

Project5

In this project, I built upon the Python graphics coding we started learning in last week's project4. For project 5, we made the programming more complex and the resulting images were thus more intricate. I used lists to define collages and something called "alpha blending" to overlap the images with varying amounts of transparency. Using my knowledge of parameters and lists within lists allowed me to easily create collages. Also, I used more "import sys" and placePixmap locations. The first goal of the project was to write functioning code to create a collage with different images, in different locations, with different transparencies, and even no background. The second goal was to create a Facebook like cover-photo by aligning the collage in a horizontal fashion.

Task 1 asked me to add a sixth item the collage list we created in Lab5. The new item in the list(NoBkg) was a Boolean value(true or false) to determine whether or not the image would contain background pixels. Next, using commented code directions, I was asked to create a new function called buildCollage() in my collage.py file that basically incorporated all the items from the list and displayed them. Here is a sample of the code from the new function:

```
def buildCollage( cList ):
    '''builds collage with input image lists
       with different filters, alpha values and locations'''
    (cols, rows) = getImageSize(cList)
    dst = graphics.Pixmap(cols, rows)
    for item in cList:
        filename = item[0]
        x0 = item[1]
        y0 = item[2]
        operator = item[3]
        alpha = item[4]
        noBkg = item[5]
        src = item[-1]
```

Then, I went on to add if and elif statements to apply correct operators/filters for all four of my filter functions depending on the operator value in the collage list. I used a program given to me (testbuildcollage.py) to test to see if my build collage function worked.

Task 2 asked me to create a function named placePixmapNoBkg() in my filter.py file. The code for this function was identical to the code for placePixmap() except for the end. The end of this new function had if and else statements that had to do with color. Since the background of my photo me.ppm was a green screen, I wrote if green values were more than red and blue than do not copy the background pixels. Here is what the code looked like:

```
def placePixmapNoBkg(dst, src, x, y, alpha):
    ''' used to position Pixmap in desired location
        now with alpha blending capabilities
        with green background removed'''
    for i in range(src.getHeight()):
        for j in range(src.getWidth()):
            (r1, g1, b1) = src.getPixel(j, i)
            (r2, g2, b2) = dst.getPixel(j+x, i+y)
            rnew = r1*alpha + (1.0-alpha)*r2
            gnew = g1*alpha + (1.0-alpha)*g2
            bnew = b1*alpha + (1.0-alpha)*b2
            if g1<(1.5*r1) or g1<b1:
                dst.setPixel(j+x, i+y, (rnew, gnew, bnew))
            else:
                dst.setPixel(j+x, i+y, (r2, g2, b2))
```

Task 3 then asked me to add the function I just coded in task 2 to my function buildCollage() from task 1 in my collage.py file. I used if and else statements to determine whether to copy the background pixels depending on the Boolean value of the sixth item in the collage list. Here is what the add-on code to buildCollage() looked like:

```
'''now considers boolean value of NoBkg
   and removes background if true'''
if noBkg == True:
    filter.placePixmapNoBkg(dst, src, x0, y0, alpha)
else:
    filter.placePixmap(dst, src, x0, y0, alpha)
```

Task 4 asked me to create a new file titled mycollage.py and create a single main() function that creates a collage list, calls readImages() function from Lab5, calls buildCollage() function from task 1, and then saves the resulting image into a .ppm file. The collage was asked to incorporate at least five different images, three filters, and at least one alpha blend. I decided to use four of the maine.ppm files and my me.ppm file with the green screen. I used a "True" value to remove the green screen from my image and place me in the foreground of the collage. Here is what some of the main() code looked like:

```

def main( argv ):
    if len(argv) < 6:
        print 'You need more command line files you dumb dumb'

    '''new collist with command line files'''
    collist = [ [ argv[1], 0, 0, 'swapRedBlue', .90, False, None ],
                [ argv[2], 400, 400, 'colorNegative', 0.75, False, None ],
                [ argv[3], 0, 350, 'purpleHaze', 0.65, False, None ],
                [ argv[4], 450, 0, 'moreGreen', 0.50, False, None ],
                [ argv[5], 125, 50, 'original', 1.0 , True, None]
            ]
    # call readImages
    collage.readImages( collist )
    # call buildCollage
    dst = collage.buildCollage( collist )
    # save the image
    dst.save( 'mycollage.ppm' )

```

Here is the resulting mycollage.ppm image:

Required Image1-(mycollage):



Task 5 asked me to create another new python file titled coverphoto.py and using similar code with a main() function, create a cover photo collage that is about three times as wide as it is high. Here is my resulting image:

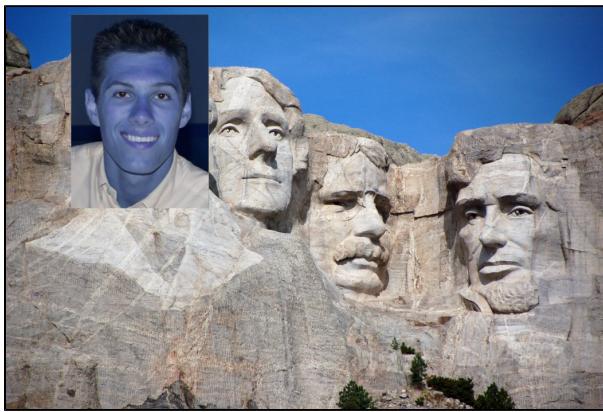
Required Image2-(coverphoto):



After finishing the required tasks, I completed a few extensions. **Extension 1** incorporated the no background pixels functions and allowed me to place myself in a random scene. I decided to put myself(me.ppm) in the "Thriller" music video by Michael Jackson(thriller.ppm). The code for this extension was very similar to the codes from tasks 4 and 5. I simply made a new main() function with a new collage list. Here is what the resulting image looked like:



Extension 2 was basically the same as extension 1 but with different background and foreground images. I took the mug shot(myface.ppm) from my Colby card and place it amongst the Presidents on Mount Rushmore(mtrushmore.ppm). Here is what it looked like:



Extension 3 was one of the suggested extensions. I created a dynamic collage that was always different depending on the number of command line files or len(argv). To do this, I had to create an if, an elif statement, and an else statement with different len(argv) values. Then, I had to create a new collage list for each conditional each with a readImage(), buildCollage(), and ultimately a saving function. Here is what the middle elif statement code looked like:

```
elif len(argv) == 4:
    '''new clist with 3 command line files'''
    clist2 = [ [ argv[1], 0, 0, 'swapRedBlue', .90, False, None ],
               [ argv[2], 600, 0, 'colorNegative', 0.75, False, None ],
               [ argv[3], 300, 0, 'purpleHaze', 0.65, False, None ],
             ]
    # call readImages
    collage.readImages( clist2 )
    # call buildCollage
    dst = collage.buildCollage( clist2 )
    # save the image
    dst.save( 'p5extension3b.ppm' )

    #for 3b i used stephcurry.ppm
```

If it there was one file, the output was snorlax.ppm. If there were three files, the output was stephcurry.ppm. If there were more than three, the output was a "Tony Overload" with myface.ppm and me.ppm. Here are the three resulting images from my third extension:





What I learned. Overall, the objective of this project was to learn how to use lists within lists to create a collection of Python graphics images and grasping how to place and orient them using `placePixmap()` function. Also, I learned about a lot more possibilities with Python graphics. For example, I know understand Alpha blending and the ability to overlap, blend, and make images more or less transparent. I know understand how to really be efficient with lists inside of lists and the flexibility of items in lists. In addition, I furthered my knowledge of `sys.argv`, loops, conditionals, and parameters. This project made me feel much more adept with Python graphics and confident in how to code using Python graphics.

I received help from TA Mike with the if and else statements in `placePixmapNoBkg()` function. I worked alongside fellow CS 151 classmate Matt Hawkins during this project and we helped each other.