Drawing Objects In Pygame

Everything in Python is an object, and in Pygame things are no different. The Pygame modules provide classes for many different objects. Three of the most commonly used objects are `Surface`s, `Color`s (note the US-style spelling) and `Rect`s. Information about each of these objects, including a list of attributes and methods, can be found on the official Pygame documentation page.

Python also provides many **drawing primitives**. These are simple functions that draw lines or shapes to a surface. All primitives must be given the name of the surface on which to draw, and a valid colour. Typically, colours are defined toward the beginning of a program so that they can be referenced by name, as in `BLACK = (0, 0, 0)`, or by purpose, as in `BORDER = (128, 0, 192)`. Other arguments are listed in the table below. Arguments surrounded by [] in the function calls are optional, and may be omitted.

Common Objects

Object	Instantiation
Surface	`Surface((width, height)[, flags=0, depth=0, masks=None])`
Color	`Color(r, g, b[, a=255])`
Rect	`Rect(left, top, width, height)` or `Rect((left, top), (width, height))`

Common Drawing Primitives (from `pygame.draw`)

Object	Instantiation
rectangle	`rect(surface, color, rect[, width=0])`
polygon	`polygon(surface, color, points[, width=0])`
circle	`circle(surface, color, center, radius[, width=0])`
ellipse	`ellipse(surface, color, rect[, width=0])`
arc	`arc(surface, color, rect, start_angle, stop_angle[, width=1])`
line	`line(surface, color, start_pos, end_pos[, width=1])`
lines	`lines(surface, color, closed, points[, width=1])`

Drawing Objects In Pygame

Answer the following questions.

1. Using the Pygame documentation, create UML class diagrams for `Surface`, `Color` and `Rect`, listing the attributes and methods in each class.

2. Why do you think Pygame provides two ways to initiate a `Rect` object?

Write programs that accomplish each task, using appropriate programming conventions.

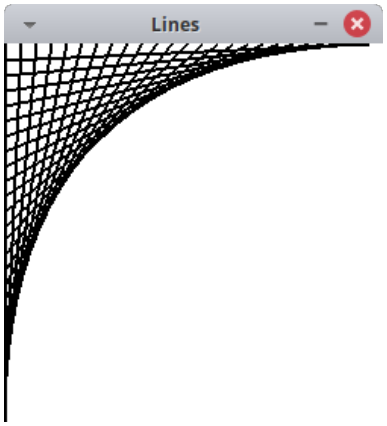
3. Create `Color` definitions for each of the following colours, then draw a rectangle for each colour to see how it looks.

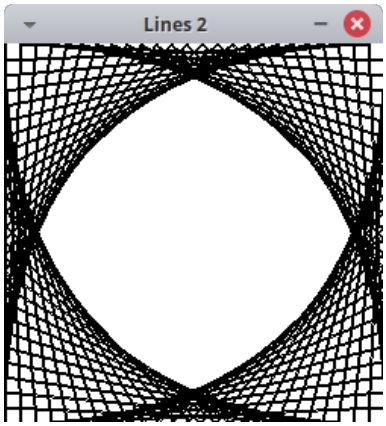
Colour	RGB	Colour	RGB	Colour	RGB
White	(255,255,255)	Yellow	(255,255,0)	Green	(0,128,0)
Black	(0,0,0)	Cyan	(0,255,255)	Purple	(128,0,128)
Red	(255,0,0)	Magenta	(255,0,255)	Teal	(0,128,128)
Lime	(0,255,0)	Maroon	(128,0,0)	Navy	(0,0,128)
Blue	(0,0,255)	Olive	(128,128,0)	Grey	(128,128,128)

4. Using Pygame's drawing primitives, draw a happy face with the following features: a yellow circular head, white elliptical eyes with blue irises and black pupils, a red triangular nose, and a black arc for a mouth. Define all colours at the beginning of your program.

5. Use the `get_width()` and `get_height()` methods in `Surface` to obtain the dimensions of the surface, then draw a circle with a diameter exactly half of the smaller dimension, centred on the surface. Test your program using different dimensions.

6. Use a loop to draw a sequence of straight lines similar to the diagram shown below left.





7. Modify your program to produce the diagram shown above right.