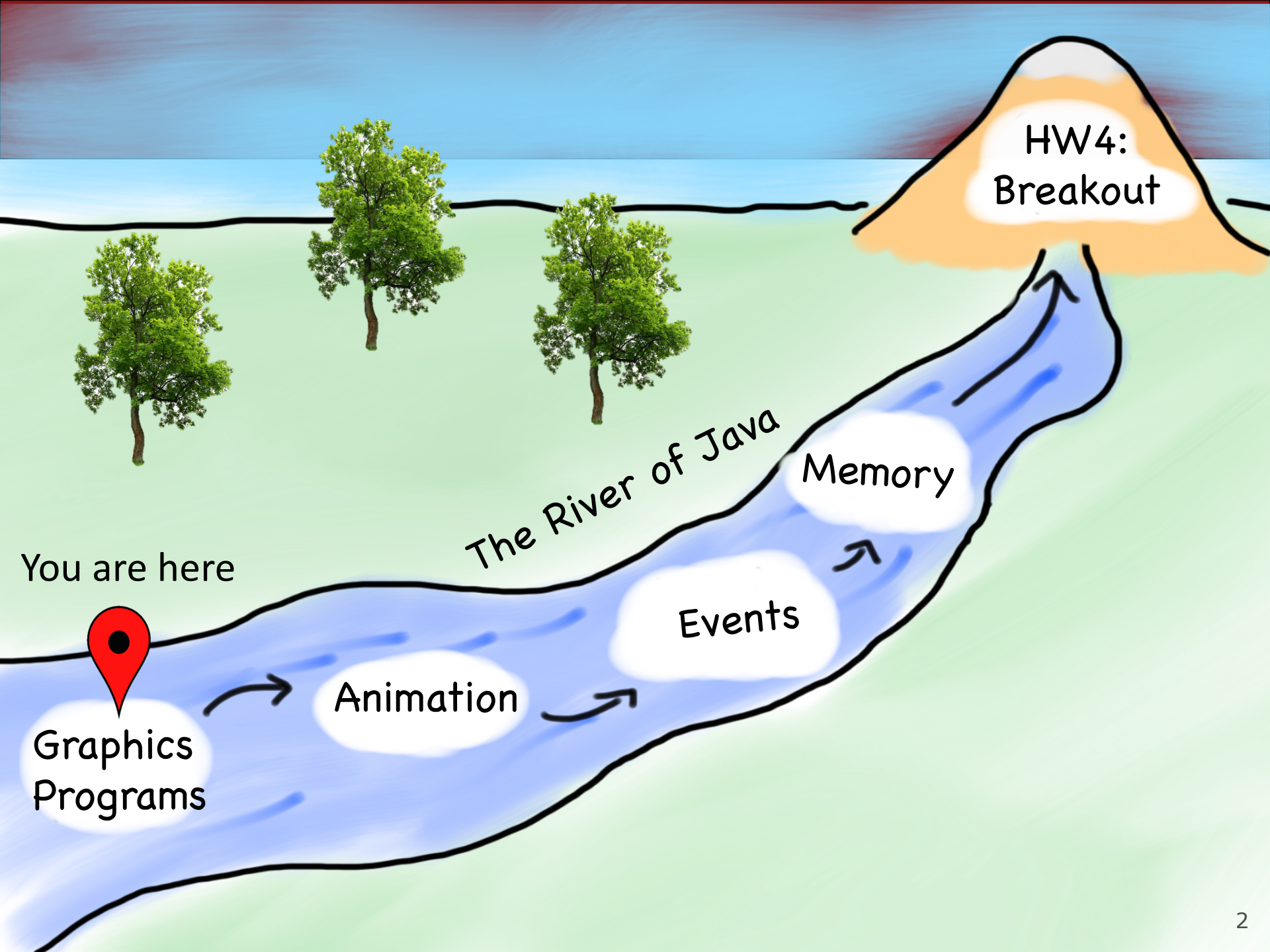
CS 106A, Lecture 12
More Graphics

reading:

Art & Science of Java, 9.4

This document is copyright (C) Stanford Computer Science and Marty Stepp, licensed under Creative Commons Attribution 2.5 License. All rights reserved.
Based on slides created by Keith Schwarz, Mehran Sahami, Eric Roberts, Stuart Reges, and others.


HW4:
Breakout
The River of Java
Memory
Events
Animation
You are here
Graphics
Programs

Plan For Today

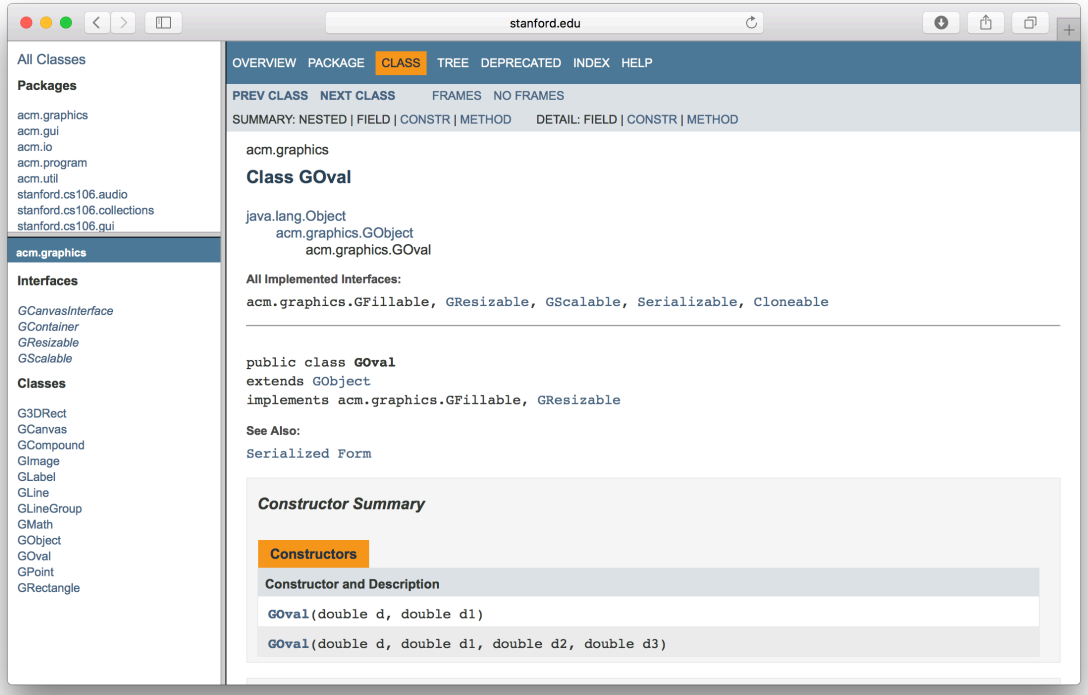
- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

Plan For Today

- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

Announcements: Docs

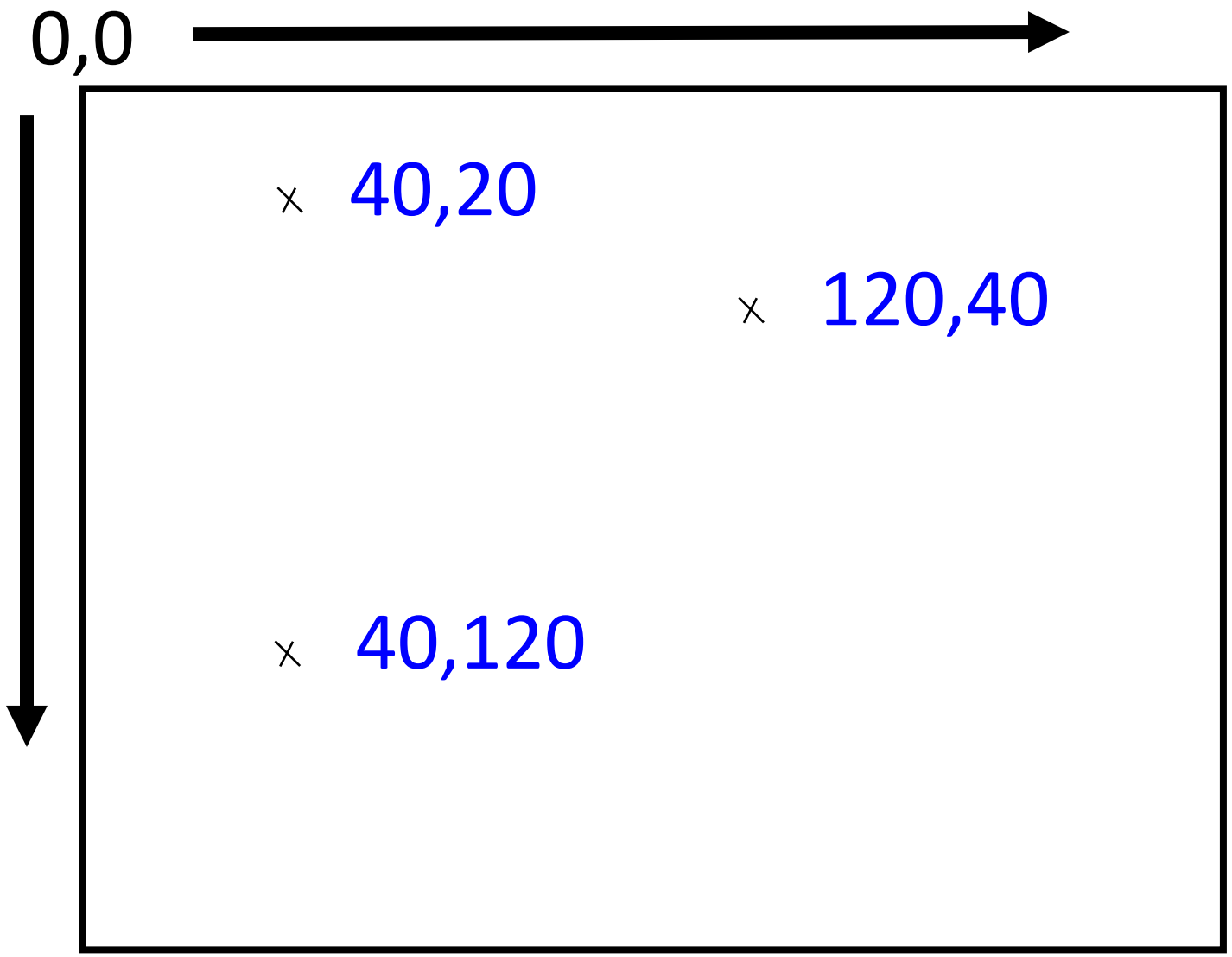
- Click the "**Stanford Library Docs**" link in the 106A website sidebar.
  - This site lists every kind of object in the Stanford libraries.
  - Click an object type on the left and see its behavior on the right.
  - These kinds of pages exist for Stanford libraries and standard Java.



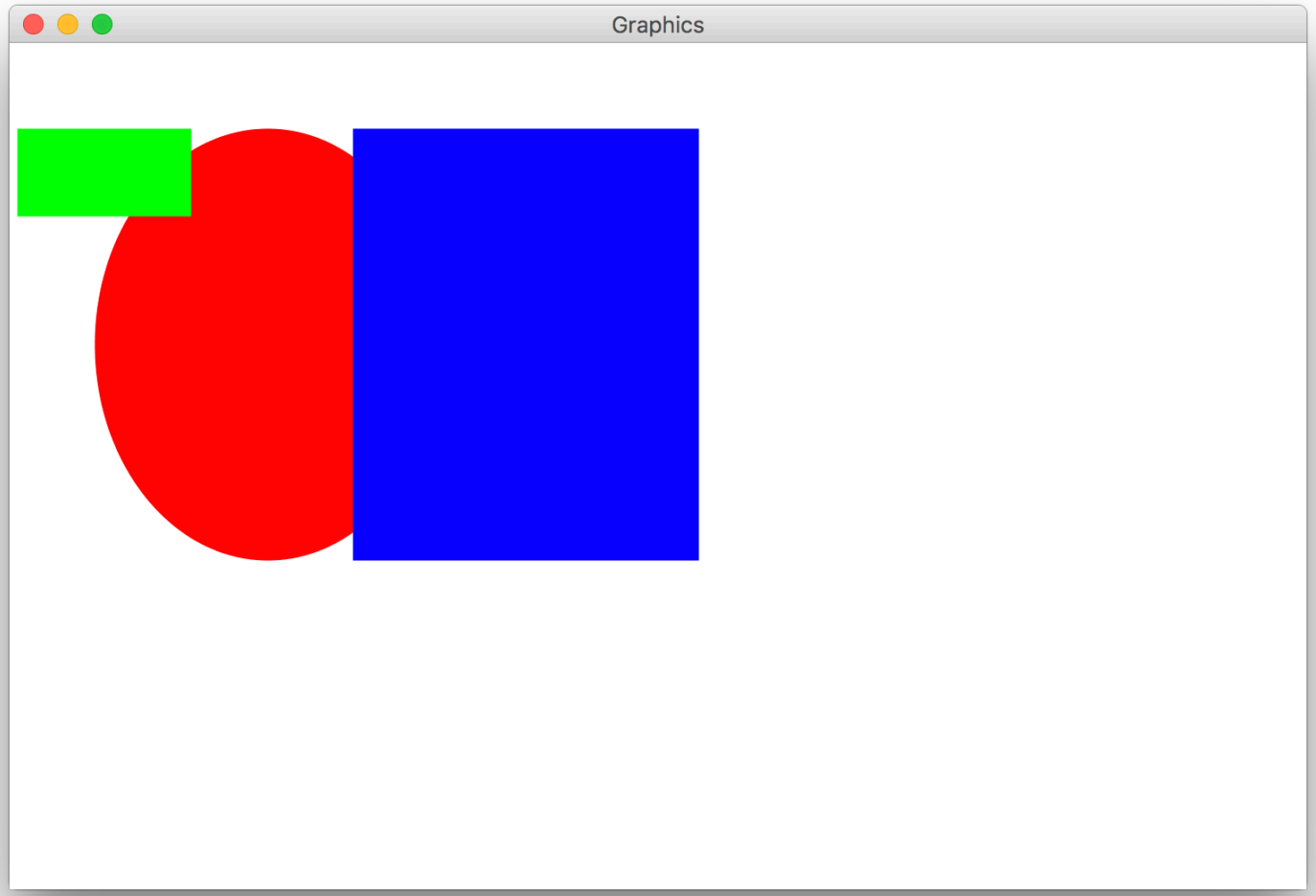
Plan For Today

- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

The Graphics Canvas

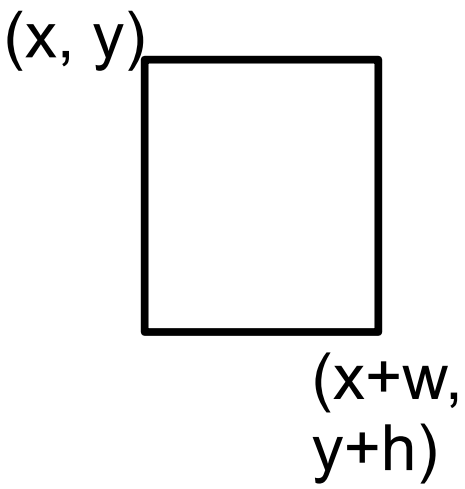


Collage Model

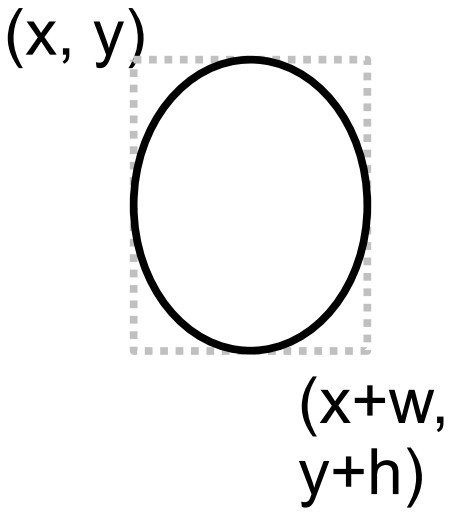


Graphical Objects

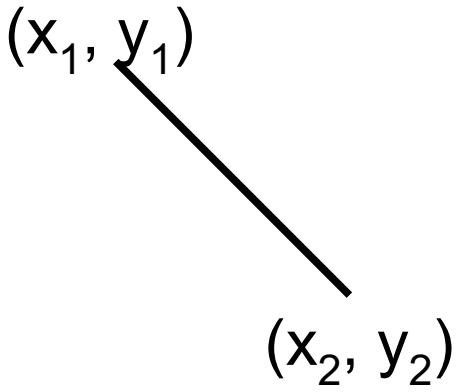
GRect



GOval



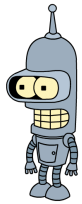
GLine



GLabel

Hello there!

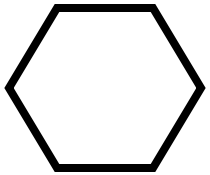
GImage



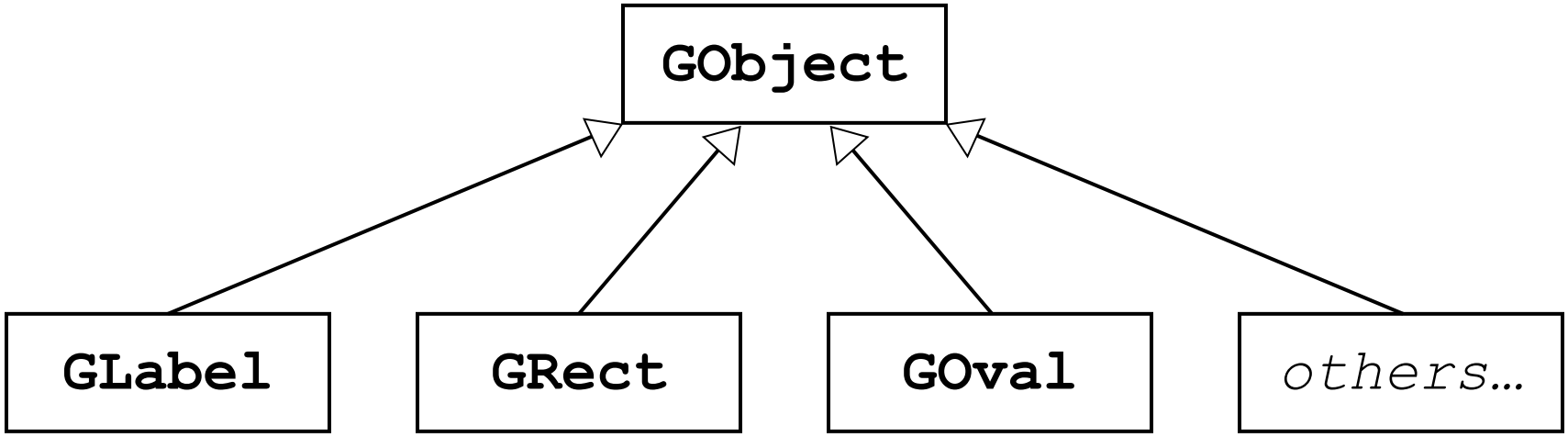
GArc

GRoundRect

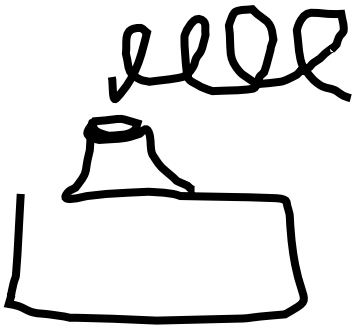
GPolygon



Graphical Objects




```
GRect myRect = new GRect(50, 50, 350, 270);
```



Primitives vs. Objects

Primitive Variable Types

```
int
double
char
boolean
```

Object Variable Types

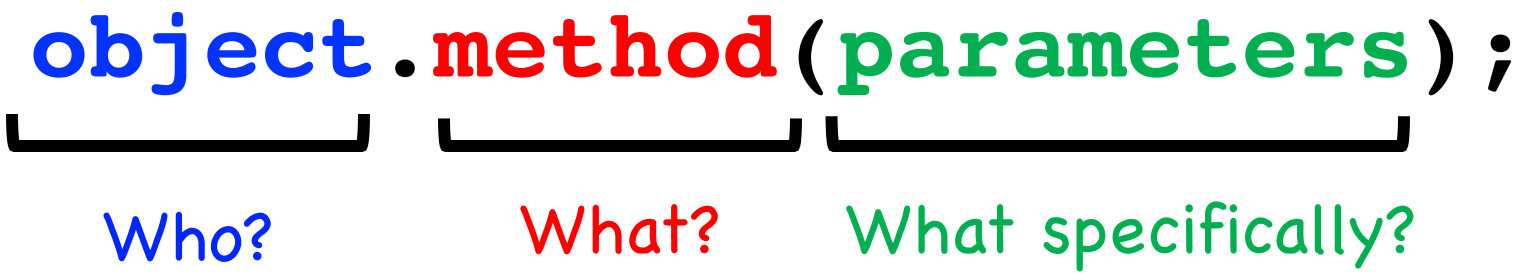
```
GRect
GOval
GLine
Scanner
...
```

Object variables:

1. Have upper camel case types
2. You can call methods on them
3. Are constructed using `new`

Methods on Graphics Objects

We manipulate graphics objects by calling methods on them:



Example:

```
rect.setColor(Color.RED);
```

GObject Methods

The following operations apply to all **`GObject`**s:

object`.`**`setColor`**`(`*color*`)` Sets the color of the object to the specified color constant.
object`.`**`setLocation`**`(`*x*`, `*y*`)` Changes the location of the object to the point (*x*, *y*).
object`.`**`move`**`(`*dx*`, `*dy*`)` Moves the object on the screen by adding *dx* and *dy* to its current coordinates.
object`.`**`getWidth`**`()` Returns the width of the object
object`.`**`getHeight`**`()` Returns the height of the object
and more…

Graphic courtesy of Eric Roberts

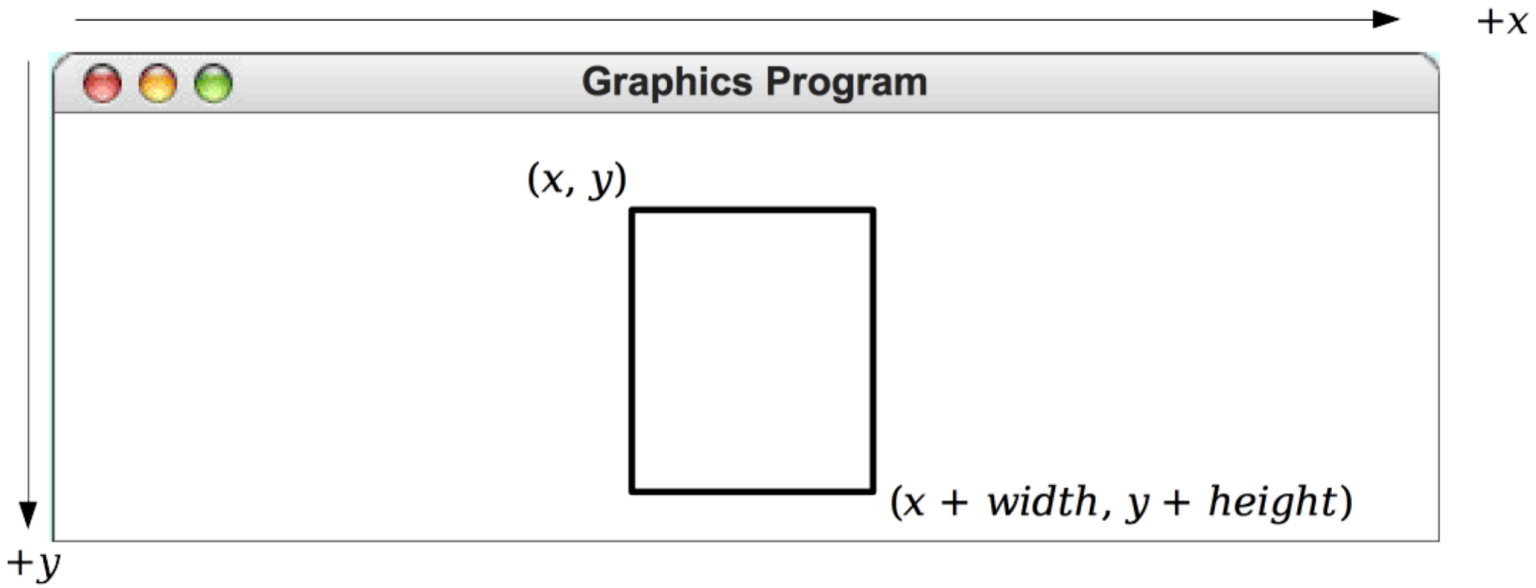
GRect

`new GRect(`*`x, y, width, height`*`);`

- Creates a rectangle with the given width and height, whose upper-left corner is at (x, y)

`new GRect(`*`width, height`*`);`

- Same as above, but defaults to (x, y) = (0, 0)



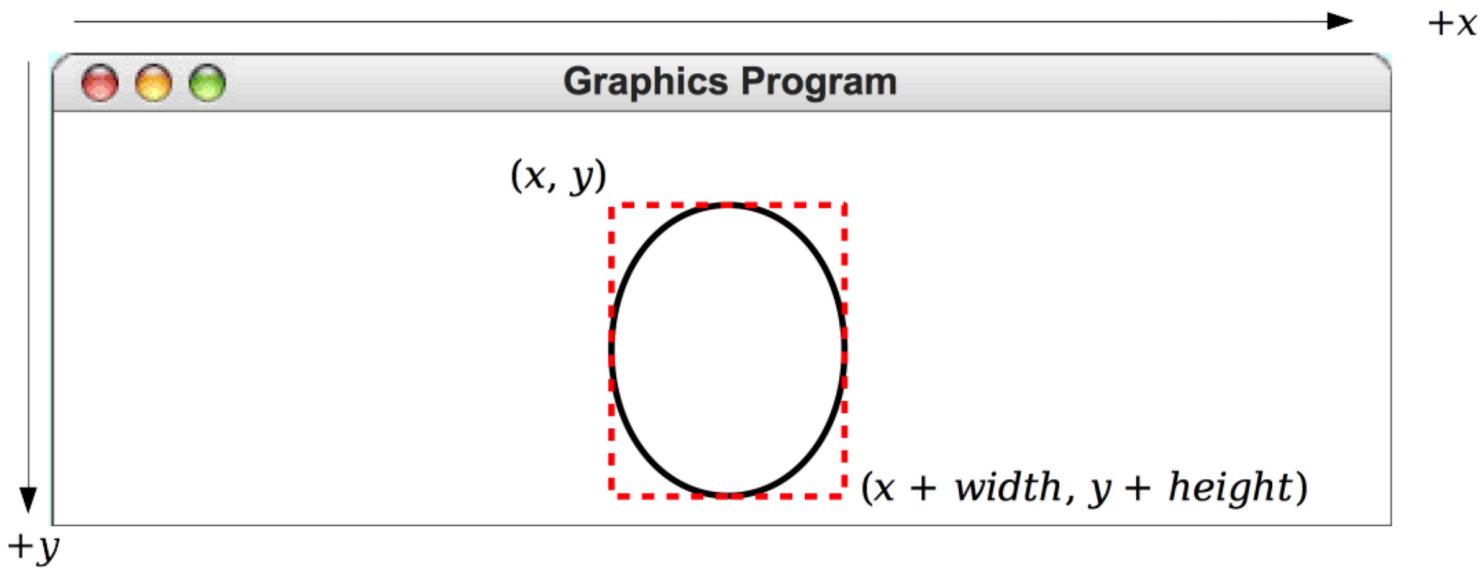
GOval

`new GOval(x, y, width, height);`

- Creates an oval that fits inside a rectangle with the given width and height, and whose upper-left corner is at (x, y)

`new GOval(width, height);`

- Same as above, but defaults to (x, y) = (0, 0)



GRect and GOval

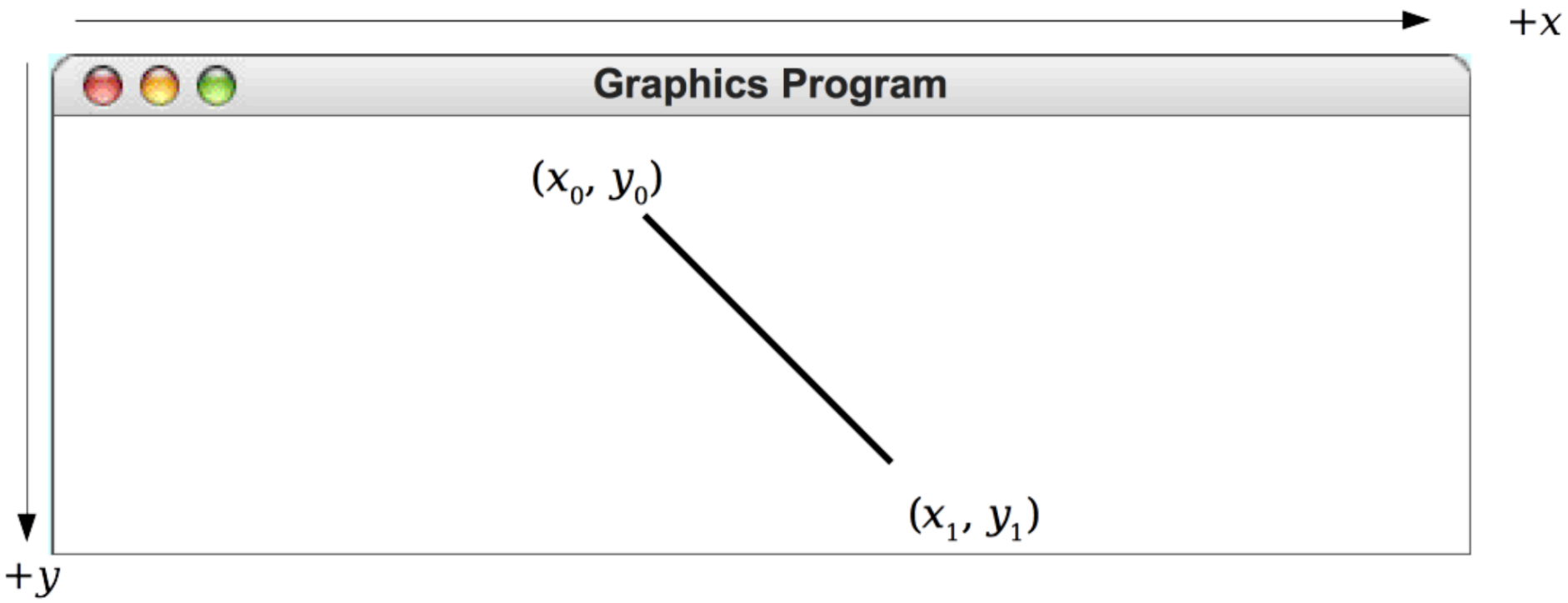
Methods shared by the **`GRect`** and **`GOval`** classes

object`.`**`setFilled(`***fill***`)`** If *fill* is **`true`**, fills in the interior of the object; if **`false`**, shows only the outline.
object`.`**`setFillColor(`***color***`)`** Sets the color used to fill the interior, which can be different from the border.
object`.`**`setSize(`***width, height***`)`** Sets the object's size to be the given width and height

GLine

`new GLine(`*`x0, y0, x1, y1`*`);`

- Creates a line extending from (x0, y0) to (x1, y1)



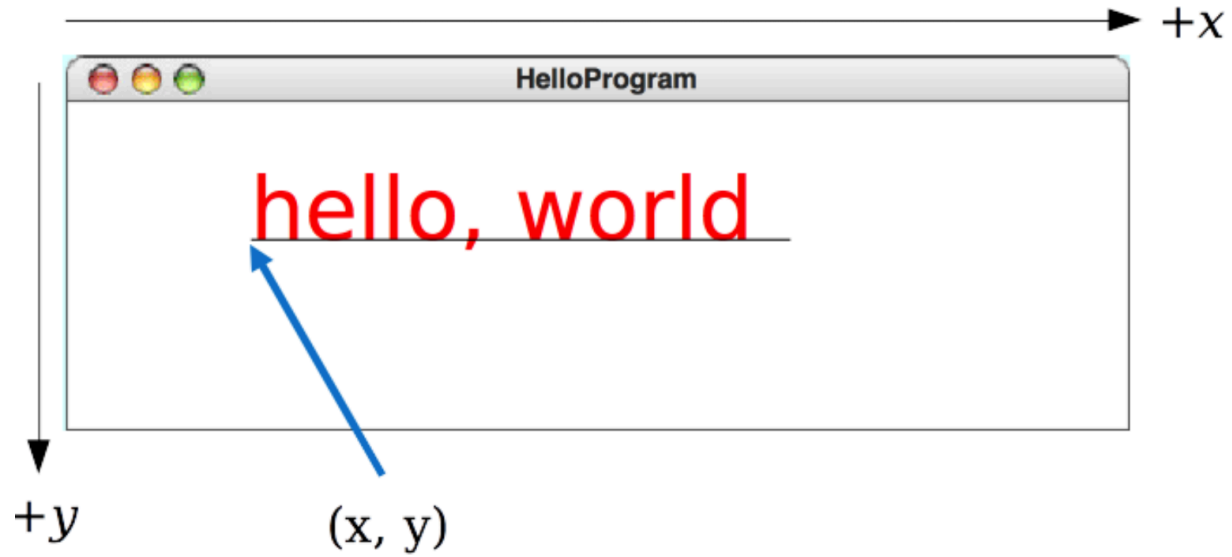
GLabel

`new GLabel(“your text here”, x, y);`

- Creates a label with the given text, whose ***baseline*** starts at (x, y). NOT positioned according to the top-left corner!

`new GLabel(“your text here”);`

- Same as above, but defaults to (x, y) = (0, 0)



GLabel Methods

Methods specific to the **`GLabel`** class

label`.`**`getDescent()`** Returns the height of the label below its baseline.
label`.`**`getAscent()`** Returns the height of the label above its baseline.
label`.`**`setFont(`***font***`)`** Sets the font used to display the label as specified by the font string.

The font is typically specified as a string in the form

"*family*-*style*-*size*"

family is the name of a font family (e.g. "SansSerif")
style is either **`PLAIN`**, **`BOLD`**, **`ITALIC`**, or **`BOLDITALIC`**
size is an integer indicating the point size

Graphic courtesy of Eric Roberts

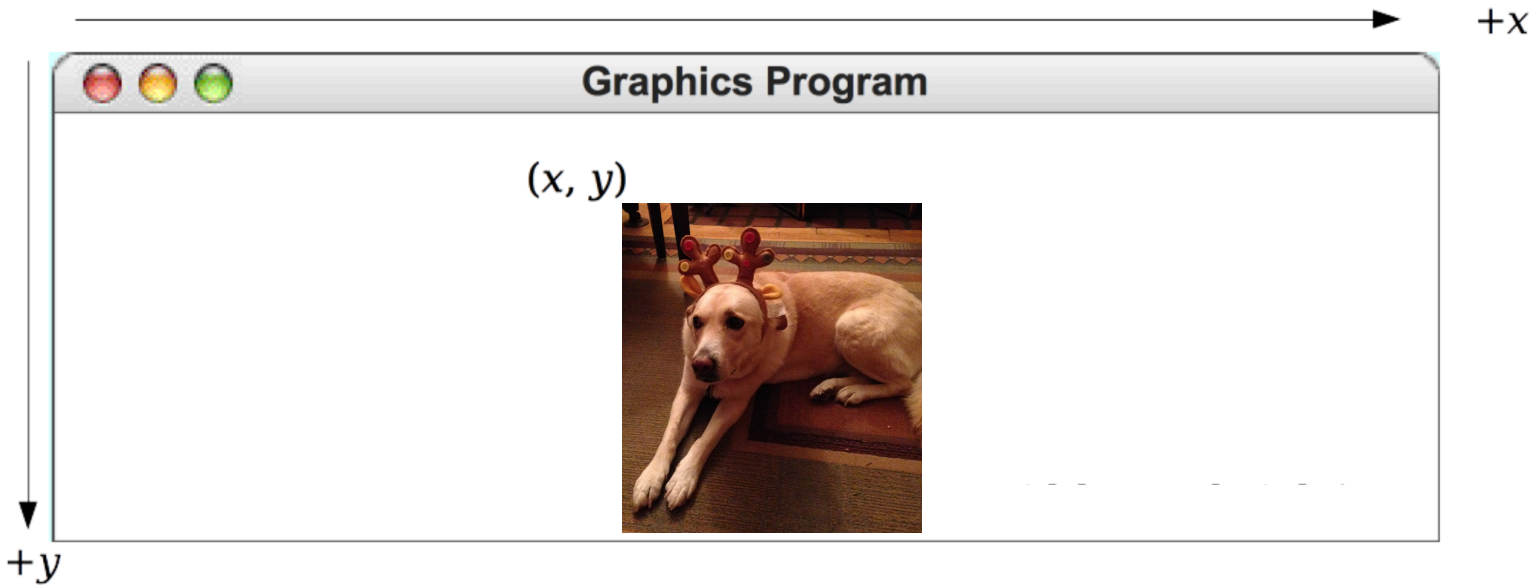
GImage

`new GImage("your filename here", x, y);`

- Creates a an image displaying the given file, whose upper-left corner is at (x, y)

`new GImage("your filename here");`

- Same as above, but defaults to (x, y) = (0, 0)



GImage Methods

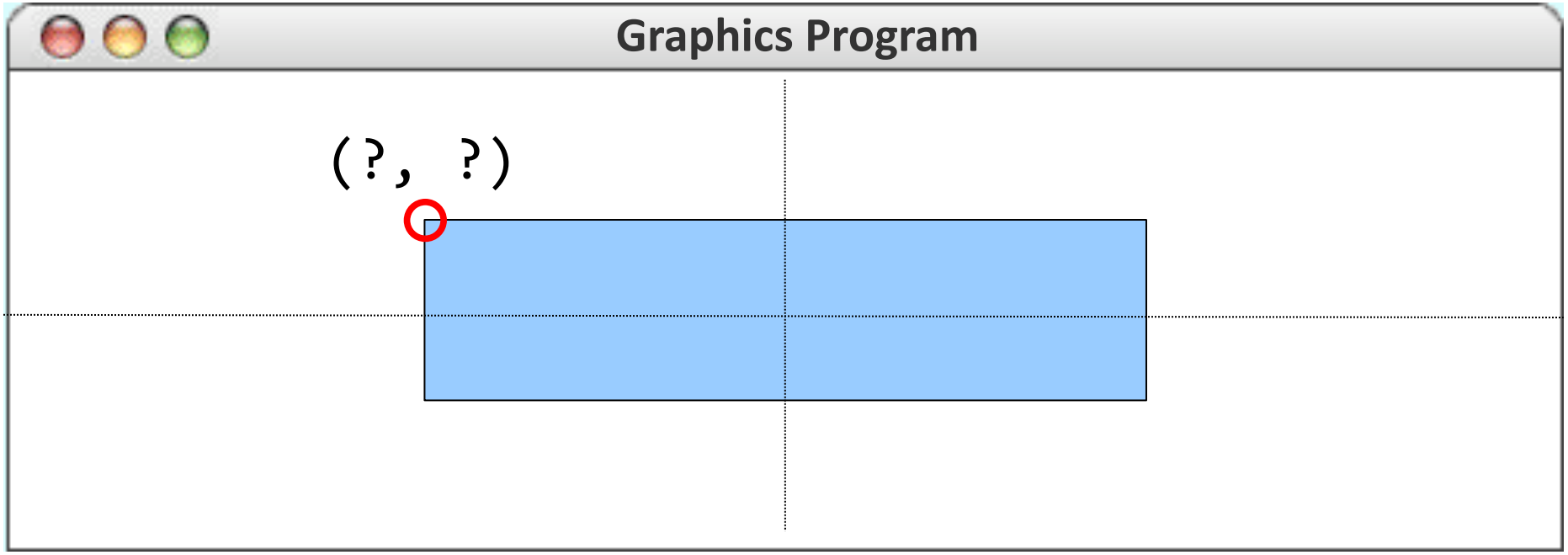
object`.setSize(`*width, height*`)`
Sets the object's size to be the given width and height

GraphicsProgram Methods

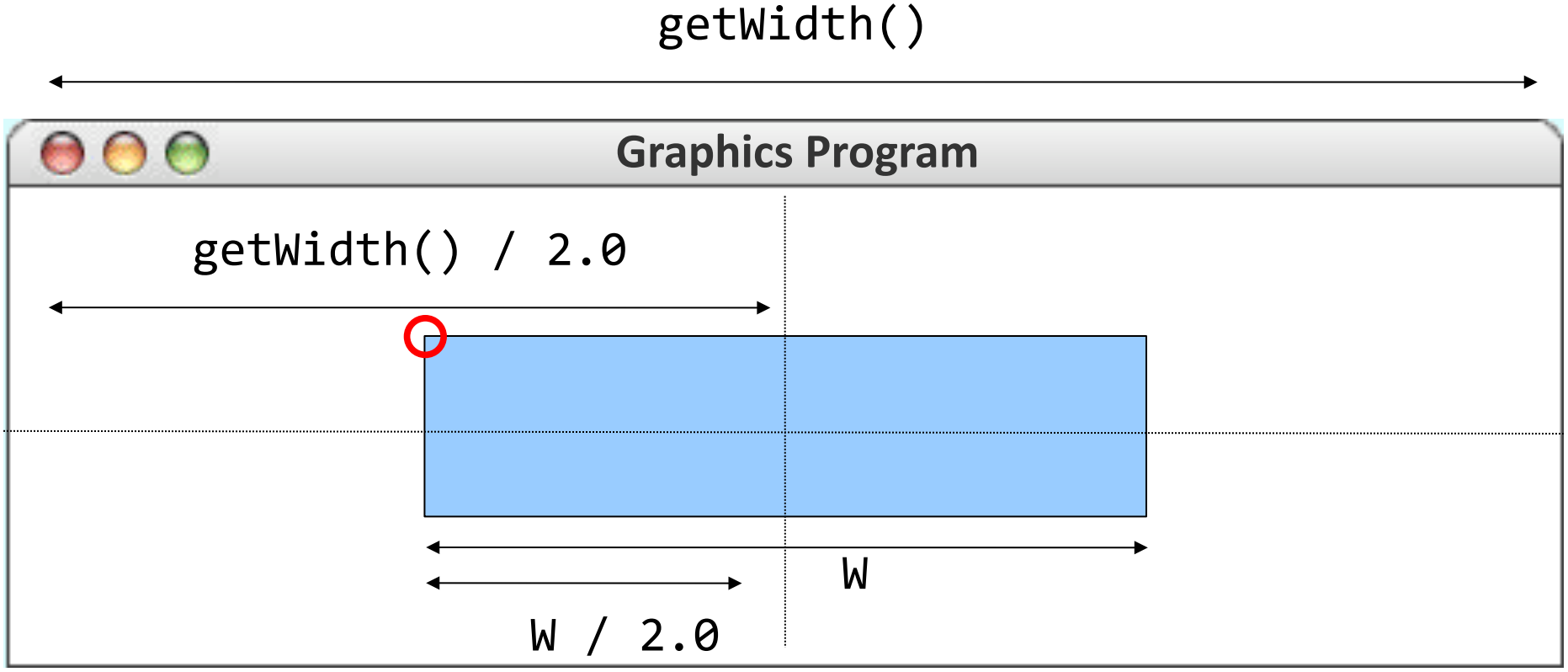
- `GraphicsProgram` contains these useful methods:

Method	Description
`add(gobj);` `add(gobj, x, y);`	adds a graphical object to the window
`getElementAt(x, y)`	return the object at the given (x,y) position(s)
`getElementCount()`	return number of graphical objects onscreen
`getWidth(), getHeight()`	return dimensions of window
`remove(gobj);`	removes a graphical object from the window
`removeAll();`	remove all graphical objects from window
`setCanvasSize(w, h);`	set size of drawing area
`setBackground(color);`	set window's background color

Recap Practice: Centering

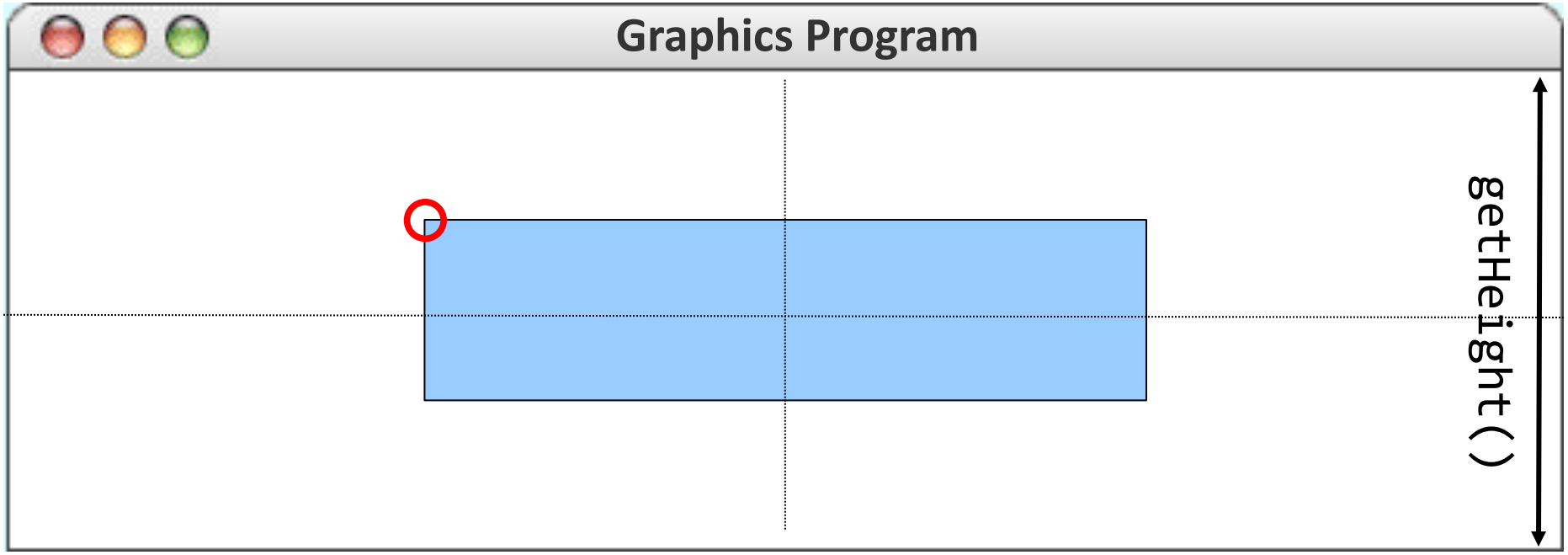


Recap Practice: Centering




```
rectangle's x value = getWidth() / 2.0 – W / 2.0
```

Recap Practice: Centering




```
rectangle's y value = getHeight() / 2.0 - H / 2.0
```

Plan For Today

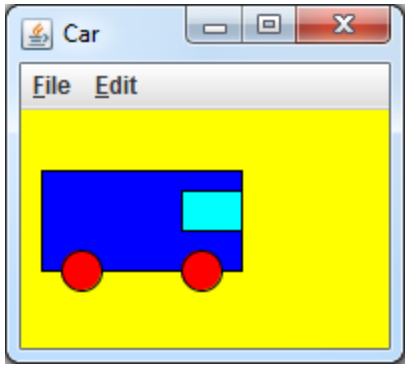
- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

GCompound

A **GCompound** contains other GObjects. It's useful when you want to do one operation on multiple GObjects at the same time.

```
GCompound compound = new GCompound();
compound.add(shape);
compound.add(shape);
...
compound.add(shape);

add(compound);
```



– E.g. you can make a GCompound to represent a car.

GCompound

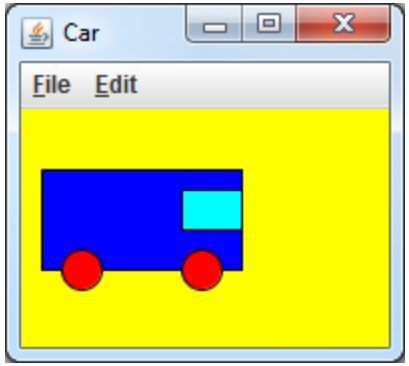
```
setBackground(Color.YELLOW);
GCompound car = new GCompound();

GRect body = new GRect(10, 30, 100, 50);
body.setFilled(true);
body.setFillColor(Color.BLUE);
car.add(body);

GOval wheel1 = new GOval(20, 70, 20, 20);
wheel1.setFilled(true);
wheel1.setFillColor(Color.RED);
car.add(wheel1);

GOval wheel2 = new GOval(80, 70, 20, 20);
wheel2.setFilled(true);
wheel2.setFillColor(Color.RED);
car.add(wheel2);

GRect windshield = new GRect(80, 40, 30, 20);
windshield.setFilled(true);
windshield.setFillColor(Color.CYAN);
car.add(windshield);
add(car);      // at  0,0!  Where we want this "sub-canvas" to go
```



Plan For Today

- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

Graphics Program "Getters"

- Methods of graphical objects that return values:

Method	Description
obj.getColor()	the color used to color the shape outline
obj.getFillColor()	the color used to color the shape interior
obj.getX()	the left *x*-coordinate of the shape
obj.getY()	the top *y*-coordinate of the shape
obj.getWidth()	number of pixels wide the shape is
obj.getHeight()	number of pixels tall the shape is

  - Example: Swapping the x/y coordinates of a shape:

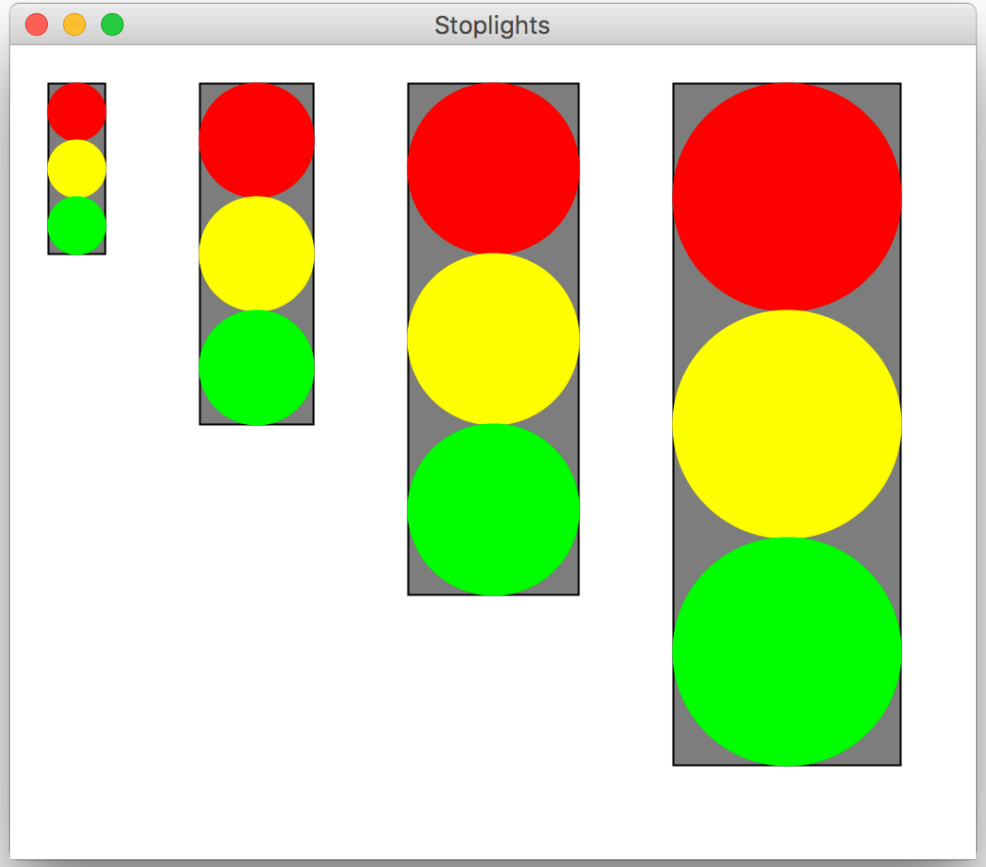
```
GRect rect = new GRect(...);
...
int rx = rect.getX();
int ry = rect.getY();
rect.setLocation(ry, rx);
```

Plan For Today

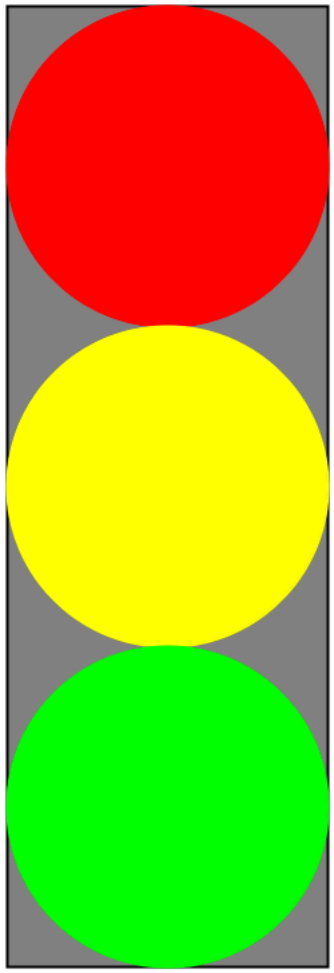
- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

Practice: Stoplights

How would you make a method for drawing stoplights of different locations and sizes?



Practice: Stoplights



What information do we need in order to draw this?

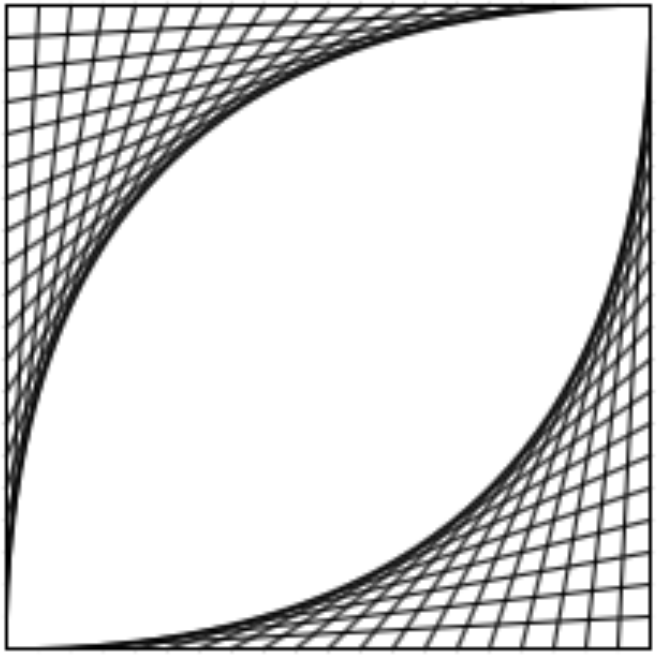
Extra Practice: Line Art

Write a graphical program **LineArt** that draws a series of lines (see lecture code for solution):

- Outer square is at (10, 30) and size 200x200
- each line is 10px apart in each dimension

coordinates of top-left lines:
- (210, 30) to (10, 30)
- (200, 30) to (10, 40)
- (190, 30) to (10, 50)
- ...
- (20, 30) to (10, 220)

coordinates of bottom-right lines:
- (210, 30) to (210, 230)
- (210, 40) to (200, 230)
- ...
- (210, 220) to (20, 230)

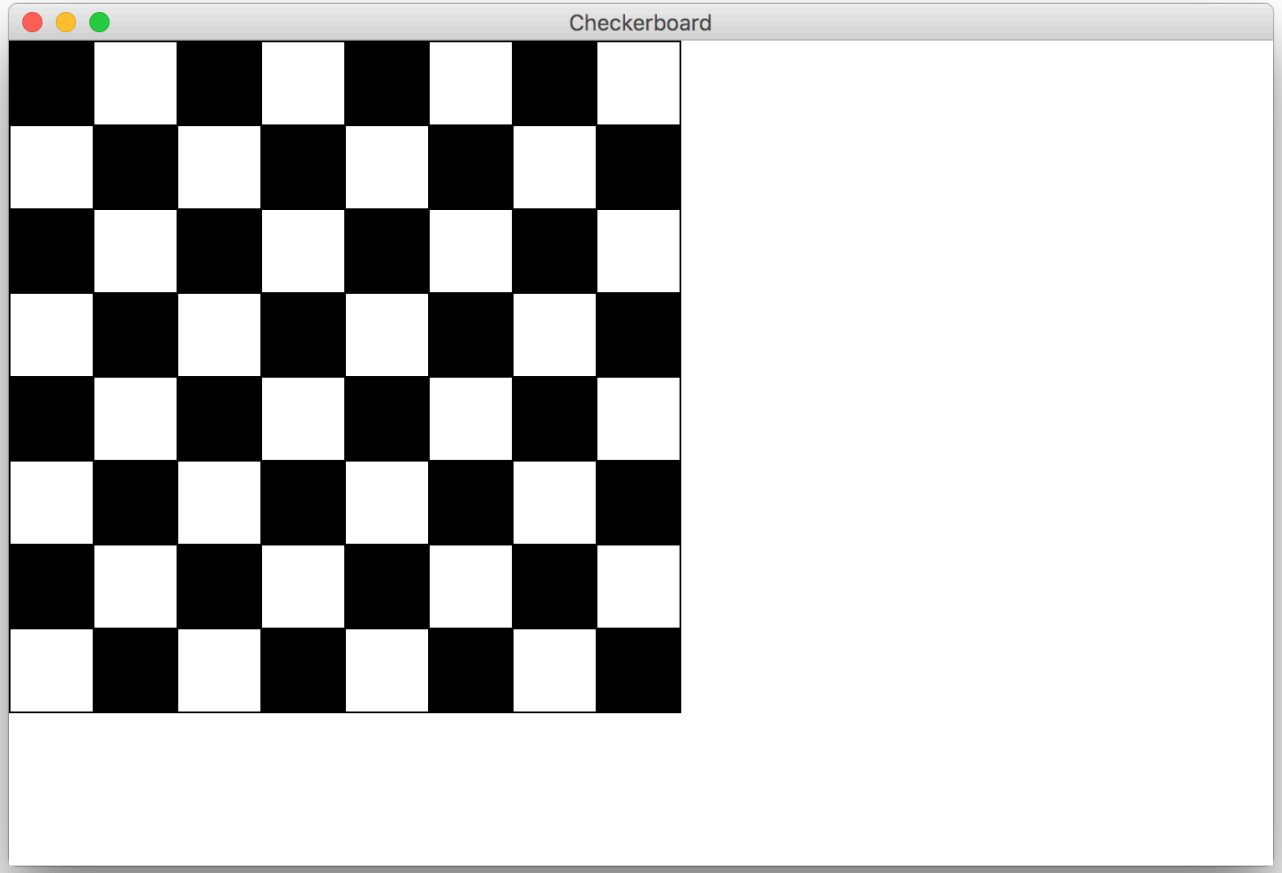


Plan For Today

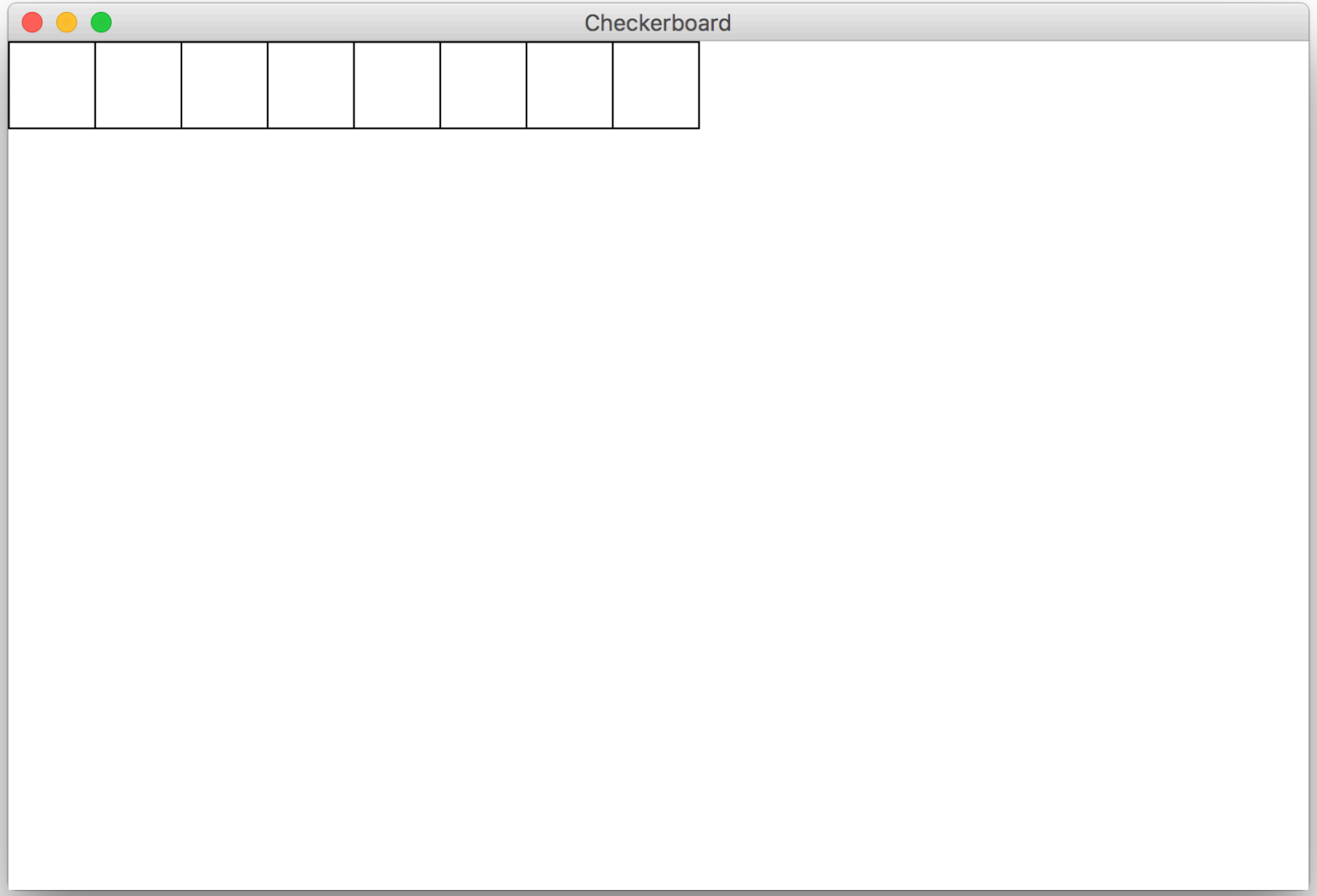
- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

Practice: Checkerboard

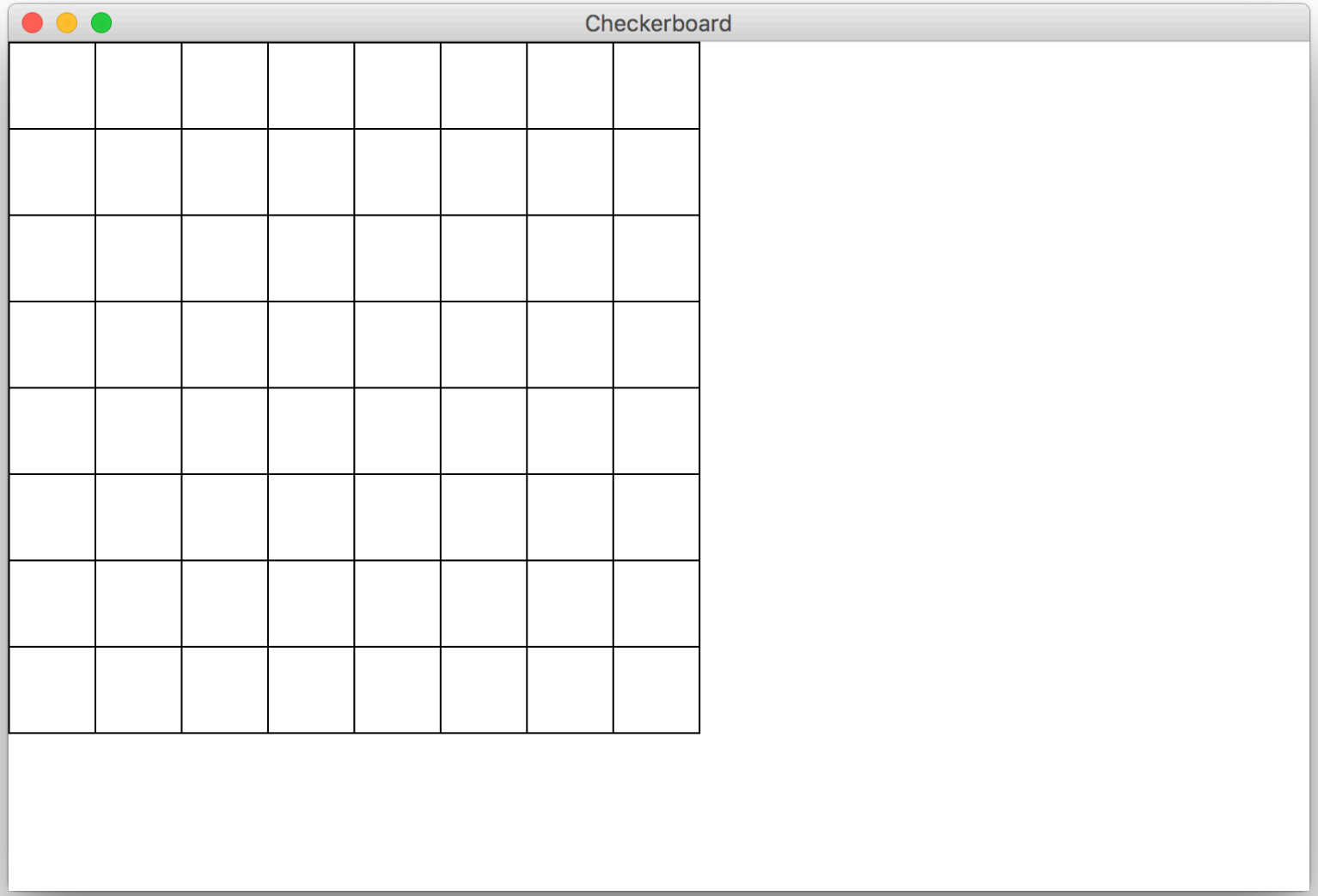
Write a graphical program named **`Checkerboard`** that draws a checkerboard pattern using GRects.



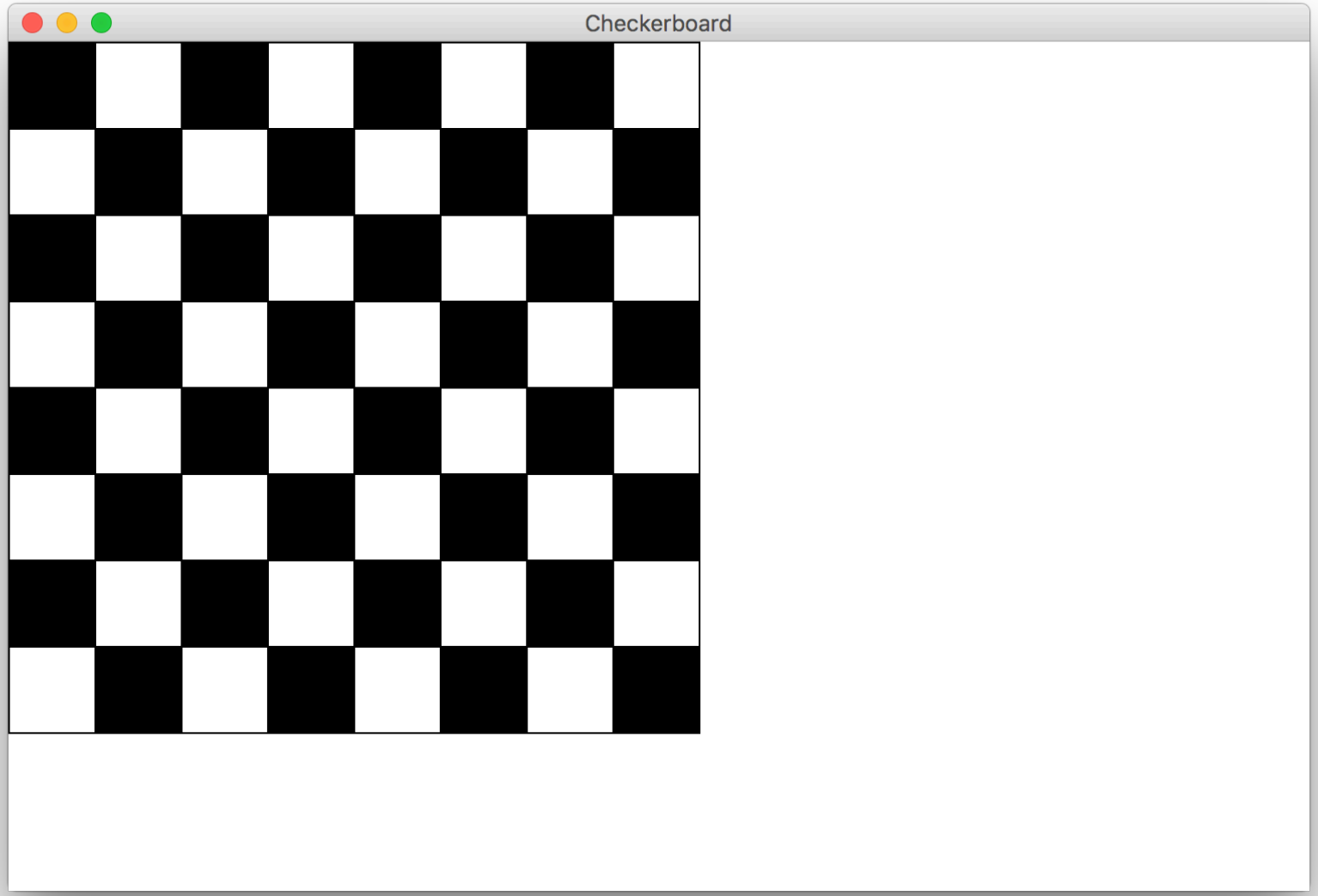
Milestone 1



Milestone 2

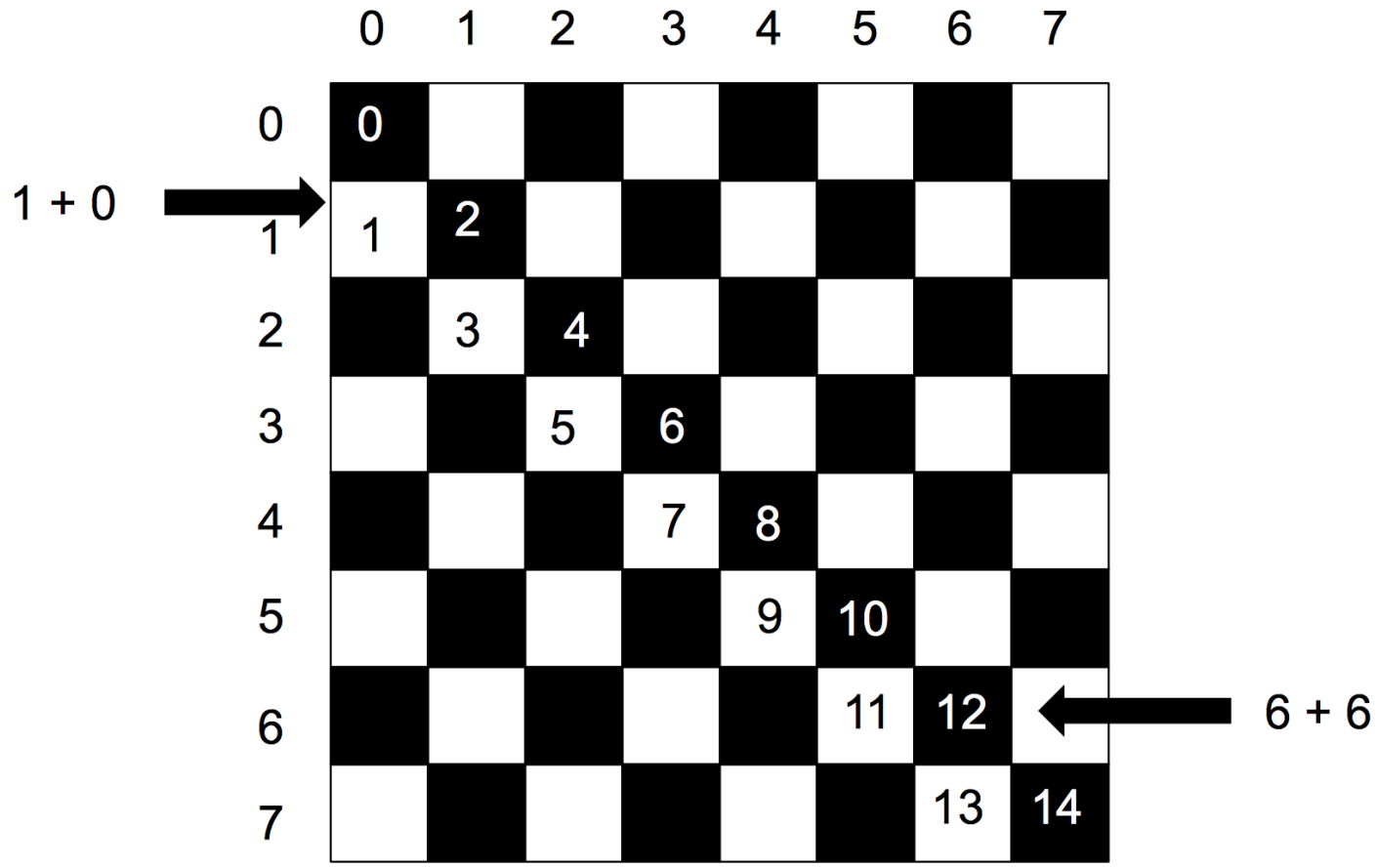


Milestone 3



Milestone 3

- Notice the pattern if we add the row and column indexes…



Recap

- Announcements
- Recap: Graphics
- GCompounds
- Getters
- Practice: Stoplights
- Practice: Checkerboard

Next time: Animation