

cs151s10proj5(Philip Prosapio)

Philip Prosapio
Spring 2010

Note- The code for this project is on the academic server under pwprosap name.

Another Note- I had to change many of my brackets to parentheses because the wiki page messed with the contents in side of the brackets.

Summary

The first task that we needed to completed was to create a function buildCollage() that would make a collage that has multiple pictures in it, and with new effects and different element. I then made a function putPixmapNoBlue() that does the same this as the putPixmap() function that we made last lab except that this new function won't transfer the blue background of the blue screen photos we use. We were then tasked to edit our buildCollage() function so that the sixth element in the collage information list would determine which function between putPixmap() and putPixmapNoBlue() to use. We then had to run our buildCollage() function with four photos in it. In the final task we needed to make another collage in which the pictures were placed in a triangular configuration.

Description of Solution

1) For the first task I followed the instructions on the website and created the buildCollage() function. For this task the key thing was to make sure each item in the collage information list was assigned to a different variable so it could be used later in the function.

for item in clist:

```
filename = item(0)
x0 = item(1)
y0 = item(2)
operator = item(3)
alpha = item(4)
useChromakey = item(5)
source = item(-1)
```

2)For the second task, I had to find a way to make it so the blue background would not be added into the collage, so I added an *if* statement to get this accomplished. I had python compare the blue values of each pixel to determine which pixels to bring in.

```
if b1 < 2*r1:
destination.setPixel(x+i, y+j, (r1*alpha + (1.0-alpha)*r2,
g1*alpha + (1.0-alpha)*g2, b1*alpha + (1.0-alpha)*b2))
```

Also notice the *alpha*'s that are in this code snippet. These affect the blending of the pictures into the collage.

3) The third task just required me to add in an *if* and *else* statement to determine what put pixmap function I wanted to be executed.

```
if useChromakey:
putPixmapNoBlue(destination, source, x0, y0, alpha)
else:
putPixmap(destination, source, x0, y0, alpha)
```

So if I entered True then it would run putPixmapNoBlue(), and if I entered False it would run putPixmap().

4) In the fourth task, I took four pictures, three from the lab and the other was the blue screen picture of me, and plugged them into the collage information list that the function buildCollage() is based off of.

```
clist = [ [ argv(1), 0, 0, 'rbswap', 0.8, False, None ],
[ argv(2), 25, 150, 'original', 1.0, True, None ],
[ argv(3), 0, 150, 'rgswap', 0.5, False, None],
[ argv(4), 150, 50, 'ynegate', 0.3, False, None]]
```

As you can tell, in each list contains the information of the filename, x and y location, what function to use on the file, the alpha blend number, my newly created put pixmap indicator, and finally the string None which is a place holder of sorts. The picture of this function is attached and is named picture1lab5.png .

5) The final task that required us to make a collage in the shape of a triangle was similar to the last task, but with a little more specifications of where to place each photo needed. By finding the height and width of the first image, I was able to place the other to images in the correct locations to make the triangle configuration. This image is attached and named picture2lab5.png .

Extensions

Extension 1- The first extension that I completed fulfills the extension that required me to make new functions that implemented different geometric organization of my pictures in my collage. For this extension I made two different functions/pictures. These are squarecollage() and dianmond(). The first one just makes a simple square out of pictures, but the second one is more complicated in which the two different pictures are used with multiple different effects to create a diamond shape. For the diamond function the big thing I needed to solve was the x and y placements of the pictures, here is some code to show how I solved that.

```
topwidth = collageList(0)(-1).getWidth()  
collageList(0)(1) = topwidth  
collageList(1)(1) = topwidth/2  
collageList(2)(1) = (3*topwidth)/2
```

The pictures for this extension are attached and named picture3lab5.png and picture4lab5.png .

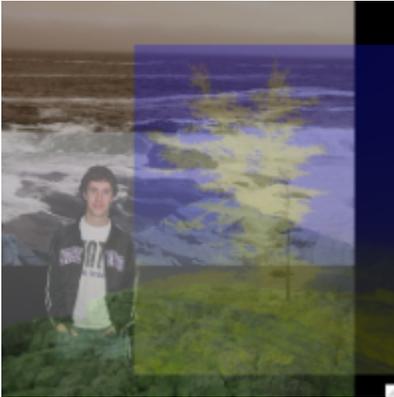
Extension 2- My second extension that I made fulfills the extension that asked me to implement new image effects that I didn't already make in lab. I made new image effects and used one of my old functions to implement the new effects. The attached picture is named picture5lab5.png .

What I Learned

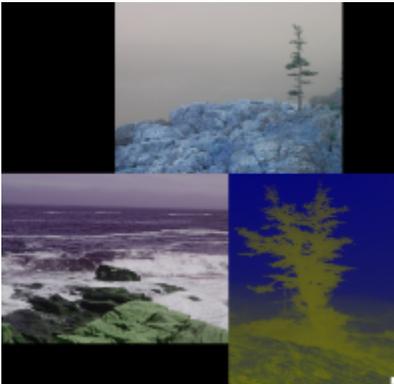
The most helpful thing that I learned has to do with lists. Learning how to assign values to items in lists, and how to call lists as a whole was a helpful skill to learn.

Pictures

picture1lab5.png



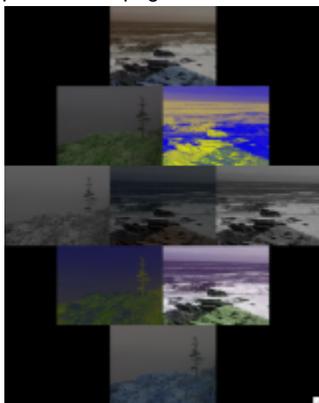
picture2lab5.png



picture3lab5



picture4lab5.png



picture5lab5.png

