

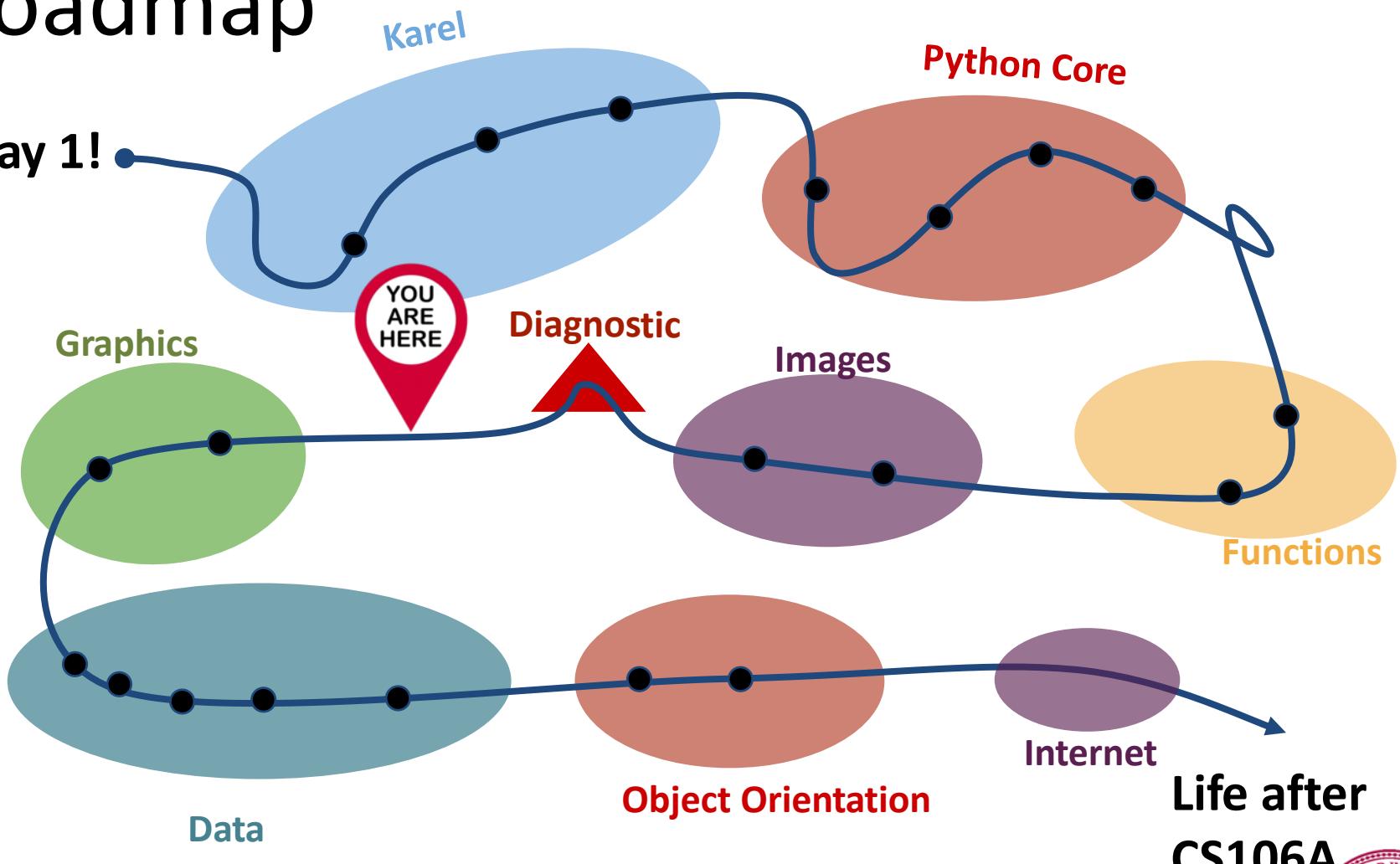


Graphics

**Chris Piech and Mehran Sahami
CS106A, Stanford University**

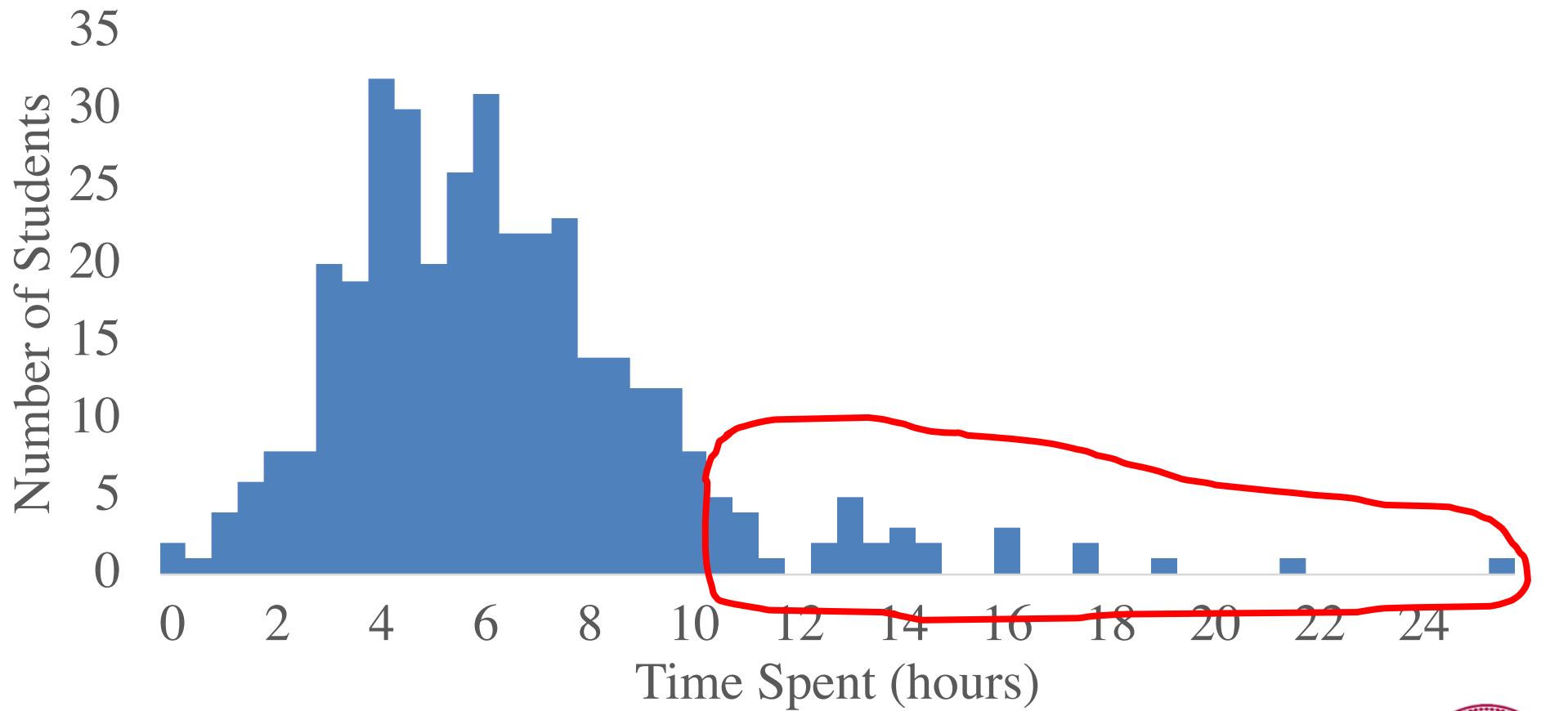
Roadmap

Day 1!



Learn by Doing

Assignment 2



Learning to Program on the Internet

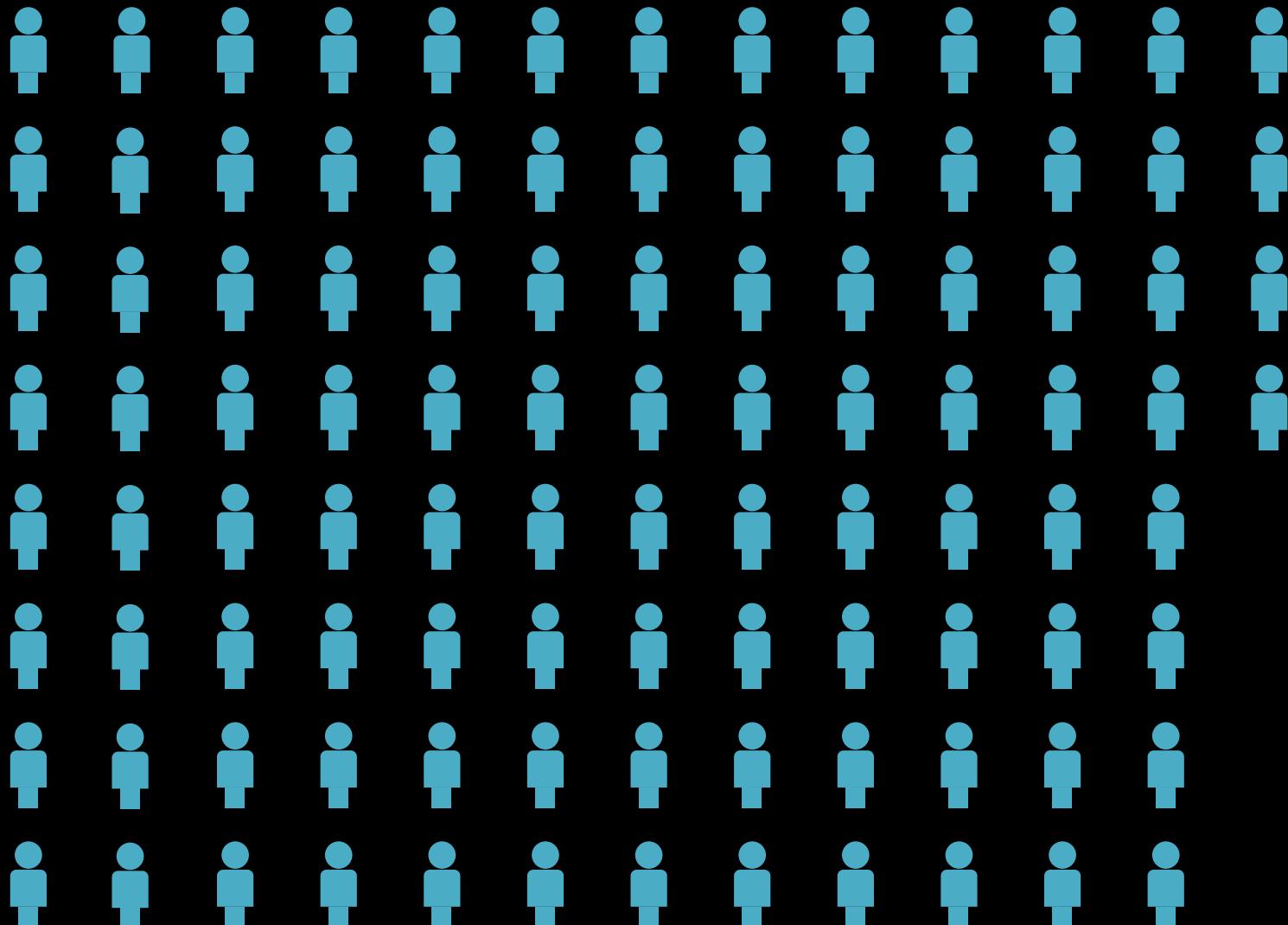


```
when run
repeat until [sunflower v]
  do [if path ahead v
    do [move forward v]
    else [turn left v]
  ]
end
```

Task

Almost a hundred thousand
unique solutions

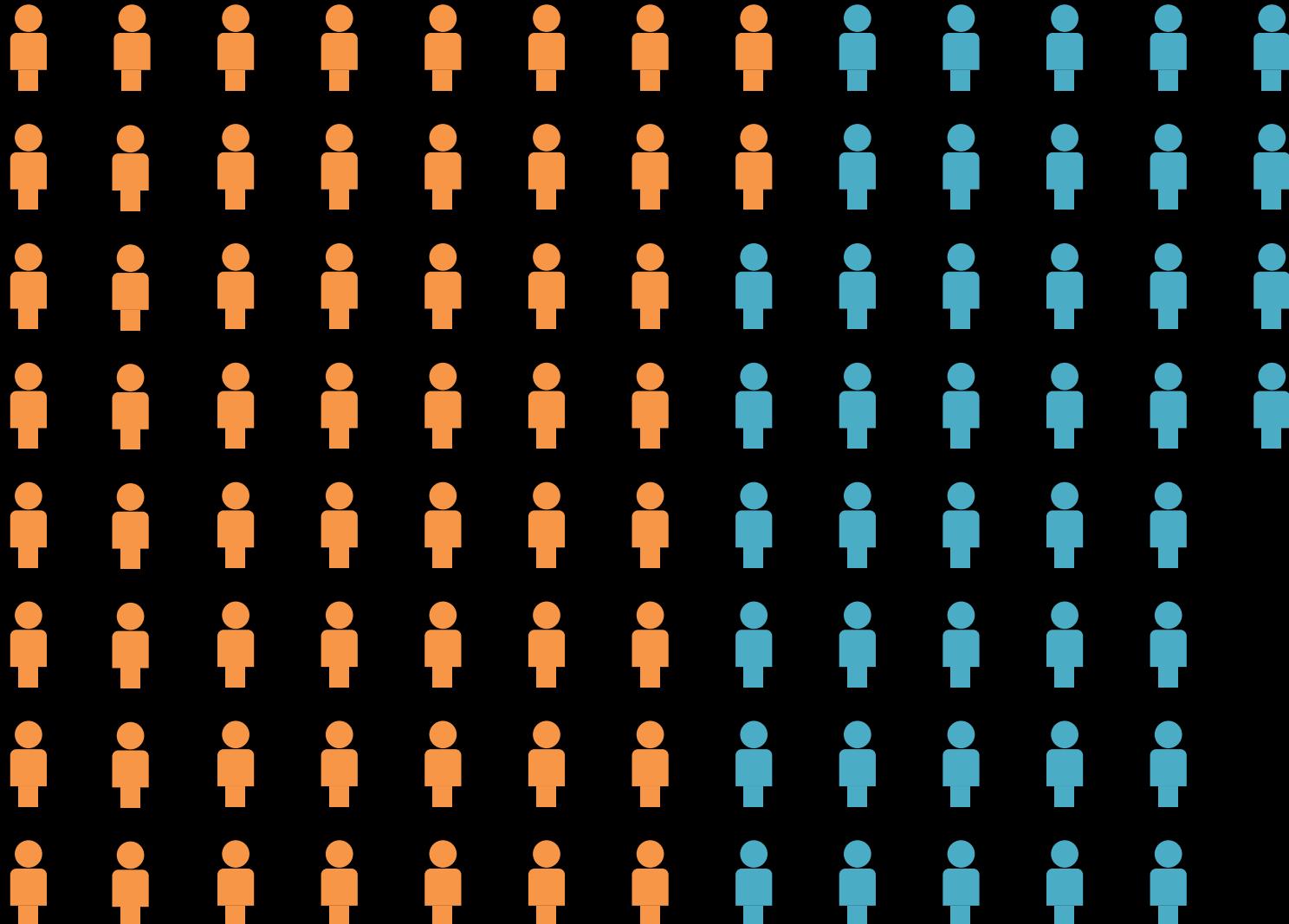
US K-12 Students



= 500,000 learners

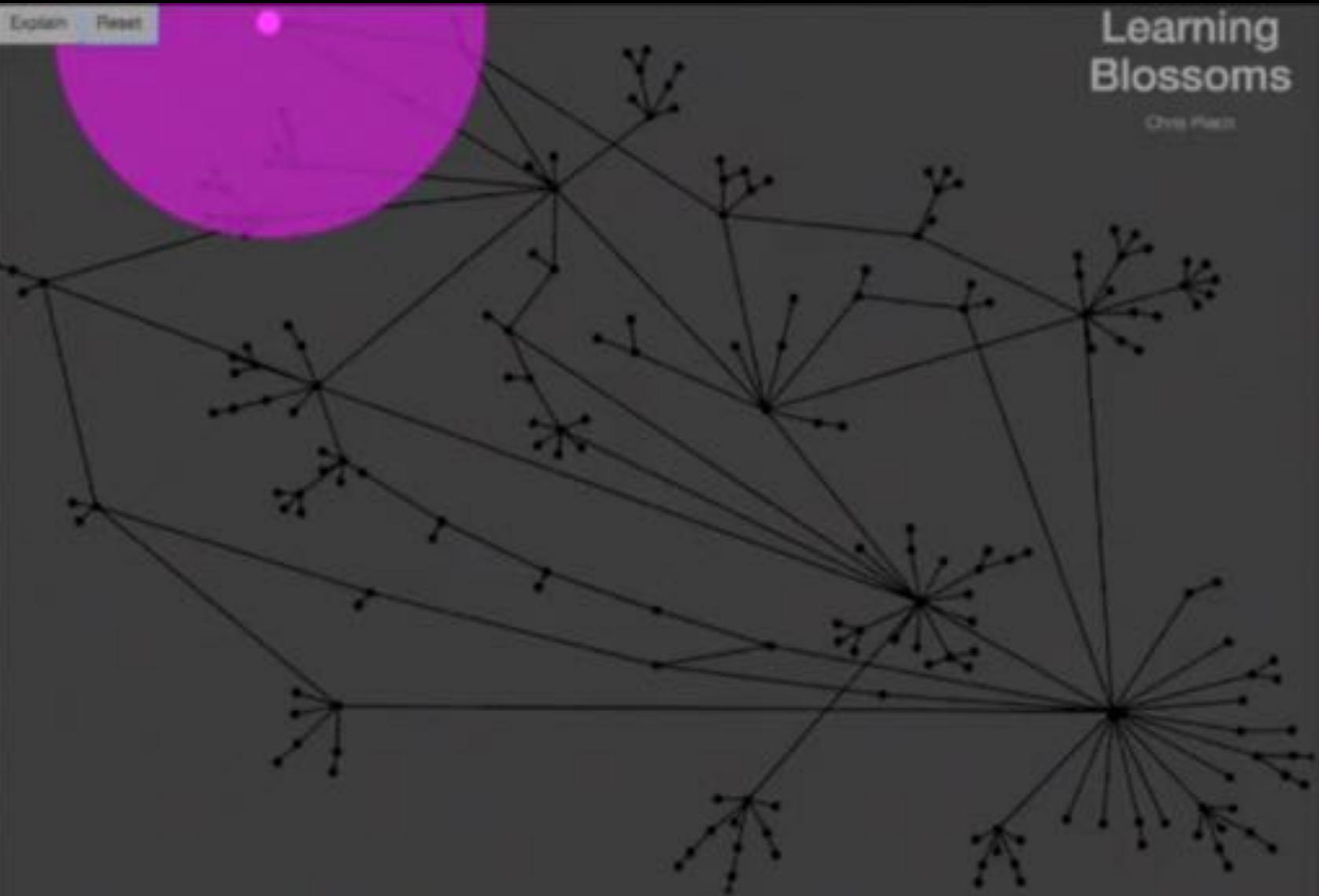


Code.org Students

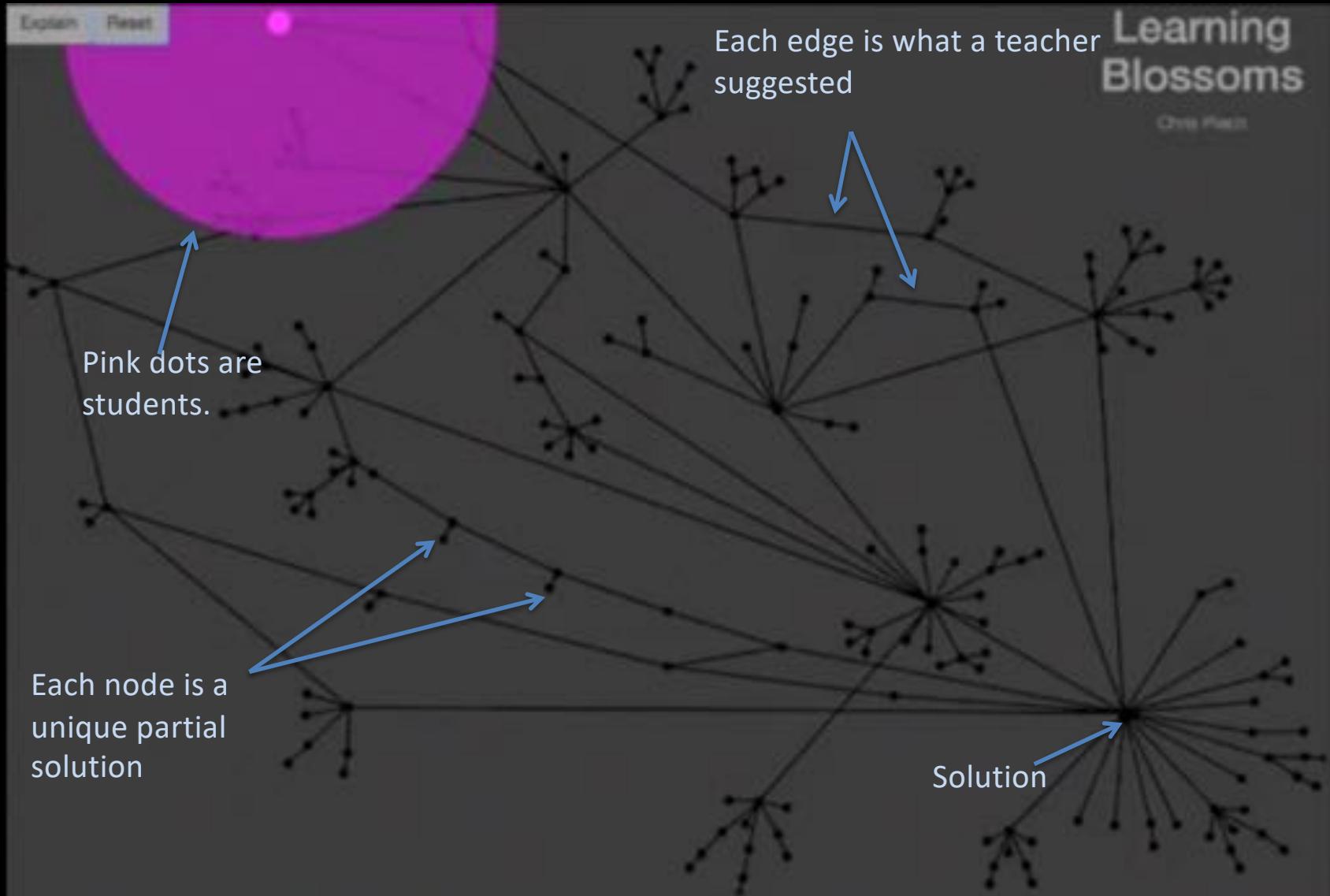


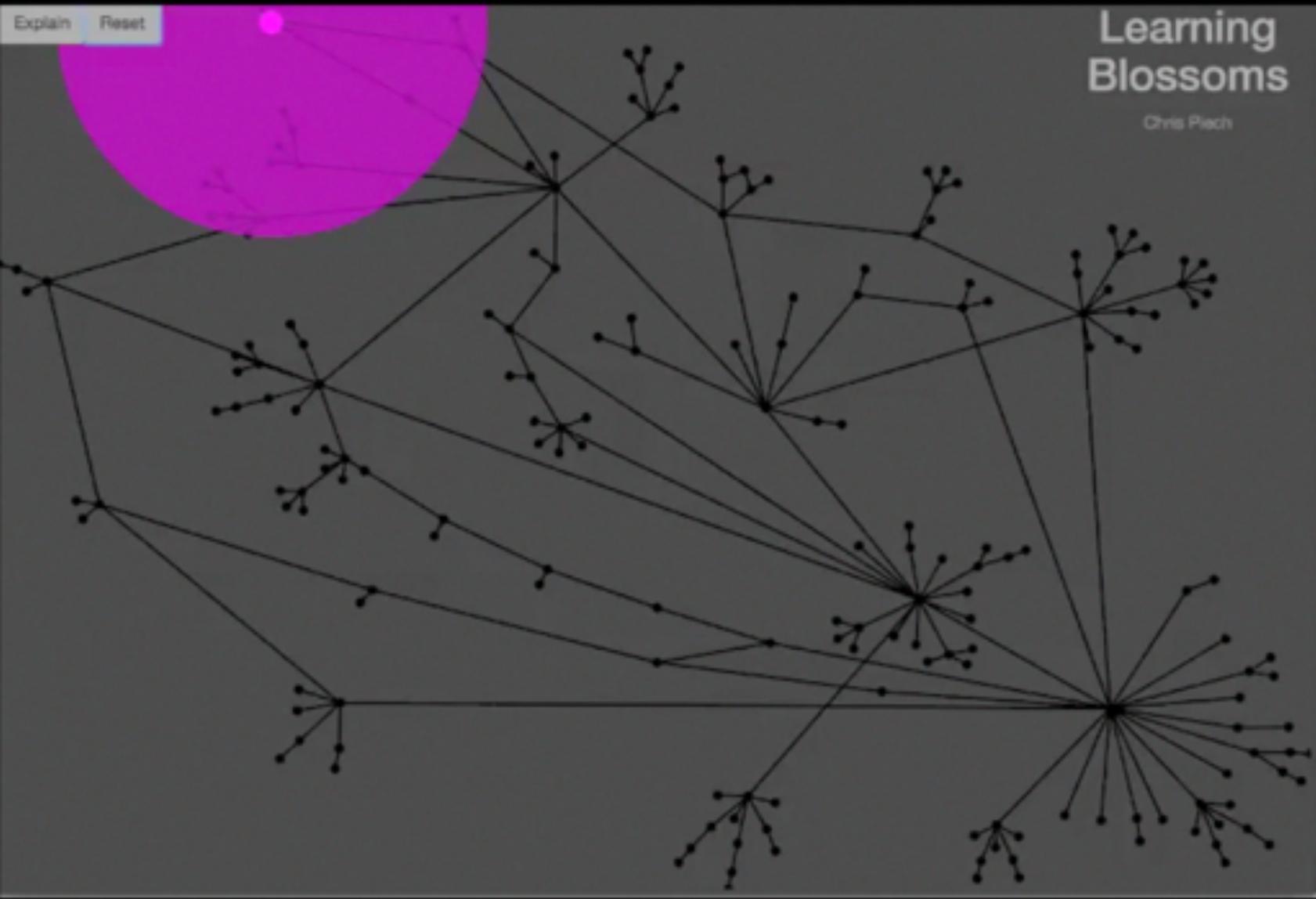
= 500,000 learners





Autonomously Generating Hints by Inferring Problem Solving Policies - Piech, Sahami et al.





Autonomously Generating Hints by Inferring Problem Solving Policies - Piech, Sahami et al.

Desirable Path Algorithm

Poisson Common Path

$$\gamma(s) = \arg \min_{p \in Z(s)} \text{Path Cost}$$
$$\text{Path Cost} = \sum_{x \in p} \frac{1}{\lambda_x}$$

Submission count of partial solution

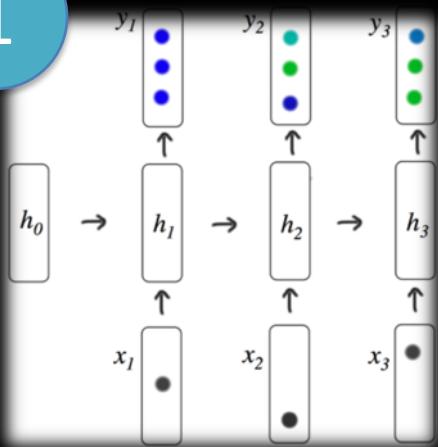
Partial solutions in the path

Paths to solution

Predicted next partial solution

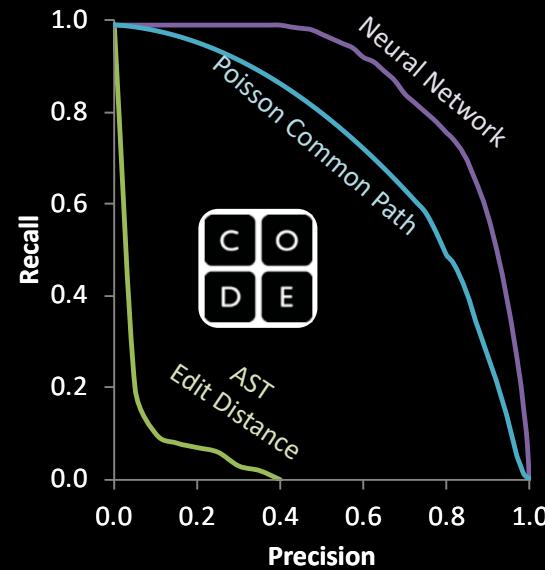
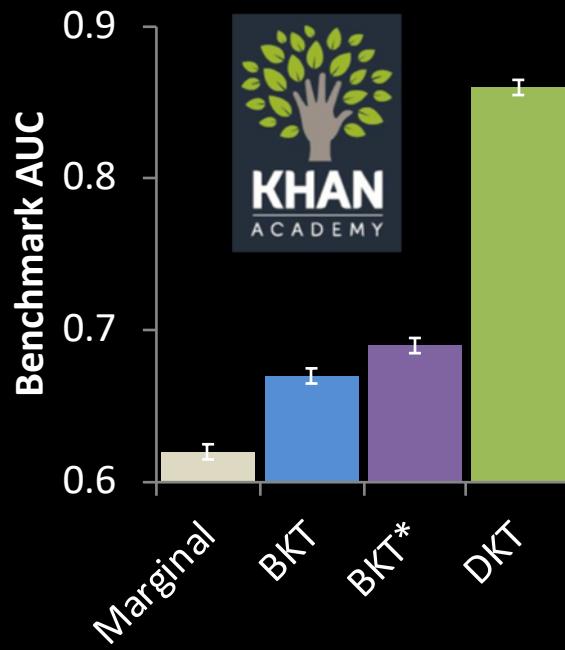
Deep Learning Algorithms

1

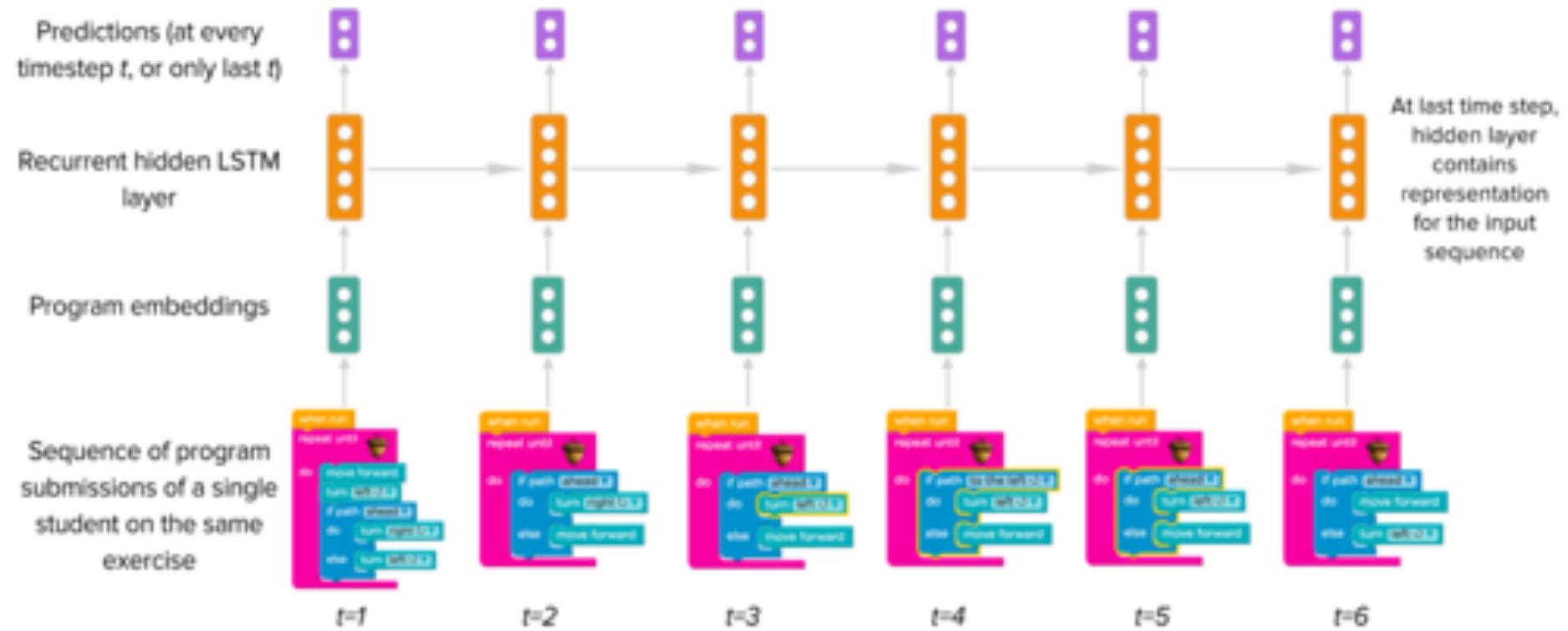


2

Program $\rightarrow \mathbb{R}^n$



Deep Learning on Trajectories

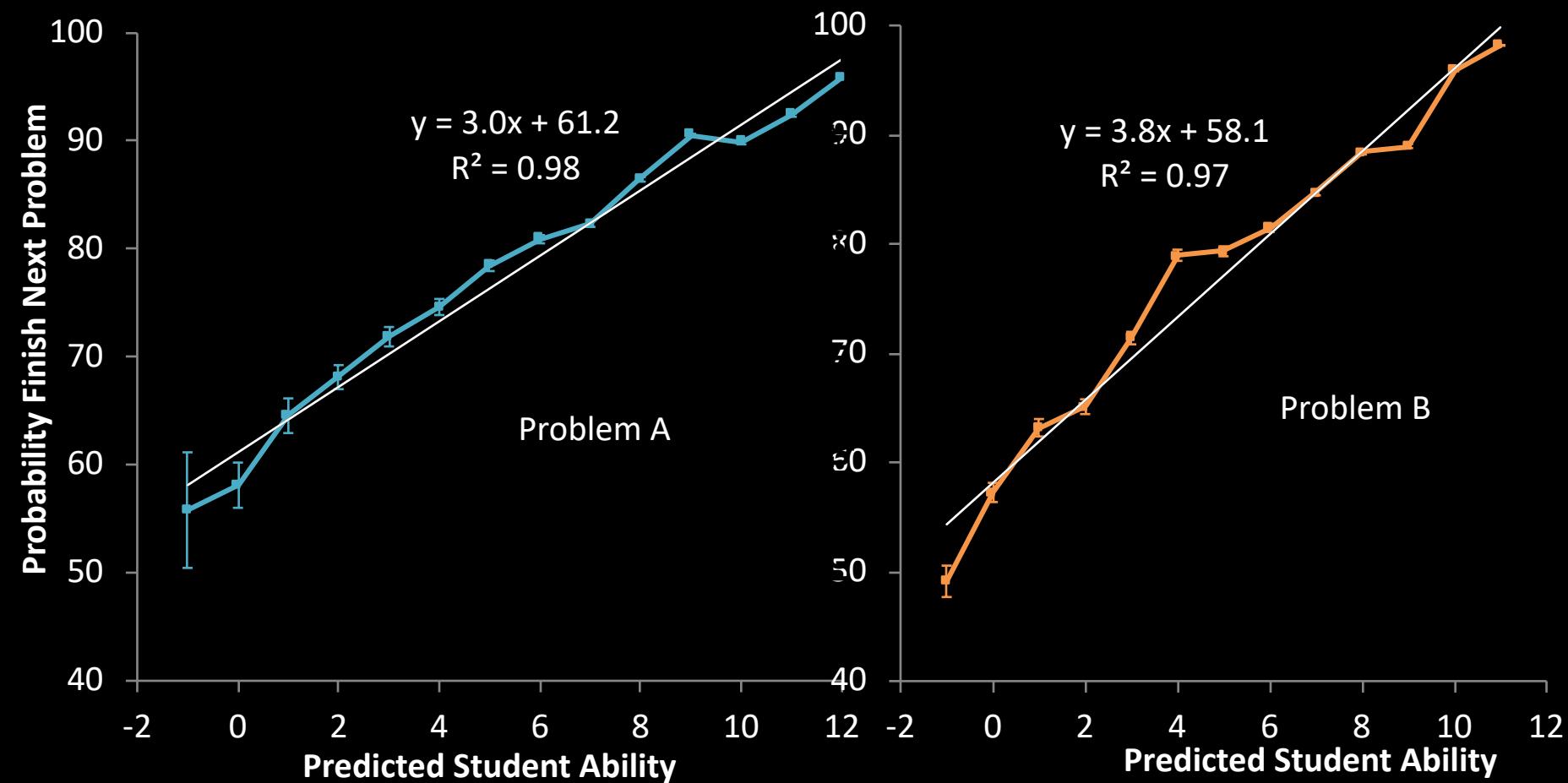


Research in collaboration with Lisa Wang

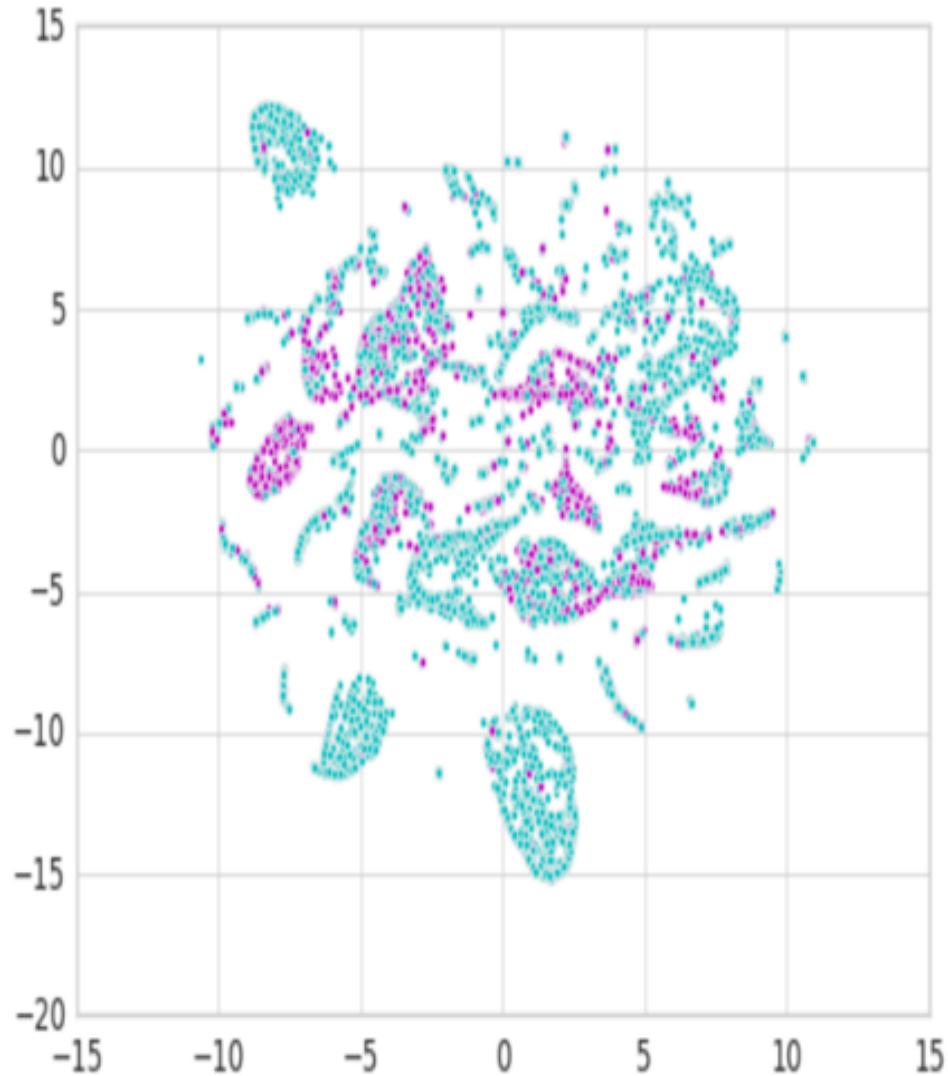
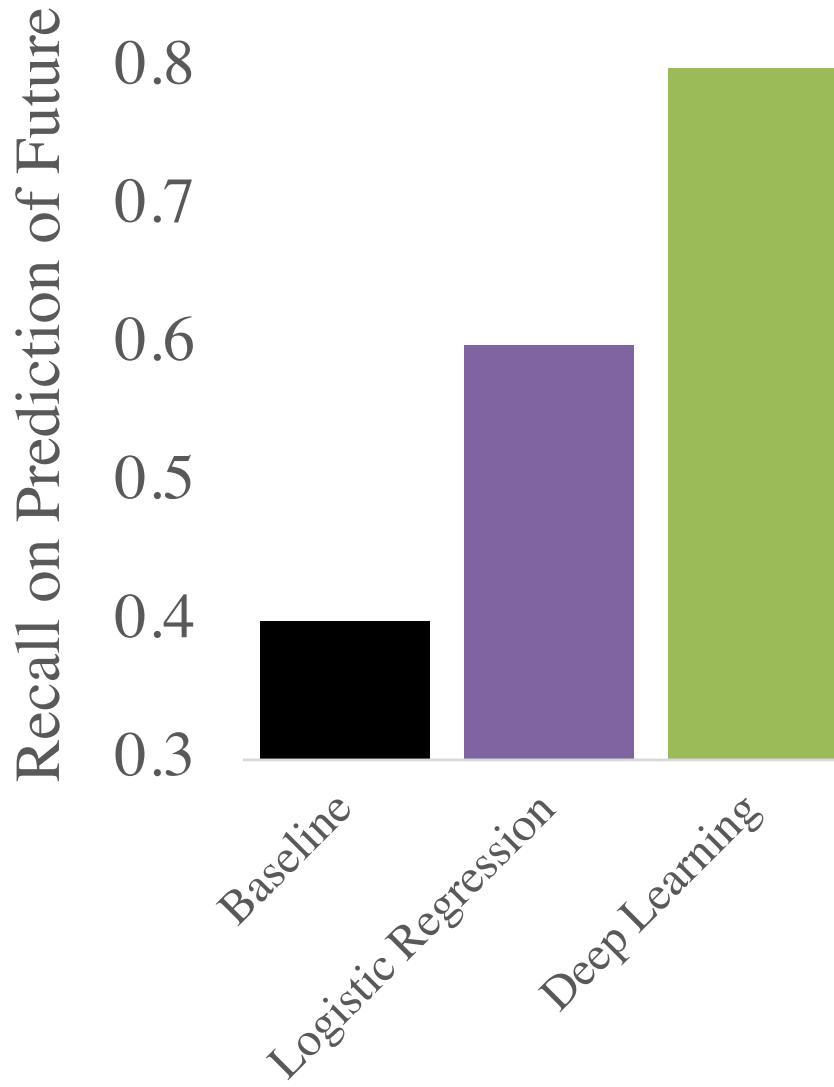
Piech + Sahami, CS106A, Stanford University



Predicts Future Success

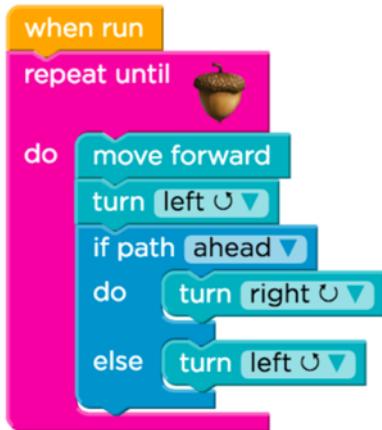


Predicts Future

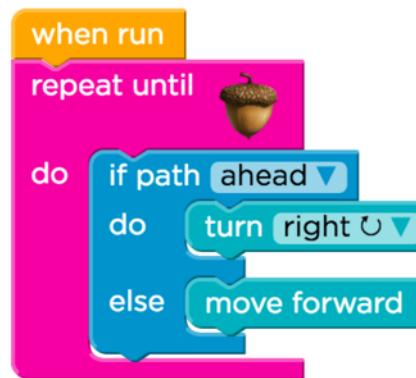


Highly Rates Grit

1. Two compound errors



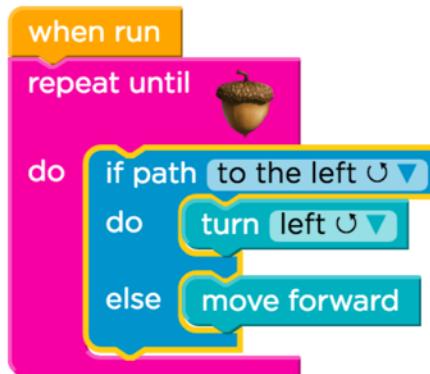
2. Solves first error



3. Starts reasonable attempt



4. Completes attempt



5. Backtracks



6. Finds solution



Review



Piech + Sahami, CS106A, Stanford University





Piech + Sahami, CS106A, Stanford University



Image processing - How is a sepi... X

← → C O stackoverflow.com/questions/1061093/how-is-a-sepia-tone-created

Products Search... Log in Sign up

Our community has been nominated for a Webby Award for Best Community Website - thank you! Show the love and [vote here](#). X

Home How is a sepia tone created? Ask Question

PUBLIC Stack Overflow

Tags Users Jobs

TEAMS What's this? Free 30 Day Trial

What are the basic operations needed to create a sepia tone? My reference point is the perl Imagemagick library, so I can easily use any basic operation. I've tried to quantize (making it grayscale), colorize, and then enhance the image but it's still a bit blurry.

11 Image-processing Imagemagick

share Improve this question follow add a comment

asked Jun 29 '09 at 23:37 user83358 854 ● 3 ● 10 ● 17

4 Answers Active Oldest Votes

24 sample code of a sepia converter in C# is available in my answer here: [What is wrong with this sepia tone conversion algorithm?](#)

The algorithm comes from [this page](#), each input pixel color is transformed in the following way:

```
outputRed = (inputRed * .393) + (inputGreen * .769) + (inputBlue * .189)  
outputGreen = (inputRed * .349) + (inputGreen * .686) + (inputBlue * .168)  
outputBlue = (inputRed * .272) + (inputGreen * .534) + (inputBlue * .131)
```

If any of these output values is greater than 255, you simply set it to 255. These specific values are the values for sepia tone that are recommended by Microsoft.

share Improve this answer follow edited May 23 '17 at 11:54 Community answered Feb 25 '12 at 23:43 Max Galkin 15.8k ● 9 ● 58 ● 108

You will need to use Math.Min likely. I tried doing the check for 255 after those three lines and an error will occur. I was facing the same problem earlier today when I was trying to make a sepia tone for my program... – [BigBug](#) Feb 26 '12 at 6:34

But what if I want something different to change the filter then how can I get to these values ? like my question is how we came to know about these values , do we need to just put different values again and again ? – [AHF](#) Mar 23 '14 at 15:20

add a comment

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



<https://stackoverflow.com/questions/1061093/how-is-a-sepia-tone-created>

Sepia Example

```
def main():
    image_name = input('enter an image name: ')
    image = SimpleImage('images/' + image_name)
    for pixel in image:
        sepia_pixel(pixel)
    image.show()

def sepia_pixel(pixel):
    R = pixel.red
    G = pixel.green
    B = pixel.blue
    pixel.red = 0.393 * R + 0.769 * G + 0.189 * B
    pixel.green = 0.349 * R + 0.686 * G + 0.168 * B
    pixel.blue = 0.272 * R + 0.534 * G + 0.131 * B
```



Sepia Example

```
def main():
    image_name = input('enter an image name: ')
    image = SimpleImage('images/' + image_name)
    for y in range(image.height):
        for x in range(image.width):
            pixel = image.get_pixel(x, y)
            sepia_pixel(pixel)
    image.show()

def sepia_pixel(pixel):
    R = pixel.red
    G = pixel.green
    B = pixel.blue
    pixel.red = 0.393 * R + 0.769 * G + 0.189 * B
    pixel.green = 0.349 * R + 0.686 * G + 0.168 * B
    pixel.blue = 0.272 * R + 0.534 * G + 0.131 * B
```



Sepia Example

```
def main():
    image_name = input('enter an image name: ')
    image = SimpleImage('images/' + image_name)
    for y in range(image.height):
        for x in range(image.width):
            pixel = image.get_pixel(x, y)
            sepia_pixel(pixel)
    image.show()

def sepia_pixel(pixel):
    R = pixel.red
    G = pixel.green
    B = pixel.blue
    pixel.red = 0.393 * R + 0.769 * G + 0.189 * B
    pixel.green = 0.349 * R + 0.686 * G + 0.168 * B
    pixel.blue = 0.272 * R + 0.534 * G + 0.131 * B
```



Sepia Example

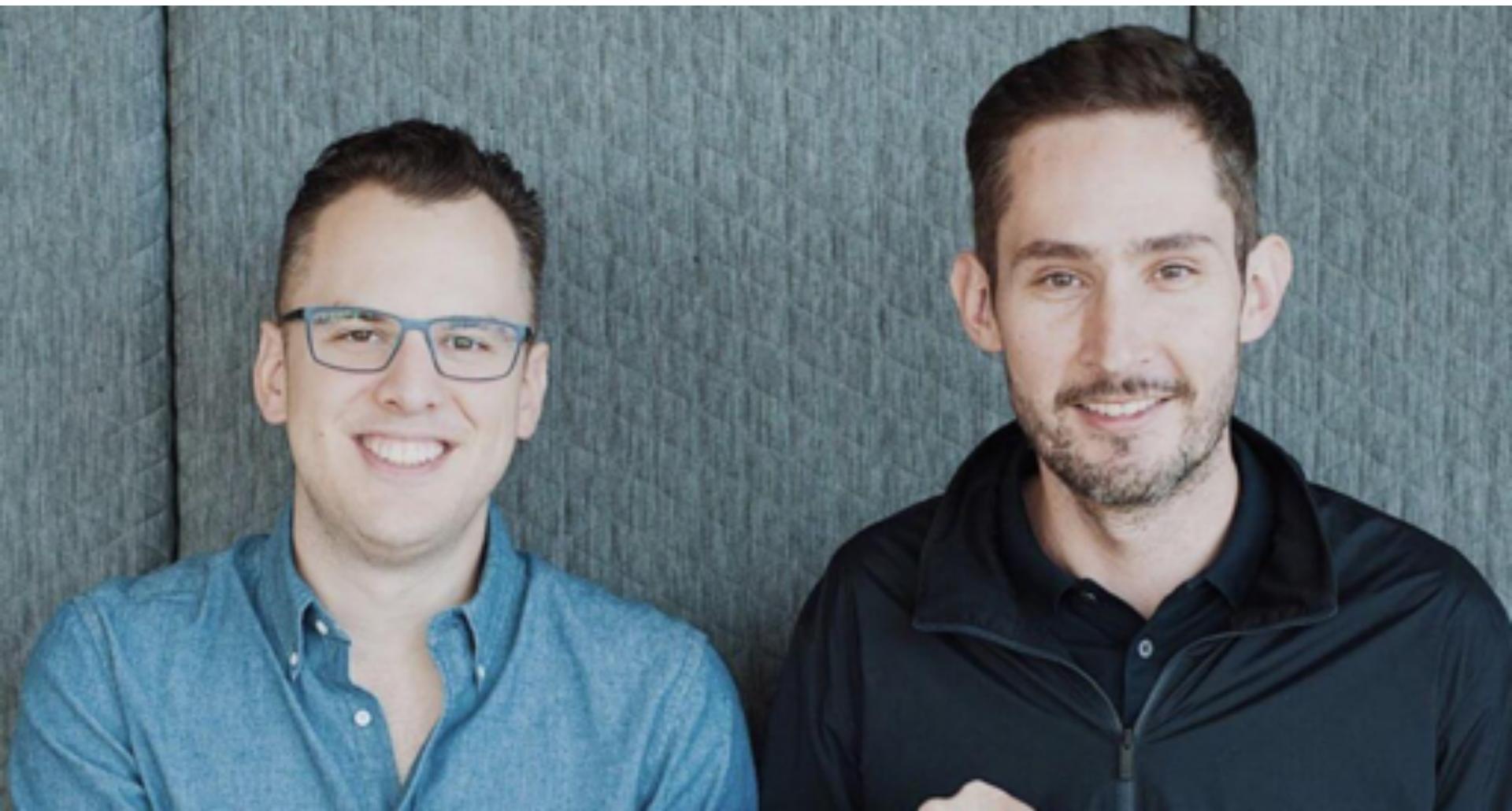
```
def main():

    for y in range(600):
        for x in range(800):
            print(x, y)
```



Mike Krieger

Kevin Systrom



Piech + Sahami, CS106A, Stanford University



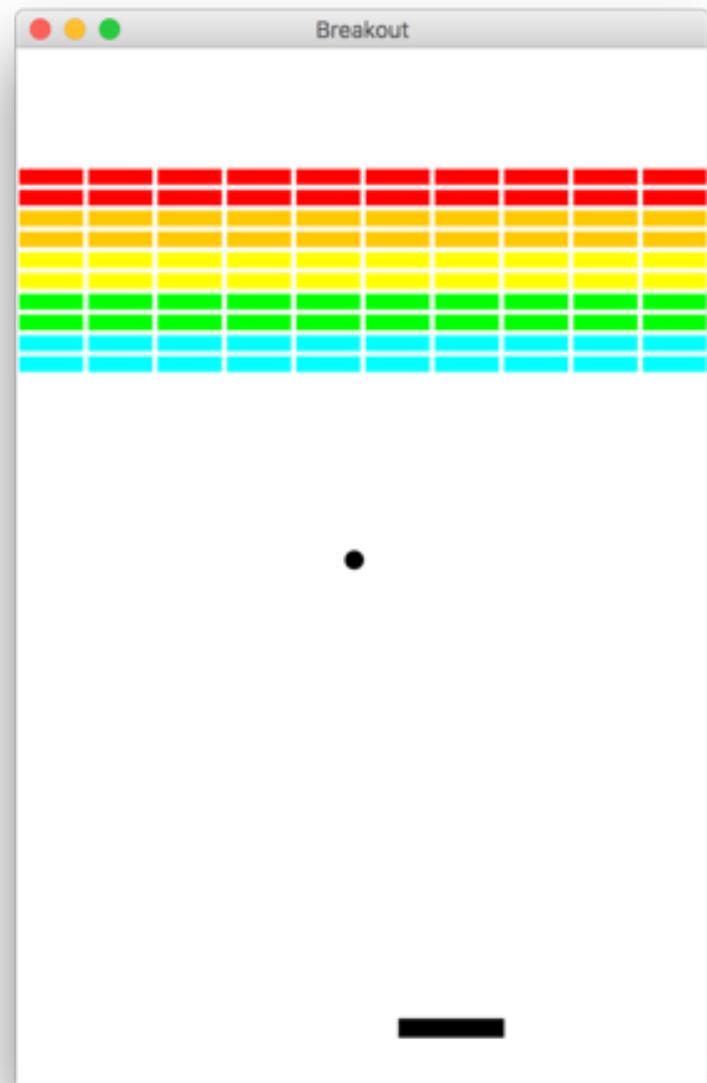
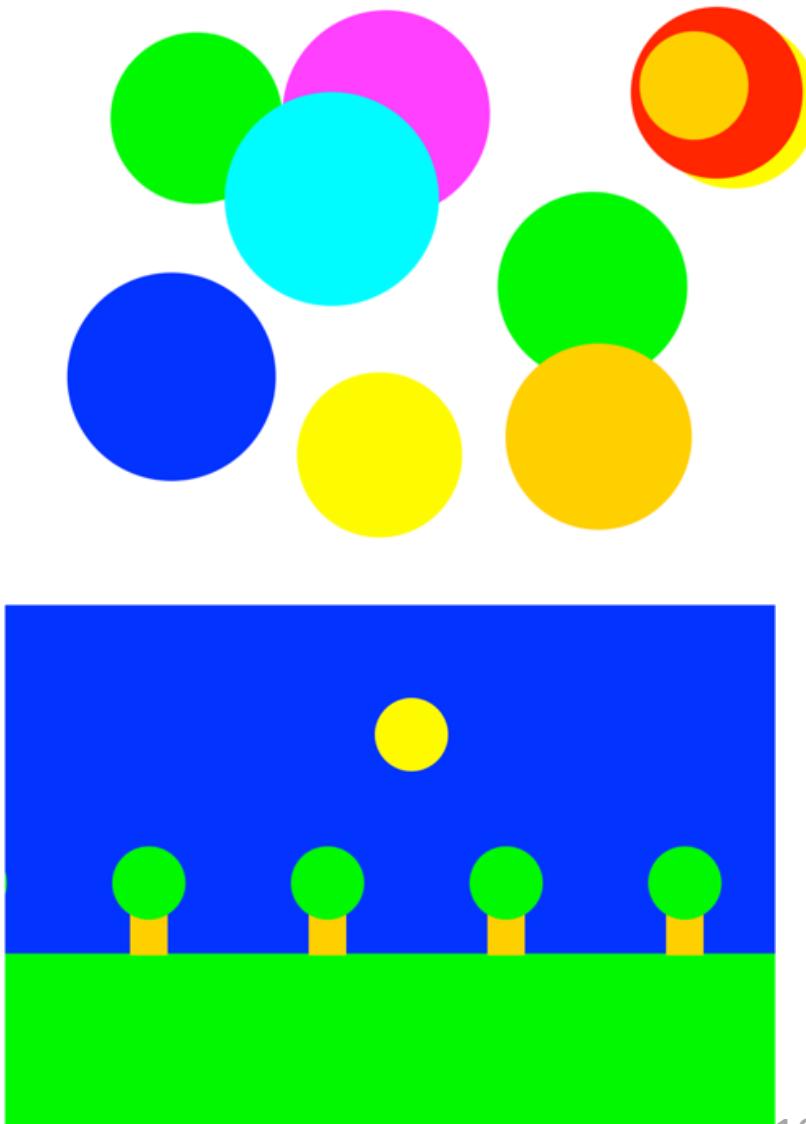
End Review

Today's Goal

1. How do I draw shapes?



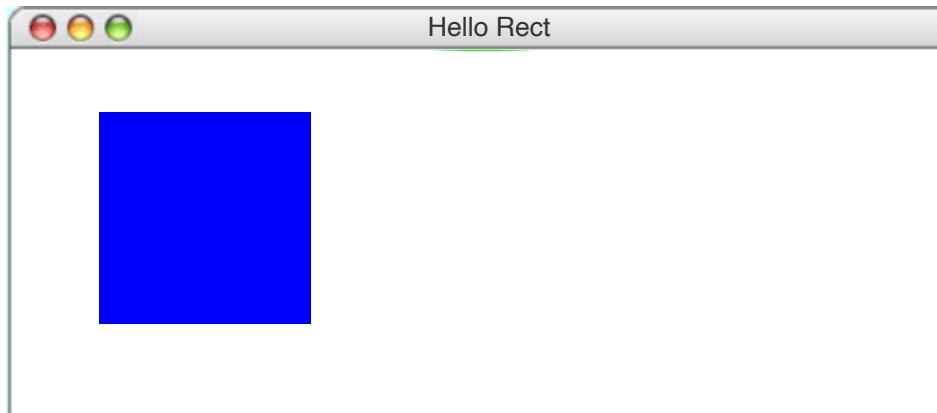
Graphics Programs



Draw a Rectangle

the following `main` method displays a blue square

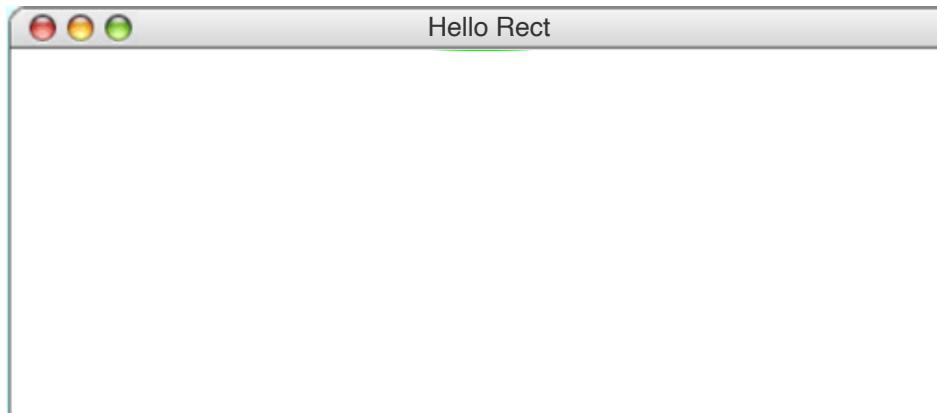
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
    canvas.mainloop()
```



Draw a Rectangle

the following `main` method displays a blue square

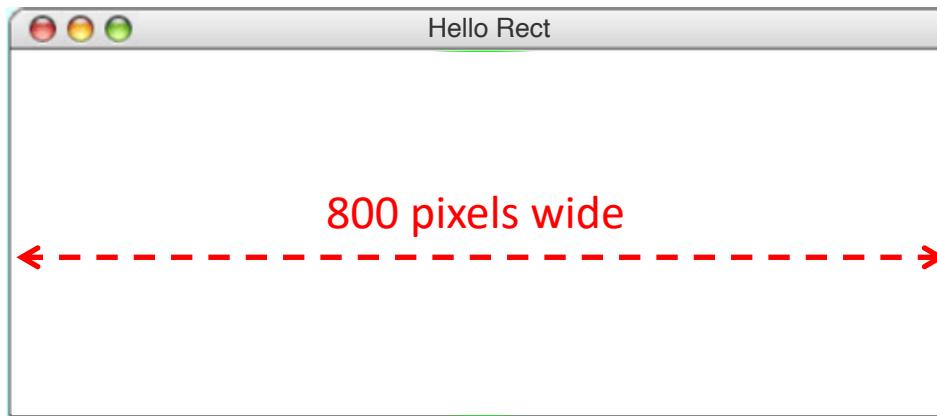
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

the following `main` method displays a blue square

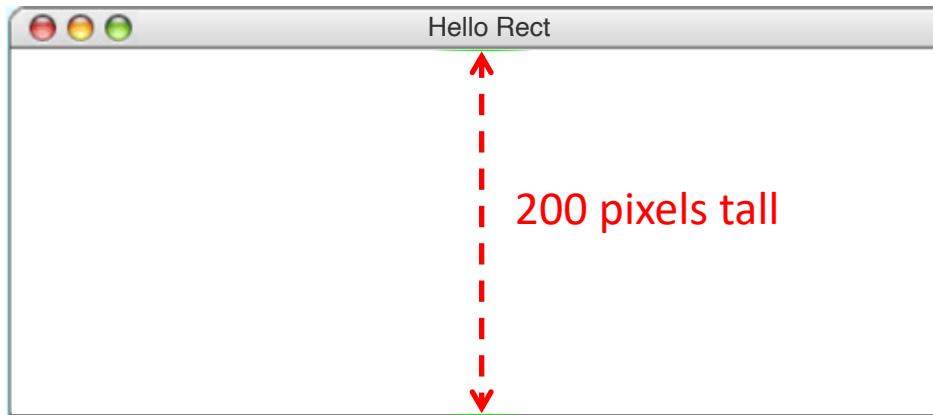
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

the following `main` method displays a blue square

```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

the following `main` method displays a blue square

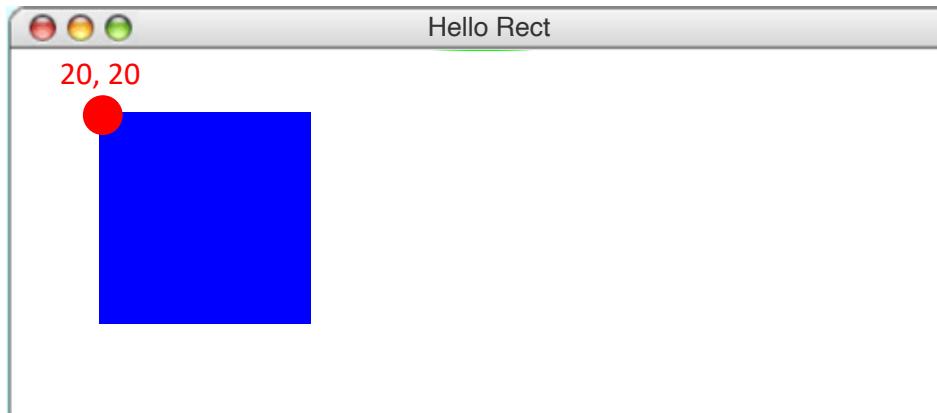
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

the following `main` method displays a blue square

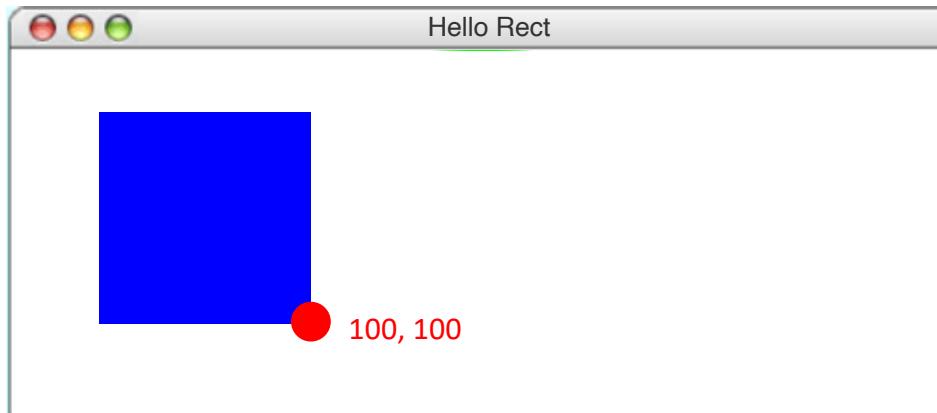
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
```



Draw a Rectangle

the following `main` method displays a blue square

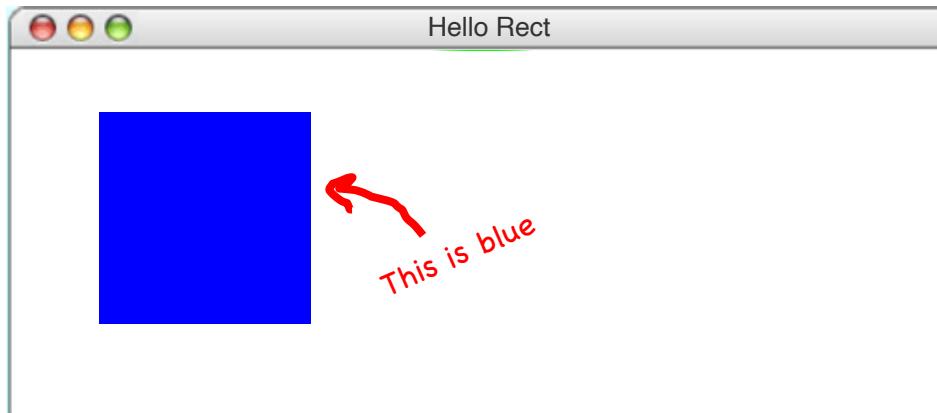
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
```



Draw a Rectangle

the following `main` method displays a blue square

```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
```



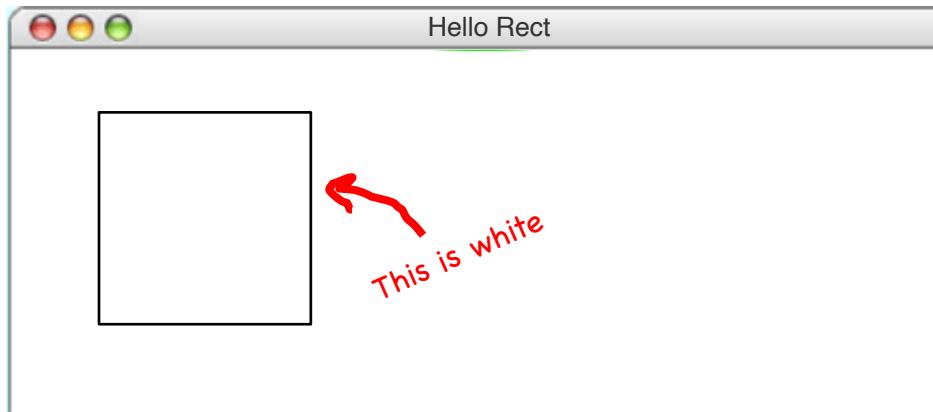
Aside: Named Arguments
This argument is named as filled. It allows functions to have arguments which you can ignore if you want a default value.



Draw a Rectangle

the following `main` method displays a blue square

```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100)
```



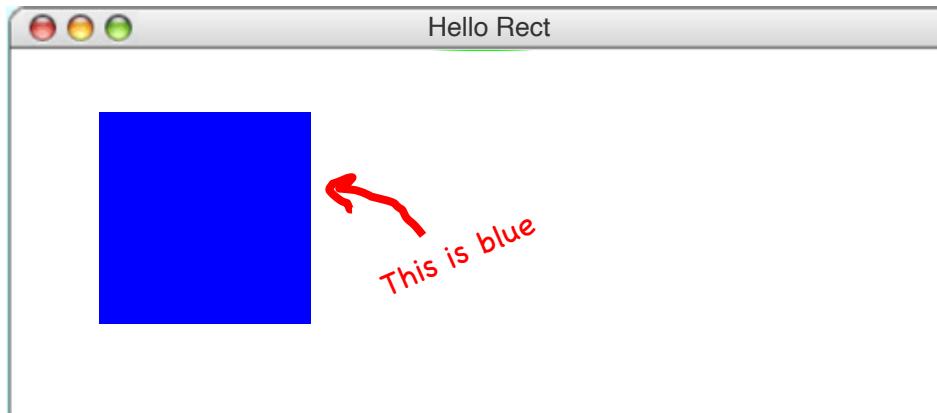
Aside: Named Arguments
This argument is named as filled. It allows functions to have arguments which you can ignore if you want a default value.



Draw a Rectangle

the following `main` method displays a blue square

```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
```



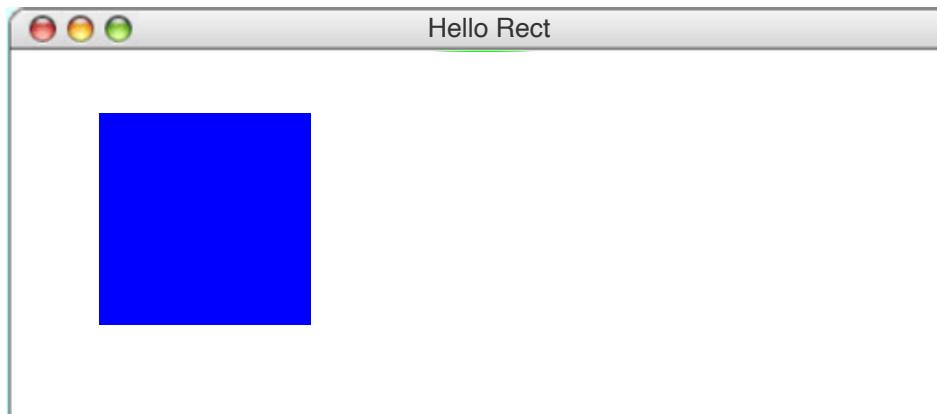
Aside: Named Arguments
This argument is named as filled. It allows functions to have arguments which you can ignore if you want a default value.



Draw a Rectangle

the following `main` method displays a blue square

```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
    canvas.mainloop()
```



TK Natural Graphics



Graphics Coordinates

0,0

x 40,20

x 120,40

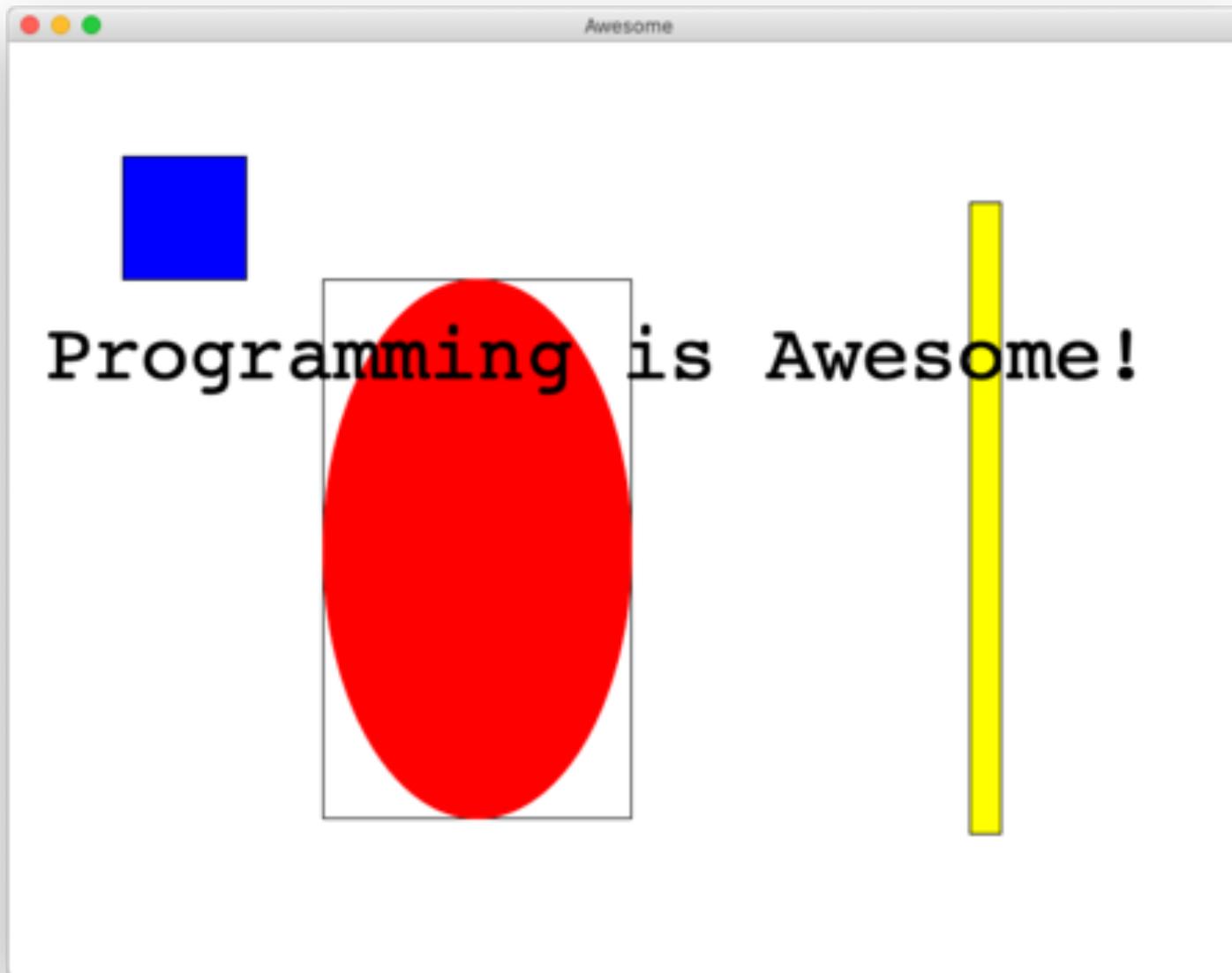
x 40,120

CANVAS_WIDTH

CANVAS_HEIGHT



Rectangles, Ovals, Text



- `canvas.create_line()`
- `canvas.create_oval()`
- `canvas.create_text()`



- `canvas.create_line(x1, y1, x2, y2)`
- `canvas.create_oval()`
- `canvas.create_text()`



- `canvas.create_line(x1, y1, x2, y2)`

- `canvas.create_oval()`
- `canvas.create_text()`



The first point of the
line is `(x1, y1)`



- `canvas.create_line(x1, y1, x2, y2)`
- `canvas.create_oval()`
- `canvas.create_text()`



The second point of the
line is `(x2, y2)`



- **canvas.create_line(x1, y1, x2, y2)**

- **canvas.create_oval()**

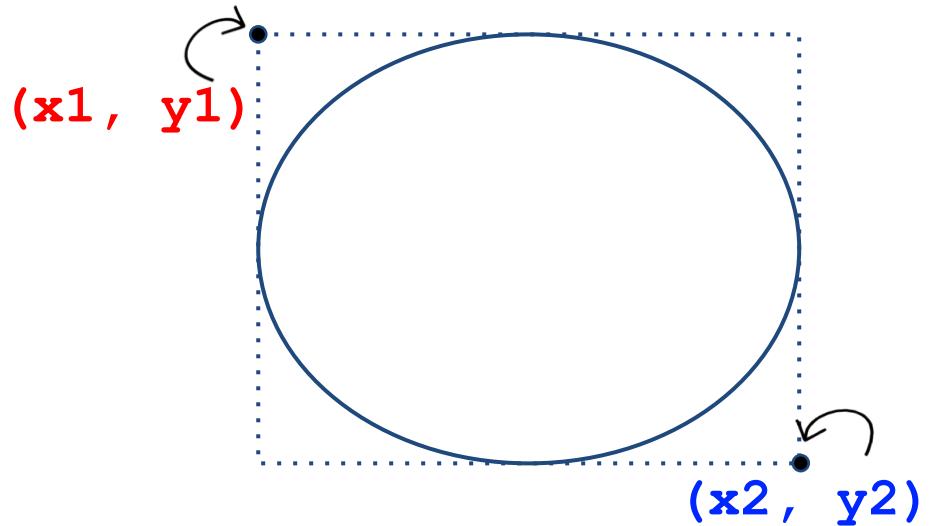
- **canvas.create_text()**



- `canvas.create_line()`
- `canvas.create_oval(x1, y1, x2, y2)`
- `canvas.create_text()`



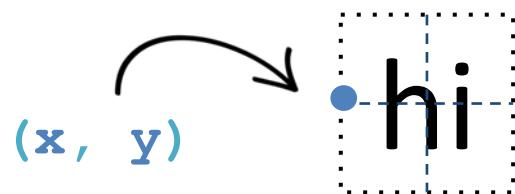
- `canvas.create_line()`
- `canvas.create_oval(x1, y1, x2, y2)`
- `canvas.create_text()`



- `canvas.create_line()`
- `canvas.create_oval()`
- `canvas.create_text(x, y, text='hi')`



- `canvas.create_line()`
- `canvas.create_oval()`
- `canvas.create_text(x, y, text='hi', anchor='w')`



Pedagogy

The screenshot shows a web browser window for CS106A. The URL is localhost:8000/examples/. The page title is "Programming is Awesome". A red circle highlights the "Examples" tab in the navigation bar. Below the title, it says "Written by Chris Peich". The main content displays a screenshot of a Java application window titled "ProgrammingAwesome". Inside the window, there is a blue square at the top left, a yellow vertical rectangle on the right, and the text "Programming is Awesome!" in the center. A red oval highlights the red oval shape in the text.

Here is the corresponding code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ProgrammingAwesome extends JPanel {
    // draw the screen in the picture above
    public void paint(Graphics g) {
        // half the height of the screen.
        double centerY = getPreferredSize().height / 2;

        // make and add a blue square
        Rectangle blueSquare = new Rectangle(0, 0, 80, 80); // width and height are 80
        blueSquare.setCenterX((centerX - 80) / 2); // make the square blue
        blueSquare.setFilled(true); // fill the square
        add(blueSquare, 70, 70); // add the square to the screen

        // make and add a yellow rectangle
        Rectangle yellowRect = new Rectangle(0, 0, 80, 80);
        yellowRect.setCenterX((centerX - 80) / 2);
        yellowRect.setFilled(true); // make the rectangle yellow
        add(yellowRect, 80, 80);

        // make and add a red oval
        OvalShape redOval = new OvalShape(50, 50); // width and height
        redOval.setCenterX((centerX - 50) / 2);
        redOval.setFilled(true); // make the oval red
        add(redOval, 200, 200);

        // make and add a rectangle which fits around the red oval
        Rectangle redOutline = new Rectangle(120, 120, 100, 100);
        add(redOutline, 200, 200);
    }
}
```

The screenshot shows a web browser window for CS106A. The URL is localhost:8000. The page title is "CS106A". A red circle highlights the "Stanford Java Lib" link in the "RESOURCES" sidebar.

CS106A: Programming Methodologies
Stanford University: Winter 2019
Monday, Wednesday, Friday 3:30pm - 4:20pm in NVIDIA Auditorium

RESOURCES

- CS106A Info
- Course Schedule
- Style Guide
- Piazza
- Eclipse
- Karel Book
- Stanford Java Lib
- Blank Karel Project
- Videos
- LaiR Hours

ANNOUNCEMENTS

Section Assignments & Late Signups
3 days ago

For those who submitted section preferences by 5PM on Sunday, we are now making section assignments; as a reminder, sections start this week! You can change your assigned section via cs198.stanford.edu). If you were unable to submit by the 5PM Sunday deadline, the late signup form is available on the course website once you log in.

If you would like to individually switch to a different section because of other constraints please request a swap via cs198.stanford.edu. You can use this form to join the section of a partner who is another section. If you have any questions, email Brahm.

ASSIGNMENTS

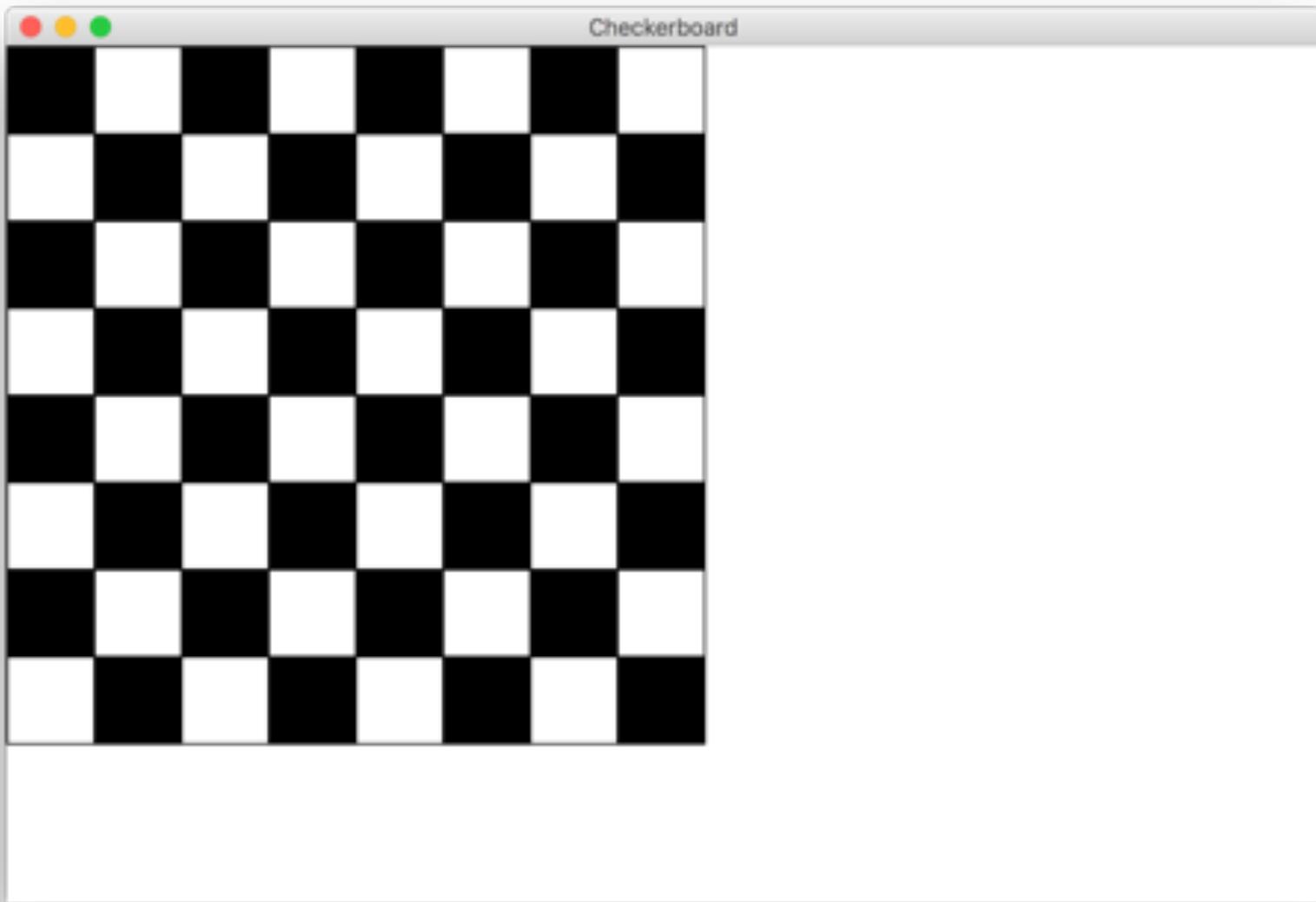
- Asn 2: Simple Java
- Asn 1: Karel

EXAMS

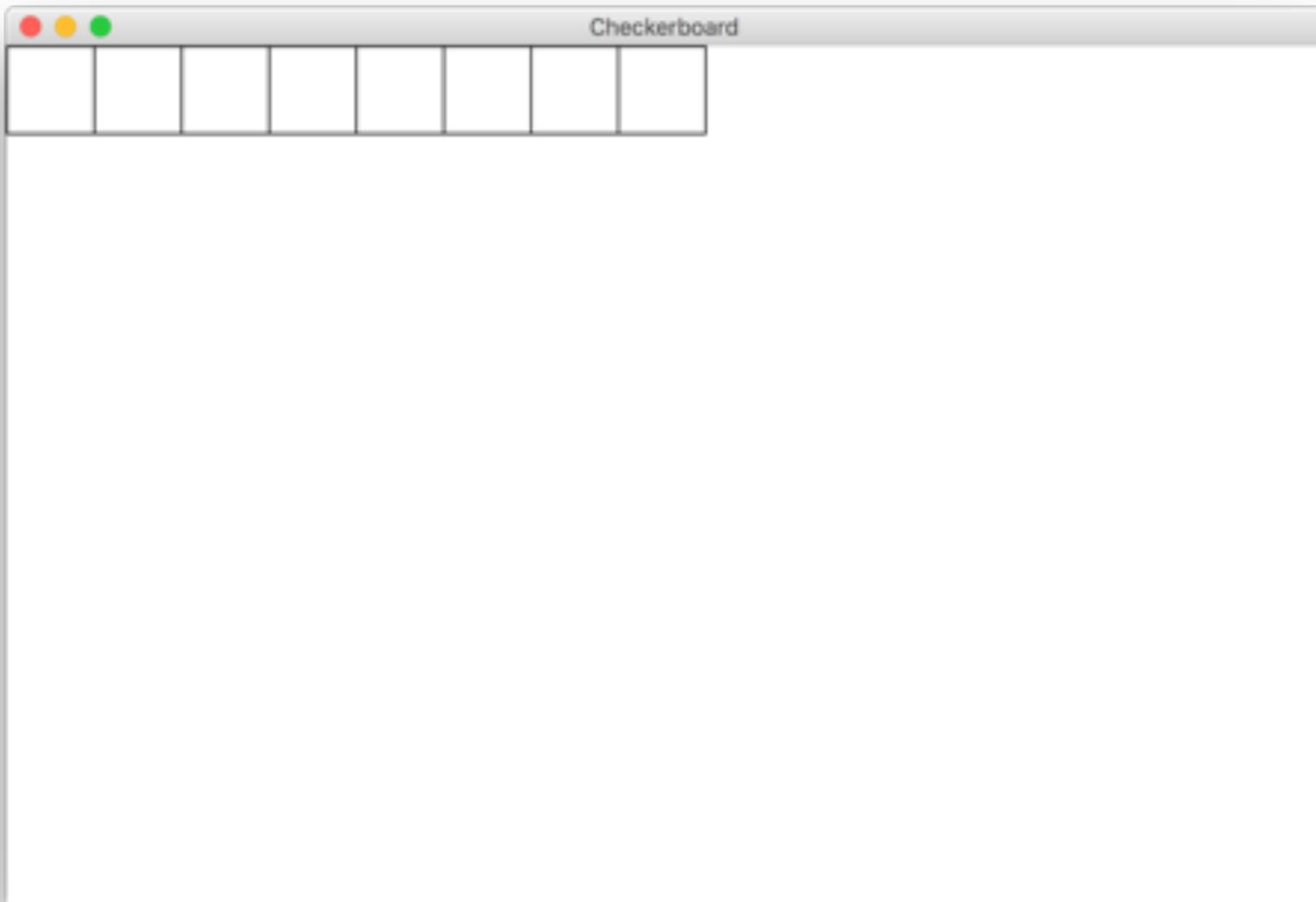
Section Signups Open Until 5PM Sunday 1/13
8 days ago

Section signups are now open! Click on the "Section" tab at the top of the page to "Section Signup" to submit your preferences. As a reminder, signups will be processed on a **come first serve** basis. As such, you may modify your preferences any time before the Sunday 5PM deadline. We will notify you of your section assignment early next week.

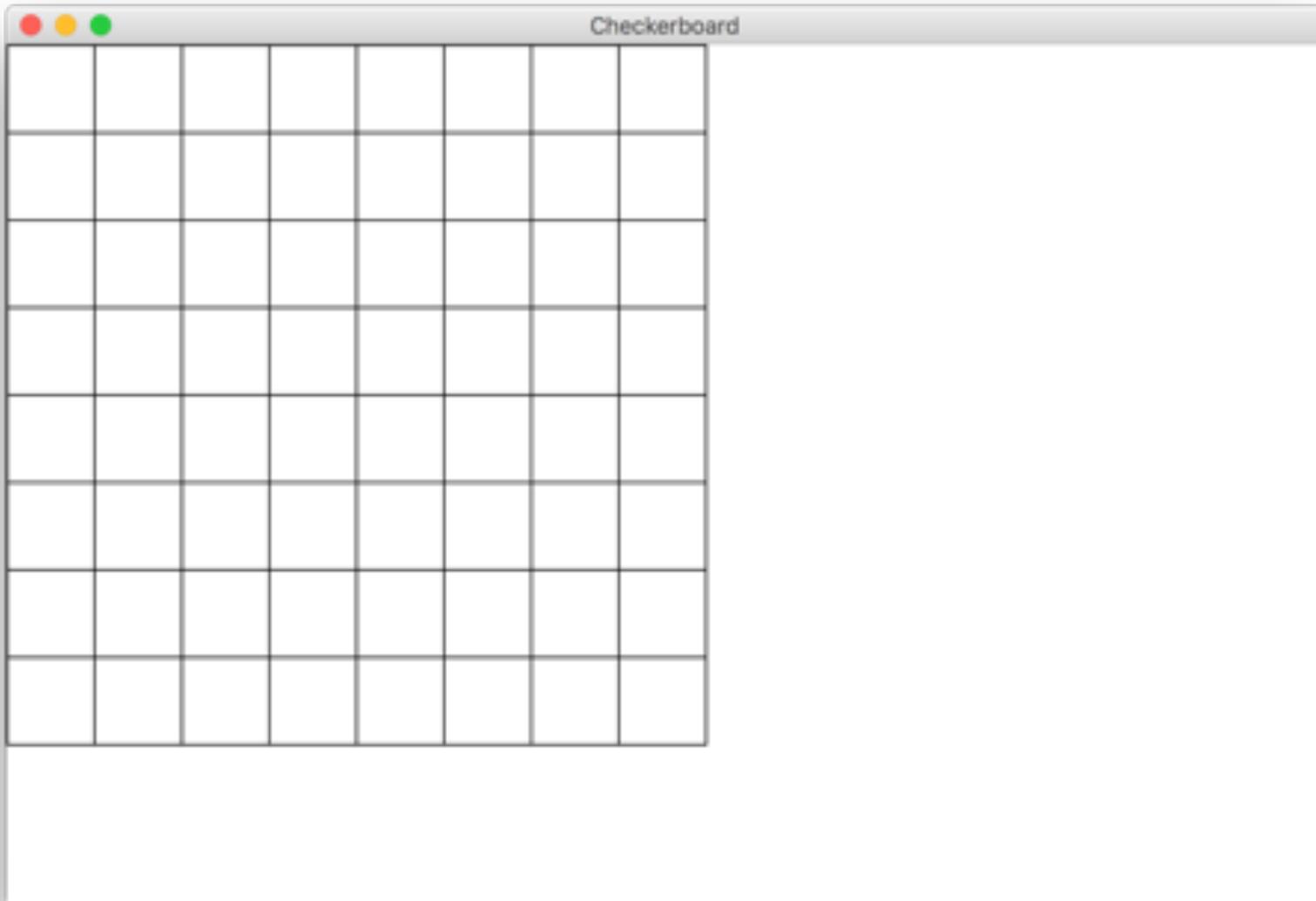
Goal



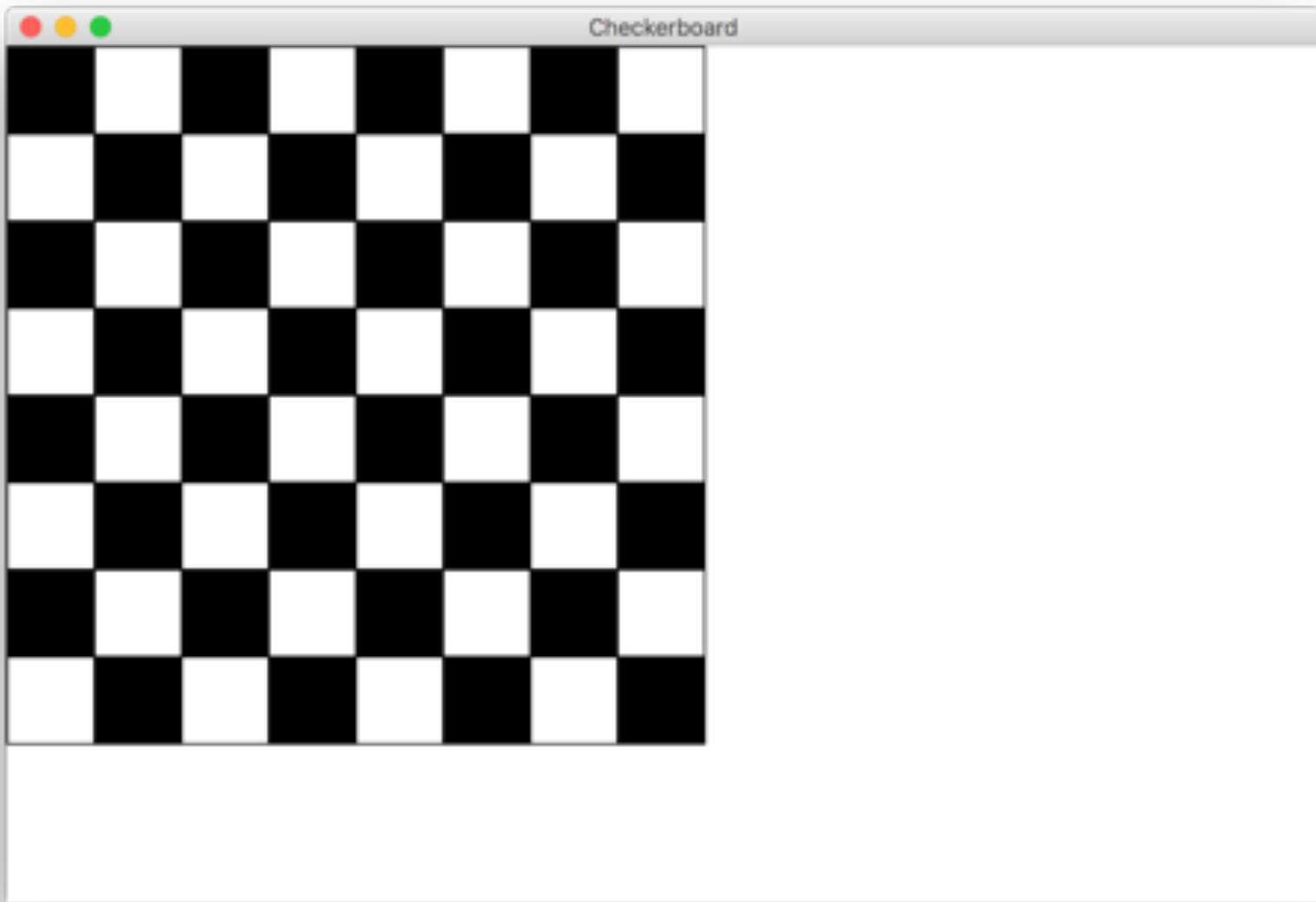
Milestone 1

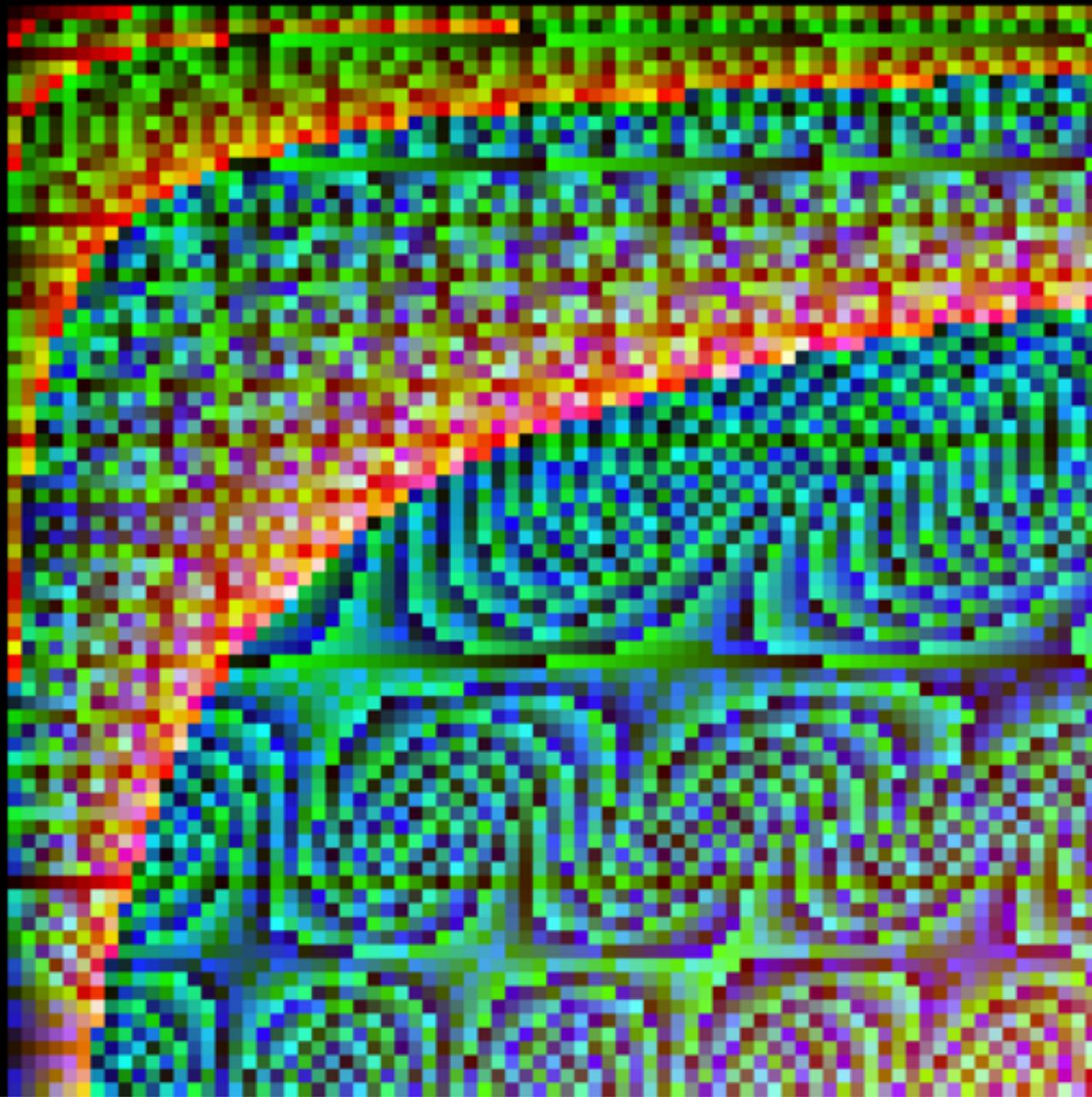


Milestone 2



Milestone 3





Teaser for next week...

Hold up!

```
def draw_square(canvas, row, col):
```

If you get a copy when you pass a parameter. Does this copy the canvas??!!

Large variables are stored using something like a URL. The URL gets copied



How do you share google docs?

The screenshot shows a Google Docs interface with the following details:

- Title:** Apollo 11 research
- Toolbar:** Includes File, Edit, View, Insert, Format, Tools, Table, Add-ons, Help, and a note that the last edit was 3 hours ago.
- Share Buttons:** Comments and Share buttons are visible in the top right.
- Document Outline:** A sidebar on the left lists sections: Document Outline, Apollo 11, Summary, The Spacecraft, Design, Command module, Service module, The People, Neil Armstrong, Buzz Aldrin, Mission Highlights, The Launch, The Landing, and Return Trip. The "Summary" section is currently selected.
- Content Area:** The main area contains the following text:

Apollo 11

Summary

This is a research paper about the Apollo 11 moon mission in which Neil Armstrong, Buzz Aldrin, and Michael Collins landed at Tranquility Base on the moon. The Apollo 11 lunar module, AKA The Eagle, landed on the moon on July 20, 1969. When they landed, the message they sent back to Mission Control was 'Tranquility Base here. The Eagle has landed.'

The Spacecraft

The Apollo 11 mission had three spacecraft: the Command Module Columbia, a Service Module, and the Lunar Module Eagle. Columbia was the only part of the spacecraft to return to Earth.

Design

The key NASA spacecraft involved in the Apollo 11 mission were the following: a Saturn V rocket, an Apollo CSM-107 (Command/Service Module) and an Apollo LM-5 (Lunar Module, AKA "The Eagle").

Command module

The Command/Service Module (CSM) was one of two spacecraft, along with the Lunar Module, used for the United States Apollo program which landed astronauts on the Moon. It was built for NASA by North American Aviation. It was launched by itself into suborbital and low Earth orbit test missions with the Saturn IB launch vehicle, and three times by itself and nine times with the Lunar Module as part of the Apollo spacecraft assembly on the larger Saturn V launch vehicle, which was capable of sending it to the Moon.

Service module

The Service Module contained oxygen, water, and electric power for the command module. It also housed the service propulsion system—the rocket engine that put the spacecraft into lunar orbit and later hoisted it back

<https://docs.google.com/document/d/1eBtnEiiI3KHe fFS-kSAOpXqeSXpbfTTMlmOgj6I9dvk/>



```
def main():
```

```
    canvas = make_canvas(...)
```

```
    draw_square(canvas)
```

```
def draw_square(canvas):
```

```
    canvas.create_rectangle(20, 20, 100, 100)
```

stack

heap

```
main
```



```
def main():
```

```
    canvas = make_canvas(...)
```

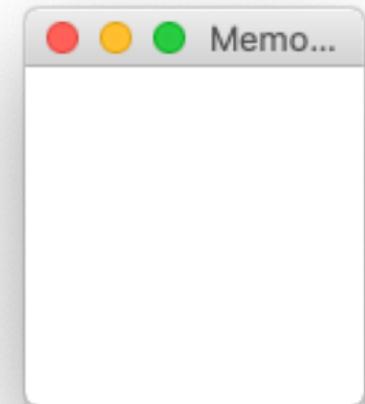
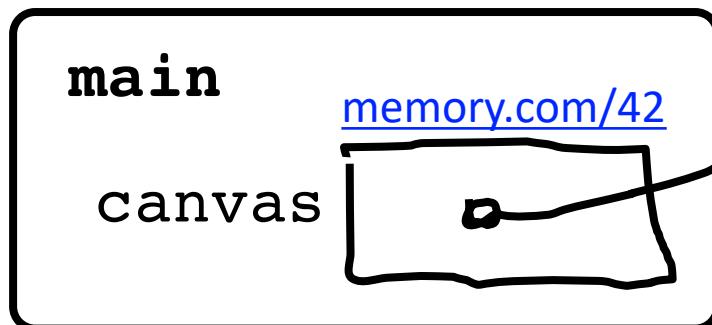
```
    draw_square(canvas)
```

```
def draw_square(canvas):
```

```
    canvas.create_rectangle(20, 20, 100, 100)
```

stack

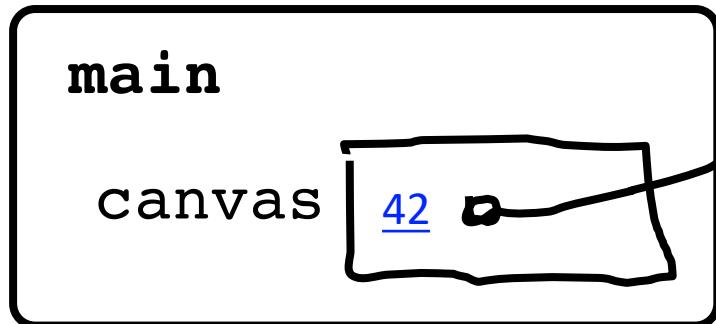
heap



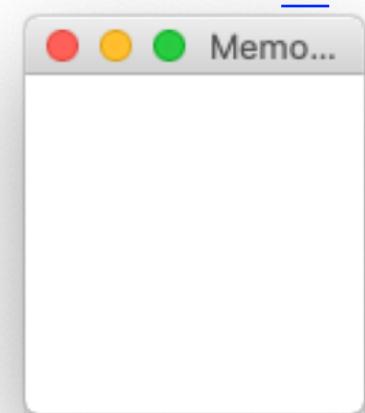
```
def main():
    canvas = make_canvas(...)
    draw_square(canvas)
```

```
def draw_square(canvas):
    canvas.create_rectangle(20, 20, 100, 100)
```

stack



heap

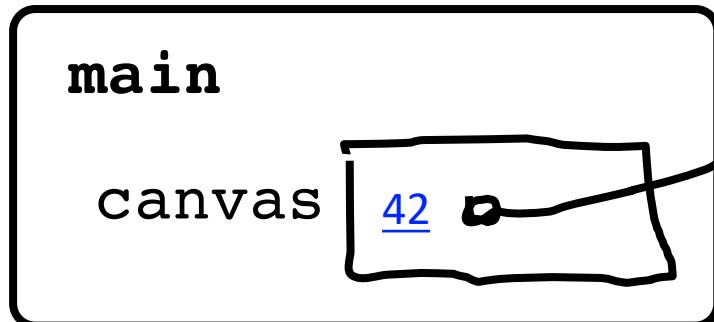


42

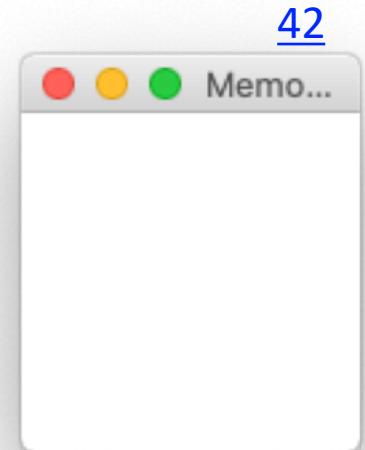
```
def main():
    canvas = make_canvas(...)
    draw_square(canvas)
```

```
def draw_square(canvas):
    canvas.create_rectangle(20, 20, 100, 100)
```

stack

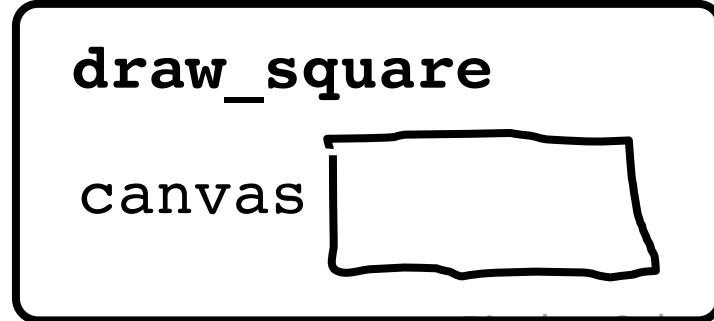


heap



42

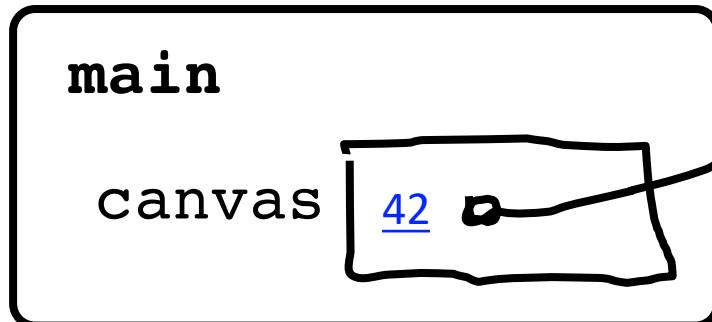
draw_square



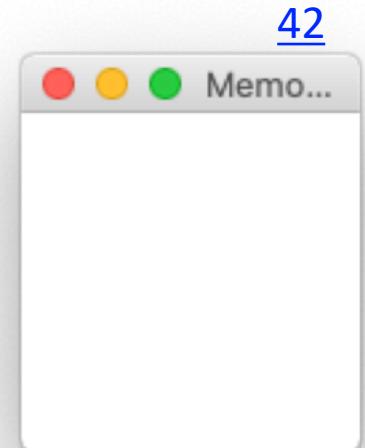
```
def main():
    canvas = make_canvas(...)
    draw_square(canvas)
```

```
def draw_square(canvas):
    canvas.create_rectangle(20, 20, 100, 100)
```

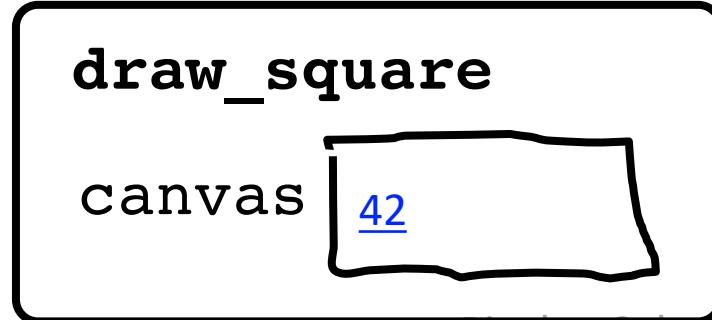
stack



heap



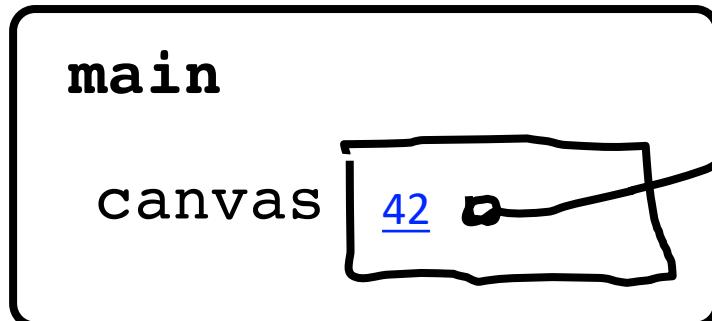
`draw_square`



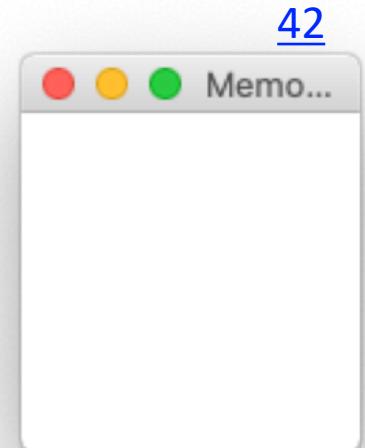
```
def main():
    canvas = make_canvas(...)
    draw_square(canvas)
```

```
def draw_square(canvas):
    canvas.create_rectangle(20, 20, 100, 100)
```

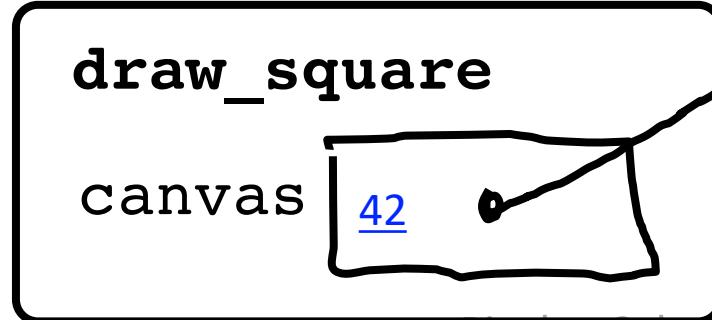
stack



heap



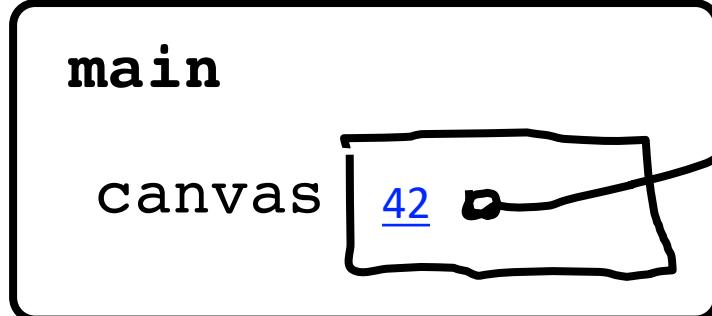
draw_square



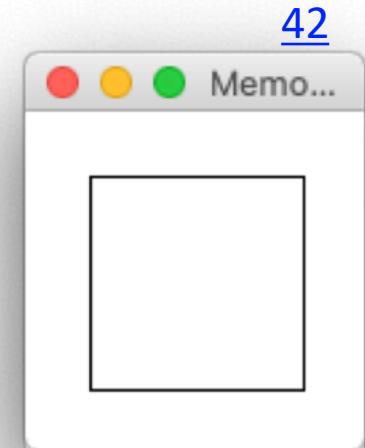
```
def main():
    canvas = make_canvas(...)
    draw_square(canvas)
```

```
def draw_square(canvas):
    canvas.create_rectangle(20, 20, 100, 100)
```

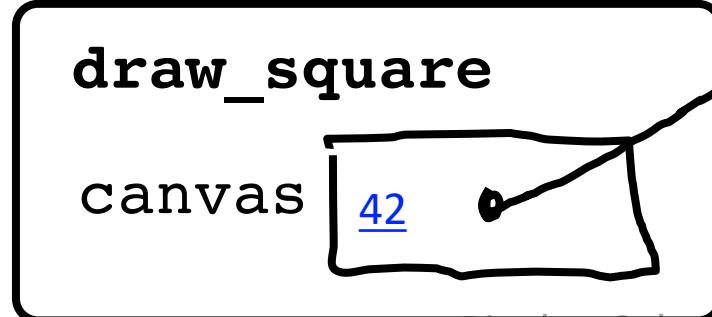
stack



heap



draw_square



```
def main():
```

```
    canvas = make_canvas(...)
```

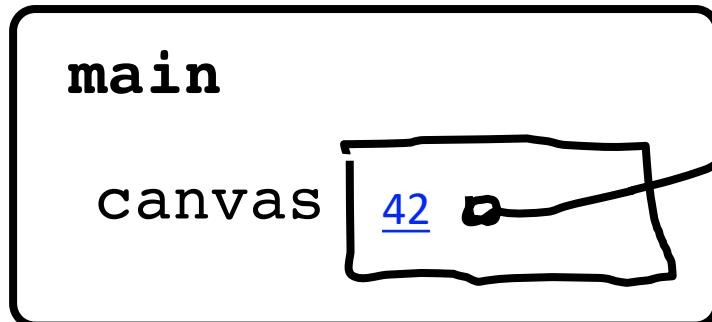
```
    draw_square(canvas)
```

```
def draw_square(canvas):
```

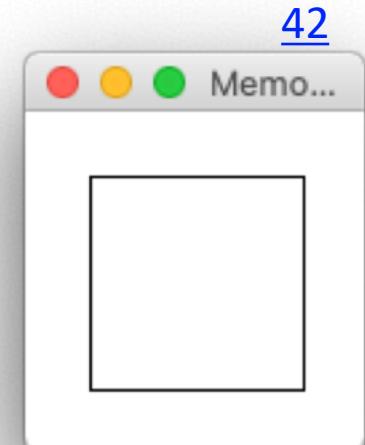
```
    canvas.create_rectangle(20, 20, 100, 100)
```



stack



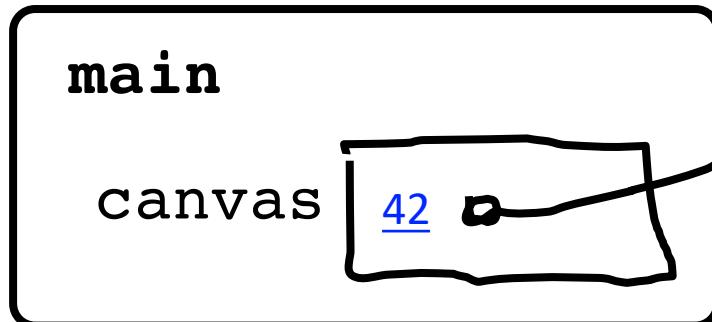
heap



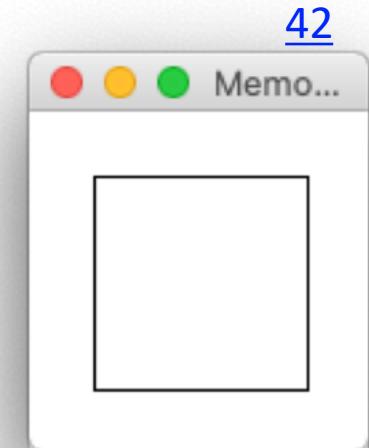
```
def main():
    canvas = make_canvas(...)
    draw_square(canvas)
```

```
def draw_square(canvas):
    canvas.create_rectangle(20, 20, 100, 100)
```

stack



heap





Large variable types are
stored as memory
addresses

(which are like memory
URLs)

