

Project4

In this project, I used a lot of new code and started a completely new topic in Python. Using something called Python graphics functions, I was able to create and display images. Along with just simply displaying the image, editing individual pixels allowed me to adjust the colors of each image. Also, I gained more experience with "import sys." The goal of this project was to begin using simple Python graphics function. I was asked to put this to use through four steps and two main tasks -- create a "Warhol-like" collage and change the pixels of Green-Screen image.

Task 1 asked me to create a new function called placePixmap() that positioned my Pixmap image/file at a specific position. Using a template provided by my professors, I coded a function that included two for loops that received the height and width of the Pixmap and the ability to set the location of the top left corner of the image. The function took four inputs -- (dst, src, x, y). "Dst" was the original smaller Pixmap, x referred to x-location, y referred to y-location, and finally "src" was the new bigger Pixmap used in the new part of **task 1**. The final component to **task 1** was to make a duplicate of the "dst" image and print it twice inside the bigger "src" Pixmap. Here is what the code looked like:

```
# Task 1
# create placePixmap file
# takes four arguments
# places pixmap file at 0,0
# then have it copy and display to the right of original image

# this places Pixmap image/file at given location
def placePixmap( dst, src, x, y ):
    ''' used to position Pixmap in desired location '''
    for i in range(src.getHeight()):
        for j in range(src.getWidth()):
            (r, g, b) = src.getPixel( j, i )
            dst.setPixel( x+j, y+i, (r,g,b) )
```

I then tested my code to see if it produced a duplicate, side by side image. Our professors provided us with a simple testing python file to see if our placePixmap() function worked properly. Using a save function in Python graphics, I was able to save the produced image as a .ppm file titled duplicate.ppm. Here is what it looked like:



Task 2 asked me to created three more filter functions like the filter we created during Lab4 titled swapRedBlue(). To do this, I used basically the same code from the swapRedBlue function, but I simply rearranged the amounts of r, g, and b in the last line of code. I created three new filter functions titled colorNegative(), moreGreen() and purpleHaze() to add to the filter collection with swapRedBlue(). These filter functions were to be used later in my project.

Task 3 was the pinnacle task of this project. I was asked to make a Warhol program titled warhol.py. Using a simply template of commented code provided by the professors, I defined a new function called main() that took a command line argv parameter. In the function, I defined a single Pixmap as pmap = graphics.Pixmap(filename). The file name was the argv[1] second thing typed in the terminal command line. Next, I cloned the pmap four times with four different names. Then, I assigned a different filter to each separate image. After that, I defined a new variable "big" as big = graphics.Pixmap(2*pmap.getWidth(), 2*pmap.getHeight()). What that did was create a Pixmap to place the four images within that is two times a wide and two times as tall as the original image to fit the four. Finally, I used my placePixmap() function from my filter.py file to place each of the four images at a separate corner of the Warhol masterpiece. I had to import my filter.py first. Each filter.placePixmap() had a little different parameters to make sure it was placed in the correct spot. Here is what the last bit of code looked like:

```
# places the four images at different locations
# into the 'big' pixmap
# each of the four images gets a different filter
filter.placePixmap( big, map1, 0, 0 )
'''put the 4 filtered images into the big one'''
filter.placePixmap( big, map2, 0, pmap.getHeight() )
filter.placePixmap( big, map3, pmap.getWidth(), 0 )
filter.placePixmap( big, map4, pmap.getWidth(), pmap.getHeight() )

# save the resulting image as a file name 'warhol.py'
big.save( 'warhol.ppm' )
```

Notice, I used big.save() again to automatically save the resulting image as warhol.ppm. Here is what the warhol piece looked like:



Task 4 asked me to take the green screen image of myself that I took in class a couple weeks back and change all the green background pixels. To do this, I have to loop over every pixel and selected and change only the pixels that are very green. I used a couple for loops to scan through all the pixels in the image and then I used if conditional statements to change the color of the green pixels. Here is what a snippet of my code

```

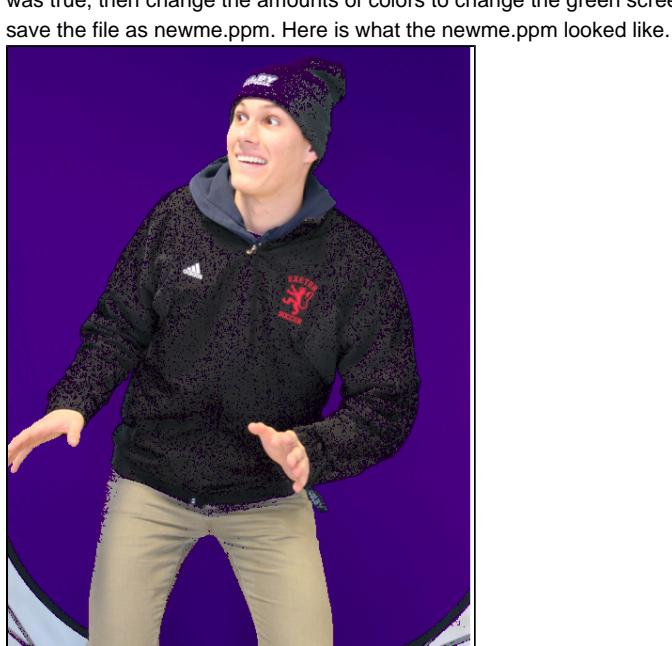
pm = graphics.Pixmap( 'me.ppm' )
w= pm.getWidth()
h = pm.getHeight()
for i in range(w):
    for j in range(h):
        (r, g, b) = pm.getPixel( i, j )
        if g > r and g > b:
            pm.setPixel( i, j, (0.5*r, 0, 0.7*b) )

filename = argv[1]

pm.save( 'newme.ppm' )

```

looked like:



The green screen was changed to a purple screen.

After I finished the required tasks for the project, I completed a few extensions. For **extension 1** I did something similar to task 4 with the Warhol collage but with my green screen picture and a picture of a monster! I partially tackled one of the suggested extensions by changing the shape of my Warhol from a square to a rectangle and thus changing some of the code. This also partially satisfies the suggested extension to give my Warhol code multiple command line arguments to use multiple images in the collage. I decided to make a collage with myself and Godzilla! Here is what it looked like:



For a second extension, **extension 2**, I decided to do the suggested extension to do something more creative than replacing just the green from the green screen. I decided to divide the green screen up into four sections and change each section to a different and separate color. To do this, I created four separate for loops and I simply adjusted the pixel scan to width/2 thru height/2 and all the other three of the combinations to scan for each for loop. Here is what the resulting extension image looked like:



I started a third extension, **extension 3**, but didn't get it done well because of time. But, what I was trying to attempt was one of the suggested extensions that placed my me.ppm image inside another image with a different colored background. I used a bigger Godzilla picture as the background Pixmap. To do this, I made a new function called placeNoGreenPixmap() that changed the background to white... or I thought. Running out of time, I will show you what I got. To start, here is some code from my attempted third extension:

```
def placeNoGreenPixmap( me, monster, x, y ):
    ''' used to position Pixmap in desired location '''
    for i in range(monster.getHeight()):
        for j in range(monster.getWidth()):
            (r, g, b) = monster.getPixel( j, i )
            if g > r and g > b:
                me.setPixel( x+j, y+i, (255, 255, 255) )
def main(argv):
    if len(argv) <2:
        print "Usage: python show.py <filename>"
        exit()
    monster = graphics.Pixmap( 'biggodzilla.ppm' )
    me = graphics.Pixmap( 'me.ppm' )
    '''read in one pixmap'''

    extension3 = graphics.Pixmap(monster.getWidth(), monster.getHeight())
    filter.placePixmap( extension3, monster , 0, 0 )
    placeNoGreenPixmap( extension3, me, 0, 0 )
```

BUT, this code has failed me in the absence of time. Here is the image it wrongfully produced :((:



Whether this is what I was trying to do or not, the resulting image turned out kinda cool.

What I learned. Overall, the objective was to make a collection of images with adjusted pixel colors within a single image while gaining a grasp on simple Python graphic functions like pixmap placement and pixel editing. I learned the basics of Python graphics and how to use, position, scale, and edit(on a basic level) Pixmaps and Python images. In addition, I furthered my knowledge on for loops, conditionals, sys package, and parameters. This was a good introduction into Python graphics and the variety of functions that come along with it.

I received help from Professor Maxwell with Task4 and the locating of specific colored pixels. He also helped me with some of my filter ideas and edits.