Gage Hamilton

Completed



Original



```
from mediaComp import *

#This program will make a collage of an image of a lightning bolt,
#but many of the images will have unique edits.
#Written by Gage Hamilton on 10/19/26

def collage():
    pic = makePicture(getMediaFolder("Silver_City.jpg"))
    sign = makePicture(getMediaFolder("Signature.jpg"))
    pic = copyToNewPicture(pic)
    scaledPic = scale(pic, 1/3)
    mirrorVertical(pic)
    edgePic = duplicatePicture(scaledPic)
    edgeDetect(edgePic, 19)
    copyEditsToNewPicture(pic, edgePic, 25, 25)
    darkPic = duplicatePicture(scaledPic)
    darkPic = darkenPicture(darkPic)
    darkPic = darkenPicture(darkPic)
    copyEditsToNewPicture(pic, darkPic, 575, 25)
    posterPic = duplicatePicture(scaledPic)
    posterize(posterPic)
    copyEditsToNewPicture(pic, posterPic, 25, 350)
    cyanPic = duplicatePicture(scaledPic)
    cyanotype(cyanPic)
    copyEditsToNewPicture(pic, cyanPic, 575, 350)
    chromakey(sign, pic)
    copyEditsToNewPicture(pic, sign, 352, 476)
    pictureTool(pic)


#This function copies the source picture onto an empty one, as asked in the
#algorithm.
def copyToNewPicture(source_picture):
    canvas = makeEmptyPicture(900, 600)
    target_x = 0
    for source_x in range(0, getWidth(source_picture)):
        target_y = 0
        for source_y in range(0, getHeight(source_picture)):
            source_pixel = getPixelAt(source_picture, source_x, source_y)
            color = getColor(source_pixel)
            target_pixel = getPixelAt(canvas, target_x, target_y)
            setColor(target_pixel, color)
            target_y = target_y + 1
        target_x = target_x + 1
    return canvas
```

```
def scale(source_picture, factor):
    new_width = int(getWidth(source_picture) * factor)
    new_height = int(getHeight(source_picture) * factor)
    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, int(source_x),int(source_y)))
            setColor(getPixelAt(new_picture, new_x, new_y), color)
            source_y = source_y + 1/factor
        source_x = source_x + 1/factor
    return new_picture


def mirrorVertical(picture):
    width = getWidth(picture)
    mirror_point = width // 2
    for x in range(mirror_point, width):
        for y in range(0, getHeight(picture)):
            left_pixel = getPixelAt(picture, x, y)
            right_pixel = getPixelAt(picture, width - x - 1, y)
            color = getColor(left_pixel)
            setColor(right_pixel, color)


def edgeDetect(picture, threshold):
    for pixel in getPixels(picture):
        x = getX(pixel)
        y = getY(pixel)
        if y < getHeight(picture) - 1 and x < getWidth(picture) - 1:
            bottom_right_pixel = getPixelAt(picture, x+1, y+1)
            this_luminance = luminance(pixel)
            bottom_right_luminance = luminance(bottom_right_pixel)
            if abs(bottom_right_luminance - this_luminance) > threshold:
                setColor(pixel, blue)


def luminance(pixel):
    red_value = getRed(pixel)
    green_value = getGreen(pixel)
    blue_value = getBlue(pixel)
    return (red_value + green_value + blue_value) // 3


def copyEditsToNewPicture(source, edit_picture, x, y):
    target_x = x
    for source_x in range(0, getWidth(edit_picture)):
        target_y = y
        for source_y in range(0, getHeight(edit_picture)):
            source_pixel = getPixelAt(edit_picture, source_x, source_y)
            color = getColor(source_pixel)
            target_pixel = getPixelAt(source, target_x, target_y)
            setColor(target_pixel, color)
            target_y = target_y + 1
        target_x = target_x + 1
    return source
```

```
def darkenPicture(picture):
    for x in range(getWidth(picture)):
        for y in range(0, getHeight(picture)):
            pixel = getPixelAt(picture, x, y)
            color = getColor(pixel)
            color = makeDarker(color)
            color = makeDarker(color)
            setColor(pixel, color)
    return picture


def posterize(source):
    for pixel in getPixels(source):
        if luminance(pixel) < 50:
            setColor(pixel, black)
        elif 50 <= luminance(pixel) <= 150:
            setColor(pixel, darkGray)
        elif 151 <= luminance(pixel) <= 200:
            setColor(pixel, lightGray)
        else:
            setColor(pixel, white)


def cyanotype(pic):
    for pixel in getPixels(pic):
        intensity = (getRed(pixel) + getGreen(pixel) + getBlue(pixel)) // 3
        setColor(pixel, makeColor(intensity, intensity, intensity))

    for pixel in getPixels(pic):
        if getBlue(pixel) < 63:
            newBlue = getBlue(pixel) * 2
        elif 63 < getBlue(pixel) < 191:
            newBlue = getBlue(pixel) * 1.3
        else:
            newBlue = getBlue(pixel) * 1.2
        newRed = getRed(pixel) * 0.75
        newGreen = getGreen(pixel) * 0.75
        setColor(pixel, makeColor(newRed, newGreen, newBlue))


def chromakey(source, background):
    for source_pixel in getPixels(source):
        x = getX(source_pixel)
        y = getY(source_pixel)
        if (200 <= getRed(source_pixel) <= 255 and 200 <= getGreen(source_pixel) <=
255 and 200 <= getBlue(source_pixel) <= 255):
            background_pixel = getPixelAt(background, x + 352, y + 476)
            background_color = getColor(background_pixel)
            setColor(source_pixel, background_color)
```