Lab 1

This lab is a warmup: it will not be collected. It involves using a simple graphics package to draw pictures under program control.

Piet Mondrian was a Dutch artist who "evolved a non-representational form which he termed Neo-Plasticism. This consisted of white ground, upon which was painted a grid of vertical and horizontal black lines and the three primary colours." [1] In other words most of his stuff looked like the image shown, though in fairness he made a bit of an effort to vary the size, number, position and colour of the blocks. What's more, this stuff sells for silly money! Do a Google search for "auction price Mondrian" to see what I mean.

Ever keen to exploit any unmet gap in the market, the management at *Studio Six Mille Cinq Cent Sept* has decided turn a buck by cranking out Mondrian-like images on an industrial scale using Python, going forward. That's where you come in. . .

This will involve the use of a simple graphics package that will allow us to draw simple shapes (just rectangles in our case) and control various characteristics such as position and colour and so on. Download the `graphics.py` package available from the Canvas page (also from Zelle's webpage `mcsp.wartburg.edu/zelle/python`) and place it in your directory for this assignment (alongside your `lab1.py` file). There is also a useful manual that you will need.

Refer to the sample program `simplepic.py` for an example of how to work with this package. It illustrates how a rectangle may be drawn by specifying the coordinates of its corners. It also allows us to specify its colour (encoded as a string). The reference manual mentioned above provides more detail.
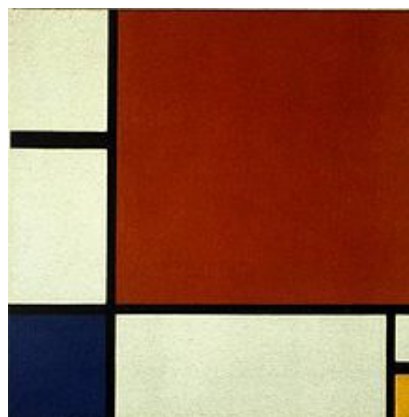

Figure 1: One of Piet's Pics.

To simplify matters we will assume that our "canvas" is 600×600 and will be divided into a regular three-by-three grid. Of these nine cells, three will be selected at random and each "painted" with a randomly-chosen colour. This boils down to painting three coloured rectangles on the "canvas".

The graphics package allows some colours to be specified by name ("red", "purple" and so on) but it also includes a function named `color_rgb (r, g, b)` to specify a broader range colours in terms of the strength of red, green and blue involved, which are represented by arguments r, g, b, that are encoded as numbers in the range 0 to 255. For example, `color_rgb (130, 0, 130)` is some shade of purple. We can generate random colours by using `color_rgb` with randomly chosen r, g and b values. Remember that you can tap into Python's random-number capabilites by importing its `random` package. See `https://docs.python.org/3/library/random.html` for a listing of these capabilites. The most relevant ones will be `randrange` or `randint`. Browse around to see what else the package

---

[1]http://en.wikipedia.org/wiki/Piet_Mondrian" Piet Mondrian

has to offer: some of these capabilities might proves useful at somke stage.

If you are feeling adventurous, feel free to enhance the basic pattern specified above. You could vary the "geometry" of the picture to chose possibilities other than 3×3 grid (e.g. 3×4, 4×3, 4×4) at random. You could alter the height and width of the individual rows and columns again at random. You could add grid lines to separate the rows and columns in some artistically pleasing shade of grey. And so on.