

# Tyler Newlin

Completed



Original



```
#Tyler Newlin  
#March 9, 2020
```

```
def collage():  
    original_picture = makePicture(getMediaPath("ssFrance.jpg"))  
    signature = makePicture(getMediaPath("signature.jpg"))  
    #Scaling the picture down so it fits within the project maximum height and width.  
    scaledDown_picture = scaleDown(original_picture, 6)  
    #Scaling down the signature a lot so it fits into the picture reasonably.  
    scaledDown_signature = scaleDown(signature, 22)  
    #making the canvas that all of the edited parts of the picture will be copied into.  
    finalCollageCanvas = makeEmptyPicture(getWidth(scaledDown_picture), getHeight(scaledDown_picture))  
    #cropping sections of the picture to be edited.  
    section_one = scaledDown_picture  
    section_two = crop(scaledDown_picture, 93, 63, 838, 569)  
    section_three = crop(scaledDown_picture, 186, 126, 745, 506)  
    section_four = crop(scaledDown_picture, 279, 189, 652, 443)  
    #Section five will be unchanged and just copied into the final collage.  
    section_five = crop(scaledDown_picture, 372, 252, 559, 380)  
    #Editing the sections of the picture.  
    #section one  
    lighten(section_one, 0, getWidth(section_one), 0, getHeight(section_one))  
    #section two  
    grayScale(section_two, 0, getWidth(section_two), 0, getHeight(section_two))  
    #section three  
    randomToneChange(section_three, 0, getWidth(section_three)/3, 0, getHeight(section_three))  
    randomToneChange(section_three, getWidth(section_three)/3, (getWidth(section_three)/3)*2, 0,  
    0, getHeight(section_three))
```

```

randomToneChange(section_three, (getWidth(section_three)/3)*2, getWidth(section_three), ↓
    0, getHeight(section_three))
#section four
negative(section_four, 0, getWidth(section_four), 0, getHeight(section_four))
#copying edited sections into the finalCollageCanvas.
copy(section_one, finalCollageCanvas, 0, 0)
copy(section_two, finalCollageCanvas, 93, 63)
copy(section_three, finalCollageCanvas, 186, 126)
copy(section_four, finalCollageCanvas, 279, 189)
copy(section_five, finalCollageCanvas, 372, 252)
#Signing the finished collage
sign(scaledDown_signature, finalCollageCanvas)
explore(finalCollageCanvas)

#Below here is all of the helper functions used in the main program. I tried to make them so that they
#aren't specific to only this picture (besides the sign() function).
def copy(picture_in, picture_out, targetx_start, targety_start):
    targetx = targetx_start
    for sourcecx in range(0, getWidth(picture_in)):
        targety = targety_start
        for sourcecy in range(0, getHeight(picture_in)):
            color = getColor(getPixel(picture_in, sourcecx, sourcecy))
            setColor(getPixel(picture_out, targetx, targety), color)
            targety = targety + 1
        targetx = targetx + 1
    return picture_out

def crop(picture_in, startX, startY, endX, endY):
    picture_out = makeEmptyPicture(endX-startX, endY-startY)
    targetx = 0
    for sourcecx in range(startX, endX):
        targety = 0
        for sourcecy in range(startY, endY):
            color = getColor(getPixel(picture_in, sourcecx, sourcecy))
            setColor(getPixel(picture_out, targetx, targety), color)
            targety = targety +1
        targetx = targetx +1
    return picture_out

def scaleDown(picture_in, scaleFactor):
    picture_out = makeEmptyPicture(getWidth(picture_in)/scaleFactor, getHeight(picture_in)/scaleFactor)
    sourcecx = 0
    for targetx in range(0,getWidth(picture_out)):

```

```

sourcey = 0
for targety in range(0,getHeight(picture_out)):
    srcpx = getPixel(picture_in, int(sourcex), int(sourcey))
    color = getColor(srcpx)
    setColor(getPixel(picture_out, targetx, targety), color)
    sourcey = sourcey + scaleFactor
sourcex= sourcex + scaleFactor
return picture_out

def grayScale(picture_in, start_x, end_x, start_y, end_y):
    for x in range(start_x, end_x):
        for y in range(start_y, end_y):
            pixel = getPixel(picture_in, x, y)
            newRed = getRed(pixel) * 0.299
            newGreen = getGreen(pixel) * 0.587
            newBlue = getBlue(pixel) * 0.114
            luminance = newRed+newGreen+newBlue
            setColor(pixel, makeColor(luminance,luminance,luminance))
    return picture_in

def randomToneChange(picture_in, start_x, end_x, start_y, end_y):
    import random
    effect_choices = [0,1,2,3]
    effect_choice = random.choice(effect_choices)
    if (effect_choice == 0):
        sepiaTint(picture_in, start_x, end_x, start_y, end_y)
    if (effect_choice == 1):
        blueTint(picture_in, start_x, end_x, start_y, end_y)
    if (effect_choice == 2):
        redTint(picture_in, start_x, end_x, start_y, end_y)
    if (effect_choice == 3):
        greenTint(picture_in, start_x, end_x, start_y, end_y)

def sepiaTint(picture_in, start_x, end_x, start_y, end_y):
    grayScale(picture_in, start_x, end_x, start_y, end_y)
    for x in range(start_x, end_x):
        for y in range(start_y, end_y):
            pixel = getPixel(picture_in, x, y)
            red = getRed(pixel)
            blue = getBlue(pixel)
            if (red < 63):
                red = red*1.1
                blue = blue*0.9

```

```

    if (red > 62 and red < 192):
        red = red*1.15
        blue = blue*0.85
    if (red > 191):
        red = red*1.08
    if (red > 255):
        red = 255
        blue = blue*0.93
    setBlue(pixel, blue)
    setRed(pixel, red)
return picture_in

def blueTint(picture_in, start_x, end_x, start_y, end_y):
    grayScale(picture_in, start_x, end_x, start_y, end_y)
    for x in range(start_x, end_x):
        for y in range(start_y, end_y):
            pixel = getPixel(picture_in, x, y)
            blue = getBlue(pixel)
            red = getRed(pixel)
            green = getGreen(pixel)
            setRed(pixel, red*0.5)
            setGreen(pixel, green*0.5)
            setBlue(pixel, blue*1.5)
    return picture_in

def redTint(picture_in, start_x, end_x, start_y, end_y):
    grayScale(picture_in, start_x, end_x, start_y, end_y)
    for x in range(start_x, end_x):
        for y in range(start_y, end_y):
            pixel = getPixel(picture_in, x, y)
            blue = getBlue(pixel)
            red = getRed(pixel)
            green = getGreen(pixel)
            setRed(pixel, red*1.5)
            setGreen(pixel, green*0.5)
            setBlue(pixel, blue*0.5)
    return picture_in

def greenTint(picture_in, start_x, end_x, start_y, end_y):
    grayScale(picture_in, start_x, end_x, start_y, end_y)
    for x in range(start_x, end_x):
        for y in range(start_y, end_y):
            pixel = getPixel(picture_in, x, y)

```

```

    blue = getBlue(pixel)
    red = getRed(pixel)
    green = getGreen(pixel)
    setRed(pixel, red*0.5)
    setGreen(pixel, green*1.5)
    setBlue(pixel, blue*0.5)
return picture_in

def negative(picture_in, start_x, end_x, start_y, end_y):
    for x in range(start_x, end_x):
        for y in range(start_y, end_y):
            pixel = getPixel(picture_in, x, y)
            red = getRed(pixel)
            green = getGreen(pixel)
            blue = getBlue(pixel)
            negColor =makeColor(255-red, 255-green, 255-blue)
            setColor(pixel, negColor)
        return picture_in

def lighten(picture_in, start_x, end_x, start_y, end_y):
    for x in range(start_x, end_x):
        for y in range(start_y, end_y):
            pixel = getPixel(picture_in, x, y)
            color = getColor(pixel)
            color = makeLighter(makeLighter(makeLighter(color)))
            setColor(pixel, color)
        return picture_in

#I made this function specific to my collage so it is located in the place I wanted.
def sign(picture_in, picture_out):
    for x in range(0, getWidth(picture_in)):
        backgroundX = x - 2
        for y in range(0, getHeight(picture_in)):
            backgroundY = y + 488
            pixel = getPixel(picture_in, x, y)
            if (getRed(pixel) < 100 and getGreen(pixel) < 90 and getBlue(pixel) < 90):
                background_pixel = getPixel(picture_out, backgroundX, backgroundY)
                setColor(background_pixel, black)
        return picture_out

```

↵ means the line is continued on the next line.