Using the canvas (Tkinter part 2)

The canvas is probably the most interesting feature that Tkinter has. It allows you to draw shapes in an almost free-form manner. Although the basic canvas commands are rather simple, they will allow you to animate objects, which allow you to make fully functional games. Understanding how to manipulate the canvas is absolutely essential before you can move on to bigger and better projects.

Creating the canvas:

After creating the root, we must construct the canvas. The function used to create the canvas is, predictably, `Canvas(master, ... )`. Although `Canvas` has a large number of options, for now we will only concern ourselves with two, `width` (which gives the width of the canvas in pixels), and `height` (which gives the height of the canvas in pixels).

If we wanted our Tkinter window to have only one canvas of height and width 300, the code we would write would look like this:

```
from Tkinter import *
root = Tk()
foo = Canvas(root, height = 300, width  = 300)
foo.pack()
root.mainloop()
```

We declared canvas as a global variable because we will want to be able to access it in the function we make later.

Drawing on the canvas:
If we want to draw a rectangle on our canvas, foo, the syntax would look like this:
`foo.create_rectangle(topLeftX, topLeftY, bottomRightX, bottomRightY, ...)`
Like all methods, there are many options to `foo.create_canvas`, but for now we will only care about one, fill. If we wanted to create a red square which was a hundred pixels wide and nested in the top corner of the canvas, the function call would look like this:
`foo.create_rectangle(0,0,100,100,fill = "red")`

Other objects that you can draw in Tkinter include ovals, lines, and text. A full list appears online.

Binding keystrokes:

If we want one of the widgets we created in Tkinter to do something whenever we press a key, we must bind that key to that widget. The syntax for binding a keystroke is:
`foo.bind(event,handler)` where `foo` is the widget, `event` is the name of the key, and `handler` is the function that we want to call when we press that key. The function should take no parameters other than `event`.

If we wanted to draw a square on the canvas whenever we pressed the space bar, our code would

look something like this:

```
from Tkinter import *
root = Tk()
foo = Canvas(root, height = 300, width = 300)

def drawSquare(event):
    global foo
    foo.create_rectangle(100,100,200,200,fill = "red")

foo.pack()
root.bind("<Space>",drawSquare)
```