

Zelle Graphics Module

Oliver W. Layton

CS151: Computational Thinking: Visual Media

Lecture 16, Fall 2020

Wednesday September 30

Switch from turtle to Zelle
Graphics Module for next 3 weeks

Analogues: GraphWin instead of Screen

We have a canvas that we draw shapes on, just like `turtle`.

- Object name for the screen/canvas is `GraphWin` (vs. `Screen` in `turtle`).
- Creating a `GraphWin` object:

```
screen = graphics.GraphWin('Title of window', width=400,  
                             height=300)
```

- To pause until a click:

```
screen.getMouse()
```

 instead of

```
screen.exitonclick()
```

- Actually closing the window requires a second command: `screen.close()`

Shape objects in Zelle's Graphics Module

Object type	Parameters	Description
Point	x <int> y <int>	Creates an object that represents a position (in pixels) within the GraphWin window.
Line	pt1 <Point>, pt2 <Point>	An object that represents a line between pt1 and pt2.
Rectangle	pt1 <Point>, pt2 <Point>	Creates a rectangle object. The parameters pt1 and pt2 represent Point objects at positions at opposite sides of the rectangle (e.g. top left, lower right)
Circle	center <Point>, radius <float>	An object that represents a circle. To create it, you give as arguments a Point object for the center position, and a radius as a float
Oval	pt1 <Point>, pt2 <Point>	Each pt is a Point that lies on opposite corners of the rectangle that bounds the oval you want to draw
Polygon	pt1 <Point>, pt2 <Point>, pt3 <Point>...	Creates a shape that connects lines through the points given by pt1, pt2, and so on.

Examples of using Zelle Graphics Objects

Rectangle object 15 pixels wide, 10 pixels high:

```
rect = graphics.Rectangle(graphics.Point(0, 0), graphics.Point(15, 10))
```

Circle object centered at $(x, y) = (50, 50)$. Radius 5:

```
circle = graphics.Circle(graphics.Point(50, 50), 5)
```

- Zelle has a different coordinate system compared to turtle:
(0,0) is the TOP-LEFT part of the canvas! No negative values.
Let's draw it out.

Shape object methods

Shape object method	Description
obj.draw(win)	Actually draws the object you create in the window (NOTE: Creating an object does not mean it shows up in the window! You need to run its draw function!)
obj.move (dx, dy)	Move the object in the window horizontally by dx, vertically by dy
obj.undraw()	Removes a drawn shape from the canvas
obj.setOutline(color)	Sets the outline edge of the shape to color, e.g. a named string like 'red' or in graphics.color_rgb(r,g,b) format
obj.setWidth(width)	Sets the width of the outline edge, in pixels
obj.setFill(color)	Sets the interior surface color, e.g. a named string like 'red' or in graphics.color_rgb(r,g,b) format

Common mistake

Creating a shape object does **NOT** automatically make it show up on the canvas! You have to call the draw method on the object to have it show up.

```
circle = graphics.Circle(graphics.Point(50, 50), 5)  
circle.draw(screen)
```

assuming we call the GraphWin object screen.

Overview of today's lab

1. Make a collage of 3 Zelle shapes of your choice. Customize the appearance of each.
2. Learn about moving Zelle shapes. Move shapes in you collage around randomly, changing their colors.
3. Make a basic animation of a snowman. Project follows same process for creating an animation.

Simple example program

Quick demo of break keyword

Quick review: Two ways to loop thru
a list