Eric Roberts
CS 151

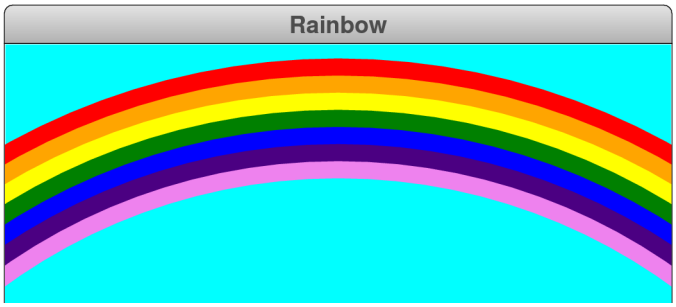
Handout #6
September 30, 2022

Problem Set #4

Due: 5:00 P.M., Friday, October 7

Problem 1 (Chapter 3, exercise 4, page 105)

Use the `GObject` hierarchy to draw a rainbow that looks something like this:



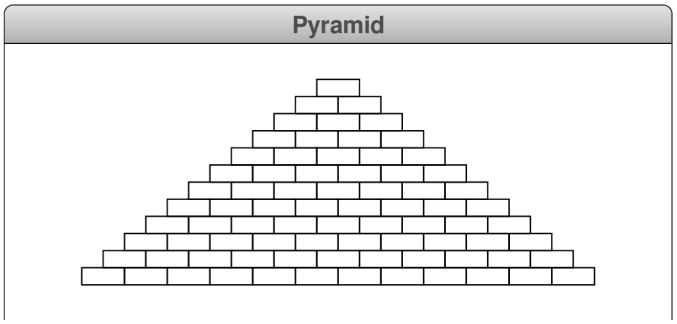
Starting at the top, the seven bands in the rainbow are red, orange, yellow, green, blue, indigo, and violet, respectively. And since Python defines it as a built-in color, it makes sense to use `"SkyBlue"` as the color of the sky. Remember that this chapter defines only the `GRect`, `GOval`, `GLine`, and `GLabel` classes and does not include a graphical object that represents an arc. It will help to think outside the box, in a more literal sense than usual.

Rather than specify the exact dimensions of each circle (and there are indeed circles here), play around with their sizes and positioning until you get something that matches your aesthetic sensibilities. The only things we'll be concerned about are:

- The top of the arc should not be off the screen.
- Each of the arcs in the rainbow should get clipped along the sides of the window, and not along the bottom.

Problem 2 (Chapter 3, exercise 7, page 107)

Write a program that displays a pyramid on the graphics window. The pyramid consists of bricks arranged in horizontal rows, arranged so that the number of bricks in each row decreases by one as you move upward, as shown in the following sample run:

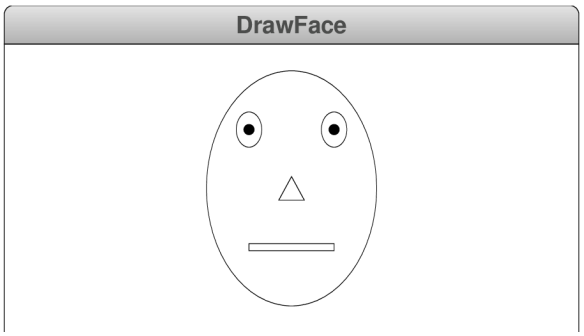


The pyramid should be centered in the window both horizontally and vertically. Your program should also use the following constants to make the program easier to change:

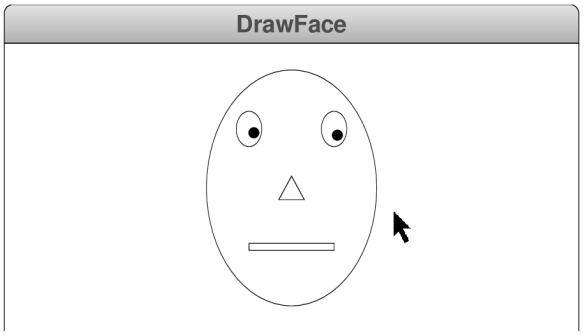
`BRICK_WIDTH`	The width of each brick
`BRICK_HEIGHT`	The height of each brick
`BRICKS_IN_BASE`	The number of bricks in the base

Problem 3 (Chapter 5, exercise 4, page 185)

Use the Portable Graphics Library to create a cartoon drawing of a face that looks like this (feel free to use `GLines` for the nose if you are confused by `GPolygon`):



Once you have this picture, add a callback function for the `"mousemove"` event so that the pupils in the eyes follow the cursor position. For example, if you move the cursor to the lower right side of the screen, the pupils should shift so that they appear to be looking at that point, as follows:



Although it doesn't matter much when the cursor is outside the face, it is important to compute the position of the pupil independently for each eye. If you move the mouse between the eyes, for example, the pupils should point in opposite directions so that the face appears cross-eyed.