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

• To pause until a click:
  
  ```python
  screen.getMouse() instead of screen.exitonclick()
  ```

• Actually closing the window requires a second command: screen.close()
## Shape objects in Zelle's Graphics Module

<table>
<thead>
<tr>
<th>Object type</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>x &lt;int&gt; y &lt;int&gt;</td>
<td>Creates an object that represents a position (in pixels) within the GraphWin window.</td>
</tr>
<tr>
<td>Line</td>
<td>pt1 &lt;Point&gt;, pt2 &lt;Point&gt;</td>
<td>An object that represents a line between pt1 and pt2.</td>
</tr>
<tr>
<td>Rectangle</td>
<td>pt1 &lt;Point&gt;, pt2 &lt;Point&gt;</td>
<td>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)</td>
</tr>
<tr>
<td>Circle</td>
<td>center &lt;Point&gt;, radius &lt;float&gt;</td>
<td>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</td>
</tr>
<tr>
<td>Oval</td>
<td>pt1 &lt;Point&gt;, pt2 &lt;Point&gt;</td>
<td>Each pt is a Point that lies on opposite corners of the rectangle that bounds the oval you want to draw</td>
</tr>
<tr>
<td>Polygon</td>
<td>pt1 &lt;Point&gt;, pt2 &lt;Point&gt;, pt3 &lt;Point&gt;…</td>
<td>Creates a shape that connects lines through the points given by pt1, pt2, and so on.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Shape object method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj.draw(win)</td>
<td>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!)</td>
</tr>
<tr>
<td>obj.move(dx, dy)</td>
<td>Move the object in the window horizontally by dx, vertically by dy</td>
</tr>
<tr>
<td>obj.undraw()</td>
<td>Removes a drawn shape from the canvas</td>
</tr>
<tr>
<td>obj.setOutline(color)</td>
<td>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</td>
</tr>
<tr>
<td>obj.setWidth(width)</td>
<td>Sets the width of the outline edge, in pixels</td>
</tr>
<tr>
<td>obj.setFill(color)</td>
<td>Sets the interior surface color, e.g. a named string like 'red' or in graphics.color_rgb(r,g,b) format</td>
</tr>
</tbody>
</table>
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 your 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