

EEN118 LAB SEVEN

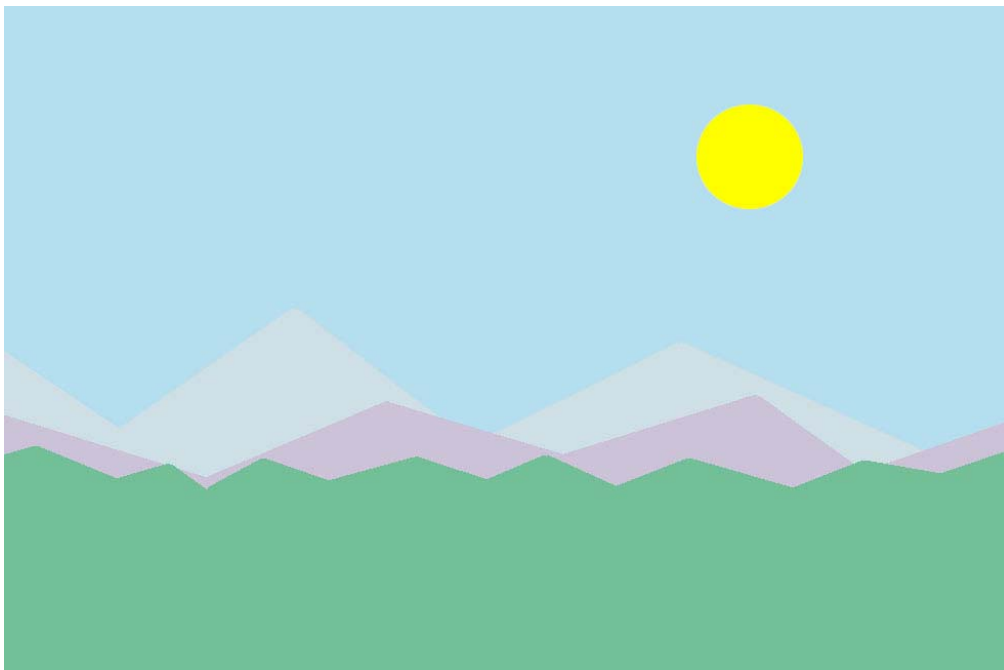
This week you are going to be creating backgrounds for cheap cartoon animations - random cityscapes and happy street scenes. More than ever, success here depends upon a step-by-step divide and conquer approach. Simple little functions that act together to produce complex results.

The details of the picture will be randomly produced. Run your programs a few times, the results should be significantly different each time.

1. *Background*

Make your program randomly choose a time of day for the scene (remember the `random_in_range` function), or let the user enter a time of day (just an hour, 0-23 would do). Shade the background in an appropriate sky colour and put a sun or a moon or something like that up there in a reasonable position for the chosen time of day.

Now, on top of that, draw some random hills and mountains. This will make the background for your street scene.



All I did was to draw a random zig-zag and colour everything below it in a faint shade of grey, the repeat the process a couple of times with different colours. It might not look very good, but this isn't an art class. It is only a background anyway.

2. *Office Buildings*

Think of a modern dull office building. From any distance it is just a bland rectangle with a few rows and columns of blank windows.



Think about how you could produce a randomly generated cityscape like that. A series of simple steps gradually creeping up on a complete solution should work out quite well:

a. *Draw a single window.*

Define a function that is given the height, width, x and y position, and colour of a window, and draws a filled rectangle exactly as described.

b. *Draw a row of windows to make a single floor.*

Define a function that is given the number of windows required, the x position of where to start, and the height, width, y position, and colour desired for each individual window. It should of course draw the row of windows representing a whole floor.

c. *Draw a whole block of windows.*

Define a function that is given the number of floors required, the number of windows on each floor, the x and y position of where to start, and the height, width, and colour desired for each individual window. It should then draw the whole array of windows in a regular grid.

d. *Draw a whole office building.*

Define a function that is given the x and y position of the bottom left corner of an office block, and its desired width.

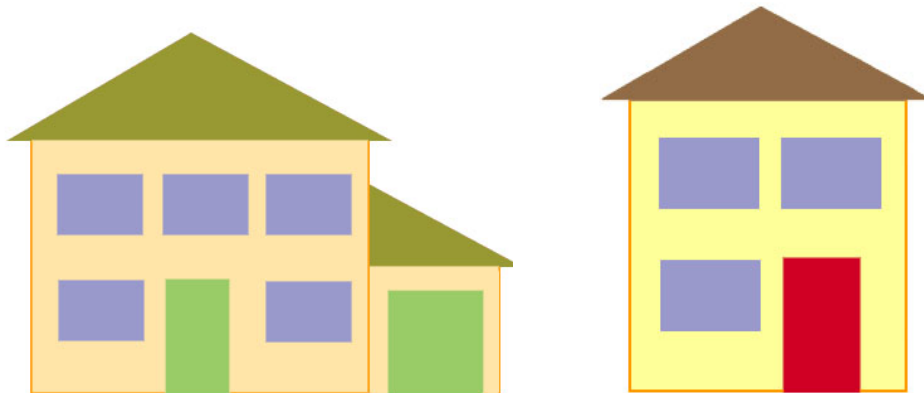
It should come up with random numbers in reasonable ranges to choose the height of the building, how many floors, how many offices per floor, the colour of the building itself, and the colour of its windows. You will need to experiment to come up with good ranges for these random numbers. I didn't do a very good job of selecting nice colours for the buildings.

e. *Draw a city scene.*

Define a function that is given the x and y position of the bottom left corner of a row of offices. It should draw one office block with a reasonably chosen random width, then if there is enough distance left before the right edge of the screen, it should continue to draw the rest of the row.

3. *Houses.*

A house is rather like a little office block. They tend to be restricted to one or two floors, and a modest number of windows on each floor, and they tend to have less garish colour schemes than offices, but a flat-roofed house should be easy to produce just by picking some of the random numbers differently.



Of course, we want something better than that. Once the position and size of a house is known, it is easy to draw a roof on top of it. But what about the front door? You can probably think of a good technique - when the bottom floor is being drawn, one of the windows can be randomly replaced with something slightly different to represent a door. And some people like to have garages for their cars.

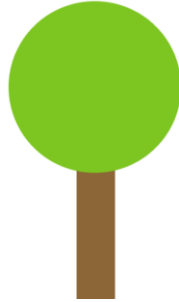
We want some architectural variety, so your program must randomly choose whether each house has a garage or not.

Make your city scene contain a random mix of houses and offices, or if you prefer, you can have a row of offices with a row of houses in front of it, just as suburbs and inner cities are often separate.

4. *Trees.*

You could imagine drawing a tree as a brown stalk with a big green circle on top. Of course, the height and width would be random numbers within a certain range, and you would probably produce a random shade of green.

The result would look terrible, but it would be a quick start. You could very easily write a function that draws a tree, and another function that draws a few trees randomly spread about your street scene. Try it.



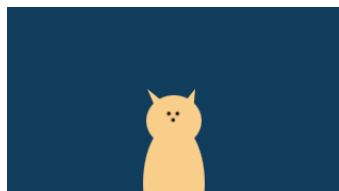
Now think how easily that could be improved. Instead of just one big stupid green circle, perhaps you could produce quite a lot of small green circles, and random positions clustered about the top of each tree.

With very little effort you can make something almost reasonable looking.



5. *Extra Credit: Make it look nice.*

As you've got a separate simple little function just for drawing one window, perhaps occasionally at random, it could put something in a window, like a cat or some curtains or whatever.



There is plenty of scope for extra credit in this lab, and the lab guys like to reward creativity. But remember that good tidy programming is essential. Pretty trees won't grow from ugly code.

When things are drawn quite small, they can be remarkably effective with very little detail. Put various things in your scene. Maybe some people in the street, some grass on the ground, an American flag on a pole, an aeroplane up in the sky...