Graphics

Day 1!

Stanford | ENGINEERING
Computer Science

Housekeeping

- Assignment 2 due tomorrow night! Grace period extends to midnight

- Breakout will be released on Friday

- Midterm is (somewhat) approaching. Information page will be posted soon.

Today

- **Recap lists**
  - **Let's do a few more list exercises!**
  - **Test with doctests**

- Introduce Graphics
  - Drawing!
  - So many new functions!

Revisit: `factorial_avg_list`

- Let's write a function, **`factorial_avg_list(nums)`**
  - Takes in a list of numbers and prints their factorials (as a list)
  - Then prints the average of all the factorials

- Decompose "make list of factorials" and "calculate the average of a list" into two helper functions

- **Try making a "return" version of "make list of factorials" and a "modify" version**

Aside: `main` function

- The **`main`** function is where a program starts

- When I run **`python3 example.py`**, it will run the code in the **`main`** function - including any function calls

Aside: `main` function

- The **`main`** function is where a program starts

- When I run **`python3 example.py`**, it will run the code in the **`main`** function - including any function calls

example.py

```
def my_helper():
  print("first line of helper!")

def main():
  print("first line of program!")
   my_helper()
```

Aside: `main` function

- The **`main`** function is where a program starts

- When I run **`python3 example.py`**, it will run the code in the **`main`** function - including any function calls

example.py

```
def func():
  print("first line of func!")

def main():
  print("first line of pgm!")
  my_helper()
```

Terminal:

```
$ python3 example.py
```

Stanford | ENGINEERING
Computer Science

Aside: `main` function

- The **`main`** function is where a program starts

- When I run **`python3 example.py`**, it will run the code in the **`main`** function - including any function calls

example.py

```
def func():
  print("first line of func!")

def main():
  print("first line of pgm!")
  my_helper()
```

Terminal:

```
$ python3 example.py
first line of pgm!
first line of func!
```

Aside: `main` function

- The **`main`** function is where a program starts

- When I run **`python3 example.py`**, it will run the code in the **`main`** function - including any function calls

Example.py

```
def func():
  print("first line of func!")

def main():
  print("first line of pgm!")
  my_helper()
```

- We are going to start writing our programs in main!

Stanford | ENGINEERING
Computer Science

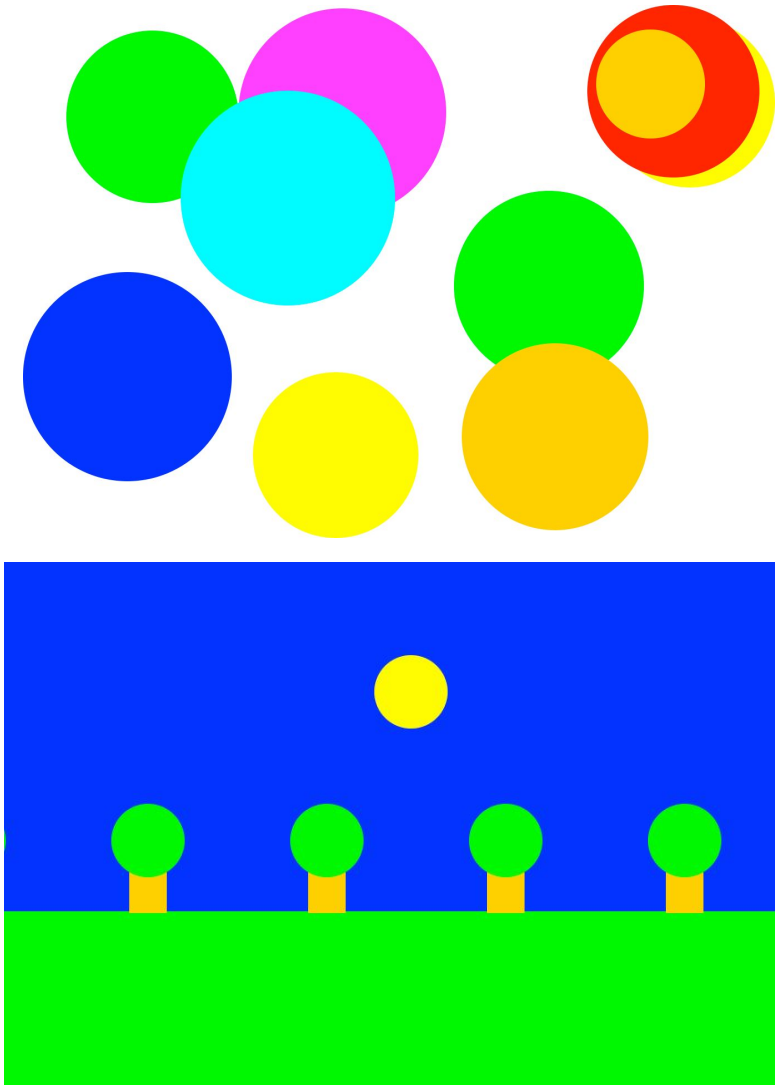
One more: `two_list`

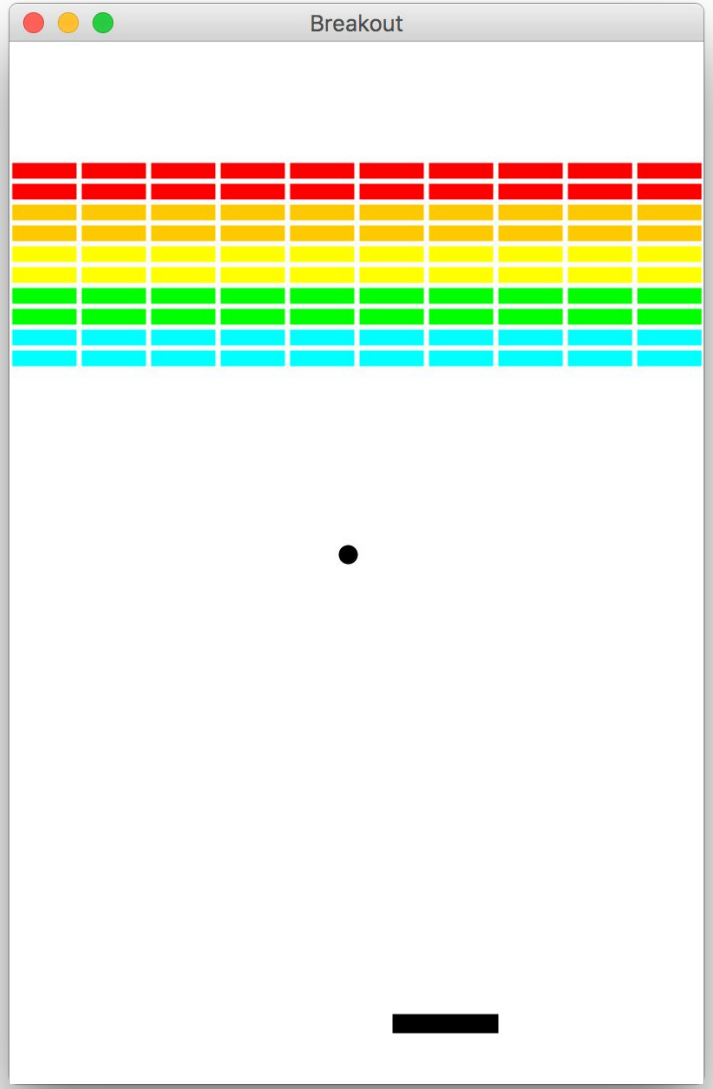
- Write a program that takes in two lists of strings
- And prints to the user a list of the largest number at each index (if a list is shorter than the other, take the number that exists)
- The two lists of numbers will be saved in variables in main. You can ignore the code above this.
- Decompose one function to call from main
- Example:

Input: [1, 1, 3, 4, 7]
[2, 0, 0, 5]

Output: [1, 2, 3, 5, 7]

Graphics Programs





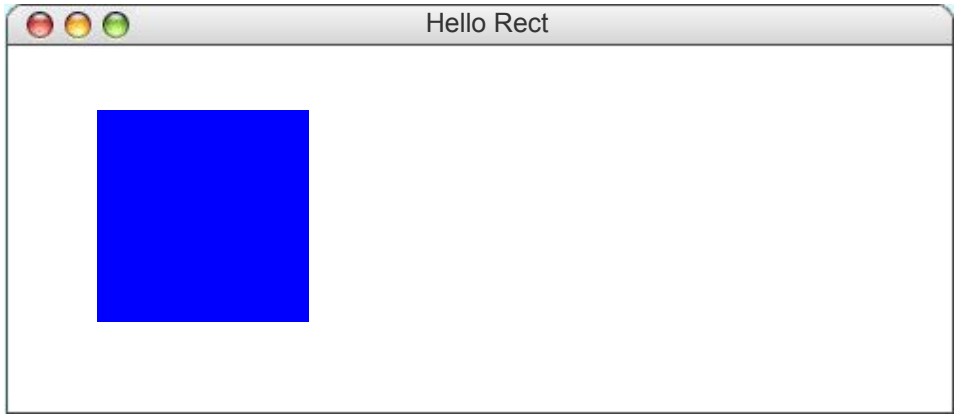
Piech + Sahami, CS106A, Stanford University

Frankie Cerkvenik, CS106A, 2023

Draw a Rectangle

The following **`main`** function displays a blue square

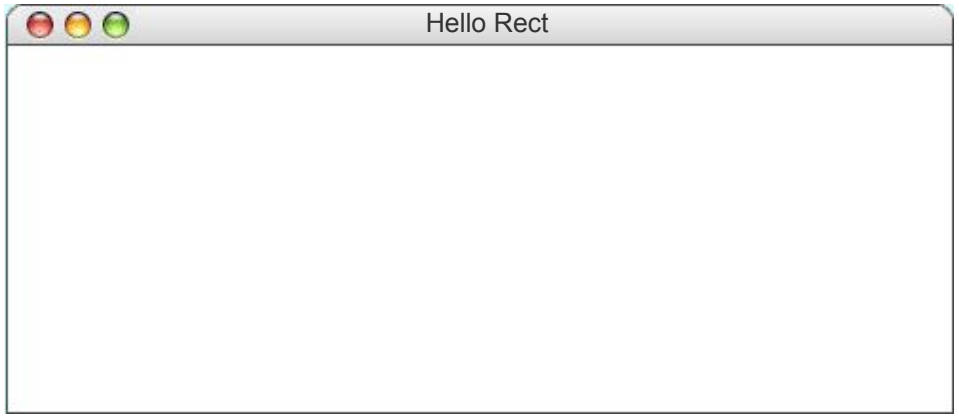
```
def main():
    canvas = Canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, "blue")
    canvas.mainloop()
```



Draw a Rectangle

`Canvas` sets up the window where our drawings will appear and **returns a reference to it** (saved in **`canvas`**)

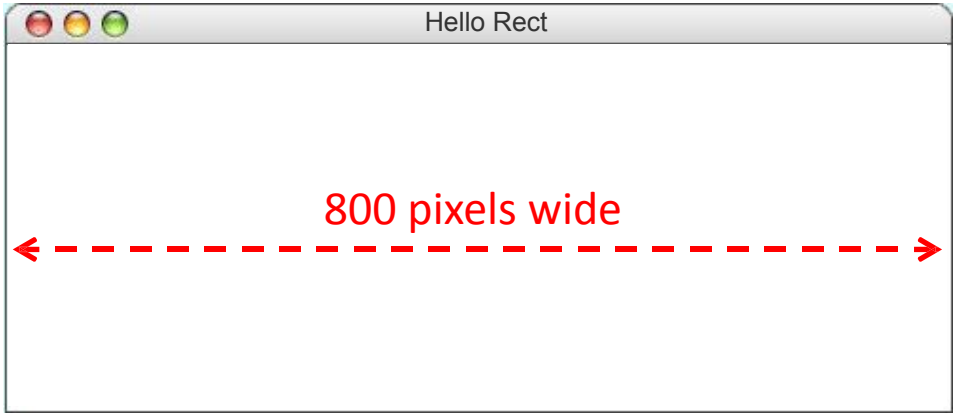
```
def main():
  canvas = Canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

`Canvas` takes in the **width** of the canvas

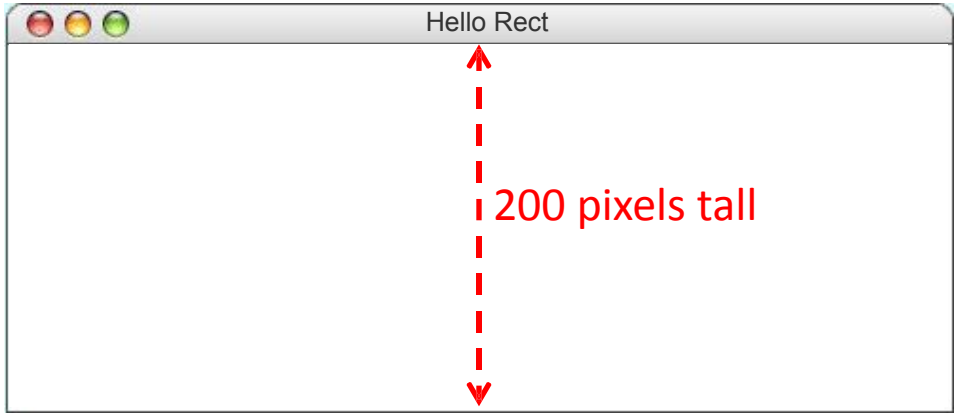
```
def main():
  canvas = Canvas(800, 200, 'Hello Rect')
```



Draw a Rectangle

`Canvas` takes in the **height** of the canvas

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



Draw a Rectangle

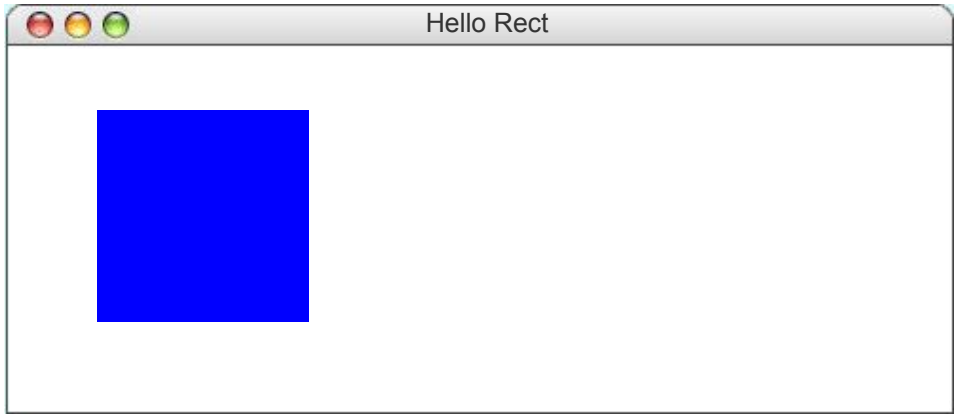
Canvas takes in the **title** of the canvas

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

Draw a Rectangle

`create_rectangle` will draw a rectangle on the canvas

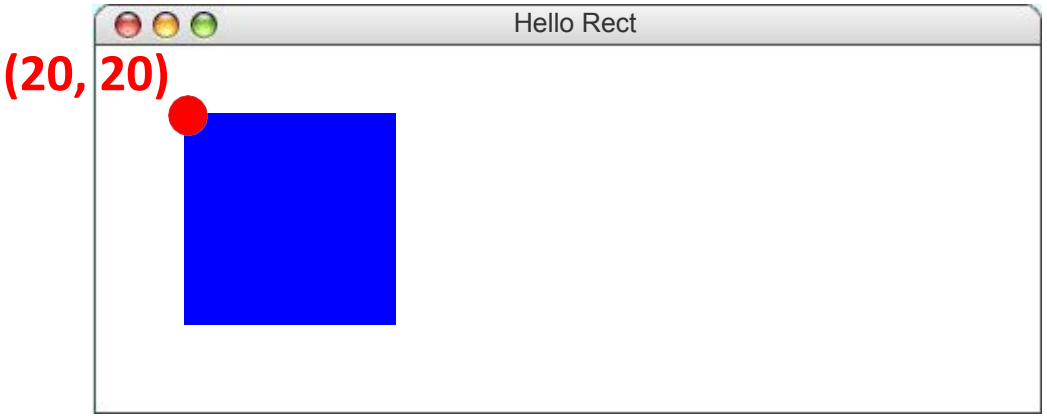
```
def main():
  canvas = Canvas(800, 200, 'Hello Rect')
  canvas.create_rectangle(20, 20, 100, 100, "blue")
```



Draw a Rectangle

`create_rectangle` takes in the **coordinate of the upper right corner** of the rectangle

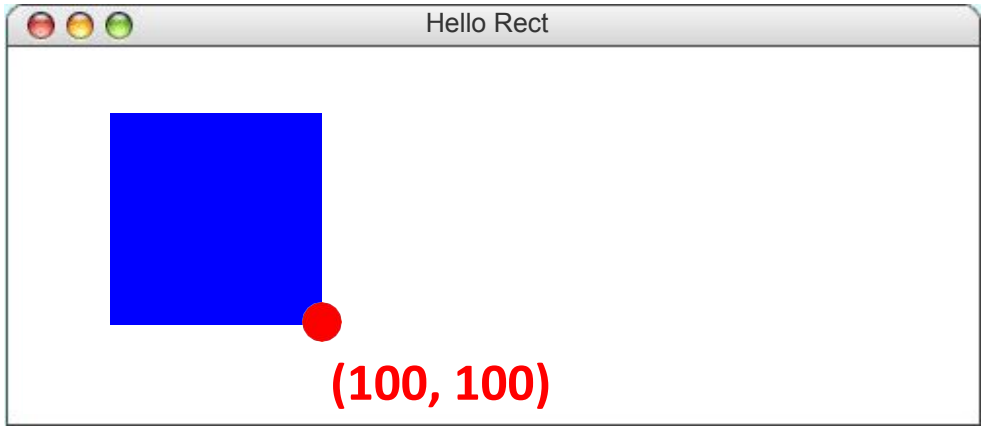
```
def main():
  canvas = Canvas(800, 200, 'Hello Rect')
  canvas.create_rectangle(20, 20, 100, 100, "blue")
```



Draw a Rectangle

`create_rectangle` takes in the **coordinate of the bottom left corner** of the rectangle

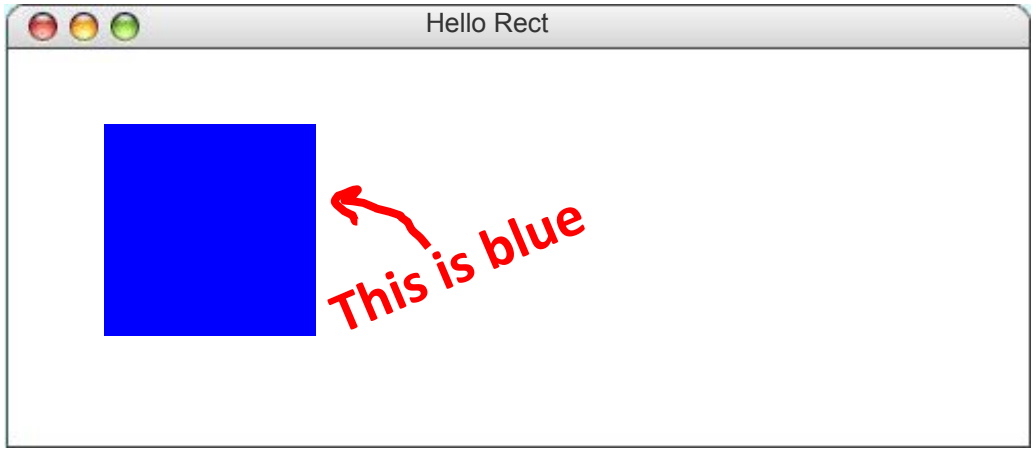
```
def main():
  canvas = Canvas(800, 200, 'Hello Rect')
  canvas.create_rectangle(20, 20, 100, 100, "blue")
```



Draw a Rectangle

`create_rectangle` takes in the **color** of the rectangle

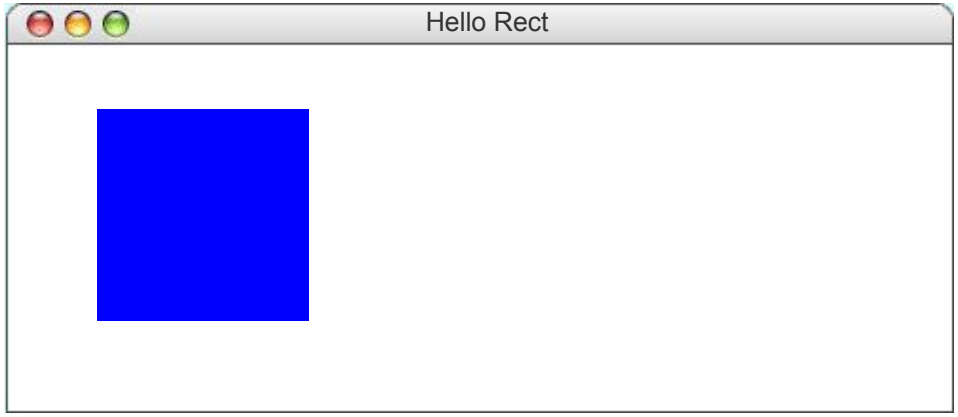
```
def main():
  canvas = Canvas(800, 200, 'Hello Rect')
  canvas.create_rectangle(20, 20, 100, 100, "blue")
```



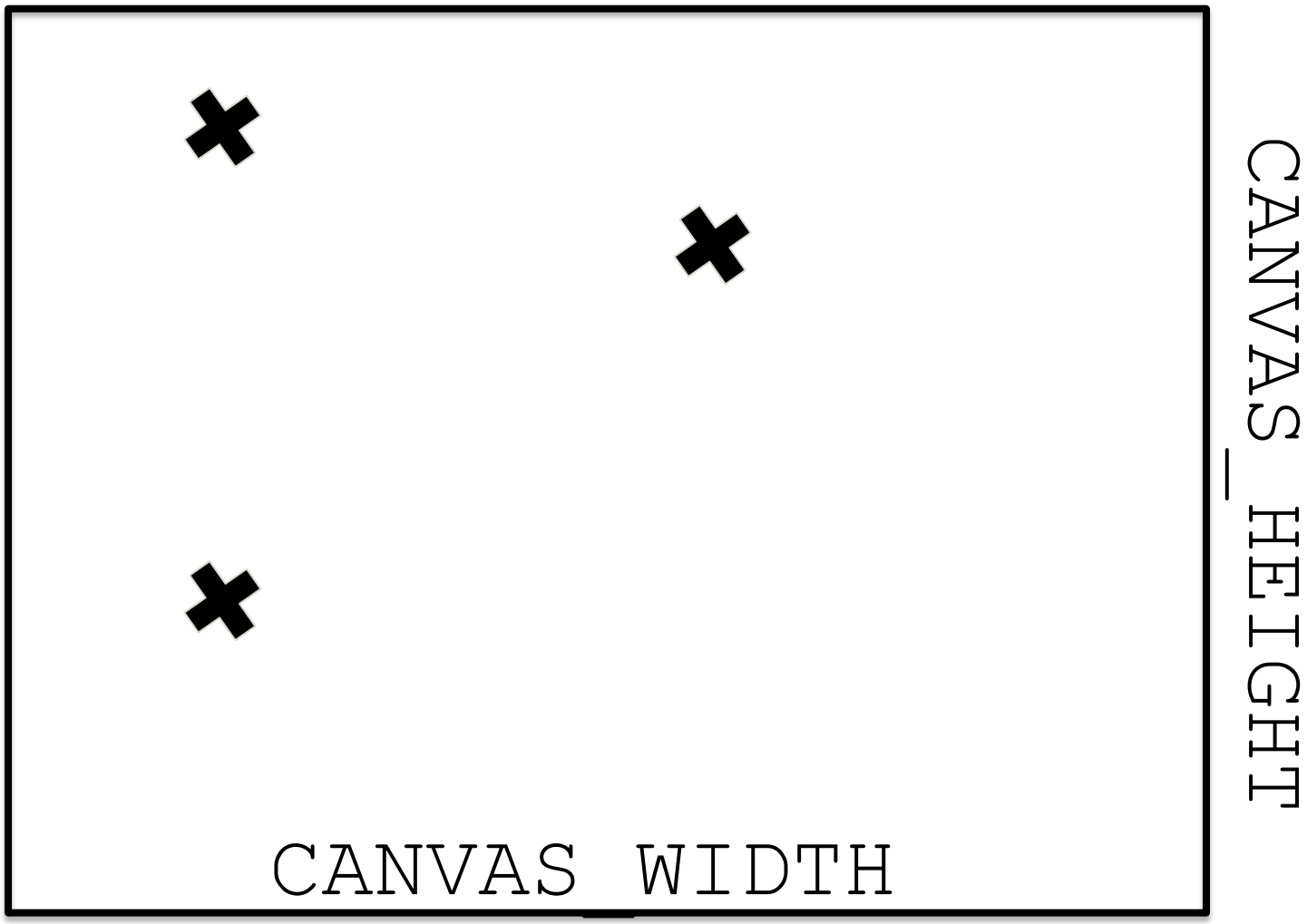
Draw a Rectangle

The mainloop function makes the canvas continue displaying until the user hits the **x** button - always include it at the end!

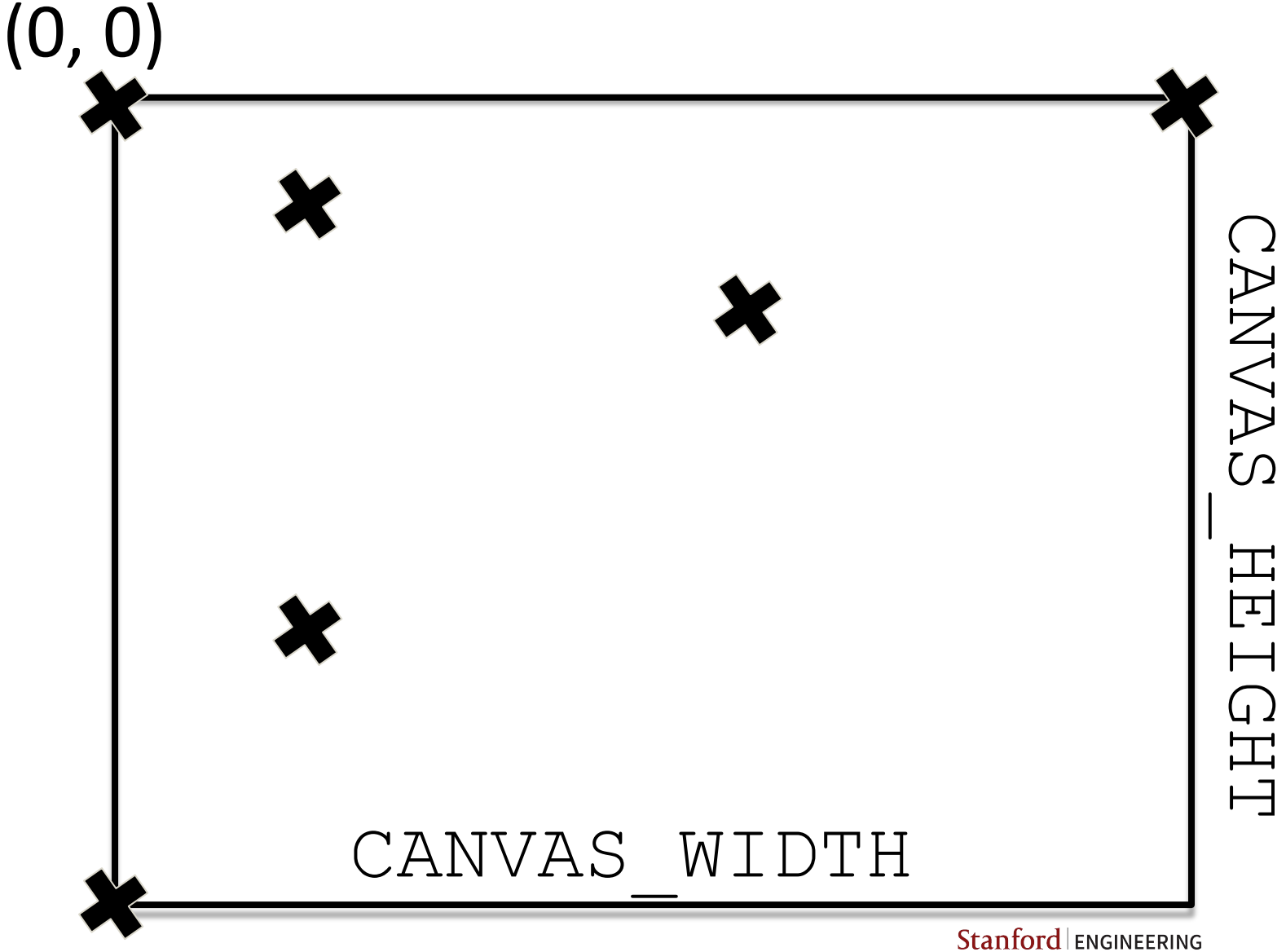
```
def main():
    canvas = Canvas(800, 200, 'Hello Rect')
    canvas.create_rectangle(20, 20, 100, 100, "blue")
    canvas.mainloop()
```



Graphics Coordinates

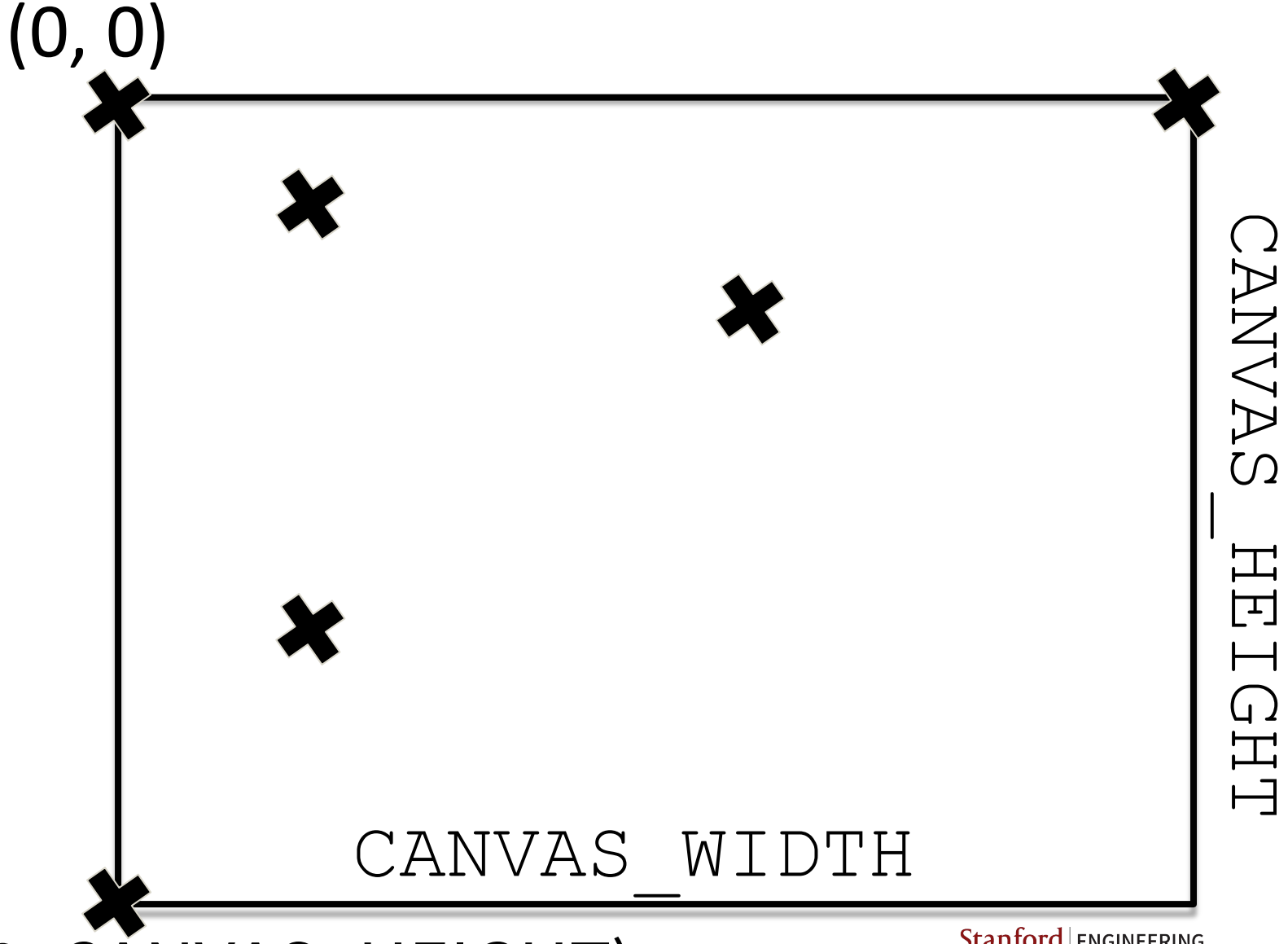


Graphics Coordinates

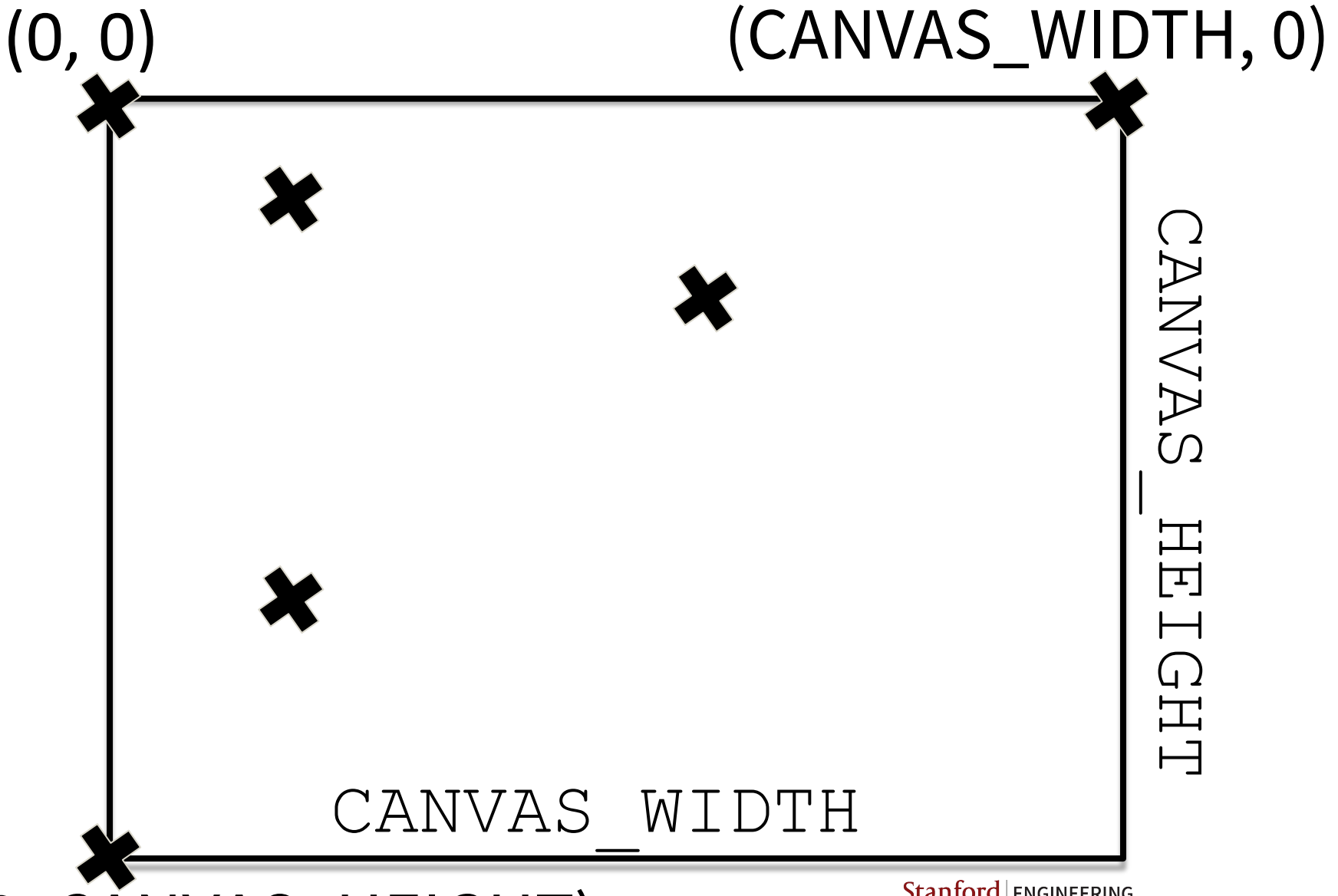


Stanford | ENGINEERING
Computer Science

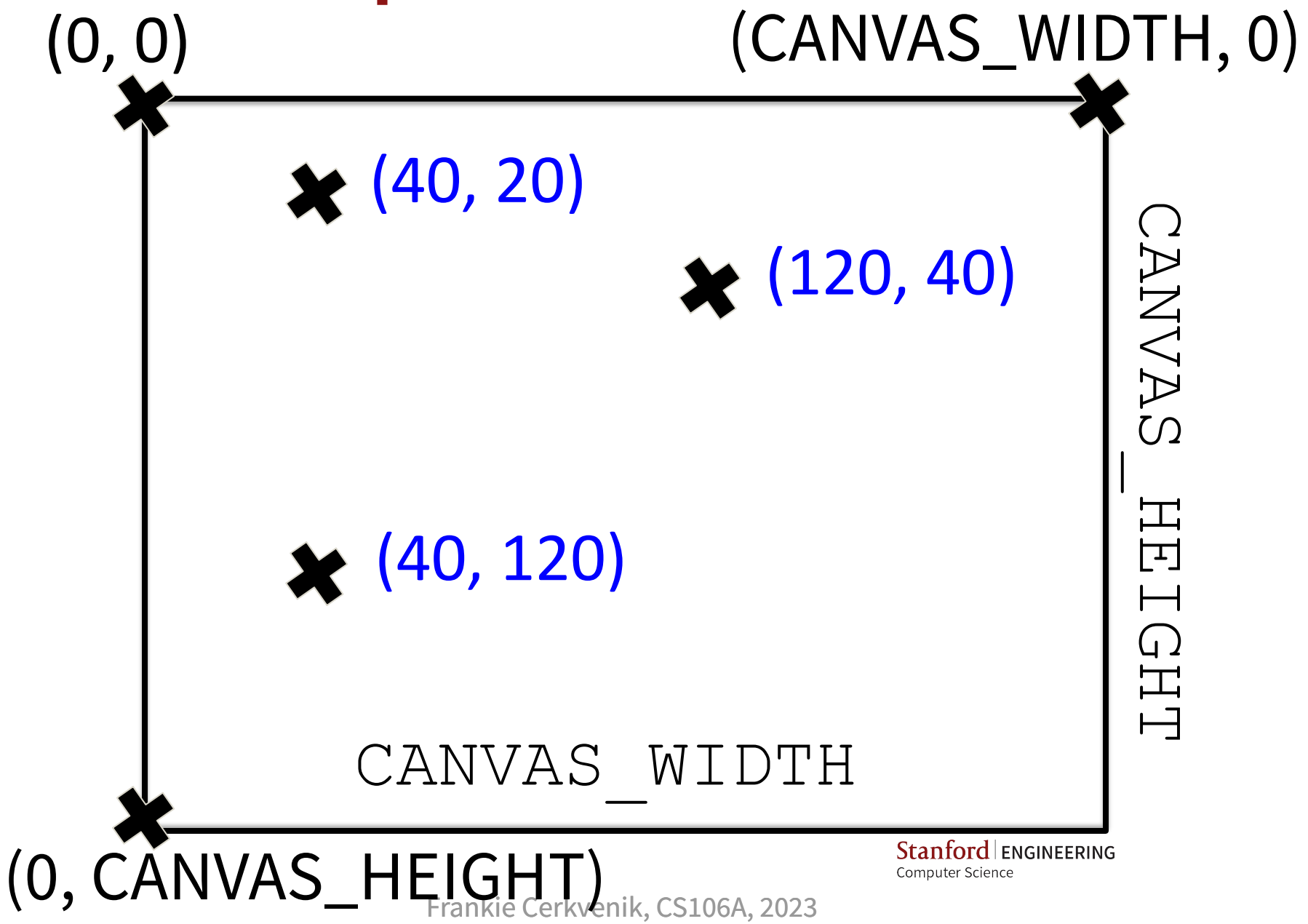
Graphics Coordinates



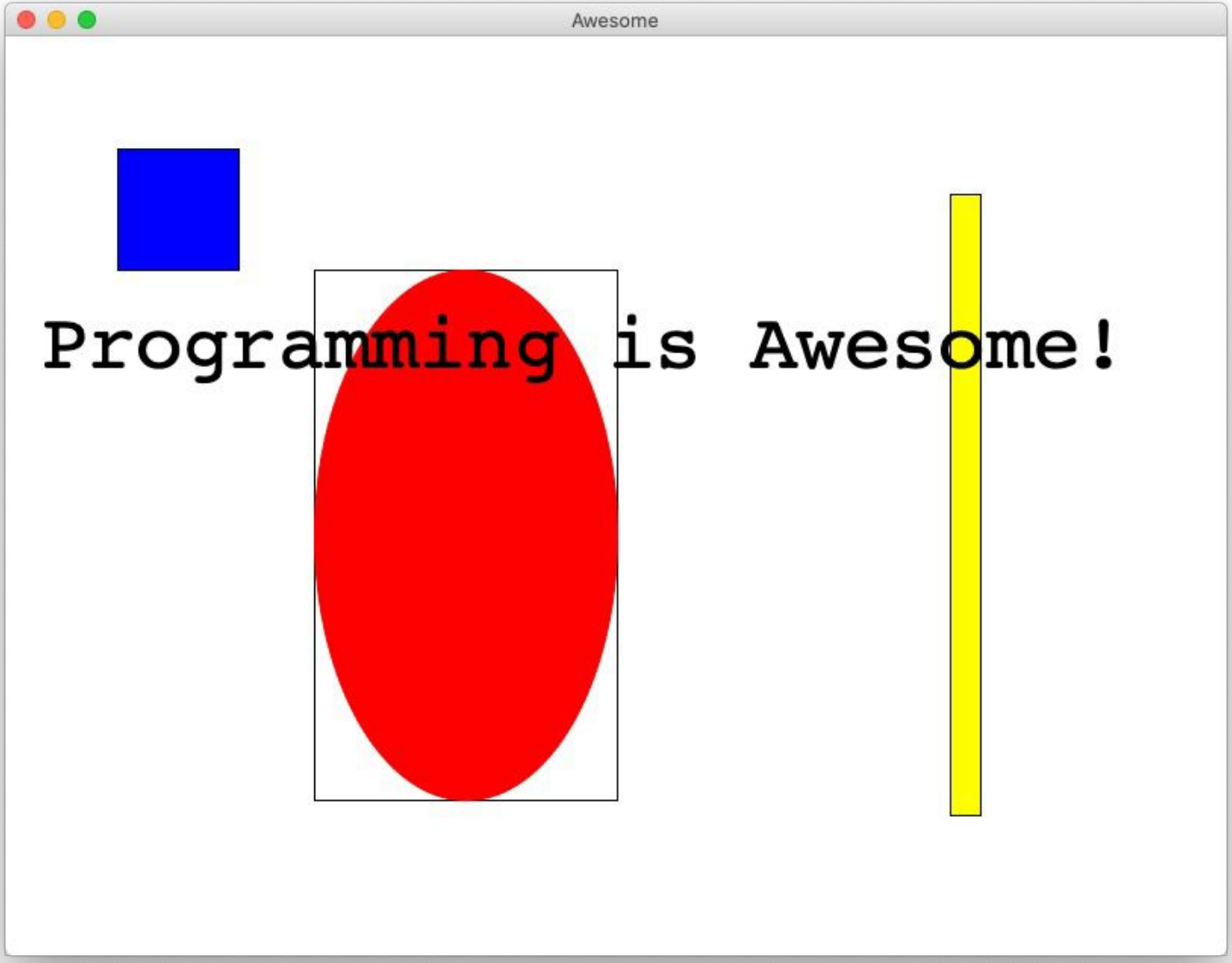
Graphics Coordinates



Graphics Coordinates



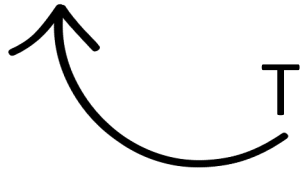
Rectangles, Ovals, Text



- `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()`

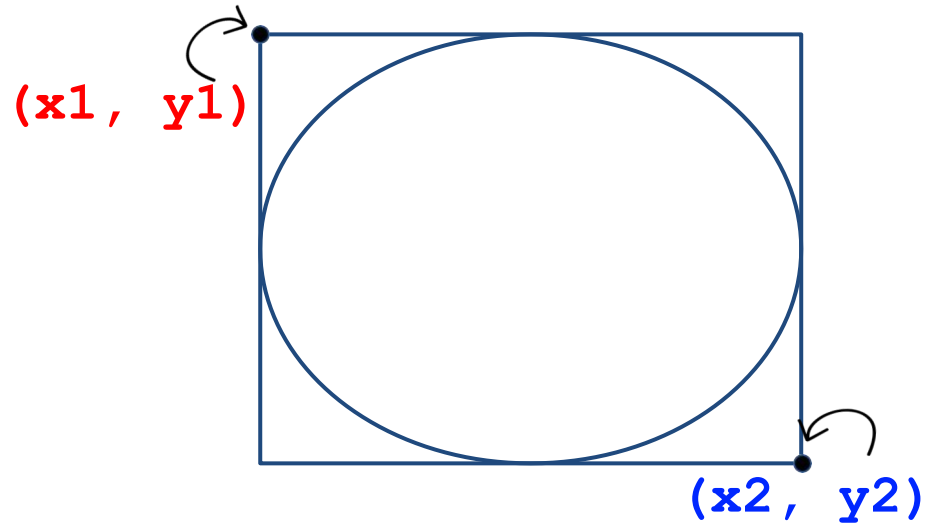
The second point of the line is `(x2, y2)`

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

Stanford | ENGINEERING
Computer Science

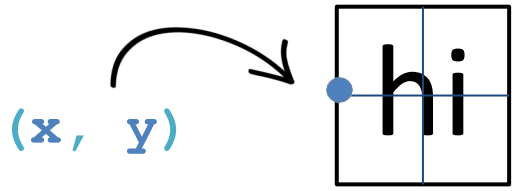
- `canvas.create_line()`
- **`canvas.create_oval(x1, y1, x2, y2)`**
- `canvas.create_text()`

- 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')`**



Code demo:

`programming_is_awesome.py`

because it is!

Aside: constants

- Constants are like variables: they store a value, but we don't change that value
- They are useful for "naming" numbers and other values that you might use multiple times throughout your code
- We usually define them at the top of files in ALL_CAPS, for every function to use

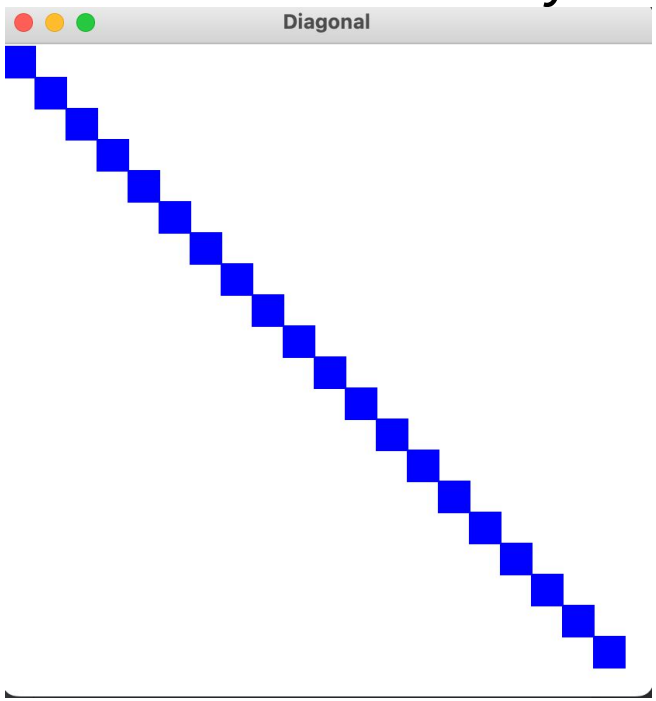
```
WIDTH = 20

def draw_square(canvas):
  canvas.create_rect(0, 0, WIDTH, WIDTH)

def main():
   canvas = Canvas(WIDTH * 2, WIDTH * 2, 'square')
   draw_square(canvas)
```

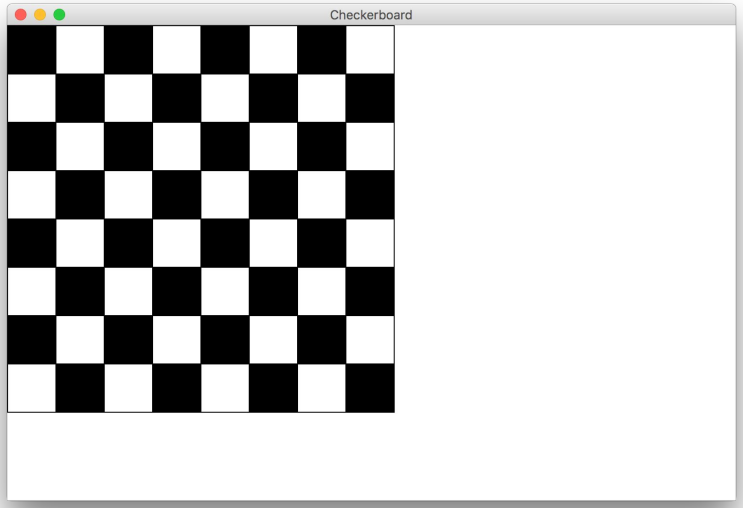
Kick it up a notch: `diagonal.py`

- Write a program, `diagonal.py`, draws a diagonal line of squares for a certain number of rows.
- The number of rows is saved in a variable named row in the main function - you can ignore the code above it (we'll learn about that later)
- Decompose a function to actually draw the squares



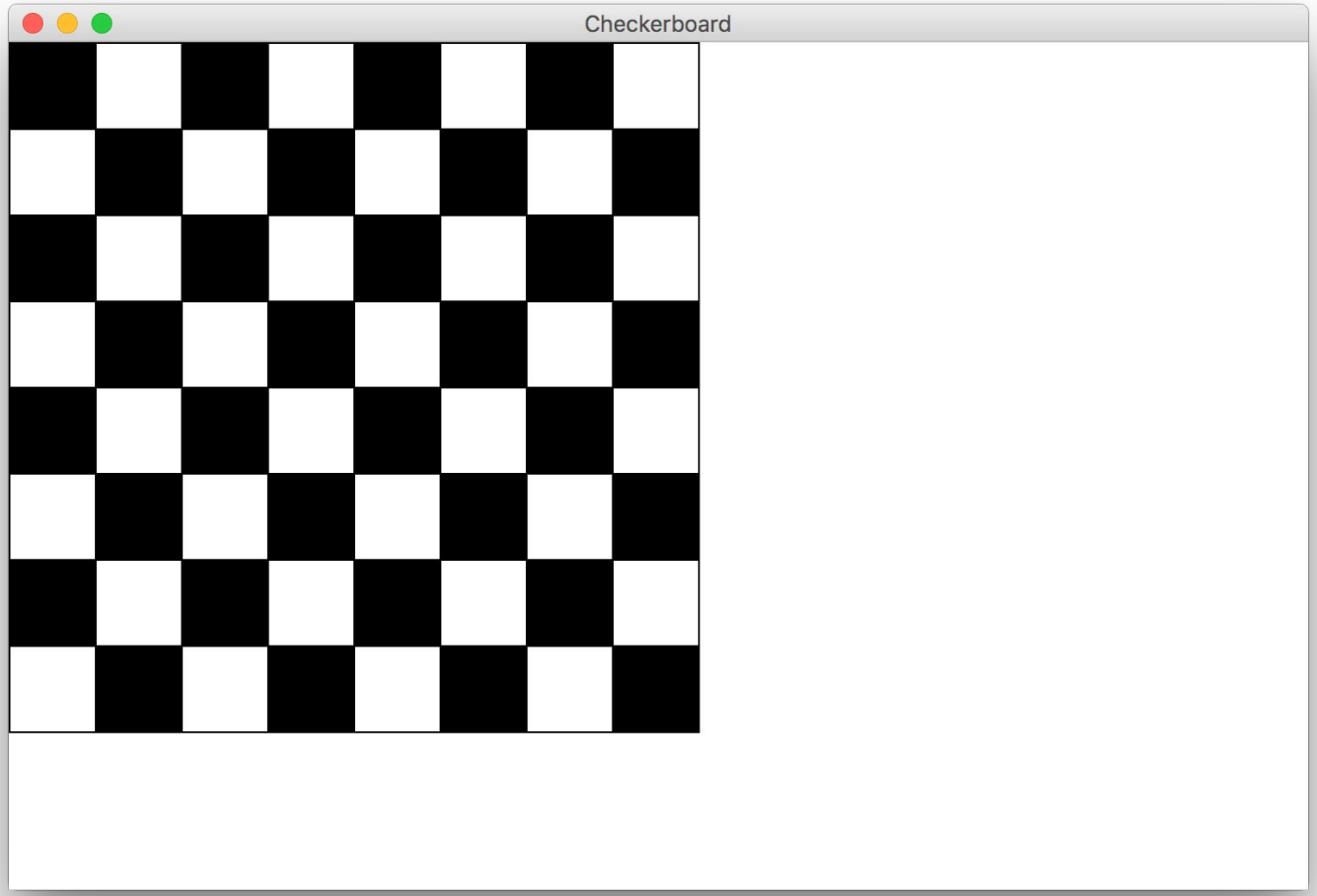
Bring it back! `checkerboard.py`

- Write a program, `checkerboard.py`, which creates a canvas and draws a square checkerboard with a certain number of rows
- You will be given the rows variable in the main function, you can ignore the code above it
- Decompose a function to actually make the checkerboard

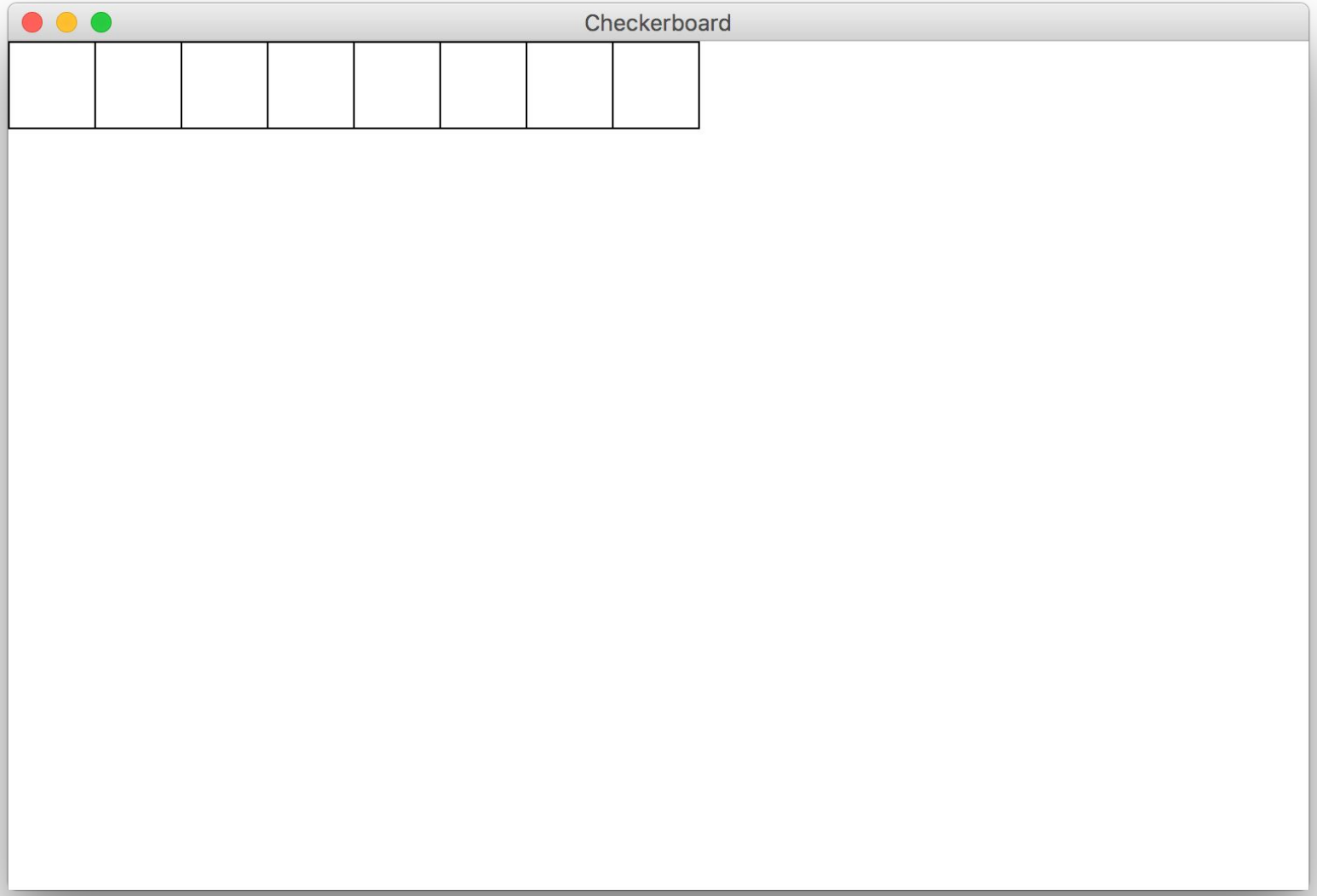


Stanford | ENGINEERING
Computer Science

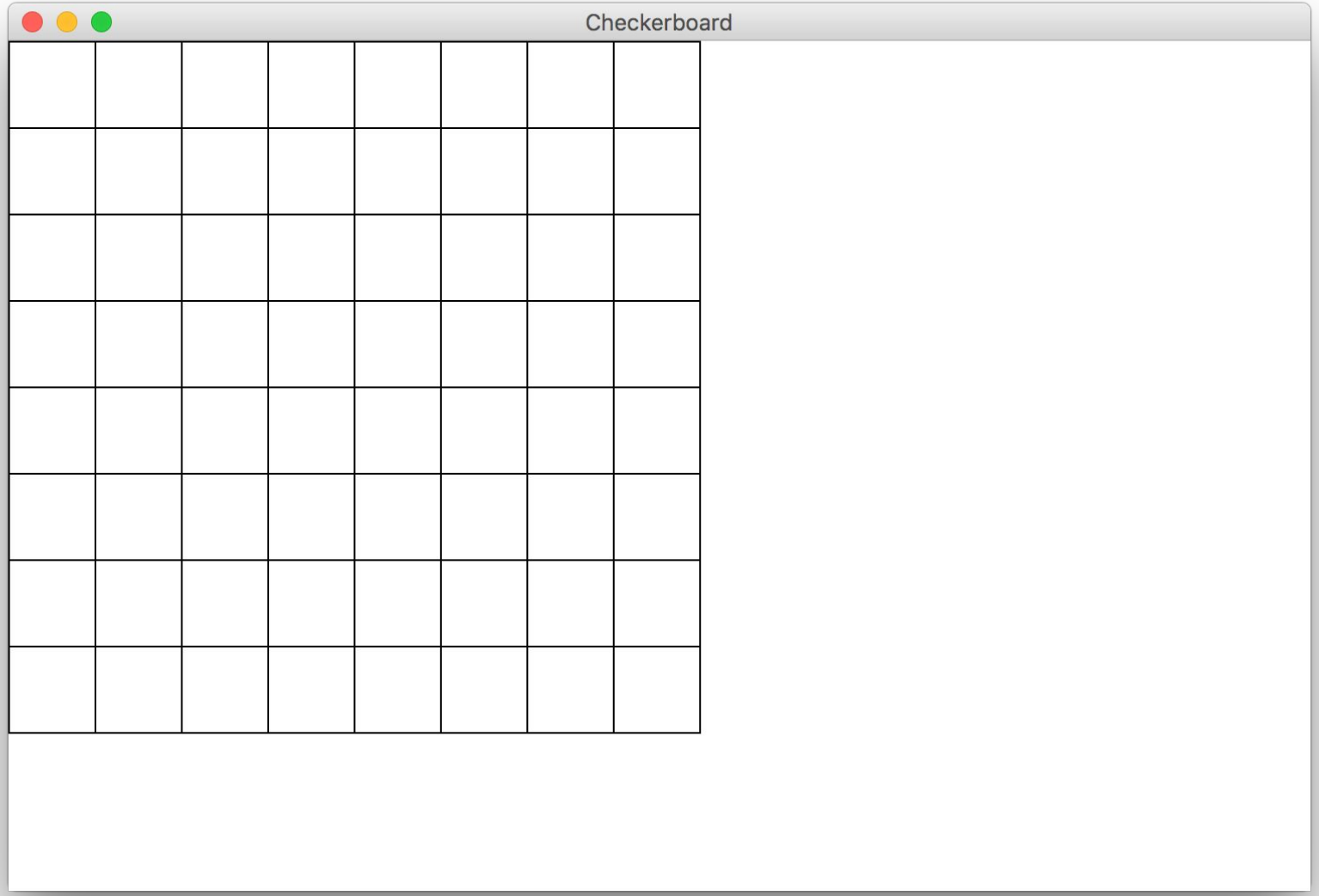
Goal



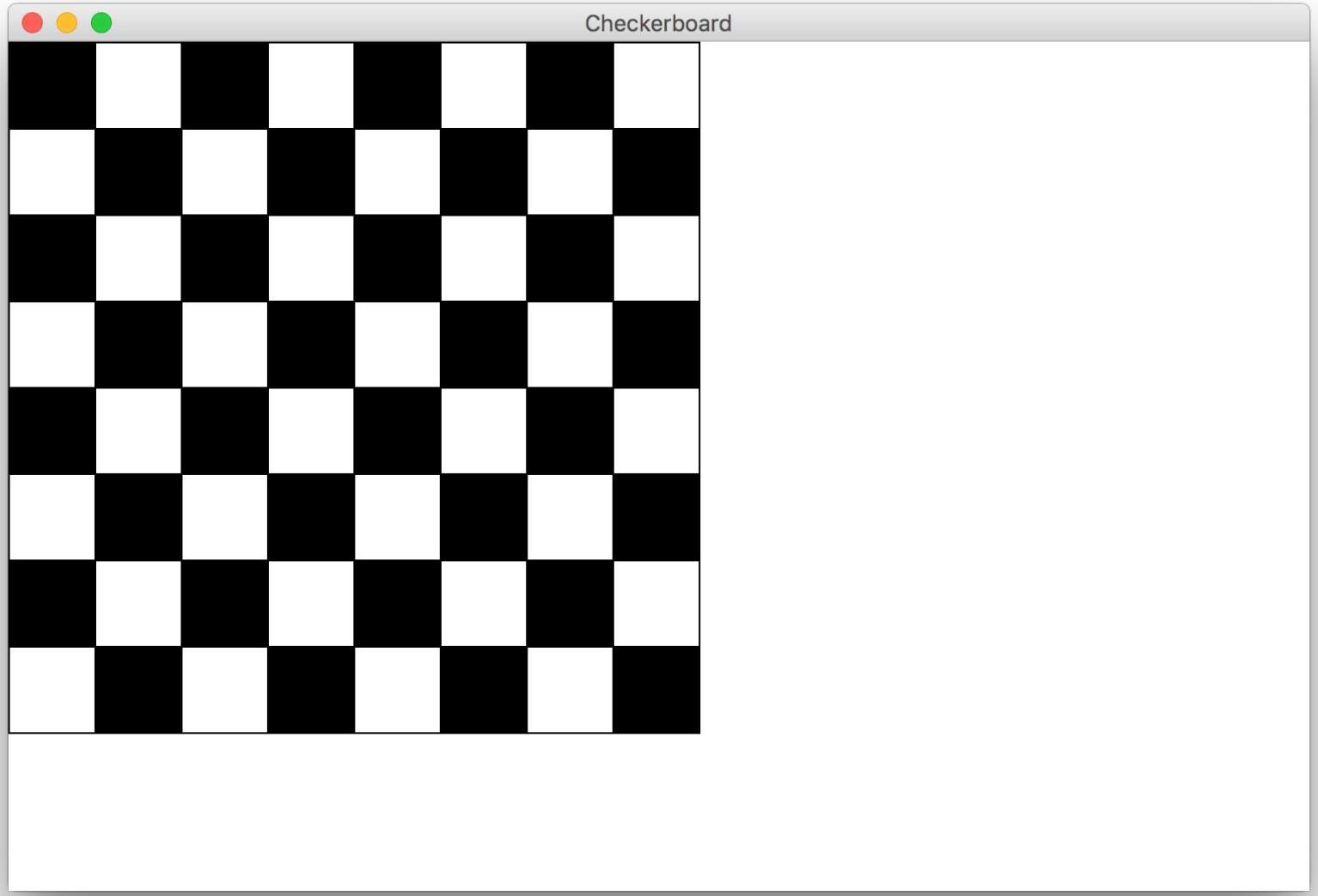
Milestone 1



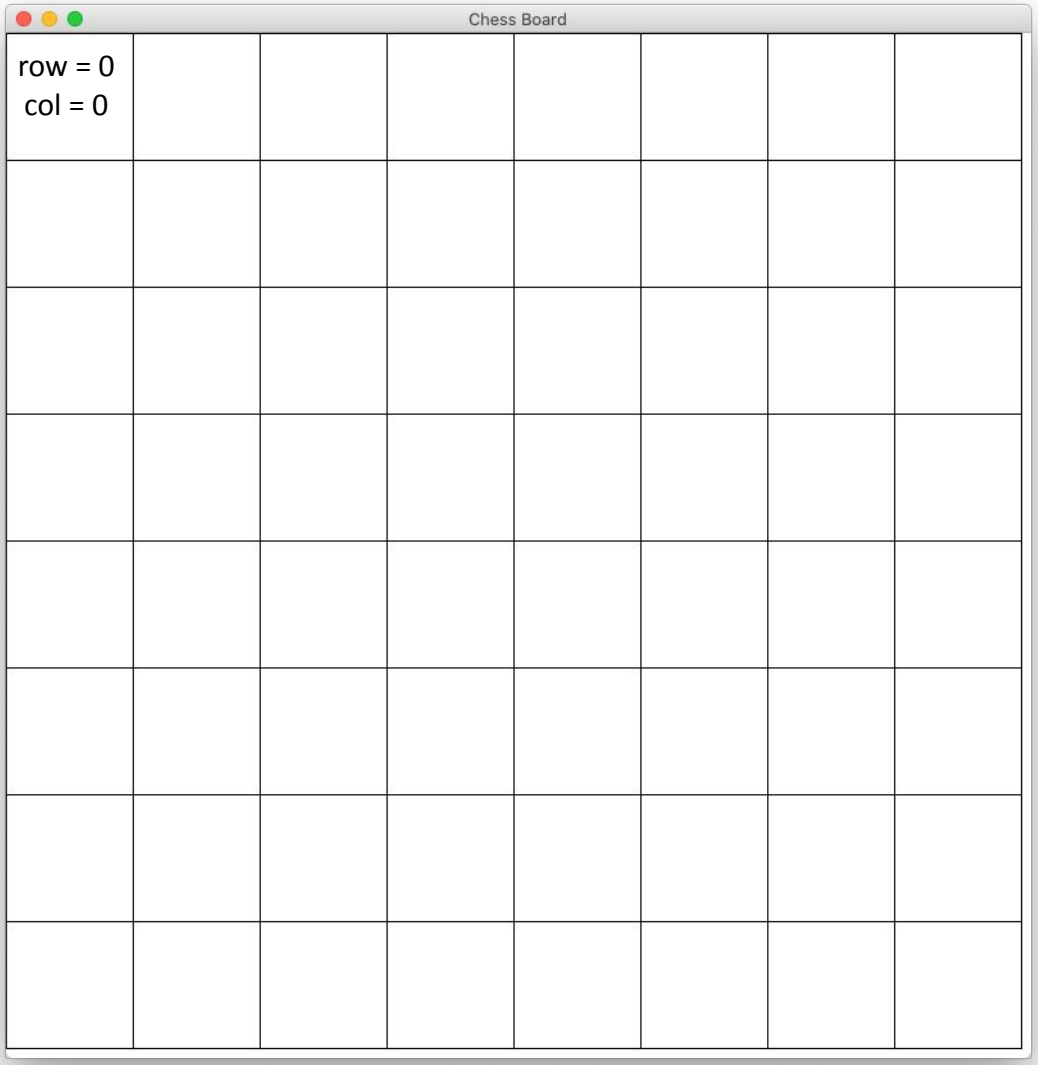
Milestone 2



Milestone 3

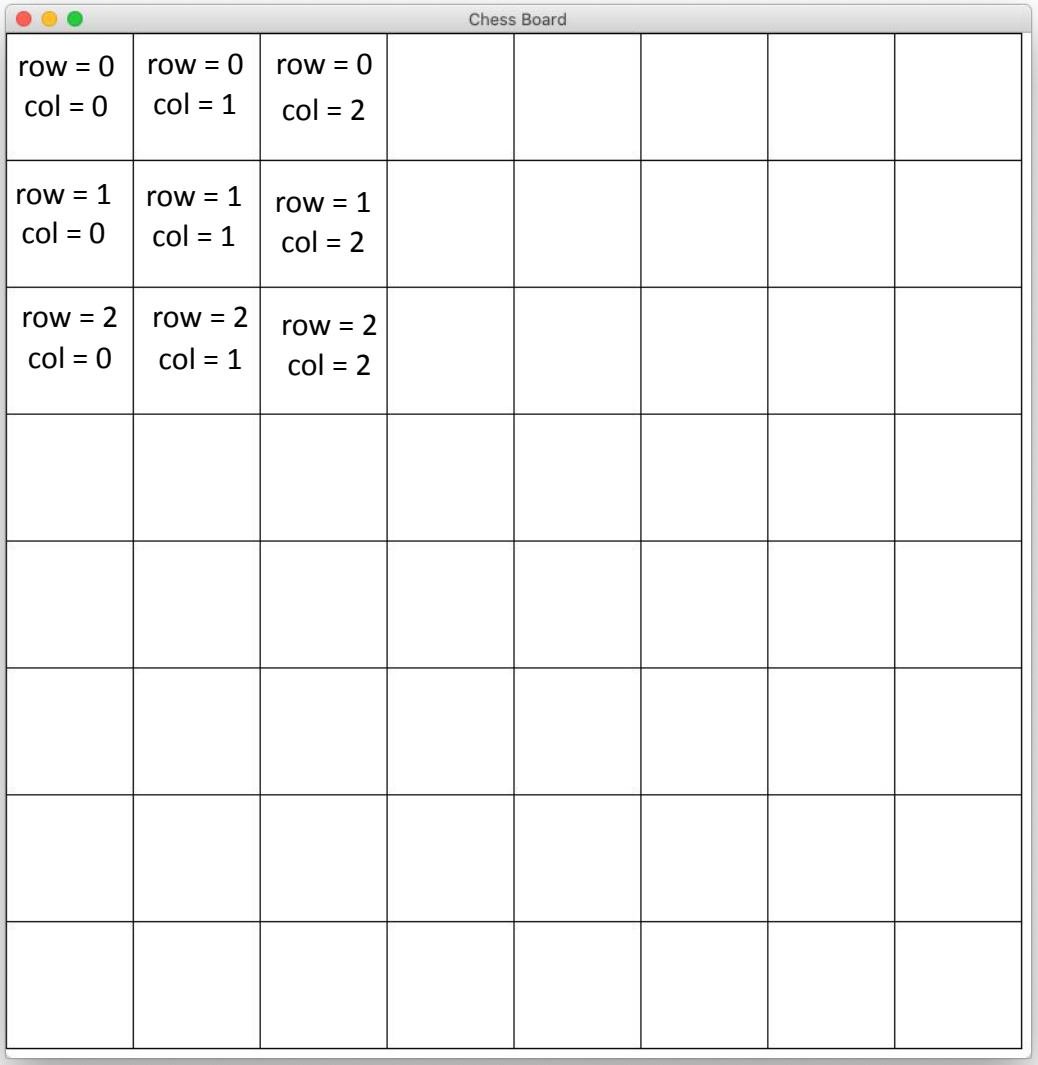


Milestone 3



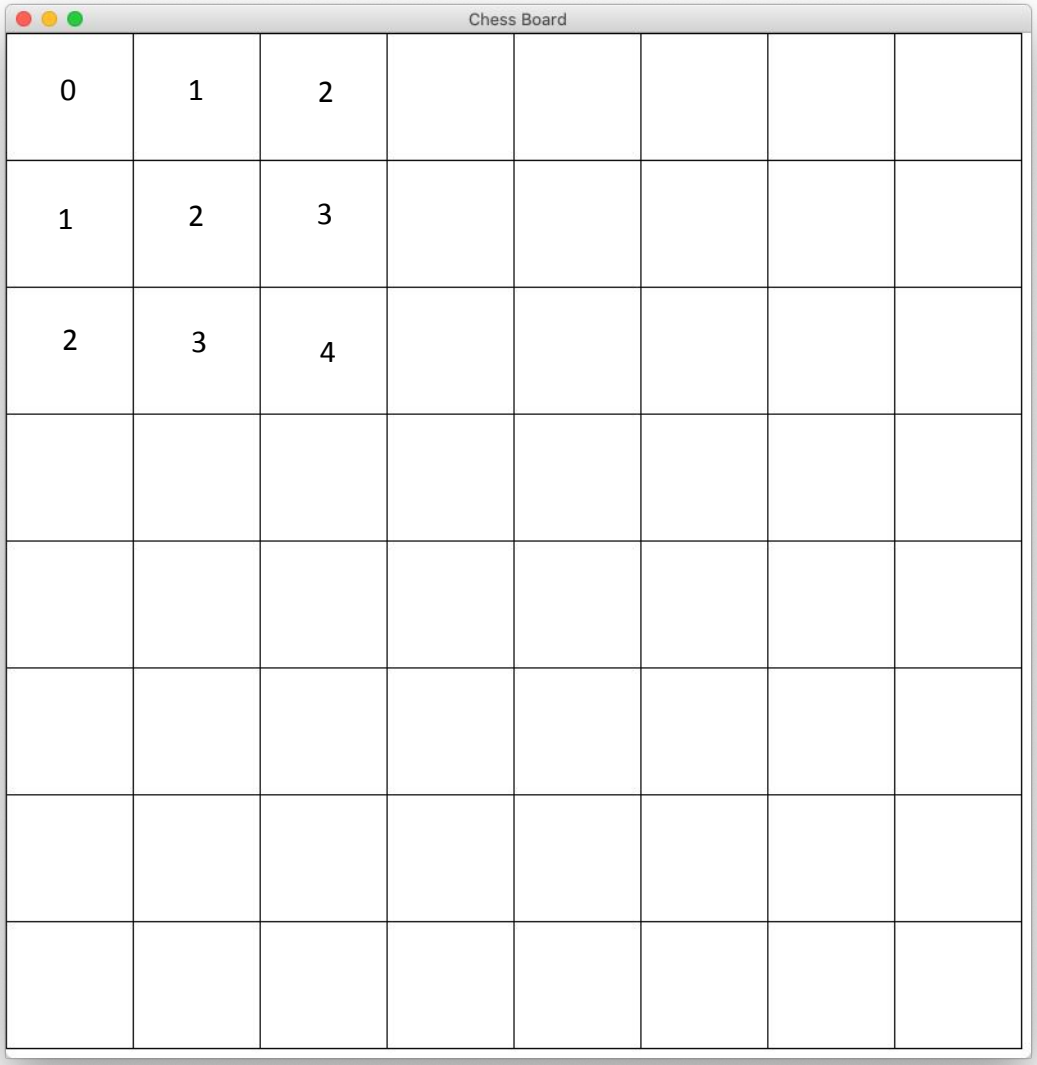
Stanford | ENGINEERING
Computer Science

Milestone 3



Stanford | ENGINEERING
Computer Science

Row + Col



Recap

- Check out the Graphics reference on the course website
- We can draw shapes!
- We can use loops etc to draw patterns of shapes!
- We will start writing our code in the main function from now on