Lecture 26:

THE GRAPHICS MODULE

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College

Announcements

- As mentioned on Slack, my Tuesday office hours this week are **rescheduled** to:

TODAY 2 - 3:30pm

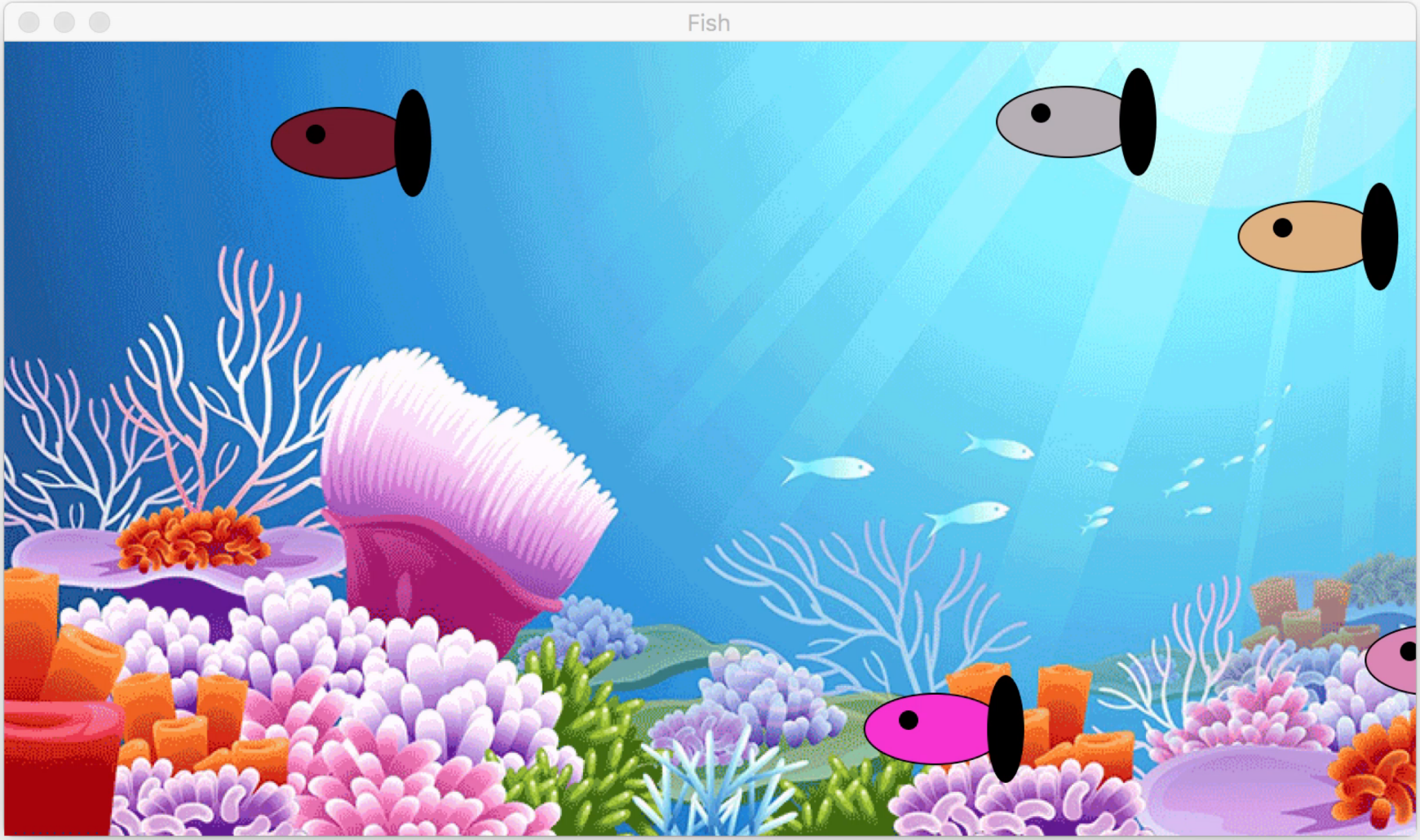
- I'll also be available on slack / Google Hangout during Tuesday hours

Outline

- Python packages (graphics)
  - A quick demo
  - Using python modules
  - Drawing pictures with **`graphics`**
  - About the final project
- Animation
- Lab: Fish Tank
- Interaction

Some of you may know…

My in-office substitute



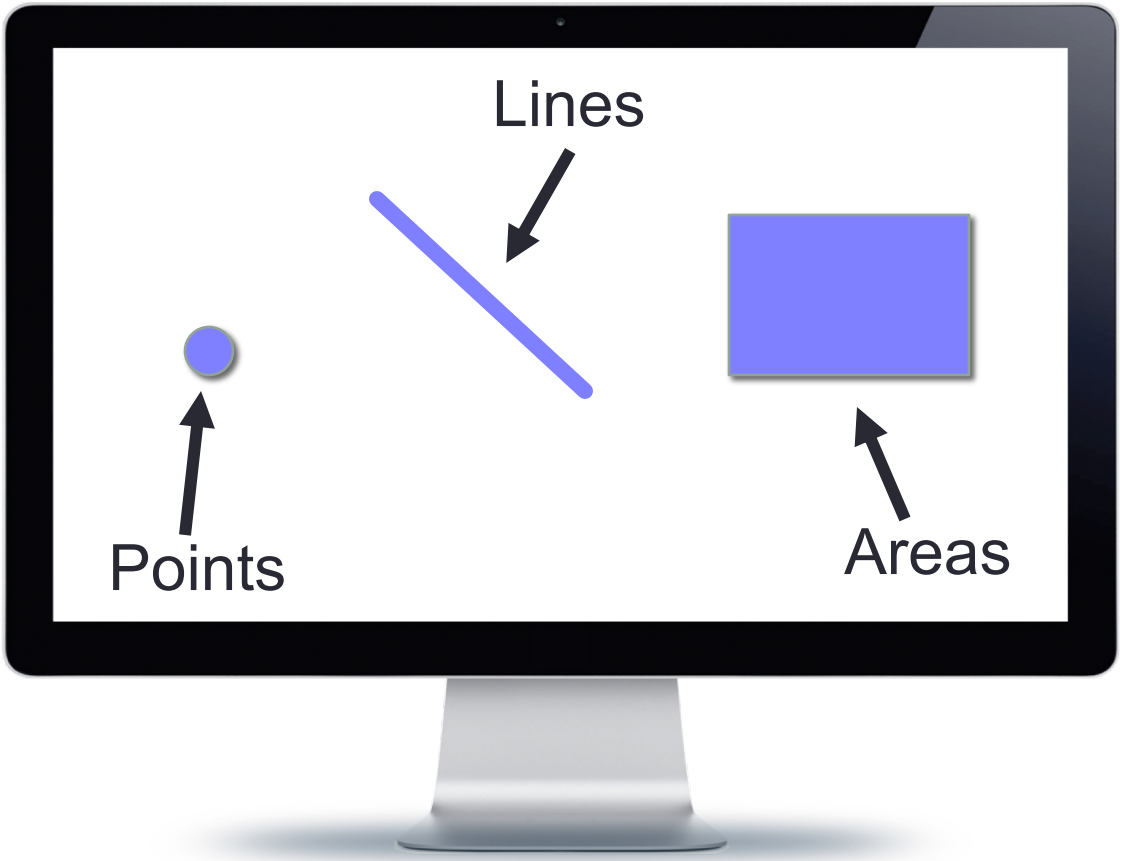
Discussion

How do you think they **built** that?

What **components** did they need?

1. Draw stuff

The images we draw are composed of marks: like ink



…this is what today will be about

2. Make it move

…more about this Wednesday

3. Get input from the user and react

…more about this Friday

Hmm…

If these are the basic components of **every game**,
it's probably the case that **someone else**
has had to **build them before**…

The **`graphics`** module*

- Two kinds of objects:
  - stuff you draw (**`Graphics`** objects)
  - stuff you draw on (**`GraphWin`** objects)

- Basic formula for drawing graphics:
  - open a graphic window (a **`GraphWin`**)
  - construct some **`Point`**, **`Line`**, **`Circle`**, **`Oval`**, **`Rectangle`**, **`Polygon`**, and **`Text`** objects
  - draw them to the window
  - close the window when you're done
  - terminate the program

- written by John Zelle to go along with his book "Python Programming: An Introduction to Computer Science" (Franklin, Beedle & Associates)
  Available from: http://mcsp.wartburg.edu/zelle/python/

Our first **graphics** program

Untitled

```
from graphics import *

def main():
    win = GraphWin("CSC111 - Graphcs Demo", 600, 400)
    c = Circle(Point(50,50), 10)
    c.draw(win)
    win.getMouse()
    win.close()

if __name__ == "__main__":
    main()
```

Ln: 9 Col: 0

Our first **graphics** program

Untitled

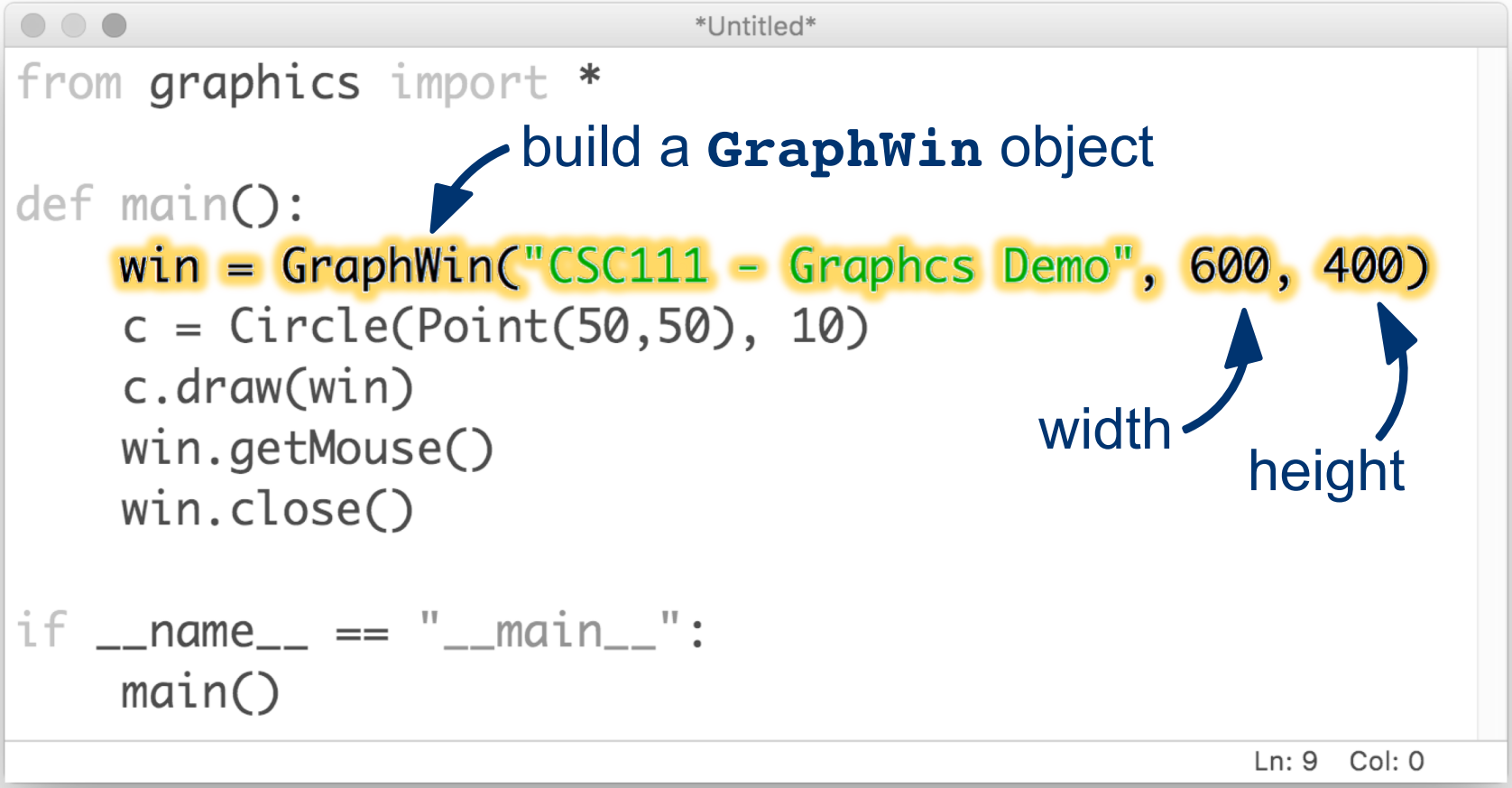
```
from graphics import *

def main():
    win = GraphWin("CSC111 - Graphcs Demo", 600, 400)
    c = Circle(Point(50,50), 10)
    c.draw(win)
    win.getMouse()
    win.close()

if __name__ == "__main__":
    main()
```

Ln: 9 Col: 0

import the module

(this method means we don't have to type "**graphics.**" in front of every method)

Our first **`graphics`** program



Our first **`graphics`** program

Untitled

```
from graphics import *

def main():
    win = GraphWin("CSC111 - Graphcs Demo", 600, 400)
    c = Circle(Point(50,50), 10)
    c.draw(win)
    win.getMouse()
    win.close()

if __name__ == "__main__":
    main()
```

construct a **`Circle`** object
(centered at (50,50) with a radius of 10)

Ln: 9 Col: 0

Our first **`graphics`** program

Untitled

```
from graphics import *

def main():
    win = GraphWin("CSC111 - Graphcs Demo", 600, 400)
    c = Circle(Point(50,50), 10)
    c.draw(win)
    win.getMouse()
    win.close()

if __name__ == "__main__":
    main()
```

draw the **`Circle`** to the **`GraphWin`**

Ln: 9 Col: 0

Our first **`graphics`** program

Untitled

```
from graphics import *

def main():
    win = GraphWin("CSC111 - Graphcs Demo", 600, 400)
    c = Circle(Point(50,50), 10)
    c.draw(win)
    win.getMouse()
    win.close()

if __name__ == "__main__":
    main()
```

wait for the user to click
(so we can actually look at what we drew)

Ln: 9 Col: 0

Our first **graphics** program

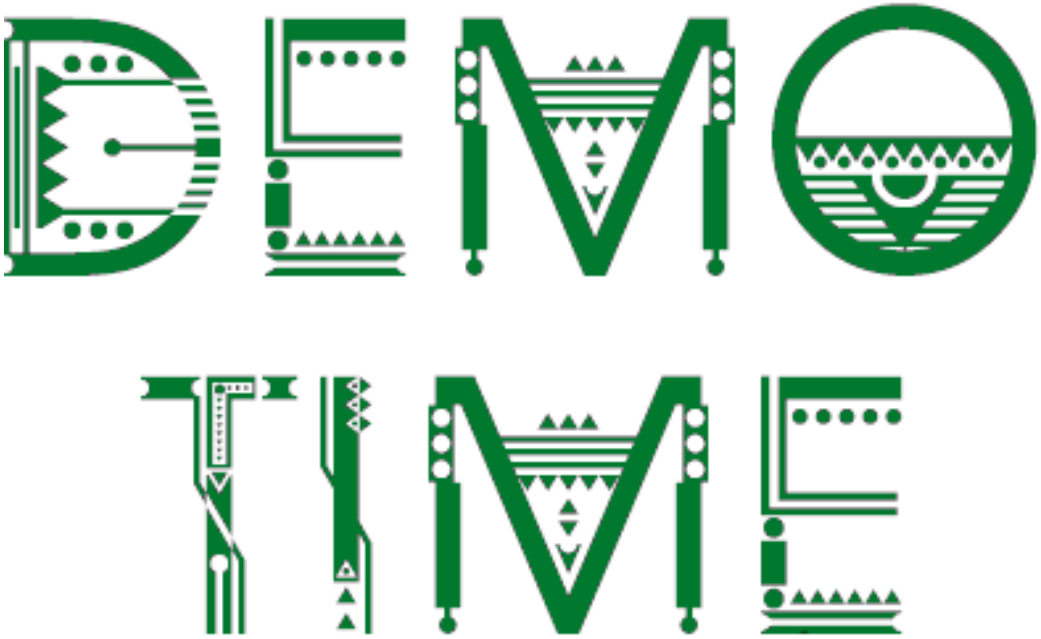
Untitled

```
from graphics import *

def main():
    win = GraphWin("CSC111 - Graphcs Demo", 600, 400)
    c = Circle(Point(50,50), 10)
    c.draw(win)
    win.getMouse()
    win.close()

if __name__ == "__main__":
    main()
```

close the **GraphWin**

Ln: 9 Col: 0

Our first **`graphics`** program



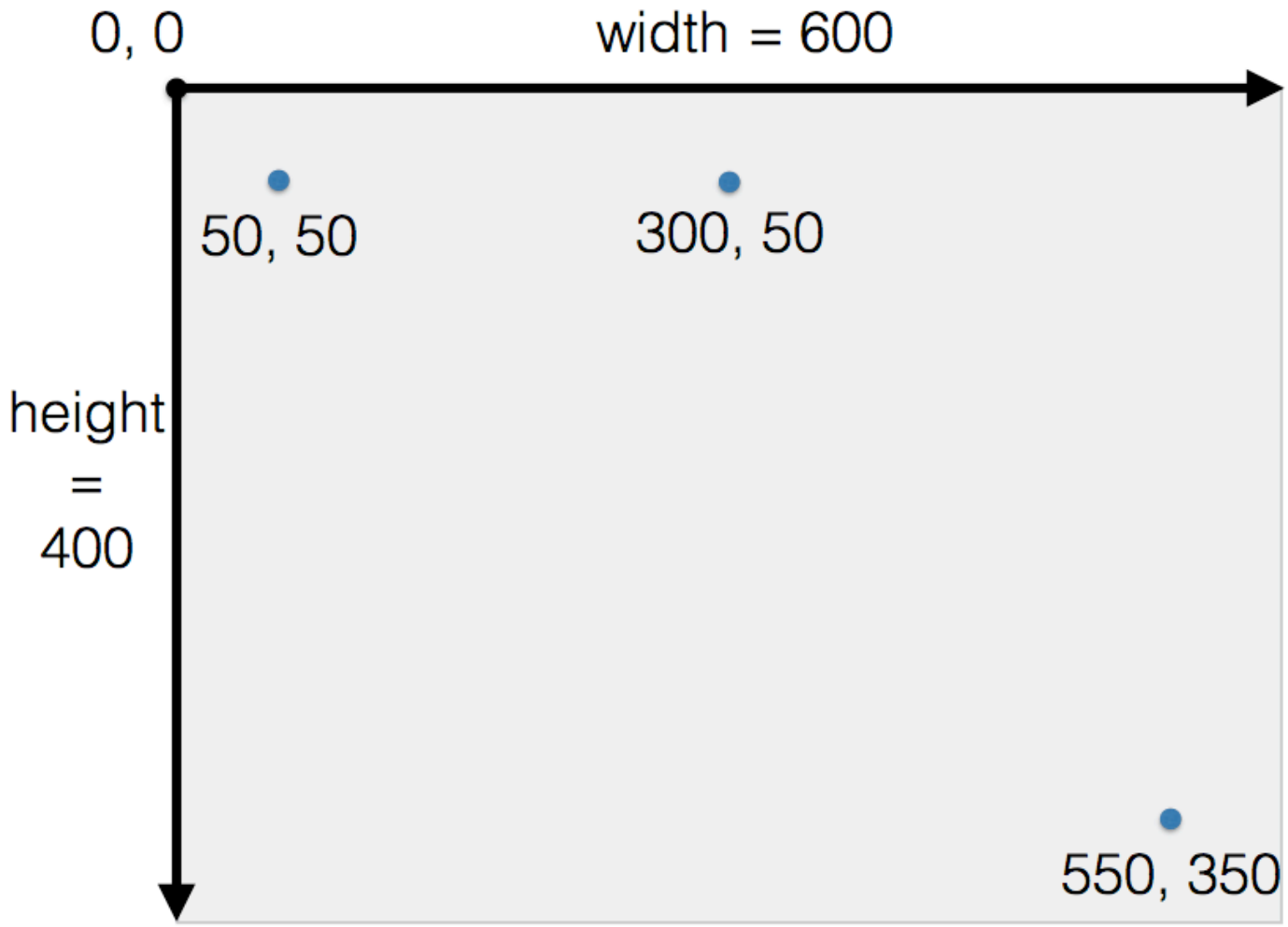
First "graphical primitives"

Points

- Used to anchor other objects (circles or rectangles)
- Defined by **x** and **y** coordinates

```
# create a point at location (50, 50)
p1 = Point(50,50)

# create a point at location (300, 50)
p2 = Point(300,50)
```

First “graphical primitives”



First "graphical primitives"

Circles

- Defined by a **center** and a **radius**
- The center is a **Point**

```
# create a circle centered at (50, 50)
# with radius 70
c1 = Circle( Point(50,50), 70 )
c1.draw( win )
```

First “graphical primitives”

Rectangles

- Defined by a **top-left,** and a **bottom-right point**

```
# create a rectangle with top-left corner
# at (5,5) and bottom-right at (50,50)

r3 = Rectangle( Point(5,5), Point( 50, 50) )
r3.draw( win )
```

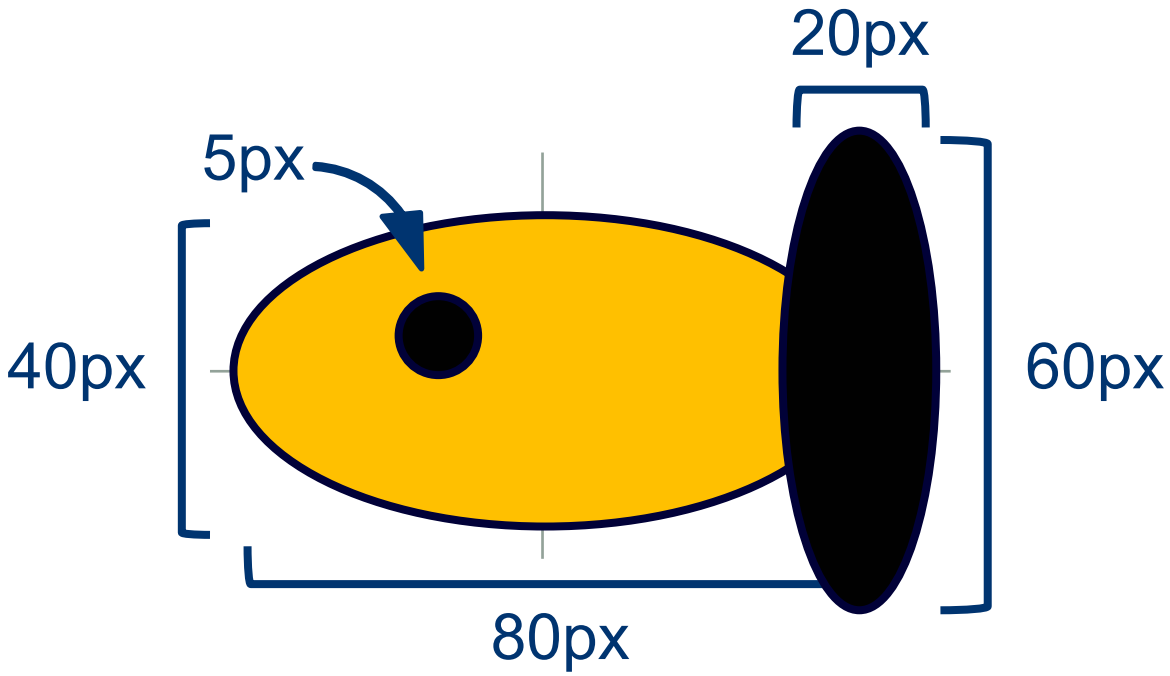
Filling an object with color

```
# create a rectangle with top-left corner
# at (5,5) and bottom-right at (50,50)

r3 = Rectangle( Point(5,5), Point( 50, 50) )
r3.setFill( "red" )
r3.draw( win )
```

What if we want a more specific color?

```
# create a rectangle with top-left corner
# at (5,5) and bottom-right at (50,50)

r3 = Rectangle( Point(5,5), Point( 50, 50) )
color = color_rgb( 200, 100, 150 )
r3.setFill( color )
r3.draw( win )
```

Okay, let's make a fish!

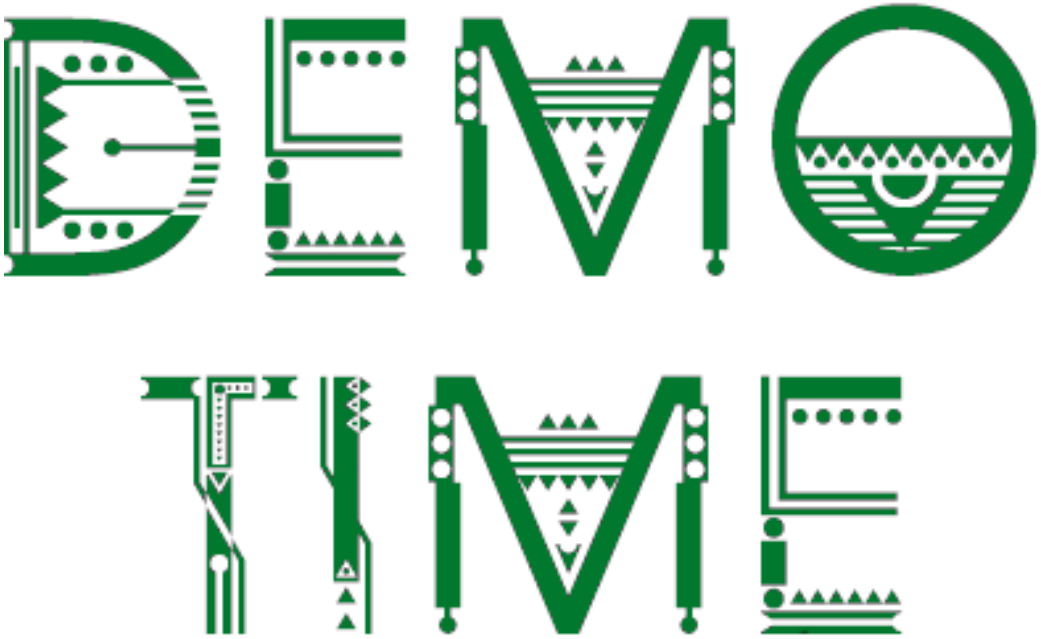


Discussion

How can we get all of these parts

to **work together**?

Making a fish



Outline

- ✓Python packages (graphics)
  - ✓A quick demo
  - ✓Using python modules
  - ✓Drawing pictures with `graphics`
  - – About the final project
- Animation
- Lab: Fish Tank
- Interaction

About the final project

- **Four weeks of class left!** (time to talk about the final)
- **Goal of the project**: apply the techniques we've learned in this class to something **you care about**
- **Ideas**:
  - an computer-generated animation
  - a custom game
  - a tool to help plan your path through the major
  - a poem generator (expanding Lab 5)
  - a choose-your-own-fractal program (expanding Lab 7)
  - a graphing calculator program (expanding Assignment 2)
  - anything else you can think of!

Final project deliverables

- Sunday Nov 18th: **Final Project Proposal**
- Sunday December 2nd: **Prototype I**
- Monday December 10th: **Prototype II**
- Wednesday December 19th: **Final Write-Up**

FP1: **final project proposal**

1. Names of **people** working on this project
2. What's the **big idea** behind this project?
3. What are the (major) **building blocks** the project will need to be successful?
4. Which do **you know how to build already**, and which ones **do you still need to figure out**?
5. Are there any **potential roadblocks**?

Coming up next

- ✓Python packages (graphics)
  - ✓A quick demo
  - ✓Using python modules
  - ✓Drawing pictures with **`graphics`**
  - ✓About the final project
- Animation
- Lab: Fish Tank
- Interaction