



Graphics

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

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

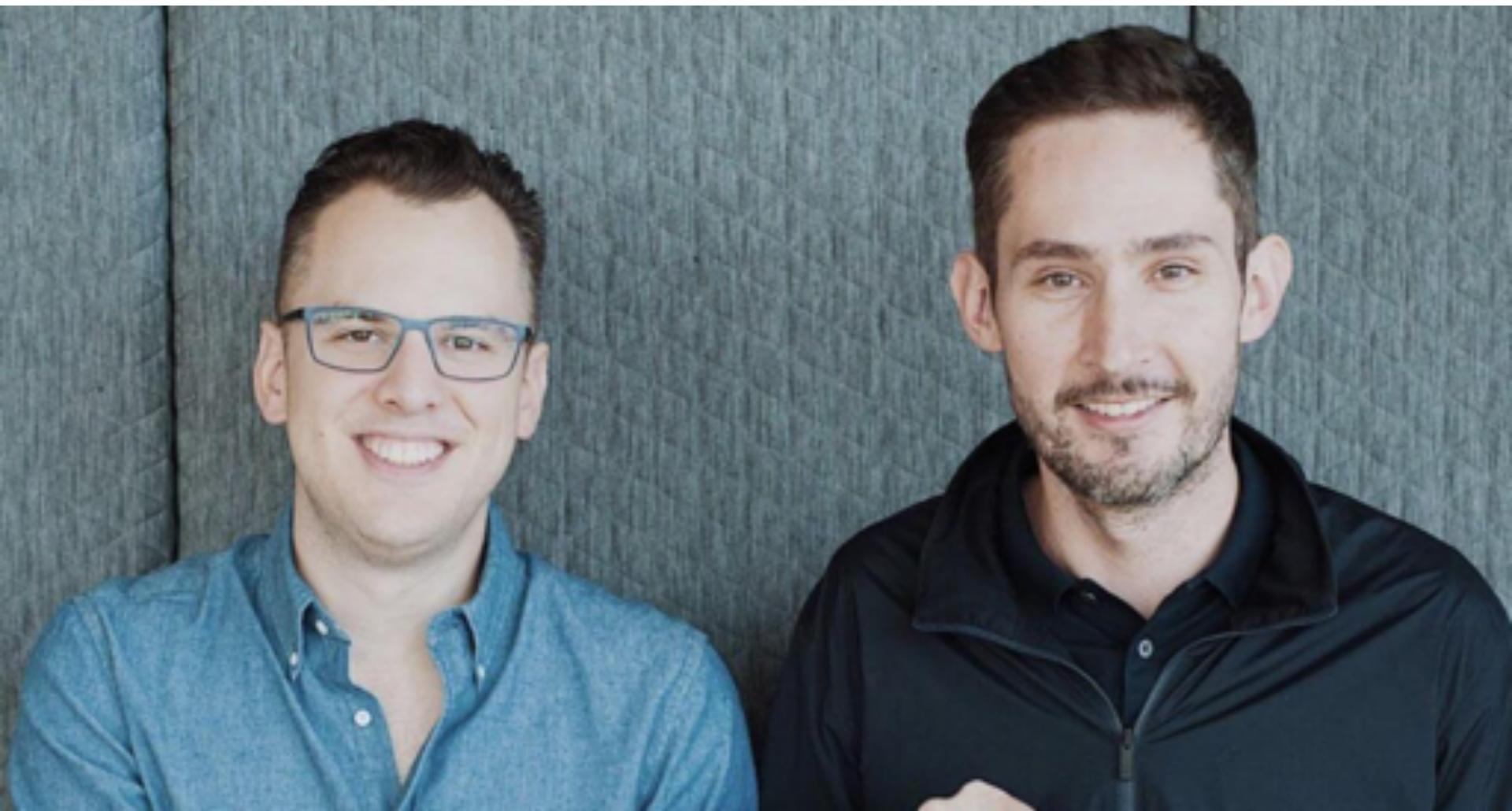
```
def main():

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



Mike Krieger

Kevin Systrom



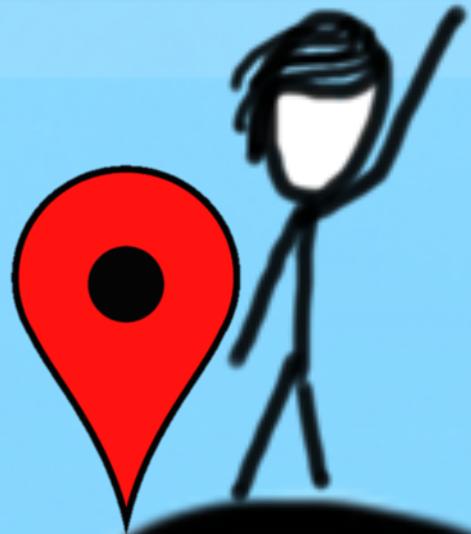
Piech + Sahami, CS106A, Stanford University



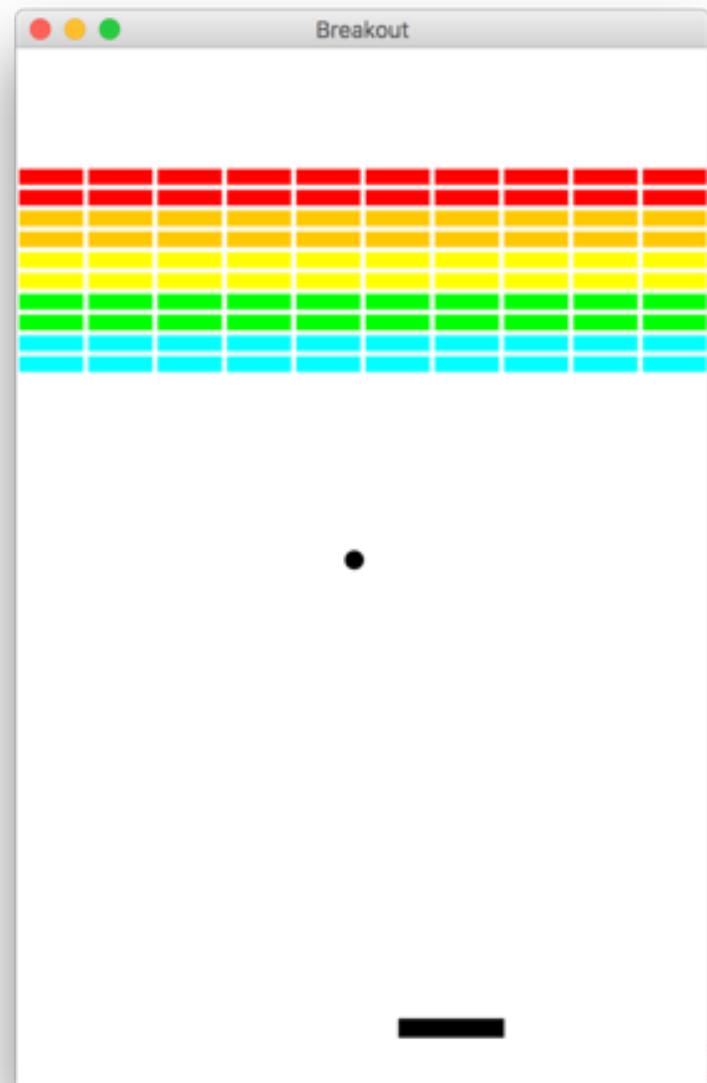
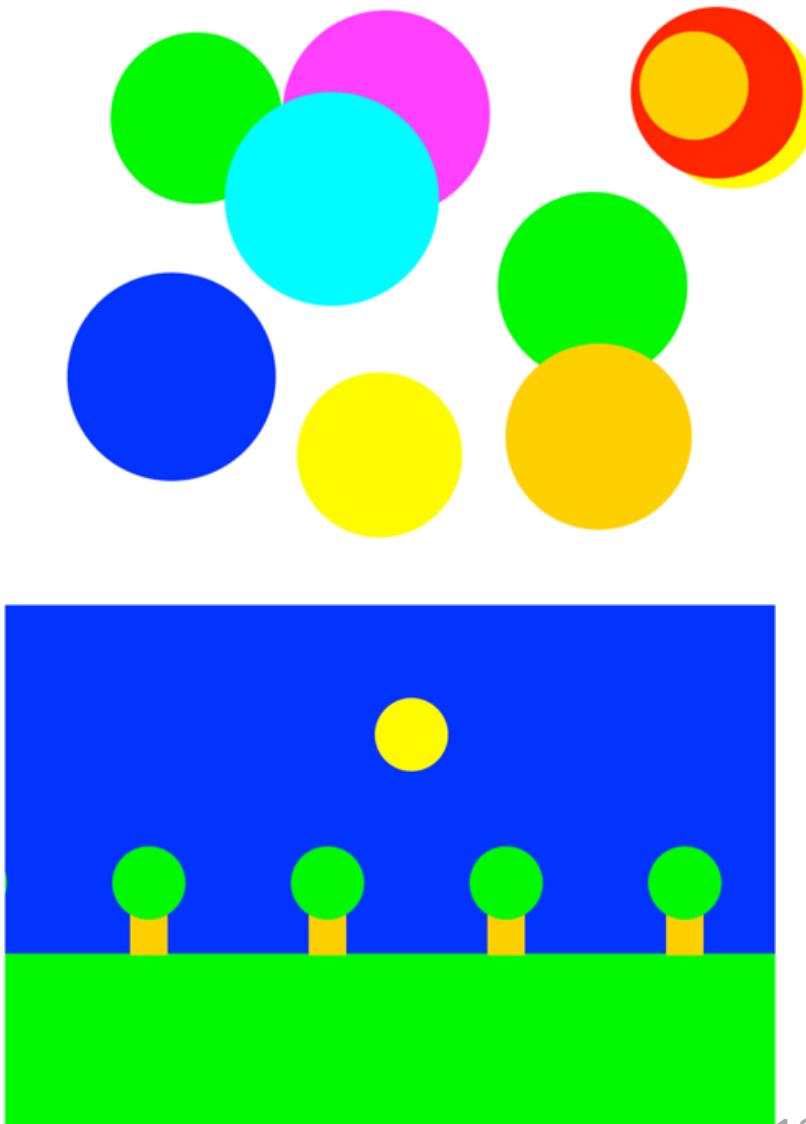
End Review

Today's Goal

1. How do I draw shapes?



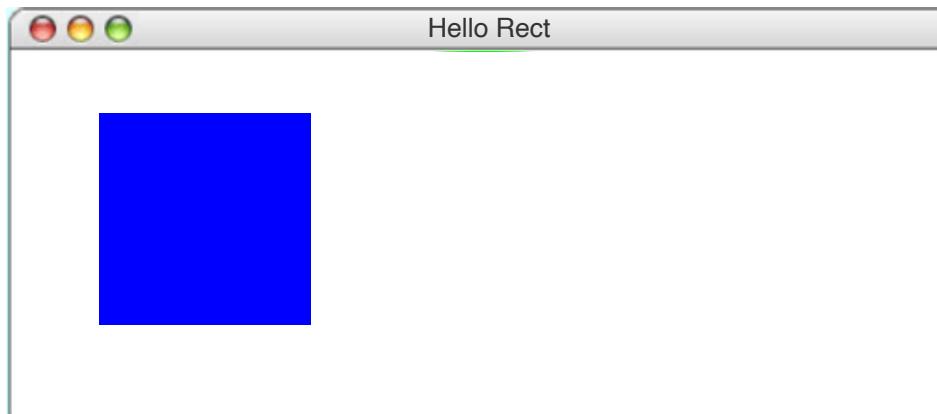
Graphics Programs



Draw a Rectangle

the following `main` method displays a blue square

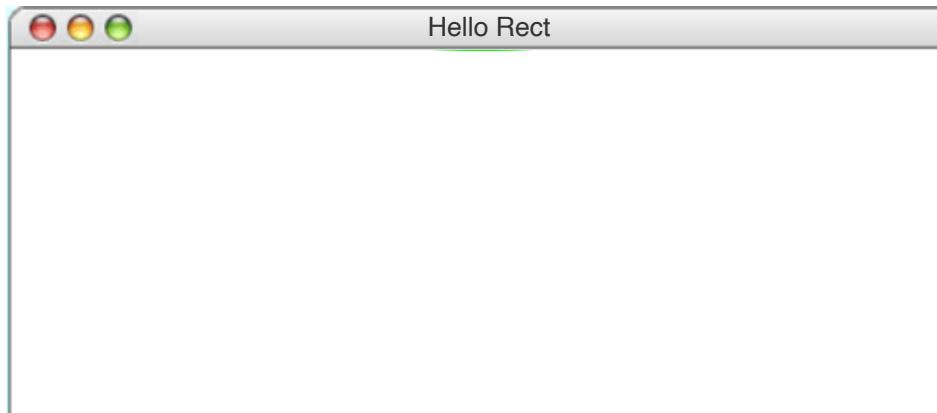
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
    canvas.mainloop()
```



Draw a Rectangle

the following `main` method displays a blue square

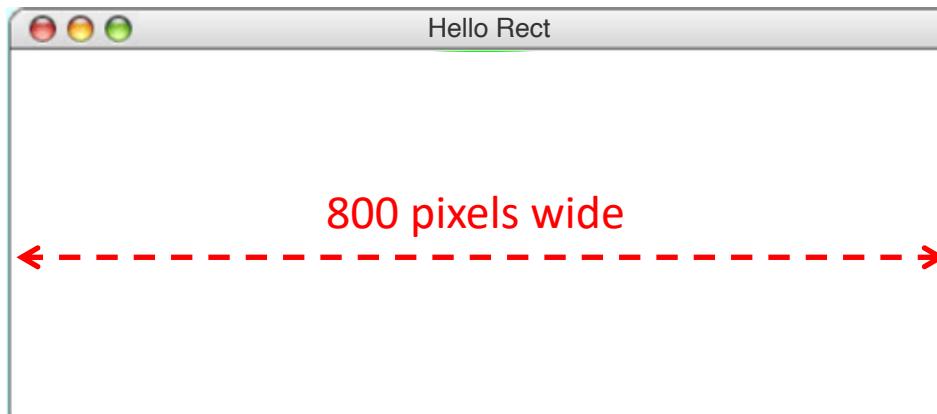
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

the following `main` method displays a blue square

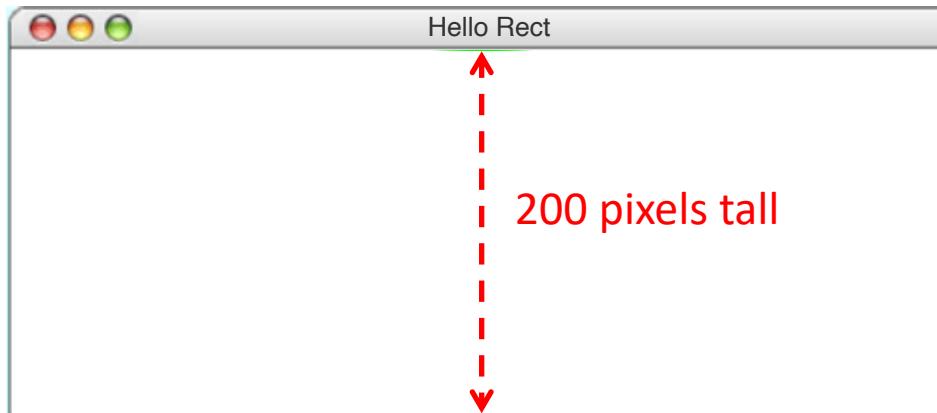
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
```



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

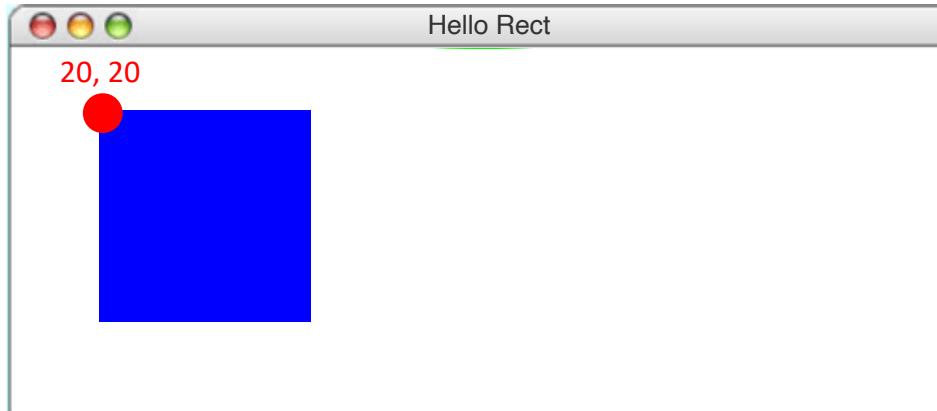
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

the following `main` method displays a blue square

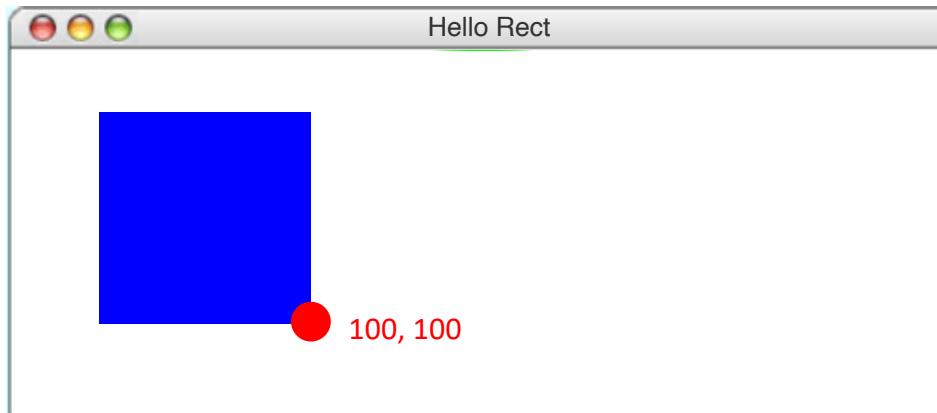
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
```



Draw a Rectangle

the following `main` method displays a blue square

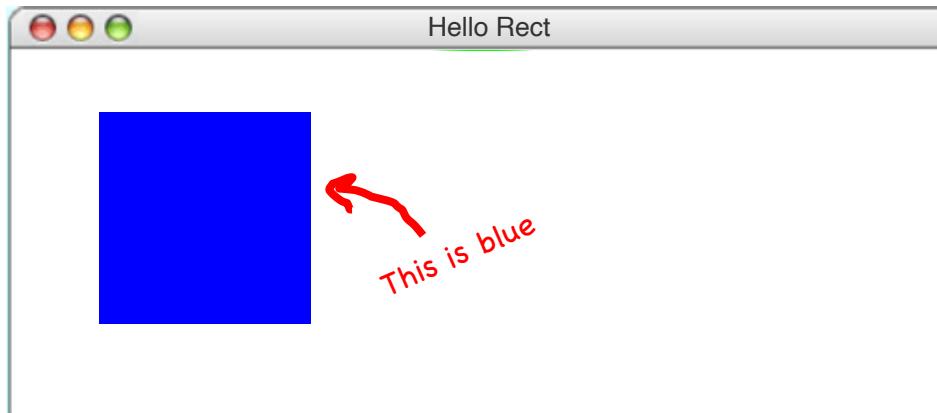
```
def main():
    canvas = make_canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, fill="blue")
```



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")
```



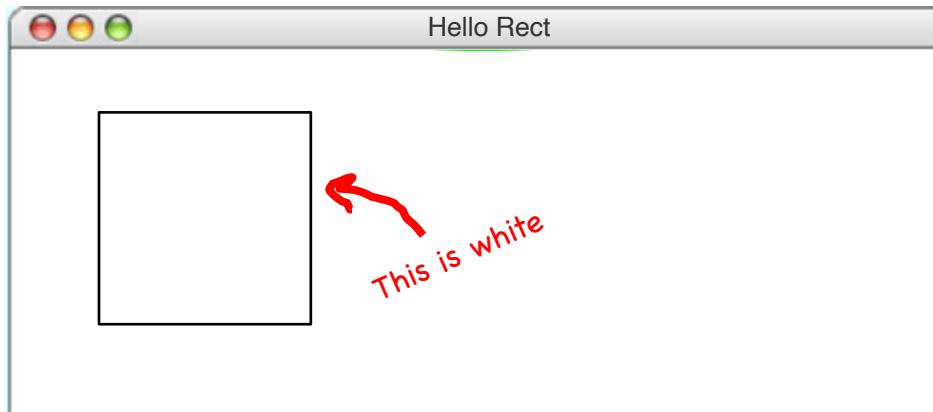
Aside: Named Arguments
This argument is named as filled. It allows functions to have arguments which you can ignore if you want a default value.



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)
```



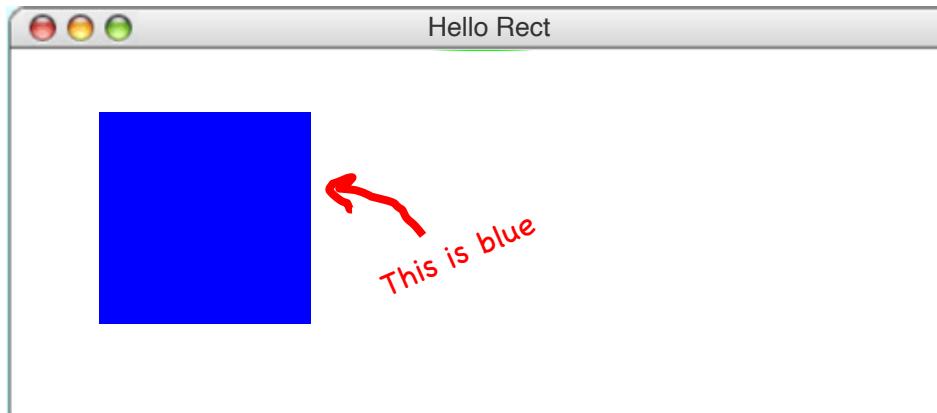
Aside: Named Arguments
This argument is named as filled. It allows functions to have arguments which you can ignore if you want a default value.



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")
```



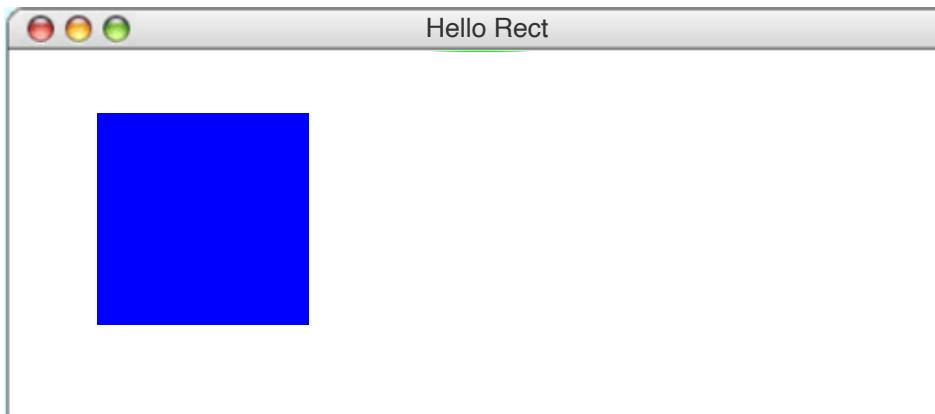
Aside: Named Arguments
This argument is named as filled. It allows functions to have arguments which you can ignore if you want a default value.



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

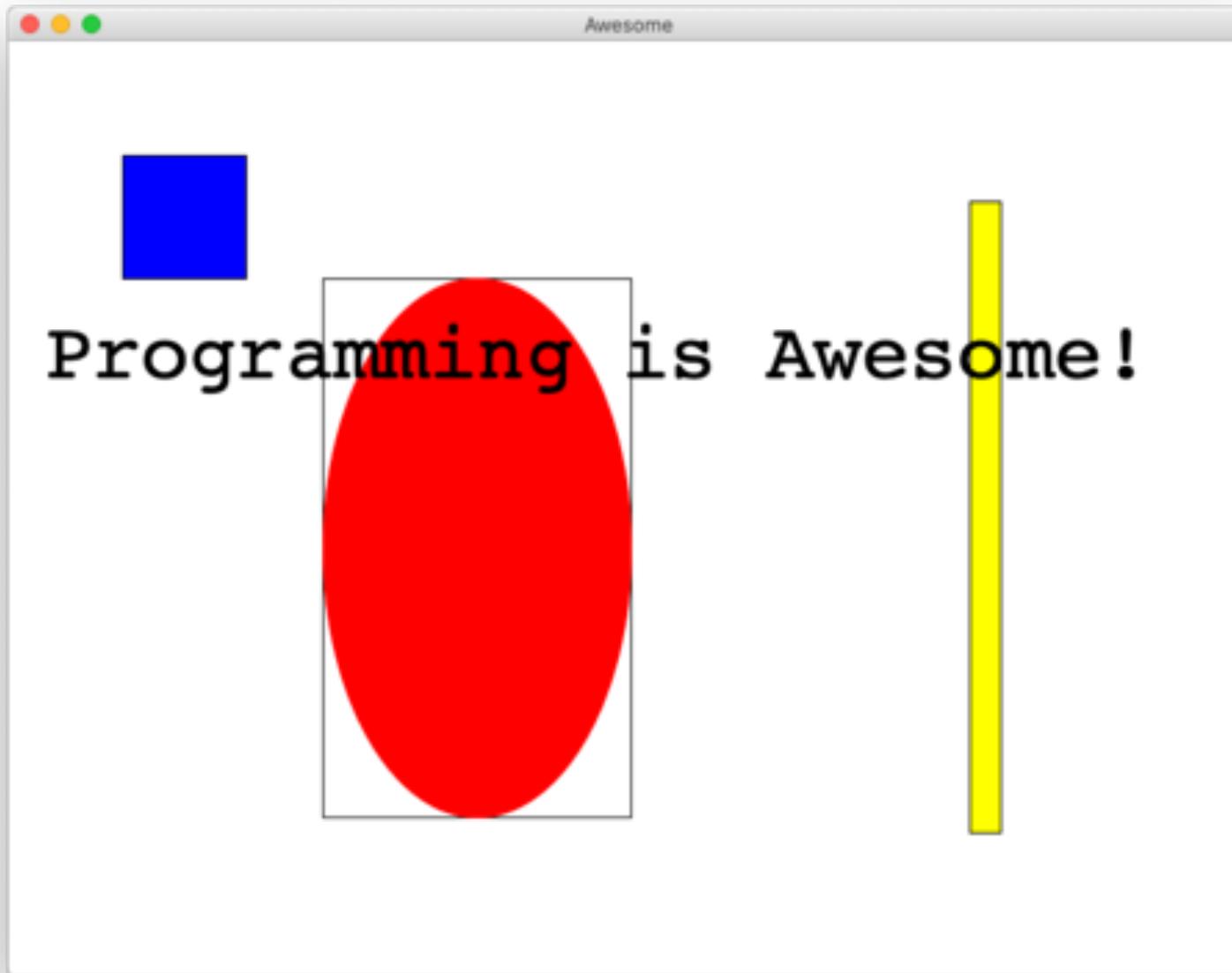
x 40,120

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()**

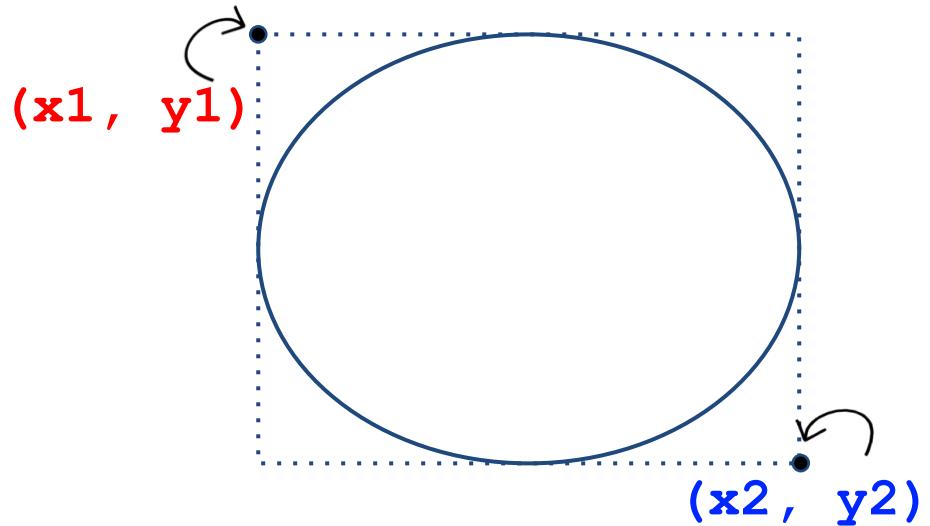
- **canvas.create_text()**



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



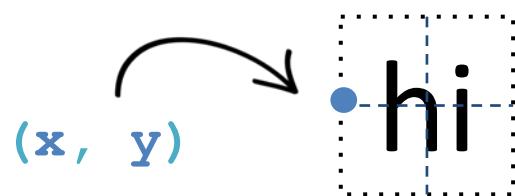
- `canvas.create_line()`
- `canvas.create_oval(x1, y1, x2, y2)`
- `canvas.create_text()`



- `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 title bar says "localhost:8000/examples/awesome/". The main content is a Tkinter application window titled "Awesome". Inside the window, there is a blue square, a red oval containing the text "Programming is Awesome!", a yellow vertical rectangle, and a small image of a dog. The entire Tkinter window is circled in red.

```
import tkinter
from PIL import ImageTk
from PIL import Image

CANVAS_WIDTH = 800
CANVAS_HEIGHT = 600

def main():
    canvas = make_canvas(CANVAS_WIDTH, CANVAS_HEIGHT, "Awesome")
    # a line for good measure!
    canvas.create_line(0, 0, 600, 600)

    # a blue square with width and height = 80
    canvas.create_rectangle(70, 70, 150, 150, fill="blue")
    # a yellow rectangle that is long and skinny
    canvas.create_rectangle(620, 100, 640, 510, fill="yellow")

localhost:8000/examples/awesome/
```

Solution

```
import tkinter
from PIL import ImageTk
from PIL import Image

CANVAS_WIDTH = 800
CANVAS_HEIGHT = 600

def main():
    canvas = make_canvas(CANVAS_WIDTH, CANVAS_HEIGHT, "Awesome")
    # a line for good measure!
    canvas.create_line(0, 0, 600, 600)

    # a blue square with width and height = 80
    canvas.create_rectangle(70, 70, 150, 150, fill="blue")
    # a yellow rectangle that is long and skinny
    canvas.create_rectangle(620, 100, 640, 510, fill="yellow")
```

Handouts ▾ Examples ▾

General Information

Course Placement

Honor Code

Installing PyCharm

Using Karel in PyCharm

Submitting Assignments

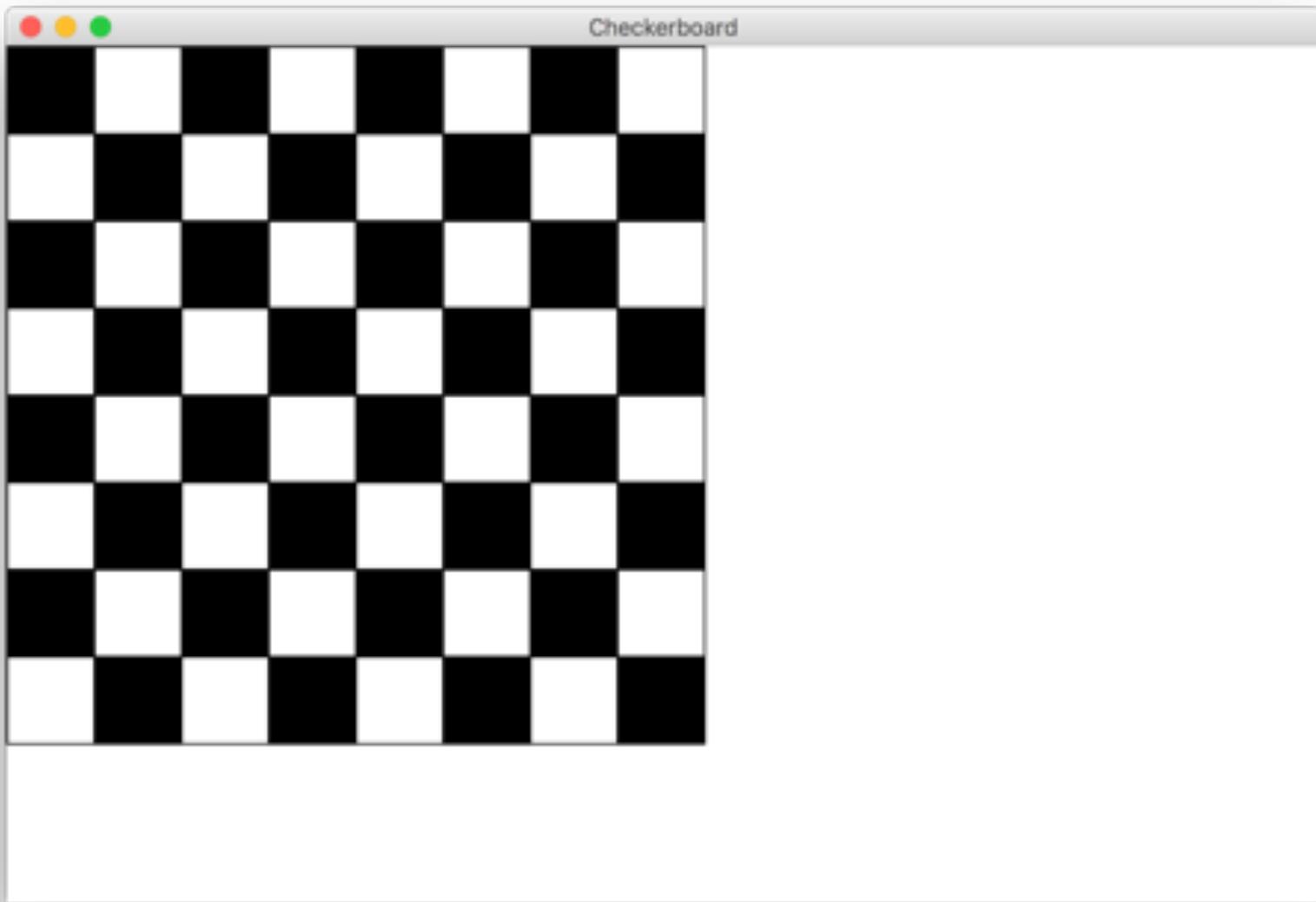
Diagnostic

Image Reference

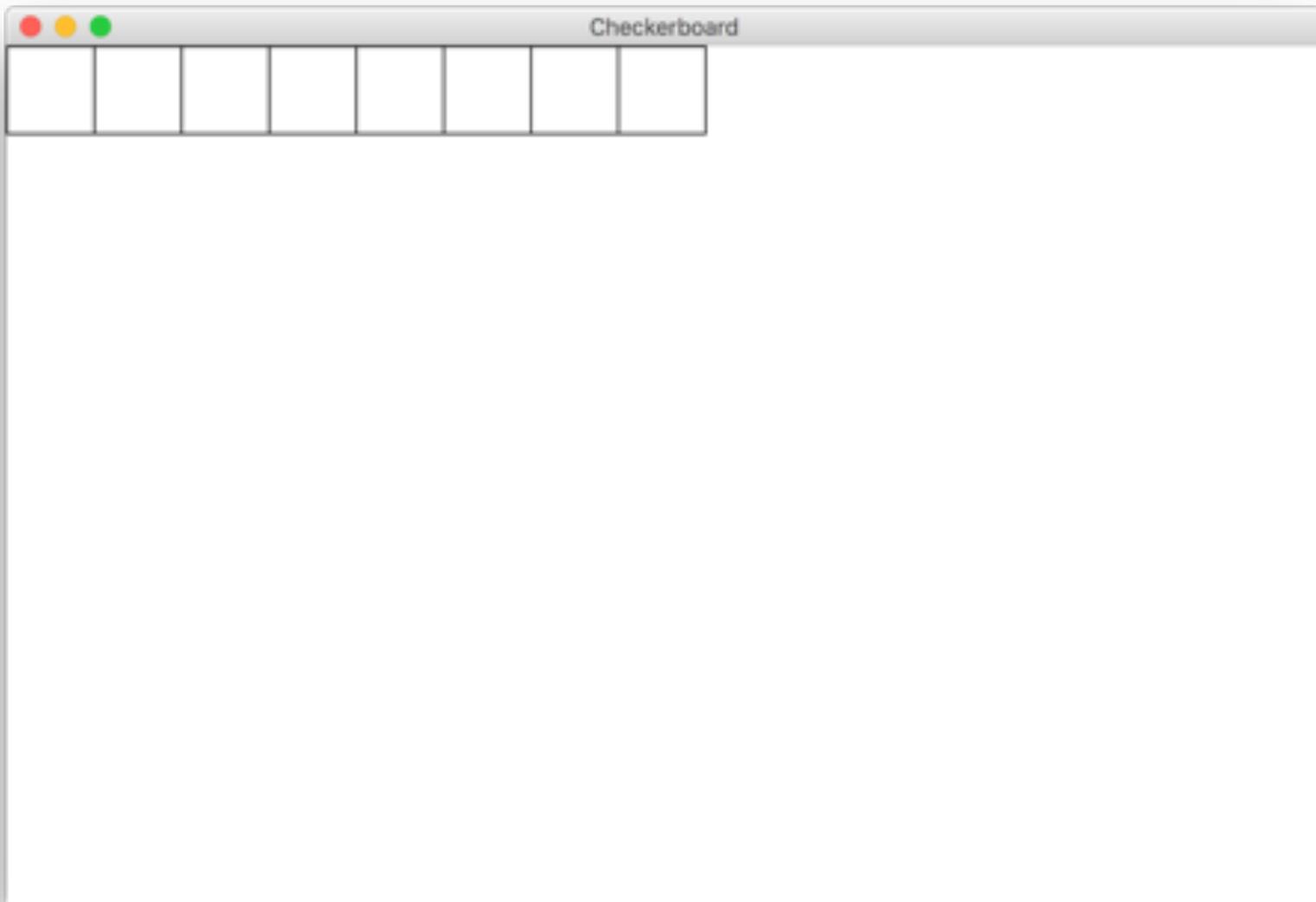
Graphics Reference



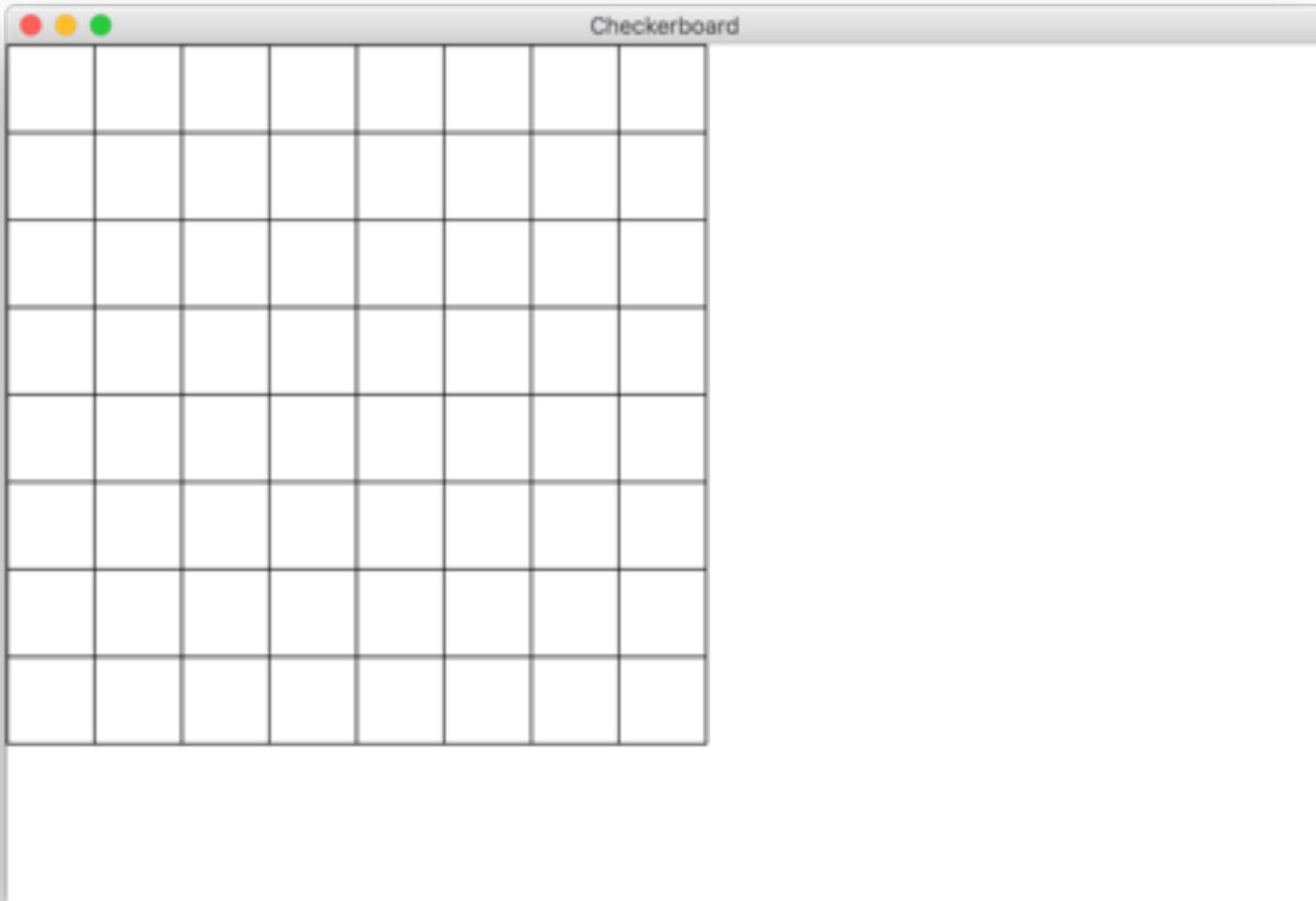
Goal



Milestone 1



Milestone 2



Milestone 3

