Andrew Sutton

Completed

Original

```
# CS 120 - Project 2: Art Show
# Author: Andrew Sutton
# Date: October 24, 2025
#
# Description:
# This program creates a collage using the same image 5 times with different effects.
# It uses the original picture instead of a plain white canvas.
from mediaComp import *

def collage():
    setMediaFolder()

    # Make the base image to work around
    background = makePicture(getMediaFolder("anime-moon.jpg"))
    canvas = scale(background, min(1000/getWidth(background), 736/getHeight(background)))
    canvasWidth = getWidth(canvas)
    canvasHeight = getHeight(canvas)
    centerX = canvasWidth // 2
    centerY = canvasHeight // 2

    # Base image to repeat for all the modified images
    originalImage = "anime-moon.jpg"

    # The different effects for the images
    effects = ["grayscale", "mirror", "lighten", "colorshift"]

    # 4 changed images that circle the original
    radiusX = canvasWidth // 3
    radiusY = canvasHeight // 3
    positions = [(centerX + radiusX, centerY), (centerX, centerY - radiusY), (centerX - radiusX, centerY), ↓
            (centerX, centerY + radiusY)]
    for i in range(4):
        pic = makePicture(getMediaFolder(originalImage))
```

↓ indicates the line is continued on the next line.

```
        pic = copyPicture(pic)

        # Apply effects to each image
        effect = effects[i]
        if effect == "grayscale":
            pic = makeGrayscale(pic)
        elif effect == "mirror":
            pic = mirrorVertical(pic)
        elif effect == "lighten":
            pic = lighten(pic)
        elif effect == "colorshift":
            pic = colorShift(pic)

        # this can change the size of the smaller images
        pic = scale(pic, 0.45)

        # Get the position from table
        posX, posY = positions[i]
        x = posX - getWidth(pic) // 2
        y = posY - getHeight(pic) // 2

        # Copy onto the empty picture
        copyInto(pic, canvas, x, y)

    # Add my signature
    signature = makePicture(getMediaFolder("mysignature.png"))
    reverseChromaKey(canvas, signature, margin=20, scaleFactor=0.15)

    # Show the collage
    show(canvas)

def copyInto(src, dest, startX, startY):
    for x in range(getWidth(src)):
        for y in range(getHeight(src)):
            destX = startX + x
            destY = startY + y
            if 0 <= destX < getWidth(dest) and 0 <= destY < getHeight(dest):
                setColor(getPixelAt(dest, destX, destY), getColor(getPixelAt(src, x, y)))

def copyPicture(pic):
    width = getWidth(pic)
    height = getHeight(pic)
    newPic = makeEmptyPicture(width, height)
```

↓ indicates the line is continued on the next line.

```
    for x in range(width):
        for y in range(height):
            setColor(getPixelAt(newPic, x, y), getColor(getPixelAt(pic, x, y)))
    return newPic

def makeGrayscale(pic):
    for x in range(getWidth(pic)):
        for y in range(getHeight(pic)):
            p = getPixelAt(pic, x, y)
            avg = (getRed(p) + getGreen(p) + getBlue(p)) // 3
            setColor(p, makeColor(avg, avg, avg))
    return pic

def mirrorVertical(pic):
    w = getWidth(pic)
    h = getHeight(pic)
    for y in range(h):
        for x in range(w // 2):
            setColor(getPixelAt(pic, w - 1 - x, y), getColor(getPixelAt(pic, x, y)))
    return pic

def lighten(pic):
    for x in range(getWidth(pic)):
        for y in range(getHeight(pic)):
            setColor(getPixelAt(pic, x, y), makeLighter(getColor(getPixelAt(pic, x, y))))
    return pic

def colorShift(pic):
    for x in range(getWidth(pic)):
        for y in range(getHeight(pic)):
            p = getPixelAt(pic, x, y)
            setColor(p, makeColor(getGreen(p), getBlue(p), getRed(p)))
    return pic

def scale(pic, factor):
    width = int(getWidth(pic) * factor)
    height = int(getHeight(pic) * factor)
    newPic = makeEmptyPicture(width, height)
    for x in range(width):
        for y in range(height):
            srcX = int(x / factor)
            srcY = int(y / factor)
            setColor(getPixelAt(newPic, x, y), getColor(getPixelAt(pic, srcX, srcY)))
```

↓ indicates the line is continued on the next line.

```
    return newPic

def reverseChromaKey(bg, sign, margin=20, scaleFactor=0.3):
    smallSign = scale(copyPicture(sign), scaleFactor)
    sigWidth = getWidth(smallSign)
    sigHeight = getHeight(smallSign)
    startX = getWidth(bg) - sigWidth - margin
    startY = getHeight(bg) - sigHeight - margin

    for x in range(sigWidth):
        for y in range(sigHeight):
            px = getPixelAt(smallSign, x, y)
            r = getRed(px)
            g = getGreen(px)
            b = getBlue(px)
            if not (r > 240 and g > 240 and b > 240):
                setColor(getPixelAt(bg, startX + x, startY + y), getColor(px))
```

↓ indicates the line is continued on the next line.