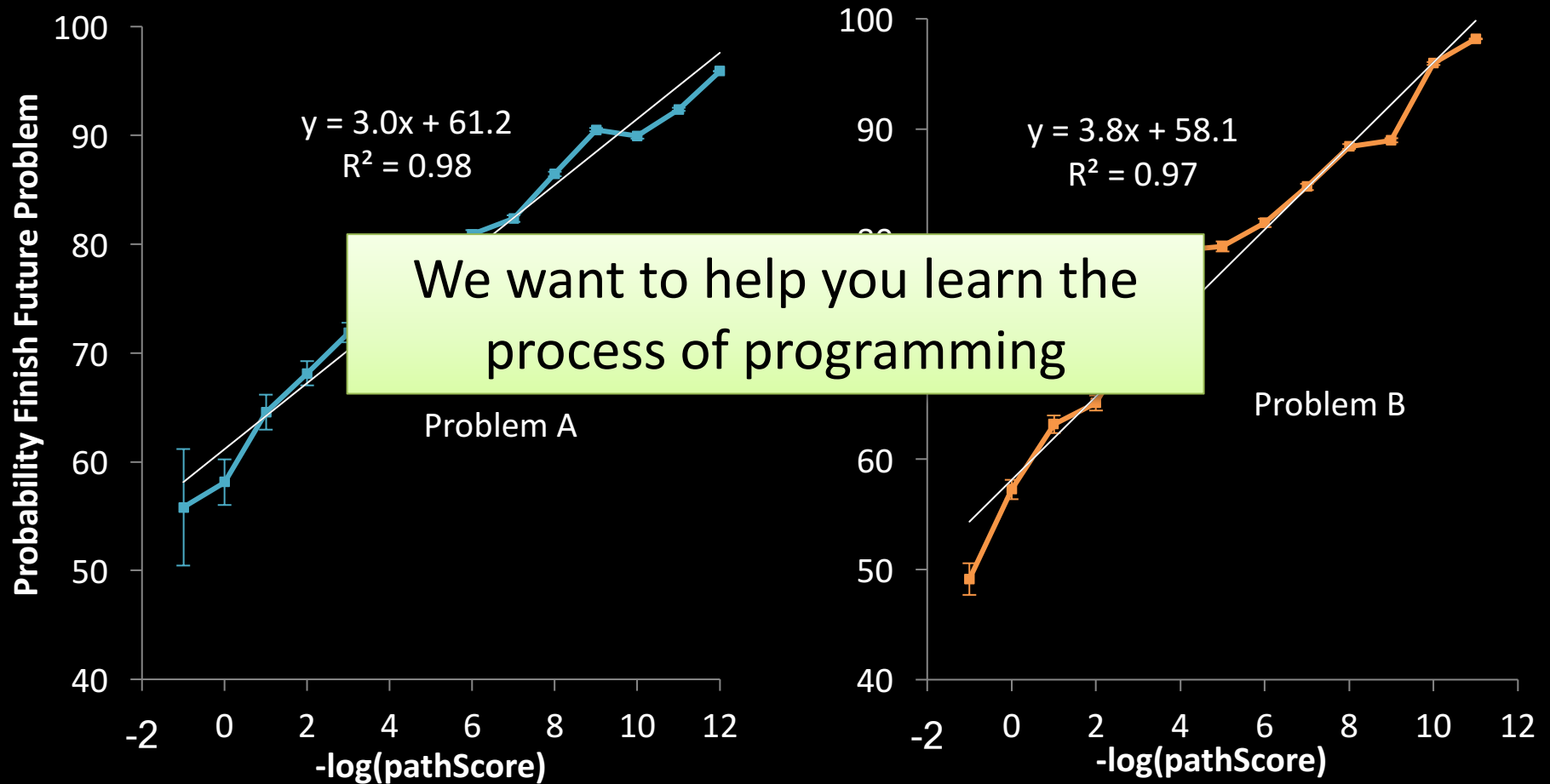


Autonomously Generating Hints by Inferring Problem Solving Policies
Chris Piech, Mehran Sahami, Jonathan Huang, Leo Guibas



Autonomously Generating Hints by Inferring Problem Solving Policies
Chris Piech, Mehran Sahami, Jonathan Huang, Leo Guibas

Process Predicts Future



Effect is large and logarithmic.

Lets Get Started



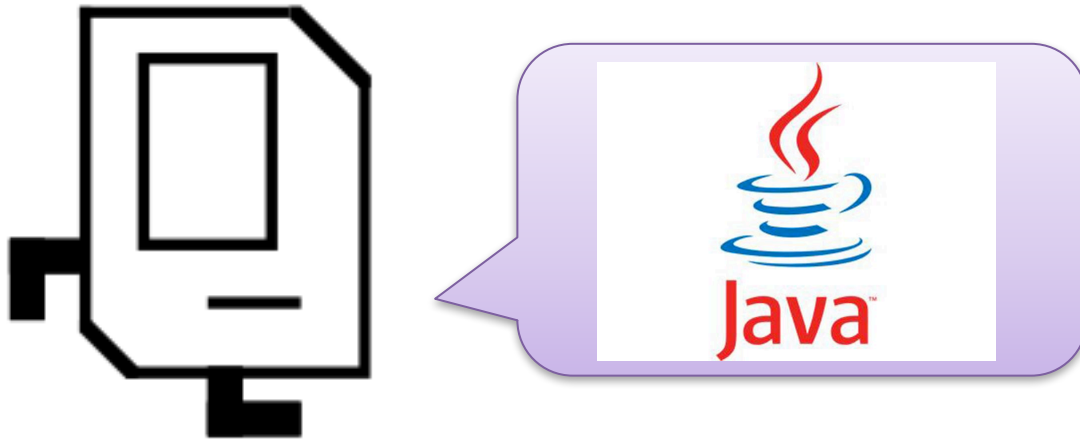
Meet Karel the Robot



Good morning



Karel Speaks Java

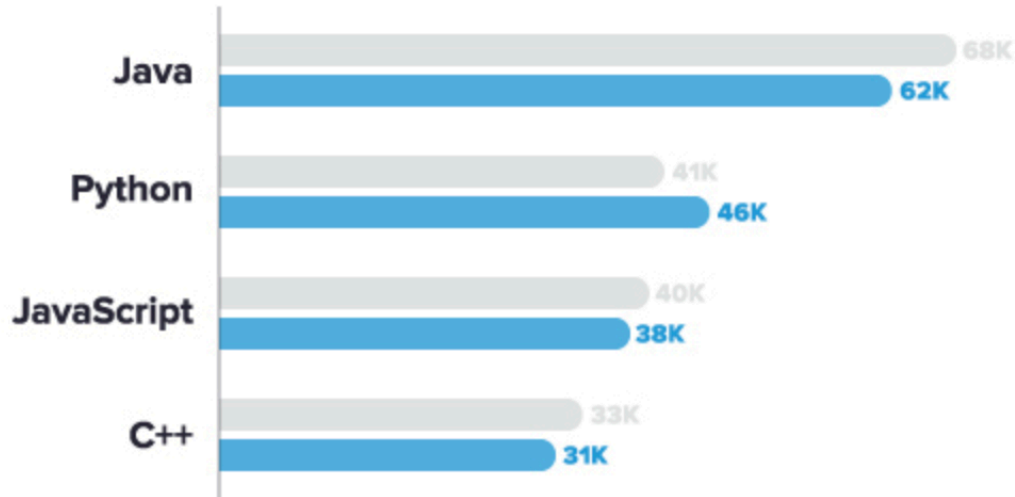


Why Java?

1

Job postings containing top languages

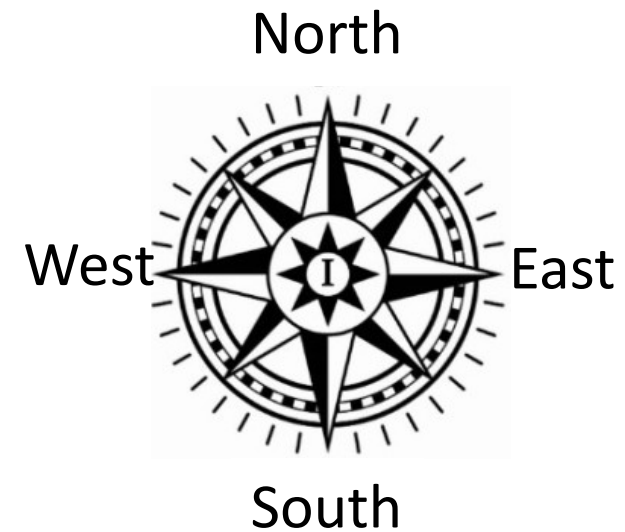
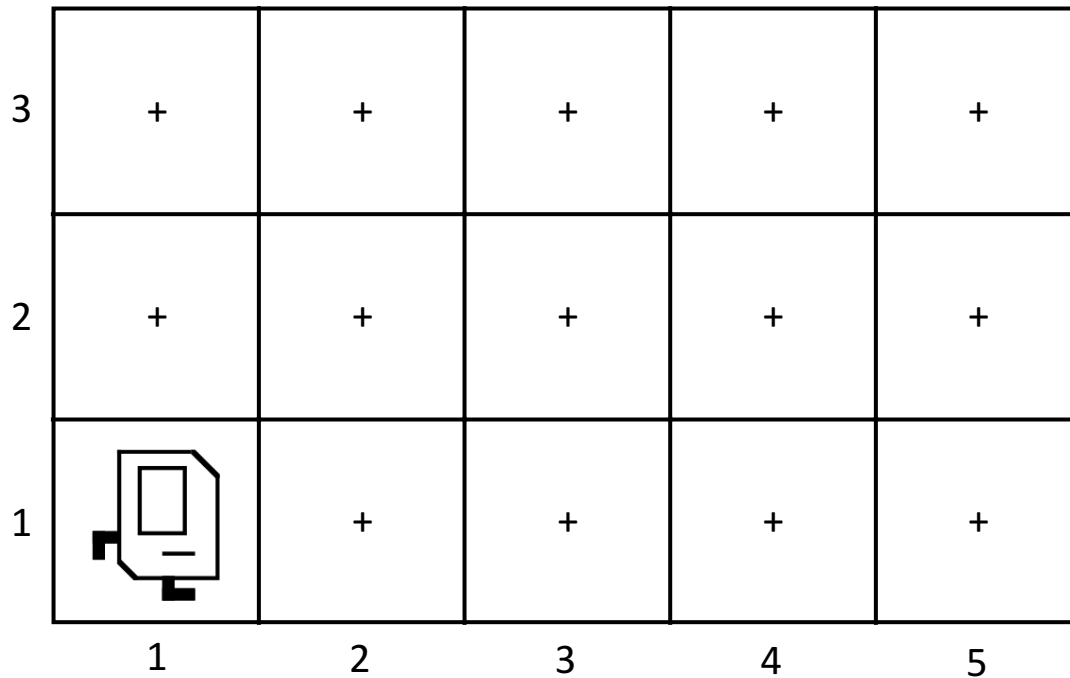
Indeed.com - November, 17th 2017



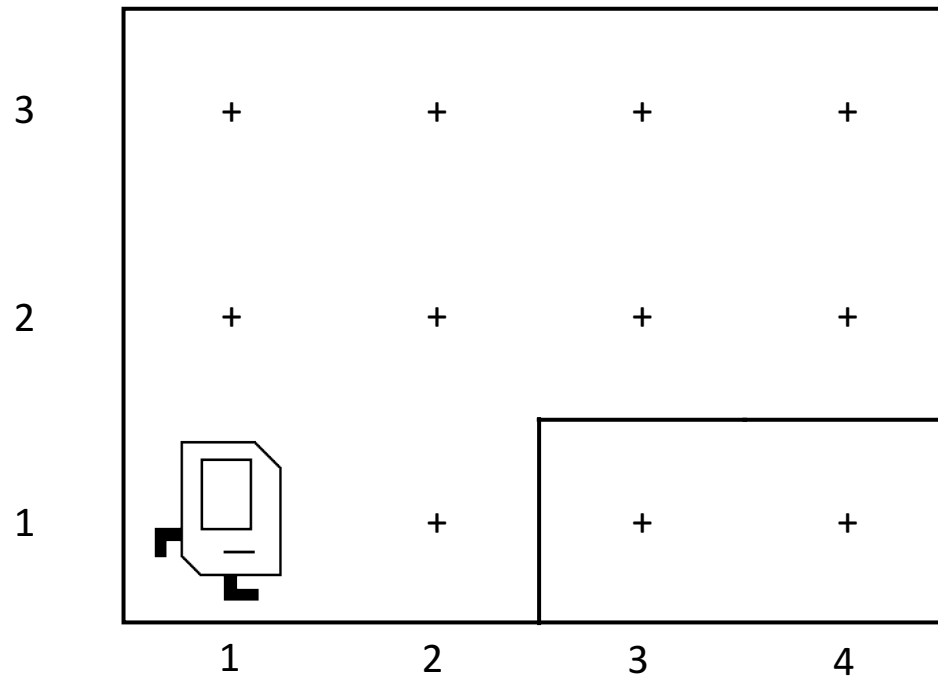
2



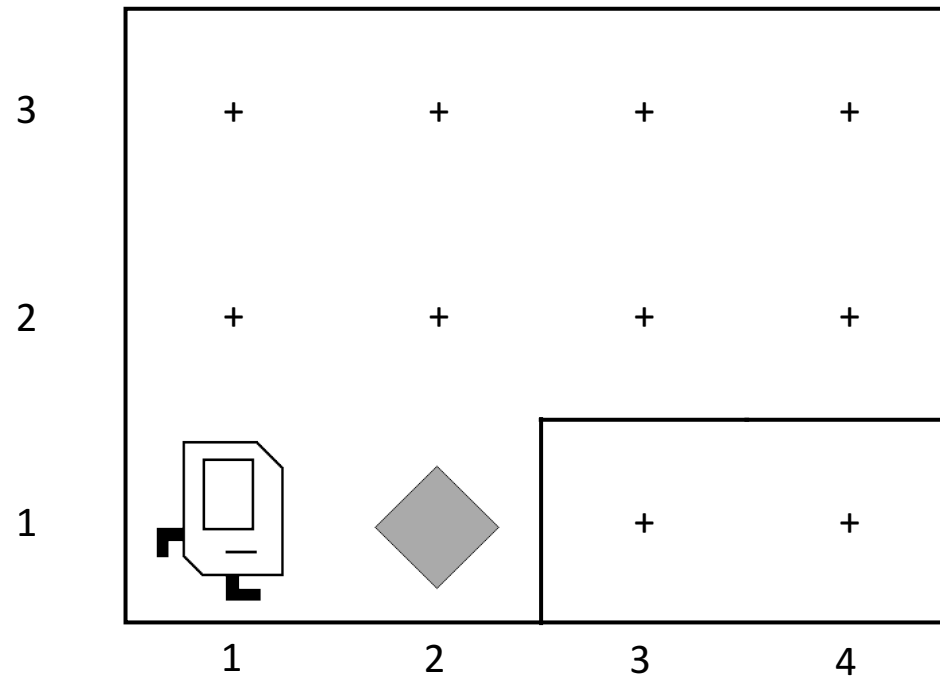
Karel's World



Walls



Beeper's



Knows Four Commands



`move ( ) ;`

`turnLeft ( ) ;`

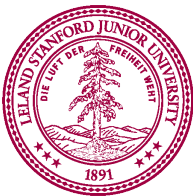
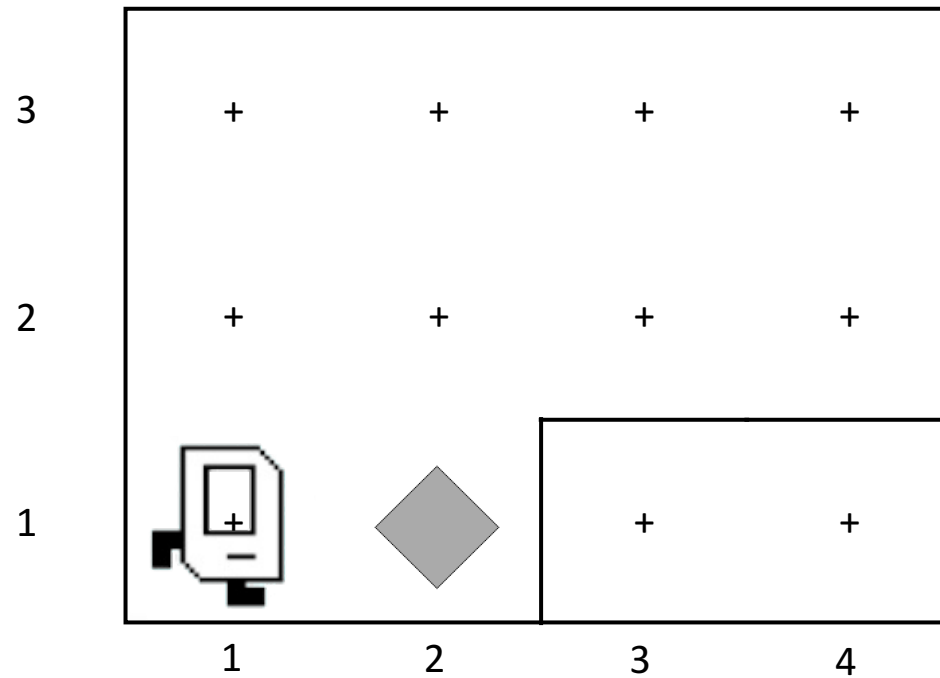
`putBeeper ( ) ;`

`pickBeeper ( ) ;`

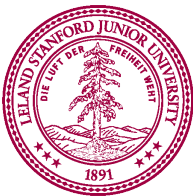
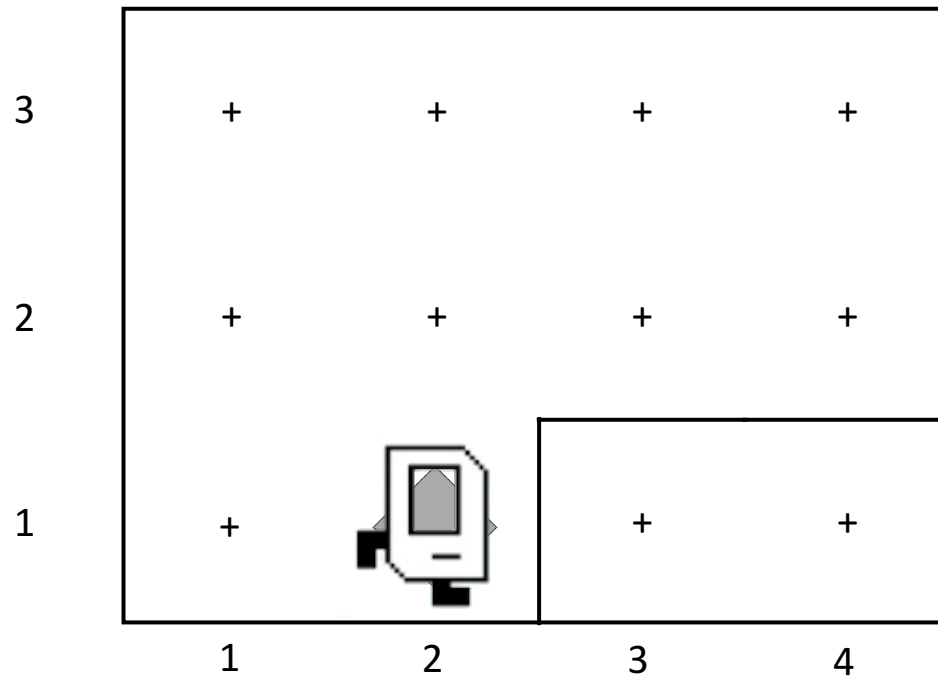


```
move ( ) ;
```

move () ;

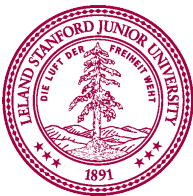
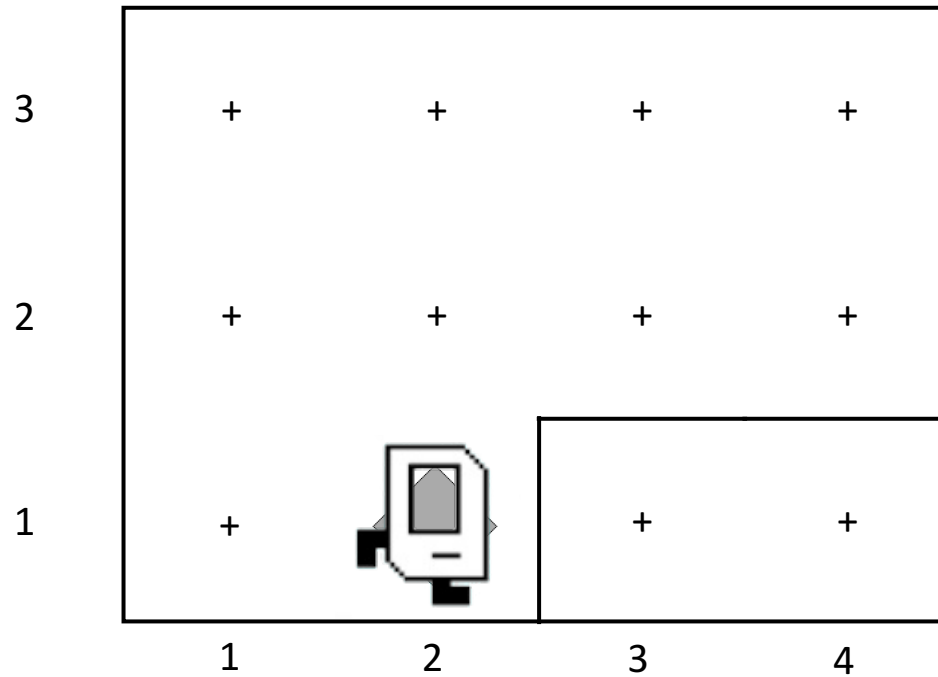


move () ;

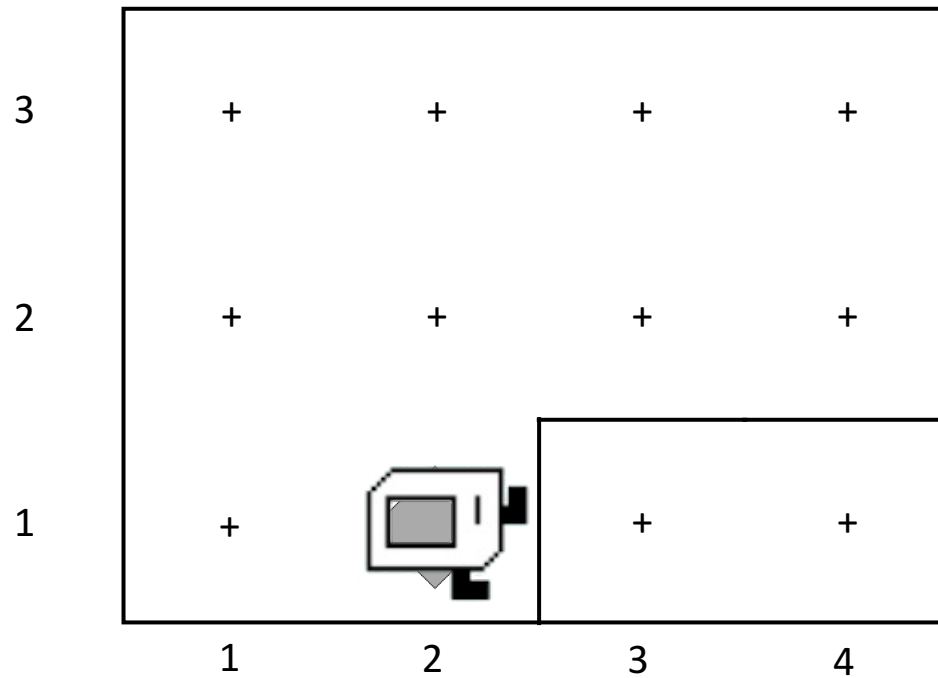



```
turnLeft( );
```

turnLeft();

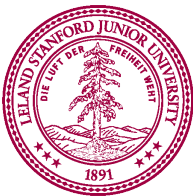
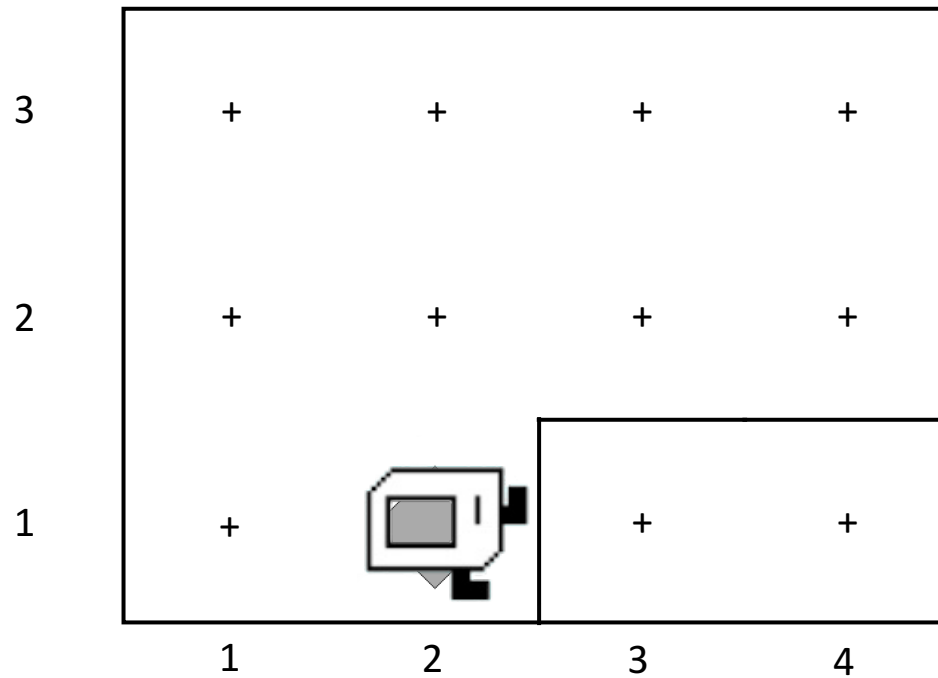


turnLeft();

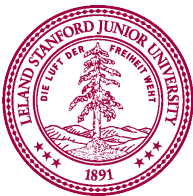
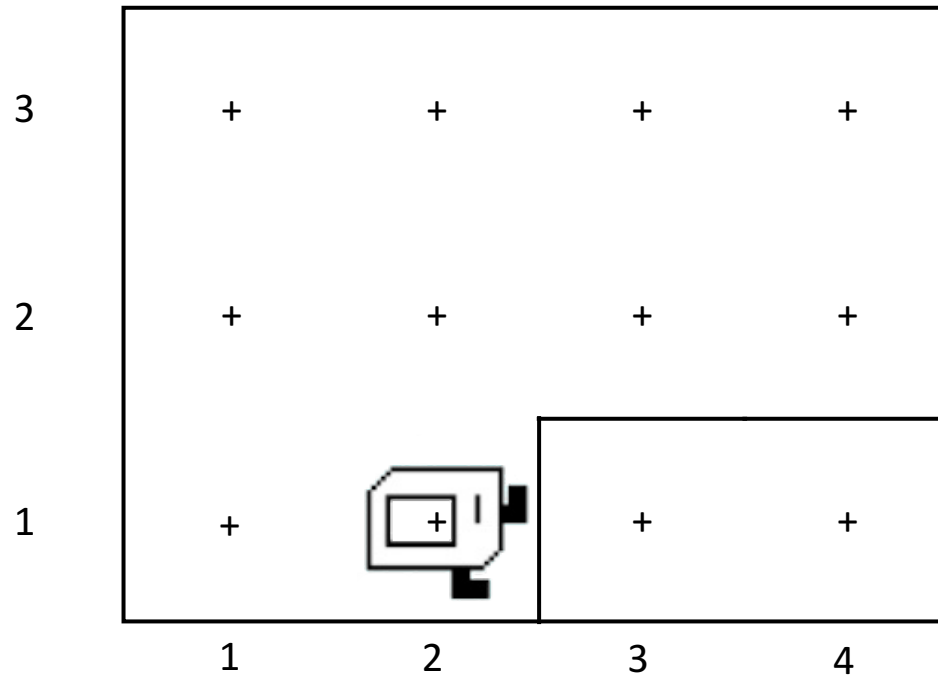


```
pickBeeper ( ) ;
```

`pickBeeper()` ;

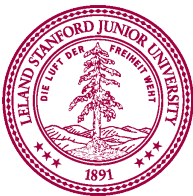
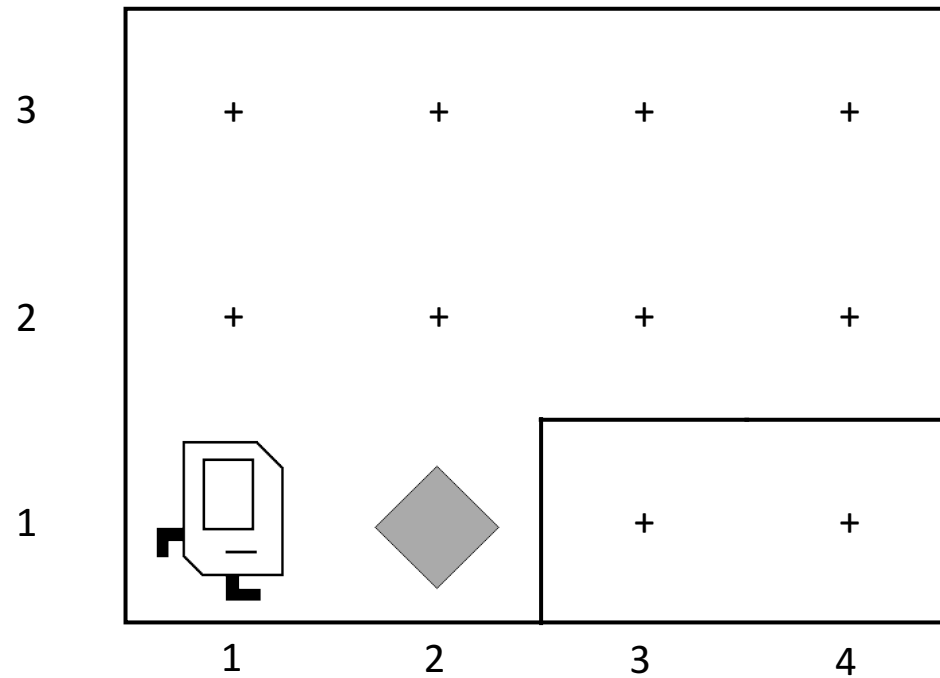


`pickBeeper()` ;

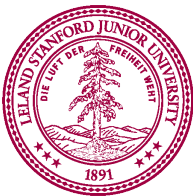
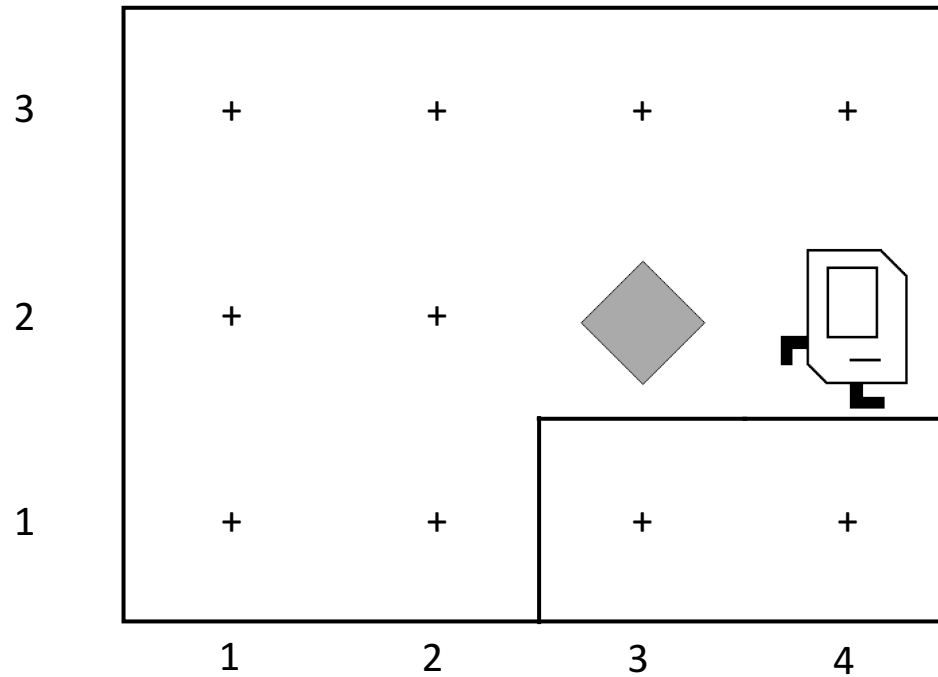


Make Sense?

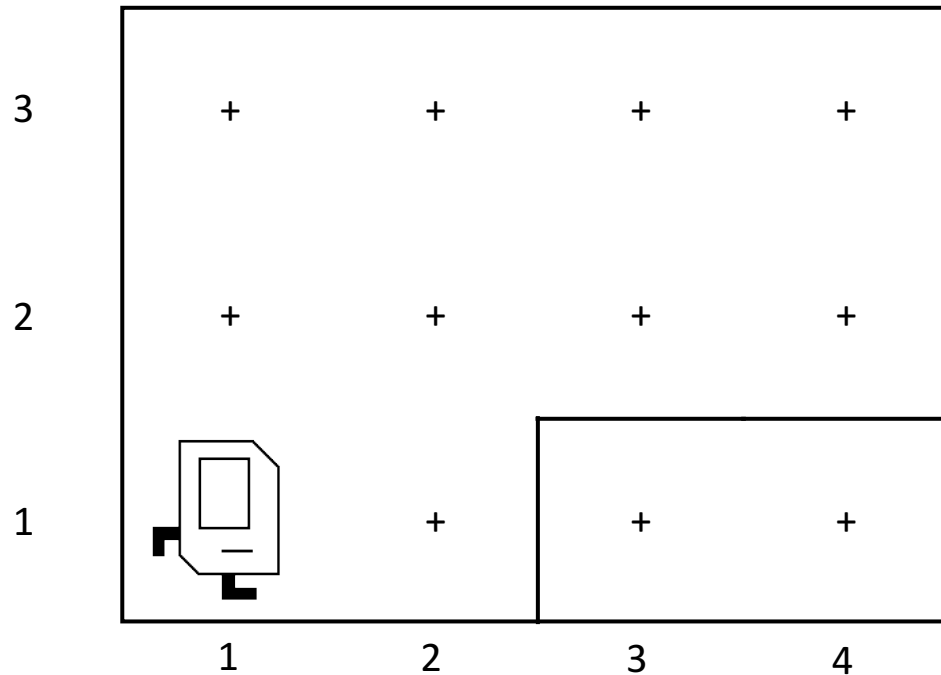
First Challenge



First Challenge



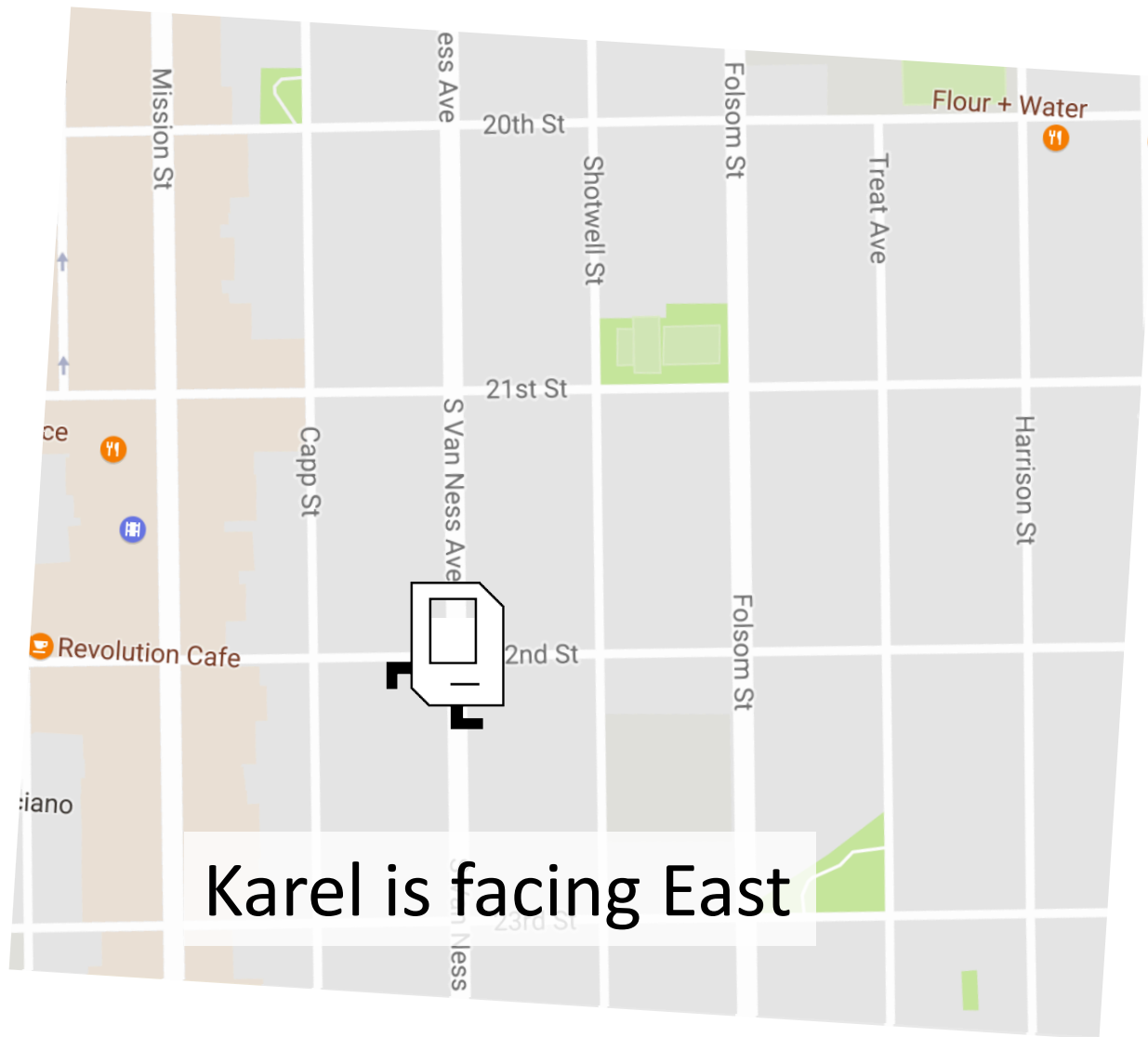
Bird's Eye View



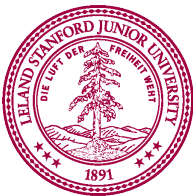
Karel is facing East



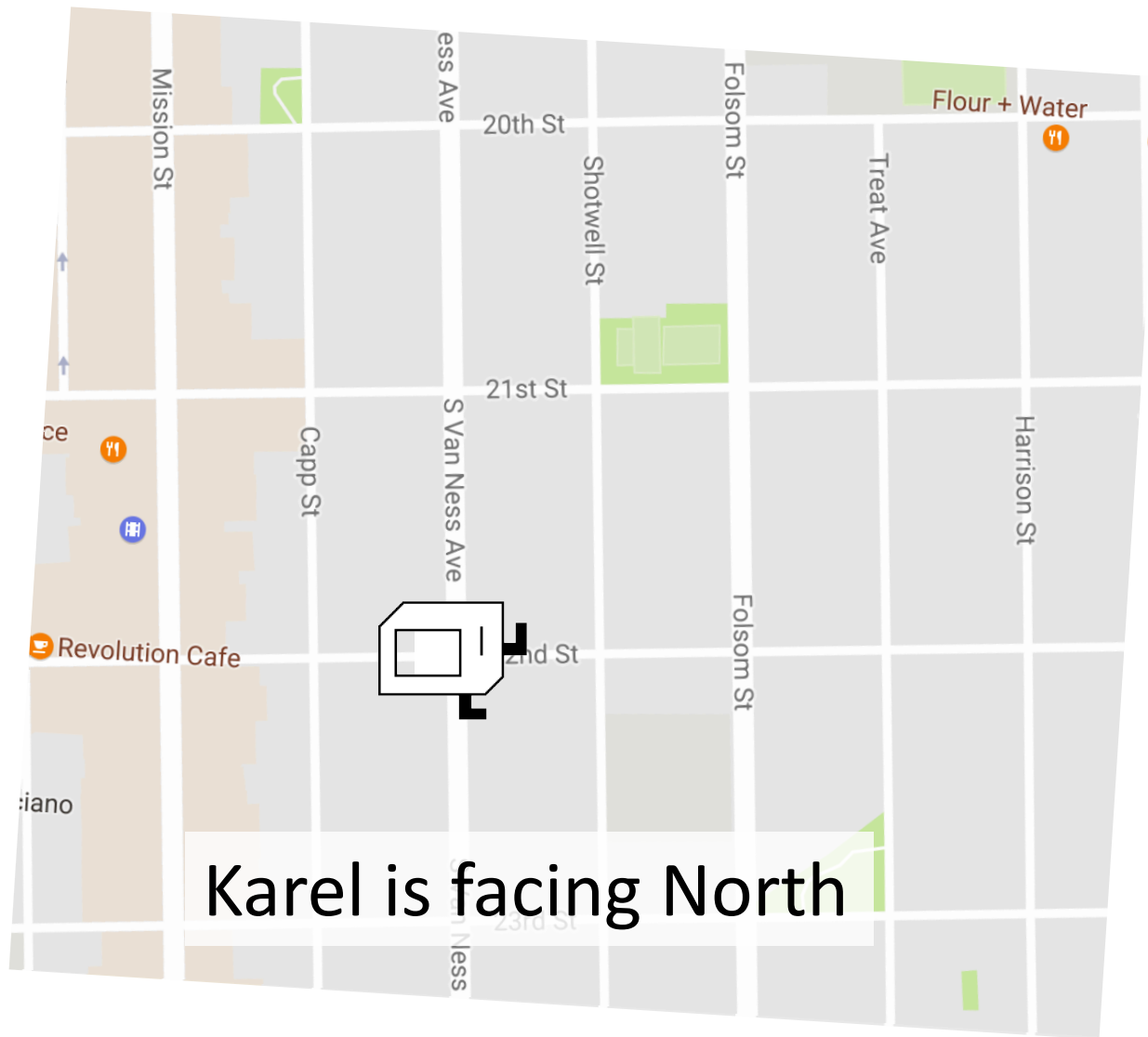
Bird's Eye View



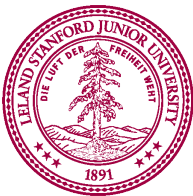
Karel is facing East



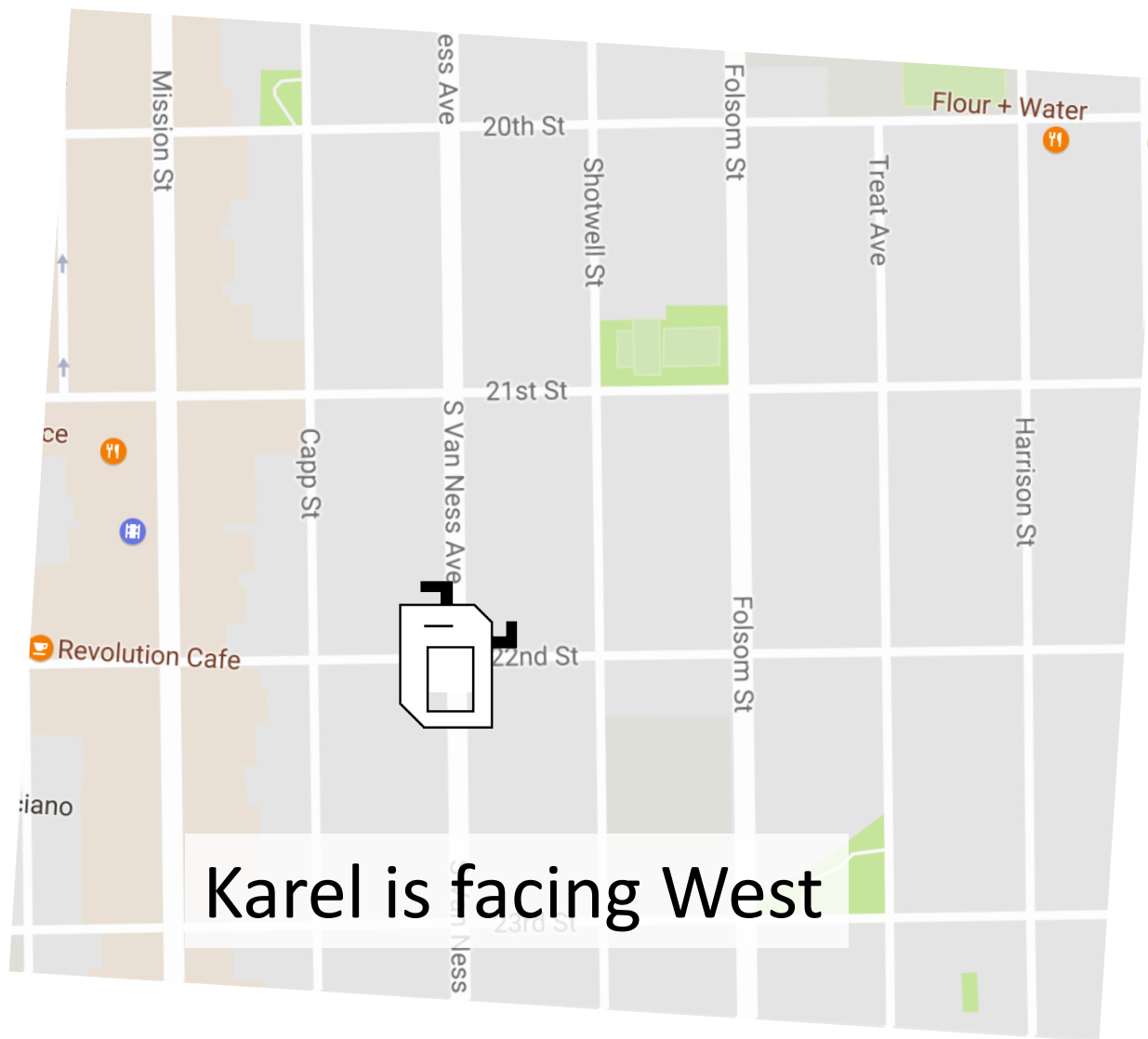
Turn Left



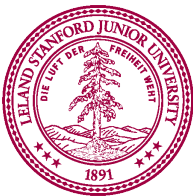
Karel is facing North



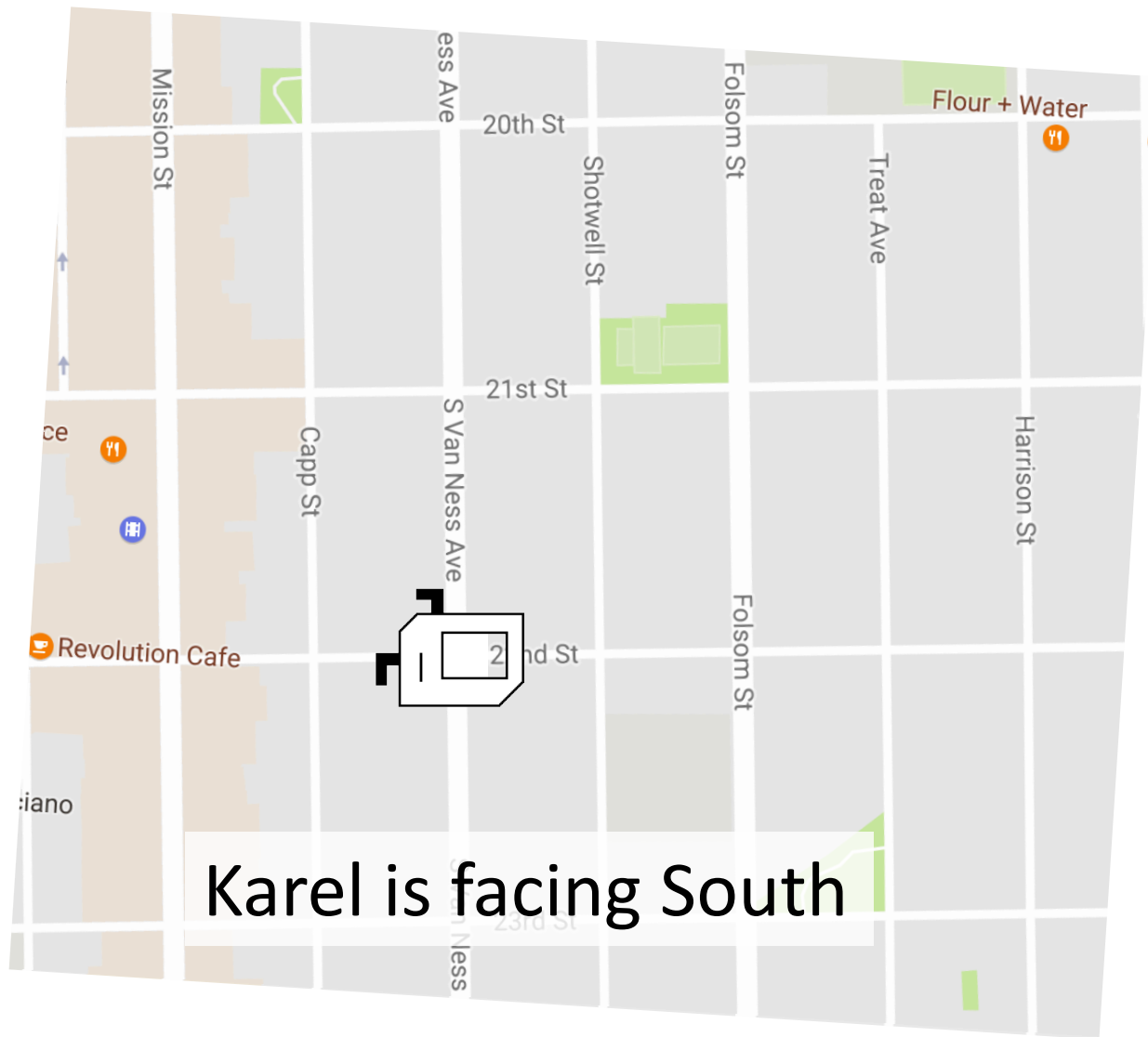
Turn Left



Karel is facing West



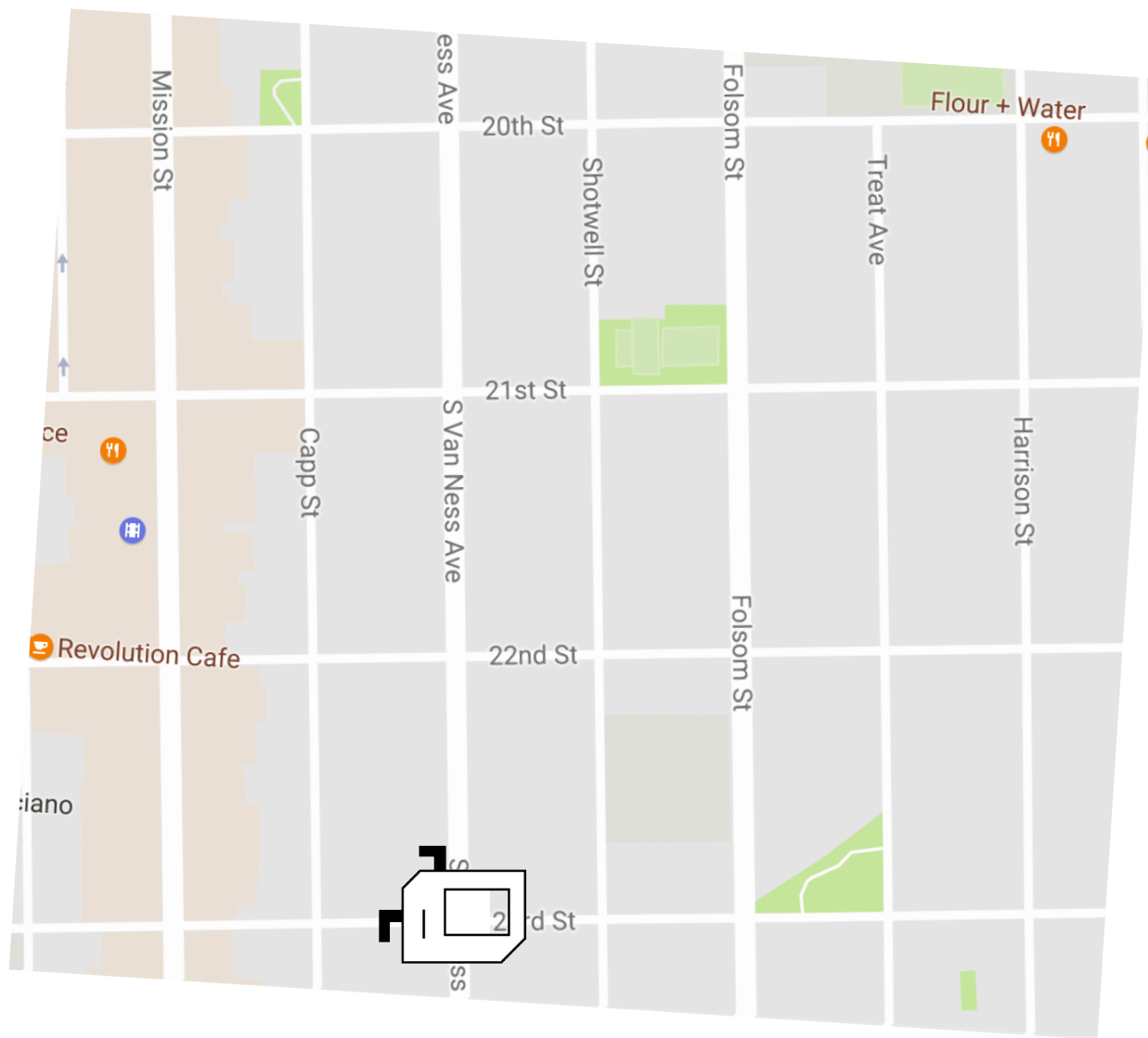
Turn Left



Karel is facing South



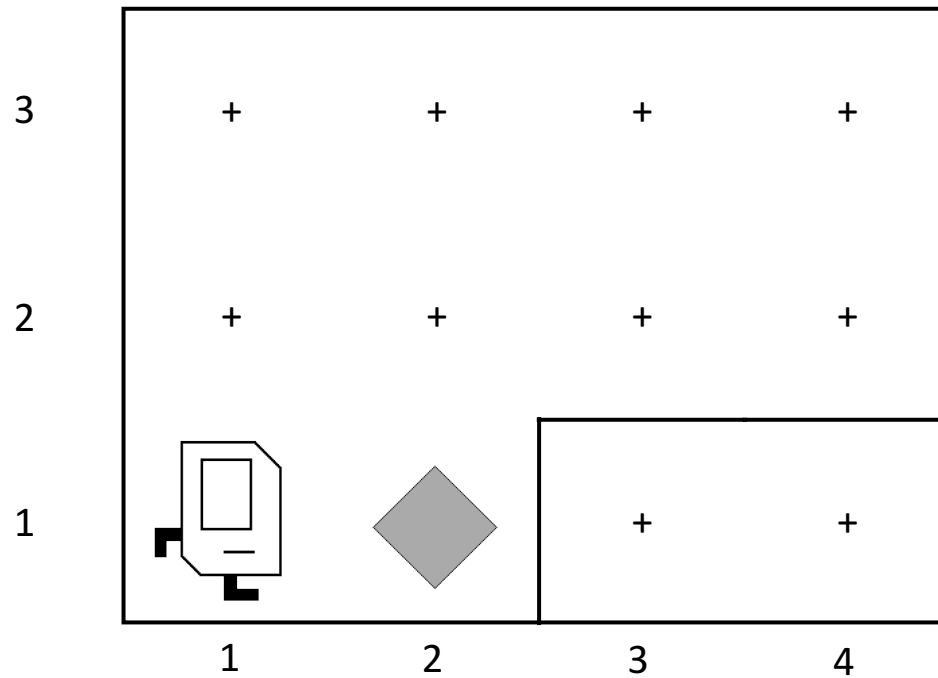
Move



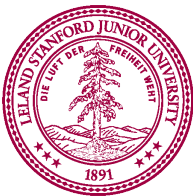
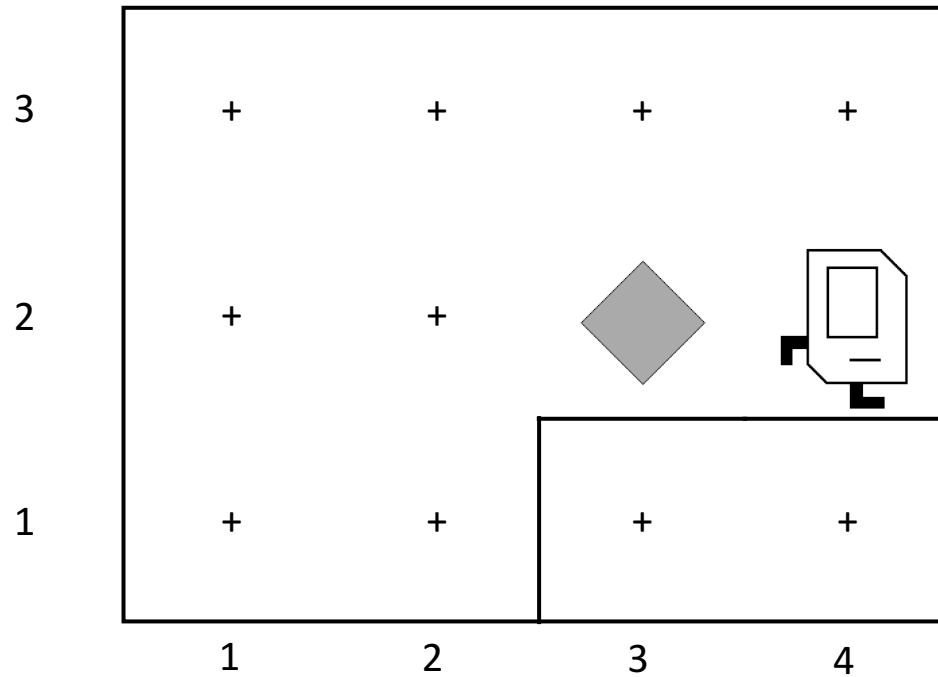
Piech, CS106A, Stanford University



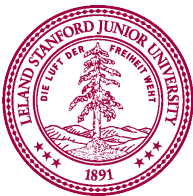
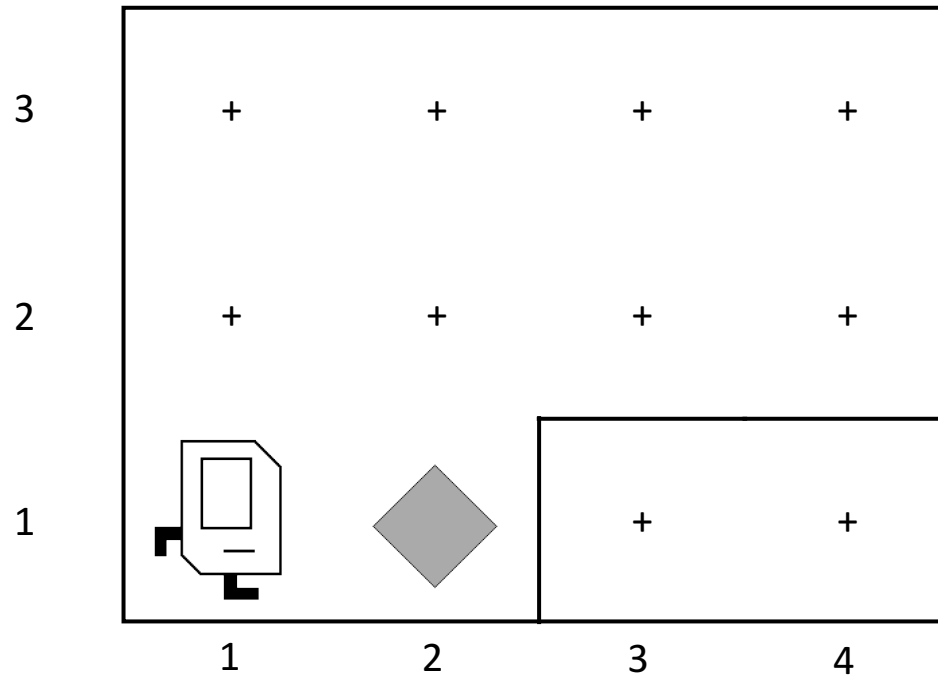
First Challenge



First Challenge



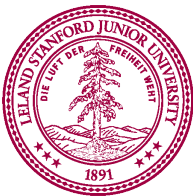
First Challenge



Learn By Doing



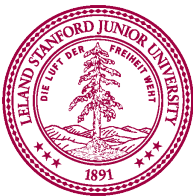
eclipse



Method Definition

```
private void name() {  
    method statements  
}
```

This adds a new
command to Karel's
vocabulary



Anatomy of a Program

Import Packages

Program



Anatomy of a Program

Import Packages

```
public class OurKarelProgram extends Karel {
```

```
}
```



Anatomy of a Program

Import Packages

```
public class OurKarelProgram extends Karel {
```

run method

helper methods

```
}
```



Anatomy of a Program

Import Packages

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



Anatomy of a Program

Import Packages

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



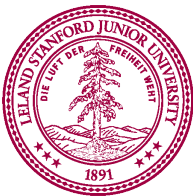
Anatomy of a Program

```
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();
    }
}
```



Anatomy of a Program

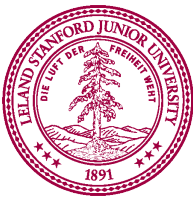
```
import stanford.karel.*;

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();
```

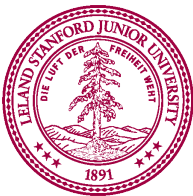
```
        turnLeft();
```

```
        turnLeft();
```

```
    }
```

```
}
```

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



Anatomy of a Program

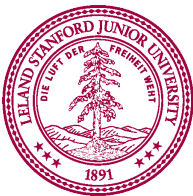
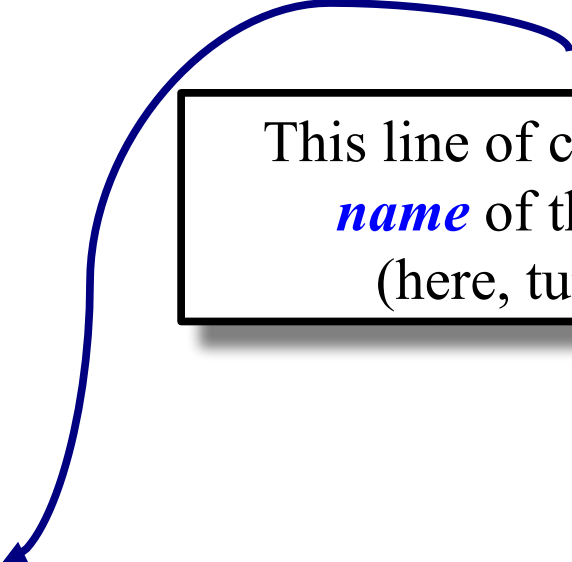
```
import stanford.karel.*;

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

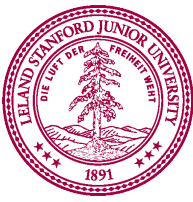
```
import stanford.karel.*;

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 a
code block



Anatomy of a Program

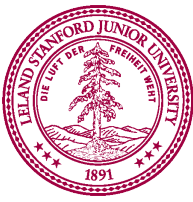
```
import stanford.karel.*;

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



Anatomy of a Program

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

```
public class OurKarelProgram extends Karel {
```

```
    public void run() {
```

```
        move();
        pickBeeper();
        move();
        turnLeft();
        move();
        turnRight();
        move();
        putBeeper();
        move();
```

```
    }
```

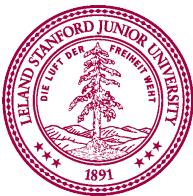
```
    private void turnRight() {
```

```
        turnLeft();
        turnLeft();
        turnLeft();
```

```
    }
```

```
}
```

This is also called a
code block



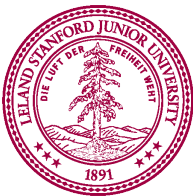
Anatomy of a Program

```
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();
```

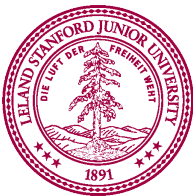
```
        turnLeft();
```

```
        turnLeft();
```

```
    }
```

```
}
```

The run method is “public” so that
Eclipse can call it.



Anatomy of a Program

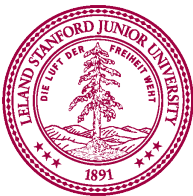
```
import stanford.karel.*;

public class OurKarelProgram extends Karel {

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

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

The turnRight method is “private” to indicate it is only visible to our current program.



Why Study CS?

Joy of Building



Interdisciplinary



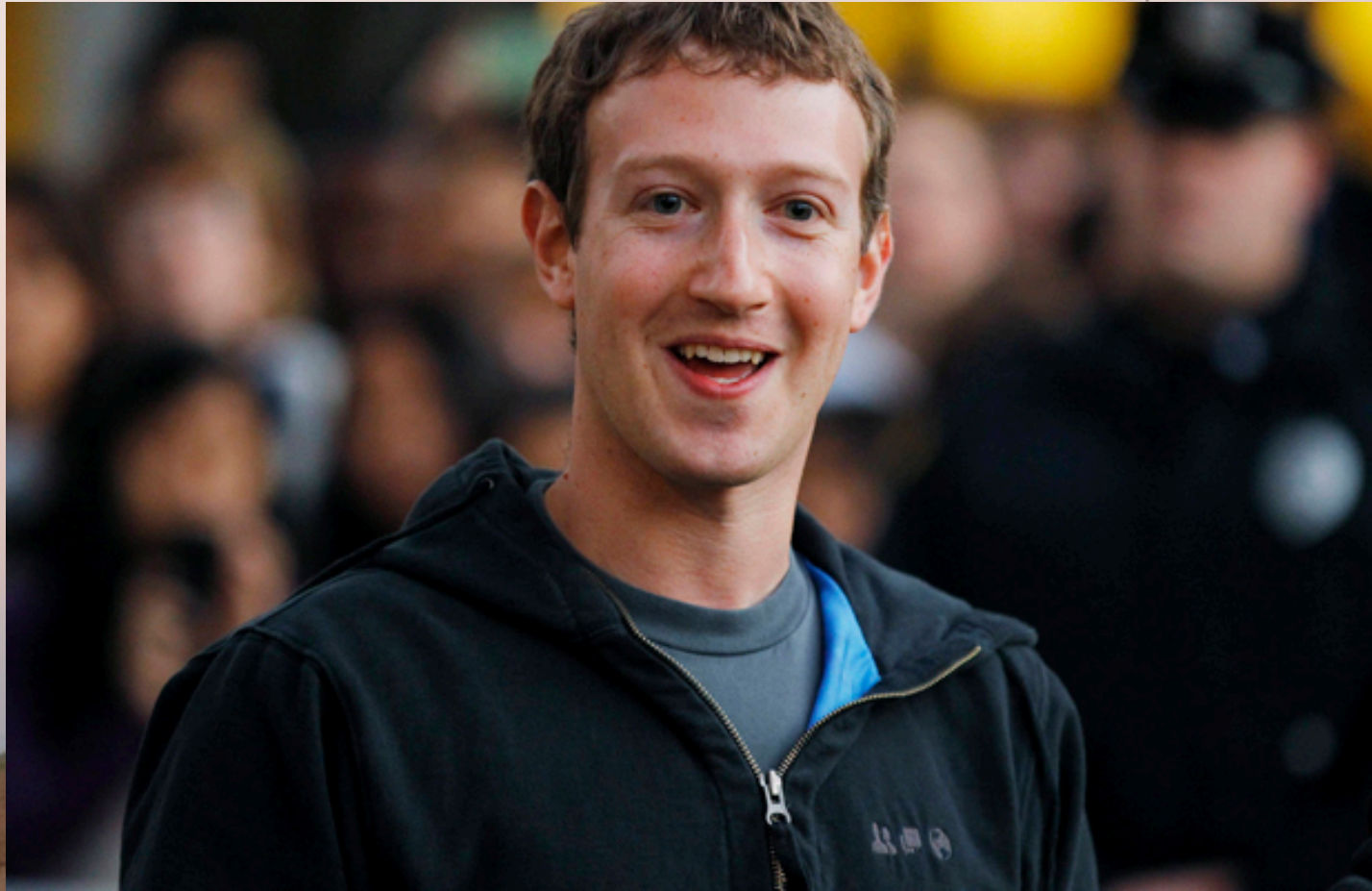
Piech, CS106A, Stanford University



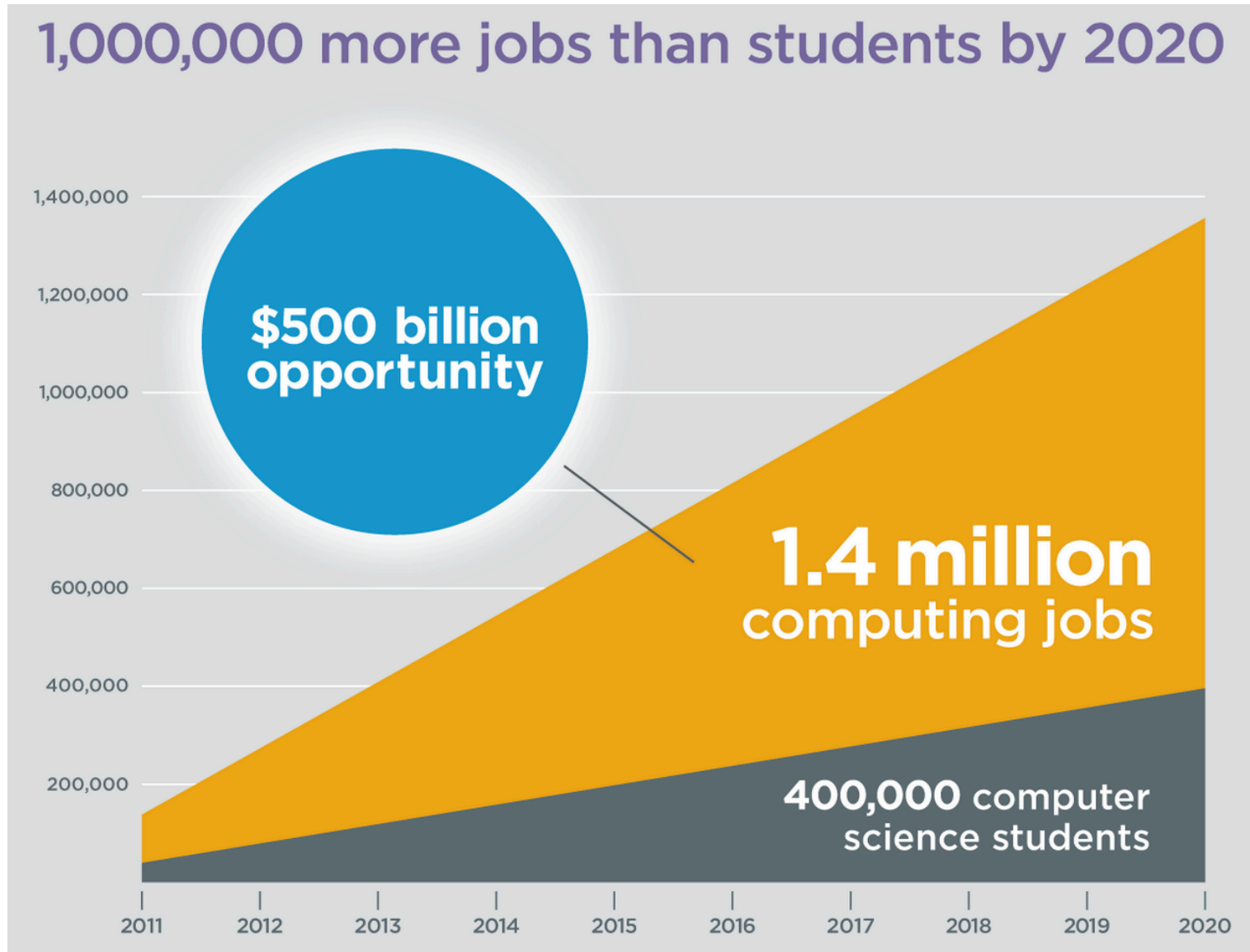
Closest Thing To Magic



Now is the Time



Oh and Its Useful



Everyone is Welcome



The End

A black and white photograph of a spinning top toy, which is a dark, polished, spindle-shaped object with a sharp point at the bottom. It is balanced on its point on a light-colored wooden surface. The background is a blurred wooden wall. The text "The End?" is overlaid in white, sans-serif font, centered horizontally and slightly to the right of the top's center.

The End?