

cs151S10proj3

Philip Prosapio
Spring 2010
CS-151

Note- Code and pictures are in pwprosap on the academic server.

Another note- for some of the code snippets, when code follows a colon it is supposed to be indented, but the wiki didn't do that, so just so you know.

Summary and description of my solution to each task

1) The first thing we were asked to do was to take some of my old code from last lab and rewrite them, by taking advantage of looping and a fill variable. This is an example of what I rewrote:

```
for i in range(2):
    turtle.forward (width)
    turtle.left (90)
    turtle.forward (height)
    turtle.left (90)
    and
    if fill:
        turtle.color(random.random(), random.random(), random.random())
        turtle.begin_fill()
```

2) In the second part, I was asked to take two more basic functions and rewrite them like I did with in part one. The functions that I rewrote for this part was my triangle function and my sun function. I added in looping and fill variables.

```
def triangle(x, y, scale, fill):
    goto(x, y)
    if fill:
        turtle.begin_fill()
    for i in range(3):
        turtle.forward(150*scale)
        turtle.left(120)
    if fill:
        turtle.end_fill()
```

3) For part three, I was asked to take two of my more complicated functions from my second lab and rewrite them again. I chose my truck function and my bridge function. With these two I mainly changed the ability to fill the objects or not.

4) The fourth part asked me to take one of my scenes from lab 2 and rewrite it so that the entire scene is parameterized with x, y, and scale parameters. Then we were asked to have the function run and produce 3 of the scenes of varying size and location on the screen. I did this, and the picture is picture1lab3.png that is attached.

This is a snippet out of the center of my code:

```
sky(x, y, scale, fill)
cliff(x, y, scale, fill)
bridge(x, y, scale, fill)
cliff2(x, y, scale, fill)
```

The key thing is that in every function I have (x, y, scale, fill) in them, and this allows me to input the location, scale, and if I want the picture to be filled.

5) The final part wanted me to place my parameterized scene into an indoor scene that I created. The idea was to move and scale the outdoor scene so it would fit into the indoor scene. This scene is named picture2lab3.png and is attached.

These are the Extensions

Extension 1) My first extension that I created fulfilled the extension that required encapsulation. I placed my previous scene from part five into a TV, so there was a scene inside of a scene inside of a scene. The difficulty came from parameterizing the scene from part 5. The biggest idea that I found out for this part was to make sure that my goto functions included scales, like this:

```
things.goto(x+125*scale, y+390*scale)
```

This made sure that no matter where or at what size I made the scene, they would always be placed in the correct order. This extension is named picture3lab3.png and is attached.

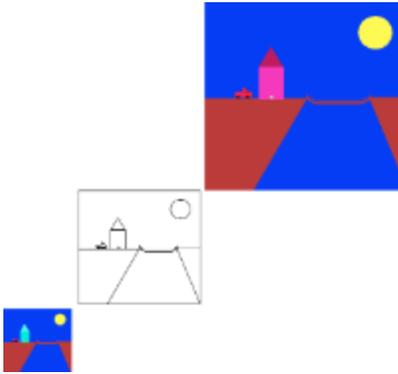
Extension 2) This second extension covers the extension that requires me to have two outdoor scenes in my indoor scene either as pictures or windows. I took my scene from part five and my night scene from lab 2 and put them into an indoor scene. Again the scaling in the lengths and the x, y locations were the key to this extension. This second extension is named picture4lab3.png and is attached.

What I learned

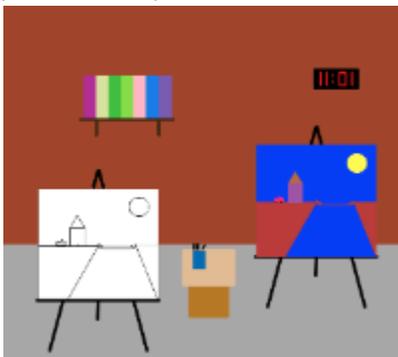
This lab truly tested my mental capacity, easily the biggest thing I learned was the concept of scaling a pre-constructed scene. This was a helpful scale to learn, but it also took a long time to learn it.

Attached photos

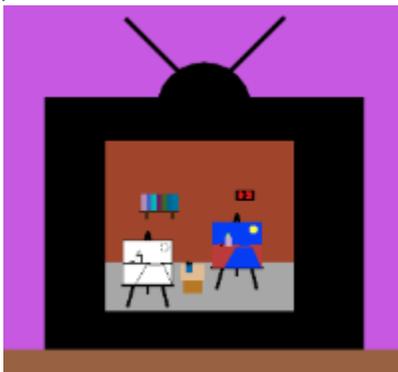
picture1lab3 for part 4



picture2lab3 for part 5



picture3lab3 for extension 1



picture4lab3 for extension 2

