

Calvin Miller

Completed



Originals



Made by Calvin Miller 10/19/2025
from mediaComp import *

```
def collage():

    picture = makePicture(getMediaFolder("butterfly.jpg"))
    background = makePicture(getMediaFolder("jungle2.jpg"))
    sig = makePicture(getMediaFolder("signature.jpg"))
    #start making and modding the butterflies
    normalButterFly = Scale(picture, 0.9) #"normal" butter fly
    #pictureTool(normalButterFly)

    oldButterFly = Scale(picture, 0.9) #greyScale normal size
    oldButterFly = blackAndWhite(oldButterFly)
    #pictureTool(oldButterFly)

    bigmadButterFly = Scale(picture, 1.15) #Scaled more, redtinted
    makeRed(bigmadButterFly)
    #pictureTool(bigmadButterFly)

    smallsadButterFly = Scale(picture, .75) #Scaled less bluetinted
    Coloraverage(smallsadButterFly)
    ColorSwap(smallsadButterFly)
    #pictureTool(smallsadButterFly)

    newButterFly = Scale(picture, 0.9) #colorsmaxed
    maxColor(newButterFly)
    #pictureTool(newButterFly)

    #start making background
    makelighter(background)
    border = makeEmptyPicture(1000, 736, green )
    Forest = Scale(background, 1.25)
    copyToNewborder(Forest, border, 100, 34)

    #start inserting butterflies
    copyToNew(normalButterFly, border, 255, 107)
    copyToNew(oldButterFly, border, 728, 124)
    copyToNew(bigmadButterFly, border, 396, 417)
    copyToNew(smallsadButterFly, border, 809, 363)
    copyToNew(newButterFly, border, 199, 498)

    #add sig
    copyToNew(sig, border, 411, 660)
    pictureTool(border)
```

↓ indicates the line is continued on the next line.

```

def copyToNewborder(source_picture, destination_picture, startx, starty):
    # mix of copy to and chroma
    for y in range(getHeight(source_picture)):
        for x in range(getWidth(source_picture)):
            setColor(getPixelAt(destination_picture, x + startx, y + starty), ↓
                    getColor(getPixelAt(source_picture, x, y)))

def copyToNew(source_picture, destination_picture, startx, starty):
    for y in range(getHeight(source_picture)):
        for x in range(getWidth(source_picture)):
            redv = getRed(getPixelAt(source_picture, x, y))
            greenv = getGreen(getPixelAt(source_picture, x, y))
            bluev = getBlue(getPixelAt(source_picture, x, y))
            light = (redv + greenv + bluev) // 3
            if light < 240:
                setColor(getPixelAt(destination_picture, x + startx, y + starty), ↓
                        getColor(getPixelAt(source_picture, x, y)))

def Scale(source_picture, scale):
    new_width = getWidth(source_picture) * scale
    new_height = getHeight(source_picture) * scale
    new_picture = makeEmptyPicture(new_width, new_height)
    source_x = 0
    for new_x in range(0, getWidth(new_picture)):
        source_y = 0
        for new_y in range(0, getHeight(new_picture)):
            color = getColor(getPixelAt(source_picture, source_x, source_y))
            setColor(getPixelAt(new_picture, new_x, new_y), color)
            source_y = int(new_y * (1 / scale))
        source_x = int(new_x * (1 / scale))
    return new_picture

def blackAndWhite(source):
    newpic = source
    for pixel in getPixels(newpic):
        red_value = getRed(pixel)
        green_value = getGreen(pixel)
        blue_value = getBlue(pixel)
        luminance = (red_value + green_value + blue_value) // 3
        if (luminance < 50):
            setColor(pixel, black)
        elif (51 <= luminance <= 150):
            setColor(pixel, darkGray)
        elif (151 <= luminance <= 200):
            setColor(pixel, lightGray)
        else:
            setColor(pixel, white)
    return newpic

def maxColor(picture):
    for pixel in getPixels(picture):
        new_red = getBlue(pixel)
        new_green = getRed(pixel)
        new_blue = getGreen(pixel)
        lightval = (new_red + new_green + new_blue) // 3
        if (lightval < 240):
            if new_red > new_blue and new_red > new_green:
                setColor(pixel, makeColor(255, new_green, new_blue))

```

↓ indicates the line is continued on the next line.

```

        elif new_green > new_blue and new_green > new_red:
            setColor(pixel, makeColor(new_red,255, new_blue))
        elif new_blue > new_green and new_blue > new_red:
            setColor(pixel, makeColor(new_red,new_green, 255))

def Coloraverage(picture):
    for pixel in getPixels(picture):
        new_red = ((getRed(pixel) + 127) // 2)
        new_green = ((getGreen(pixel) + 181) // 2)
        new_blue = ((getBlue(pixel) + 63) // 2)
        lightval = (getRed(pixel) + getGreen(pixel) + getBlue(pixel) ) // 3
        if (lightval < 240):
            setColor(pixel, makeColor(new_red, new_green, new_blue))

def makeRed(picture):
    for pixel in getPixels(picture):
        lightValue = (getRed(pixel) + getGreen(pixel) + getBlue(pixel)) // 3
        if (lightValue < 240 ):
            new_red = ((getRed(pixel) + 127) * 2)
            new_green = ((getGreen(pixel) + 30) // 2)
            new_blue = ((getBlue(pixel) + 30) // 2)

            setColor(pixel, makeColor(new_red, new_green, new_blue))

def ColorSwap(picture):
    for pixel in getPixels(picture):
        new_red = getBlue(pixel)
        new_green = getRed(pixel)
        new_blue = getGreen(pixel)
        lightval = (new_red + new_green + new_blue) // 3
        if (lightval < 240):
            setColor(pixel, makeColor(new_red, new_green, new_blue))

def makelighter(picture):
    for x in range(0, getWidth(picture)):
        for y in range(0, getHeight(picture)):
            pixel = getPixelAt(picture,x,y)
            for i in range(0, 1):
                color = getColor(pixel)
                setColor(pixel, makeLighter(color))

```