Lab 08

CSCI 110-02   Red-Eye, Sepia Tones, & Edge Detection   October 16, 2013

Name: ________________________________

We are going to add to our library of 2 dimensional processing functions. Each time the instructions as for you to show me your testing of a function I will initial that you have done that step correctly.

If you are working on your own laptop you must download the mediasources from http://coweb.cc.gatech.edu/mediaComp-teach in order to do the lab.

Preliminary setup. Use `setMediaPath()` to point to the media sources directory. Use `picture=makePicture(getMediaPath("filename.jpg"))` to load the picture

1. JES allows us to calculate the distance between two colors. By using this idea of distance, we can adjust some colors while leaving others untouched. We are going to write a function to reduce red-eye. First open and explore the picture jenny-red.jpg. Figure out a rectangle where Jenny's eyes are. We need an upper left corner and a lower right corner to define a rectangle.

   What numbers do you have for the rectangle (startX, startY, endX, endY)?

   ____

   ```
   def removeRed(pic, startX, startY, endX, endY, replacementColor):
     red = makeColor(255,0,0)
     for x in range(startX, endX):
       for y in range(startY, endY):
         currentPixel = getPixel (pic, x, y)
         if distance(red,getColor(currentPixel)) < 165:
           setColor(currentPixel, replacementColor)
   ```

   call this function by doing the following:

   ```
   jenny=makePicture(getMediaPath("jenny-red.jpg"))
   removeRedEye(jenny, 109,91,202,107, makeColor(0,0,0))
   explore(jenny)
   ```

   Now download the picture of the authors from Piazza and replace the red uniforms with blue ones.

   When it works show me ________

2. Old photographs have a yellowish tint. We sometimes wish to apply this treatment to new pictures. It's called Sepia-Tone. Some cameras can take pictures in sepia, most graphics softwares allow a picture to be changed to sepia tones. We begin with a grayscaled image. If the function to grayscale a picture is not in your library, add it now:

```
def grayscale(picture)
  for px in getPixels(picture):
    newRed = getRed(px) * 0.299
    newGreen = getGreen(px) * 0.587
    newBlue = getBlue(px) * 0.114
    luminance = newRed + newGreen + newBlue
    setColor(px,makeColor(luminance, luminance, luminance))
```

We will change high, middle and low ranges of luminance differently. The values in this function were arrived at through trial and error... finding a pleasing value for the sepia tone. Be VERY careful of the indentation.

```
def sepiaTint(picture):
   # call the grayscale function
   grayscale(picture)
   # loop through picture to tint the pixels
   for p in getPixels(picture):
     redP = getRed(p)
     blueP = getBlue(p)
     # tint shadows
     if (redP < 63):
       redP = redP * 1.15
       blueP = blueP * 0.9
     # tint midtones
     if (redP > 62 and redP < 192):
       redP = redP * 1.15
       blueP = blueP * 0.85
     # tint highlights
     if(red > 191):
       redP = redP * 1.08
       if (redP > 255):
         redP = 255
       blueP = blueP * 0.93
     # set the new color values
     setBlue(p, blueP)
     setRed(p, redP)
```

When it works show me ________

3. Blurring a picture reduces pixilization that might occur when you enlarge it. That makes blurring an image very useful. For our algorithm we will set a pixel to the average of the 4 pixels around it. Please NOTE: three lines below are wrapped because they need to fit on this page. You Must NOT break a line of code in Python!!

```
def blur(filename):
   source = makePicture(filename)
   target = makePicture(filename)
   for x in range (1 , getWidth(source) - 1)
     for y in range (1 , getHeight(source) - 1)
       top = getPixel(source, x, y-1)
       left = getPixel(source, x - 1, y)
       bottom = getPixel(source, x, y + 1)
       right = getPixel(source, x + 1, y)
       center = getPixel(target, x, y)
       newRed = (getRed(top) + getRed(left) + getRed(right) +
getRed(bottom) + getRed(center))/5
       newGreen = (getGreen(top) + getGreen(left) + getGreen(right) +
getGreen(bottom) + getGreen(center))/5
       newBlue = (getBlue(top) + getBlue(left) + getBlue(right) +
getBlue(bottom) + getBlue(center))/5
       setColor(center, makeColor(newRed, newGreen, newBlue))
   return target
```

Save the file and load the file. Choose a picture and enlarge it (code from lab 07 should be in your library) to get it fairly pixelated then run it through the blur function to see the result. You will need to use: `writePictureTo(aPicture,afilename)` where aPicture is the picture you get back from enlargePicture and afilename is a string that includes the extension .jpg then send it to your blur function. It would be best to save it to a USB drive and then set that as the media path.

When it works show me ________

4. Another common graphic software routine is edge detection. We can convert a color picture into a line drawing by finding the edges and repainting the picture as a black and white of the edges. Add these functions to your file.

```
def getLum(pixel):
   return ((getRed(pixel) + getGreen(pixel) + getBlue(pixel))/3)

def isLine(here, down, right):
   return abs(here-down) > 10 and abs(here-right) > 10

def lineDetect(filename):
   orig = makePicture(filename)
   makeBw = makePicture(filename)
   for x in range (1 , getWidth(orig) - 1):
     for y in range (1 , getHeight(orig) - 1):
       here = getPixel(makeBw, x, y)
       down = getPixel(orig, x, y + 1)
       right = getPixel(orig, x + 1, y)
       hereL = getLum(here)
       downL = getLum(down)
       rightL = getLum(right)
       if isLine(hereL, downL, rightL):
         setColor(here,black)
       if not isLine(hereL, downL, rightL:
       setColor(here,white)
   show(makeBw)
   return makeBw
```

Choose a file to send to the lineDetect function then save the result using
`writePictureTo(aPicture,afilename)`
where aPicture is the picture you get back from lineDetect and afilename is a string that includes the extension .jpg then send it to your blur function. It would be best to save it to a USB drive and then set that as the media path.

When it works show me ________

5. Drawing on a picture. There are 4 drawing commands that we can use on our pictures.
   (a) `addText(pict, x, y, string)` puts the string starting at the point (x,y) in the picture pict.

   (b) `addLine(pict,x1,y1,x2,y2)` draws a line from (x1,y1) to (x2,y2)

   (c) `addRect(pict, x, y, w, h)` draws a rectangle with the top left corner at (x,y) that is w wide and h tall.

   (d) `addRectFilled(pict, x, y, w, h, color)` draws a rectangle and fills it with the color color.

   Add the following function to your file and test it:

```
def addBox():
   beach=makePicture(getMediaPath("beach-smaller.jpg"))
   addRectFilled(beach,150,150,50,50,red)
   show(beach)
   return beach
```

   Now try each of the other ones by writing a small function like the one above.

   When they work show me ________

Hand this in to get credit for this lab