**Initial Design Plan:**

* **Greyscale**
  + Add the red, green, and blue together
  + Take the total and divide by 3
* **Flip horizontal**
  + Make a deep copy
  + Create a for loop
  + Reverse the newpixellist
* **Rotate clockwise**
  + Create two lists
  + The first list will hold the values in the columns
  + The column values will then be placed in the second list
  + This will repeat for all of the following columns
  + The second list will append all of the first list values together
* **Purple**
  + Set the green value as 0
  + Add the red and blue together
  + Take the total and divide by 2
* **Flip vertical**
  + Make a deep copy
  + Reverse rowlist

**Summary:**

* When doing the different colors, it was easy to just alter them, but the horizontal and vertical changes were a little more difficult. I decided to use the reverse method. When trying to do the horizontal flip, I ended up accidentally flipping it vertically. I found out that I needed to change what I was reversing, which makes sense to change the rowlist when flipping it horizontally. The rotate clockwise is what took the longest time for me to do. The concepts were used, but I had to add a few things such as resetting the first list, and different for loops. The end results made all of the methods work. I spent around an hour and a half programming these results.

**Implementations:**

* Making the horizontal and vertical method work
* The issue was that I was reversing the wrong things within them

**Testing:**

* bc-flowers.ppm
* bc-sign.ppm

**Files:**

* hea-L3-ppm.py
* hea-ppm.py
* bc-flowers.ppm
* bc-sign.ppm

**Errors:**

* Quit button does not work. It tends to freeze instead of just exits.

**Comments:**

* I thought this lab was fun. It was a lot simpler than I thought other than rotating the image clockwise. That was the most difficult for me. I loved the idea of creating my own filters for the images.

**CRC Card for the PPM class**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Class name: PPM class** | **PPM** | | **Class Methods:** | **Class Collaborations (other classes, etc):** | | * **\_\_init\_\_()** initializer/constructor for the class, opening a provided PPM-P3 file and setting all member attributes. * **PPM\_makeoutputfiles()** given self.inasciifile, sets self.ascii and creates both ascii and binary files for output * **PPM\_partition()** Given input parameter strng, the string to partition and ch, the character to use as the delimiter returns a triple with all characters before the delimiter, the delimiter itself if present and all of the characters after the delimiter (if any). * **PPM\_clean()** removes all single line comments, whitespace, and newline characters present in the input parameter string strng. * **PPM\_load()** takes string input parameter inasciifile as the name of the ASCII PPM-P3 (non-binary) file to load. * **PPM\_makepixellist()** creates self.pixellist, a nested list of rows of [red, green, blue] pixels from input color\_list which contains an unnested list of strings. * **PPM\_updatefrompixellist()** updates image object data and related files from input pixellist * **PPM\_convert2bin()** converts PPM-P3 to PPM-P6 using self.pixellist. * **PPM\_set\_title()** setter for title of display window. * **PPM\_make\_red(** ) colorizes current image to red by using self.pixellist. * **PPM\_greyscale()**colorizes current image to grey by adding the colors together, dividing by 3, and using self.pixellist. * **PPM\_flip\_horizontal()** flips image horizontally by using the reverse method. * **PPM\_rotateclockwise()** rotates image clockwise by using lists to store the values. * **PPM\_make\_purple()** colorizes current image to purple by adding the red and blue, diving by 2, and using self.pixellist. * **PPM\_flip\_vertical()** flips image vertically by using the reverse method. | * **PPM\_set\_up()** is a helper function which must be called at the beginning of any program which uses the PPM class, but it is not a part of the class. * **PPM\_render()** is a helper function which renders all PPM images. It is not part of the class. * **PPM\_Exception** is a Python class which enables meaningful error messages on exceptions. | | **Class Data:** | **Class Collaborations (other classes, etc):** | | * **self.root** # provided master Tkinter instance created using helper function PPM\_set\_up() * **self.inasciifile** # string which represents filename is used only for reading the provided PPM-P3 as input * **self.outasciifile** # string of filename of human readable modifications to the PPM file * **self.outbinfile**  # string of binary ppm filename needed for viewing * **self.title**  # used for the title of the display window * **self.magic** # ppm file type is often called the "magic number." It needs to be P3 to be readable * **self.comment** # creates a comment for the PPM file * **self.width** # image width in pixels * **self.height** # image height in pixels * **self.colormax** # should be set to 255 * **self.ascii**  # will store the color intensities in P3 format * **self.pixellist** # will store nested list containing pixel colors * **self.image** # reference to image window * **self.label**  #  used to place image in window | * global **tkintertoggle** is needed as global to ensure a single Tkinter instance which is needed to render image | |