

Lecture 27:

THE graphics MODULE

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College

Announcements

- A reminder: Tuesday is Otelia Cromwell Day!
- Tuesday afternoon study groups have been rescheduled (check your group discord channel for details)
- Jordan's office hours this week are rescheduled to:

Wednesday 2 - 3:30pm

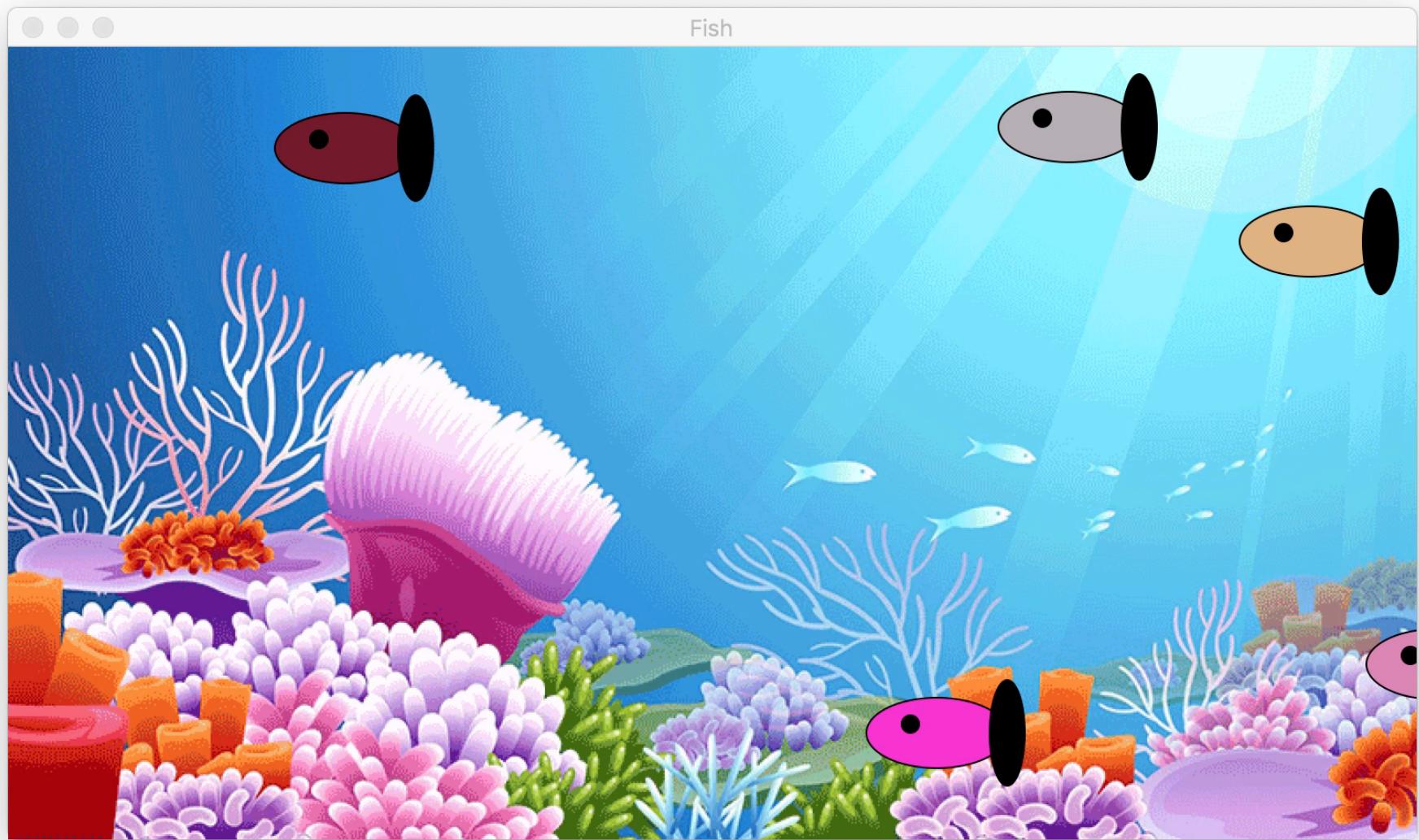
Outline

- Monday: Python packages (graphics)
 - A quick demo
 - Using python modules
 - Drawing pictures with **graphics**
 - Questions about the Midterm (time permitting)
- Wednesday: Life Skills #5/6: Code Diagrams & Prototyping
- Friday: Animation
- Monday: Interaction

Some of you may know...



My in-office substitute



Discussion

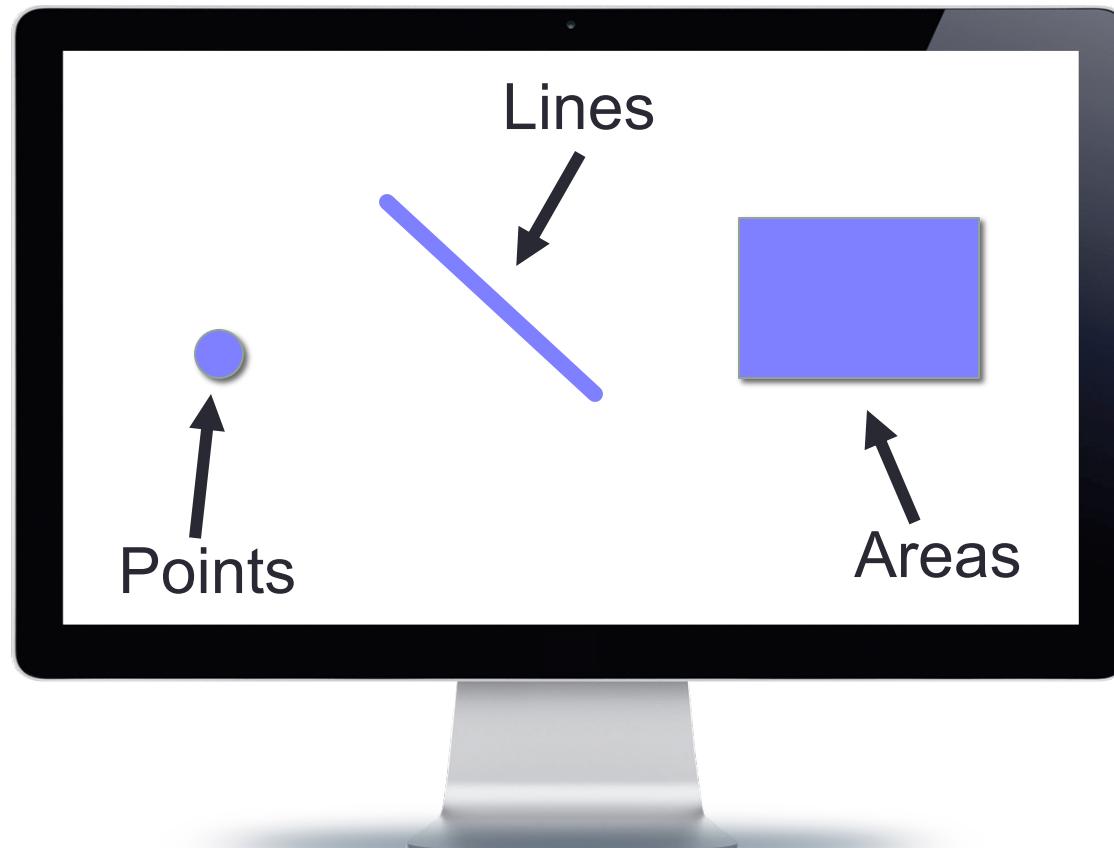
How do you think they **built** that?

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 Friday

3. Get input from the user and react

...more about this Monday

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

```
1 from graphics import *
2
3 def main():
4     win = GraphWin("CSC111 – Graphics Demo", 800, 600)
5     c = Circle(Point(50,50), 10)
6     c.draw(win)
7     win.getMouse()
8     win.close()
9
10 if __name__ == "__main__":
11     main()
```

Our first **graphics** program

```
1 from graphics import *           import the module
2
3 def main():
4     win = GraphWin("CSC111 – Graphics Demo", 800, 600)
5     c = Circle(Point(50,50), 10)
6     c.draw(win)
7     win.getMouse()
8     win.close()
9
10 if __name__ == "__main__":
11     main()
```

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

Our first **graphics** program

```
1 from graphics import *
2
3 def main():
4     win = GraphWin("CSC111 - Graphics Demo", 800, 600)
5     c = Circle(Point(50,50), 10)
6     c.draw(win)
7     win.getMouse()
8     win.close()
9
10 if __name__ == "__main__":
11     main()
```

build a **GraphWin** object

width height

Our first **graphics** program

```
1 from graphics import *
2
3 def main():
4     win = GraphWin("CSC111 – Graphics Demo", 800, 600)
5     c = Circle(Point(50,50), 10)
6     c.draw(win)
7     win.getMouse()
8     win.close()
9
10 if __name__ == "__main__":
11     main()
```



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

Our first **graphics** program

```
1 from graphics import *
2
3 def main():
4     win = GraphWin("CSC111 – Graphics Demo", 800, 600)
5     c = Circle(Point(50,50), 10)
6     c.draw(win) ← draw the Circle to the GraphWin
7     win.getMouse()
8     win.close()
9
10 if __name__ == "__main__":
11     main()
```

Our first **graphics** program

```
1 from graphics import *
2
3 def main():
4     win = GraphWin("CSC111 – Graphics Demo", 800, 600)
5     c = Circle(Point(50,50), 10)
6     c.draw(win)
7     win.getMouse()
8     win.close()    ← wait for the user to click
9
10 if __name__ == "__main__":
11     main()
```

(so we can actually look at what we drew)

Our first **graphics** program

```
1 from graphics import *
2
3 def main():
4     win = GraphWin("CSC111 – Graphics Demo", 800, 600)
5     c = Circle(Point(50,50), 10)
6     c.draw(win)
7     win.getMouse()
8     win.close() ← close the GraphWin
9
10 if __name__ == "__main__":
11     main()
```

Our first **graphics** program

DEMO

TIME

<https://repl.it/@JordanCrouser/graphics-demo>

First “graphical primitives”

Points

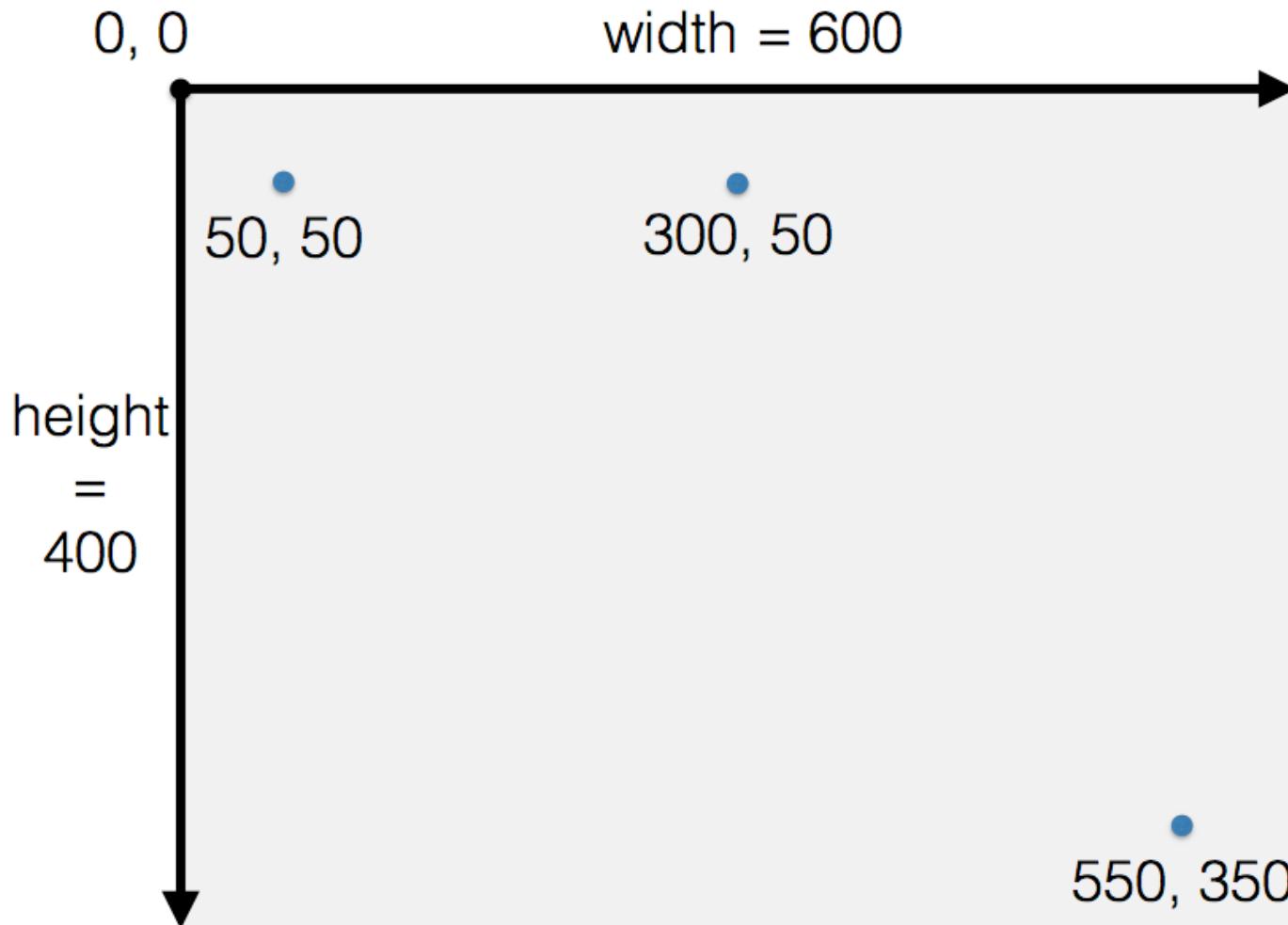
- Used to anchor other objects (circles or rectangles)
- Defined by **x** and **y** coordinates

```
# 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 )
```

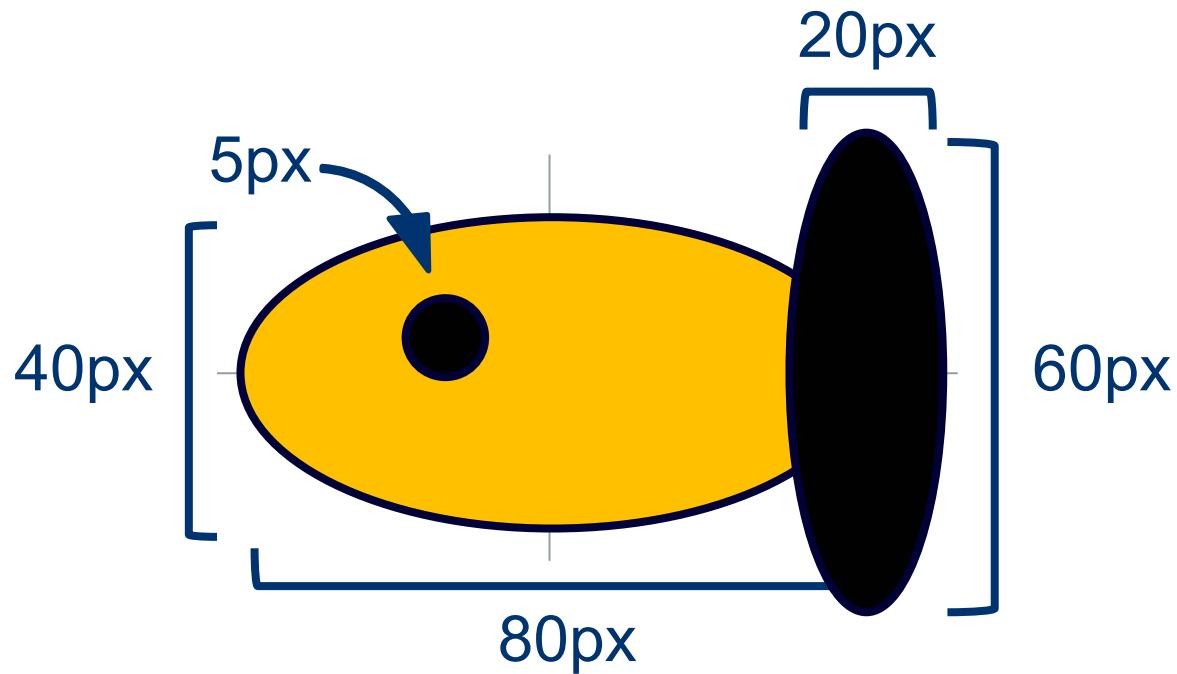


What if we want a more specific color?

```
# 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!

DEMO

TIME

Fork: <https://repl.it/@JordanCrouser/fish-demo>

Discussion

So what did your team
come up with?



AMA: midterm edition

Any lingering questions about
the Midterm?



Outline

- ✓ Monday: Python packages (graphics)
 - ✓ A quick demo
 - ✓ Using python modules
 - ✓ Drawing pictures with **graphics**
 - ✓ Questions about the Midterm (time permitting)
- Wednesday: Life Skills #5/6: Code Diagrams & Prototyping
 - Friday: Animation
 - Monday: Interaction