

CS151 Fall 2012 Lecture 16

Stephanie R Taylor

Oct 9, 2012

1 Administrative Topics

- Any questions about project 5?
- Remember to come to lab today for help with project 5.

2 Objects vs Modules

Today we are going to learn about more of the objects available in the graphics package written by John Zelle. But first I want to reiterate the difference between modules and objects.

- Module
 - What it is: a collection of functions
 - How you get one: `import`
 - How many do you have: Each file will have only one copy of each module
- Item
 - What it is: a collection of data and methods that operate on that data

- How you get one: create one using a function. For an object that will be a Circle, you create it using the function called `Circle`
- How many do you have: Each code file can create as many objects as it needs

3 Graphics Objects

Object Type	Description
Window	an object that allows you to put graphics into a window
Point	an object that allows you to specify the x-y coordinates of a graphics object
Shape Objects Type	
Circle	a circular graphics object – lets you draw a circle
Rectangle	a rectangular graphics object – lets you draw a rectangle
Oval	an elliptic graphics object – lets you draw an oval
Line	a line graphics object
Polygon	a graphics object that is constructed by providing all the vertices

The Graphwin methods we are interested in are:

- `win.getMouse()` – waits for a mouse click
- `win.close()` – closes the window
- `win.checkMouse()` – True if a mouse has recently clicked on the window

Zelle has organized the methods for the shape functions well. There are some methods (and functions) specific to each shape. Most notably, the functions used to create each object are unique:

- `Circle(centerpt, radius)`. `centerpt` is a Point specifying the location of the center of the circle. `radius` is a number.
- `Rectangle(pt1, pt2)`. each `pt` is a Point specifying opposite corners of the rectangle (e.g. upper left and lower right)

- `Oval(pt1, pt2)`. Each pt is a Point specify oppoosite corners of the rectangle that bounds the oval you want to draw
- `Line(pt1, pt2)`. Each pt is an endpoint of the line segment.
- `Polygon(pt1, pt2, pt3, ...)` or `Polygon([pt1, pt2, pt3, ...])`. Each pt is a vertex of a polygon.

There are several methods that are general - -they can be used on any of the shape objects. Suppose that `obj` is a Circle, Rectangle, Oval, Line, or Polygon. Then, the follow methods are defined:

- `obj.draw(win)` – places `obj` in the window (`win`) (which must be a `GraphWin`)
- `obj.move(dx, dy)` – moves the `obj` by a given amount
- `obj.setOutline(clr)` – sets the outline color
- `obj.setWidth(width)` – sets the width of the outline (in pixels)
- `obj.setFill(clr)` – sets the fill color

For the color-setting methods, `clr` must be either a string with a color name (e.g. 'green', 'blue', etc.) or the output of the function `color_rgb` (e.g. `graphics.color_rgb(255, 150, 20)`).

To draw a simple circle, you use code like this:

```
# Make a window big enough to hold the scene
w = 300
h = 300
win = graphics.GraphWin("Crazy Toy", w, h)

# Make a circle centered at 150,150 with radius 20
center = graphics.Point(150, 150)
c = graphics.Circle(center, 20)

# Draw the circle (i.e. put it in the window)
c.draw(win)
```

```
# Keep the window open for awhile  
win.getMouse() # Pause to view result  
win.close()    # Close window when done
```

The color/width-setting methods that allow us to do make more interesting scenes.

Below is a picture of a very simplified view of a Fisher Price Person.

```
# Make a window big enough to hold the scene
w = 300
h = 300
win = graphics.GraphWin("Crazy Toy", w, h)

# Make a person
upper_left = graphics.Point(w/2-20, 150)
lower_right = graphics.Point(w/2+20,150+60)
body = graphics.Rectangle(upper_left, lower_right)
body.setFill('blue')
body.setOutline('blue')
center = graphics.Point(w/2, 150-20)
head = graphics.Circle(center, 20)
head.setOutline(graphics.color_rgb(0,0,0))
head.setFill(graphics.color_rgb(255,0,255))

# Draw the objects.
body.draw(win)
head.draw(win)

# Keep the window open for awhile
win.getMouse() # Pause to view result
win.close()    # Close window when done
```

Let's think about the symbol table contents after this code has run. Suppose we have executed all the lines except the last line. What does the main symbol table look like?

There are 6 graphics objects (1 GraphWin, 3 Points, 1 Rectangle, and 1 Circle).

Name	Value	
w	300 <int>	
h	300 <int>	
win	→	<GraphWin>
upper_left	→	<Point>
lower_right	→	<Point>
body	→	<Rectangle>
center	→	<Point>
head	→	<Circle>