David York

Completed

Original

```
from mediaComp import *

# David York
# 2/27/2026
# Project 2 - Art Show

def collage():
    soldierFile = getMediaFolder() + "soldier-sunset-silhouette-stockcake.jpg"
    signatureFile = getMediaFolder() + "signature.jpg"

    original = makePicture(soldierFile)
    canvas = makeEmptyPicture(1000, 736)

    base = scalePicture(original, 300, 168)

    originalPic = duplicatePicture(base)

    grayPic = duplicatePicture(base)
    grayscale(grayPic)

    warPic = duplicatePicture(base)
    warFilter(warPic)

    edgePic = duplicatePicture(base)
    edgeDetect(edgePic, 18)

    sepiaPic = duplicatePicture(base)
    sepia(sepiaPic)

    peacePic = duplicatePicture(base)
    peaceFilter(peacePic)

    splitPic = duplicatePicture(base)
    twoFuturesFilter(splitPic)
```

```
    bg = scalePicture(original, 1000, 736)
    darken(bg, 0.45)
    copyInto(bg, canvas, 0, 0)

    copyInto(grayPic, canvas, 40, 35)
    copyInto(warPic, canvas, 350, 35)
    copyInto(edgePic, canvas, 660, 35)

    centerX = (getWidth(canvas) - getWidth(originalPic)) // 2
    centerY = (getHeight(canvas) - getHeight(originalPic)) // 2
    copyInto(originalPic, canvas, centerX, centerY)

    copyInto(sepiaPic, canvas, 40, 533)
    copyInto(peacePic, canvas, 350, 533)
    copyInto(splitPic, canvas, 660, 533)

    signature = makePicture(signatureFile)
    signature = scalePicture(signature, 220, 70)
    reverseChromakey(signature, canvas, 750, 655)

    show(canvas)

def duplicatePicture(pic):
    newPic = makeEmptyPicture(getWidth(pic), getHeight(pic))
    for x in range(getWidth(pic)):
        for y in range(getHeight(pic)):
            oldPixel = getPixelAt(pic, x, y)
            newPixel = getPixelAt(newPic, x, y)
            setColor(newPixel, getColor(oldPixel))
    return newPic

def copyInto(source, target, startX, startY):
    for x in range(getWidth(source)):
        for y in range(getHeight(source)):
            targetX = startX + x
            targetY = startY + y

            if targetX >= 0 and targetX < getWidth(target) and targetY >= 0 and targetY < getHeight(target):
                sourcePixel = getPixelAt(source, x, y)
                targetPixel = getPixelAt(target, targetX, targetY)
                setColor(targetPixel, getColor(sourcePixel))
```

```
def scalePicture(pic, newWidth, newHeight):
    scaled = makeEmptyPicture(newWidth, newHeight)

    oldWidth = getWidth(pic)
    oldHeight = getHeight(pic)

    xScale = float(oldWidth) / newWidth
    yScale = float(oldHeight) / newHeight

    for x in range(newWidth):
        for y in range(newHeight):
            oldX = int(x * xScale)
            oldY = int(y * yScale)

            if oldX >= oldWidth:
                oldX = oldWidth - 1
            if oldY >= oldHeight:
                oldY = oldHeight - 1

            oldPixel = getPixelAt(pic, oldX, oldY)
            newPixel = getPixelAt(scaled, x, y)
            setColor(newPixel, getColor(oldPixel))

    return scaled

def grayscale(pic):
    for p in getPixels(pic):
        avg = int((getRed(p) + getGreen(p) + getBlue(p)) / 3)
        setRed(p, avg)
        setGreen(p, avg)
        setBlue(p, avg)

def darken(pic, factor):
    for p in getPixels(pic):
        setRed(p, int(getRed(p) * factor))
        setGreen(p, int(getGreen(p) * factor))
        setBlue(p, int(getBlue(p) * factor))

def sepia(pic):
    grayscale(pic)
    for p in getPixels(pic):
        red = getRed(p)
        green = getGreen(p)
        blue = getBlue(p)
```

```
        setRed(p, min(255, int(red * 1.25)))
        setGreen(p, min(255, int(green * 1.05)))
        setBlue(p, min(255, int(blue * 0.75)))

def posterizeValue(value):
    if value < 64:
        return 31
    elif value < 128:
        return 95
    elif value < 192:
        return 159
    else:
        return 223

def posterize(pic):
    for p in getPixels(pic):
        setRed(p, posterizeValue(getRed(p)))
        setGreen(p, posterizeValue(getGreen(p)))
        setBlue(p, posterizeValue(getBlue(p)))

def warFilter(pic):
    for p in getPixels(pic):
        r = getRed(p)
        g = getGreen(p)
        b = getBlue(p)

        r = min(255, int(r * 1.25))
        g = int(g * 0.65)
        b = int(b * 0.55)

        setRed(p, posterizeValue(r))
        setGreen(p, posterizeValue(g))
        setBlue(p, posterizeValue(b))

def peaceFilter(pic):
    for p in getPixels(pic):
        r = getRed(p)
        g = getGreen(p)
        b = getBlue(p)

        avg = int((r + g + b) / 3)

        setRed(p, min(255, avg + 45))
        setGreen(p, min(255, avg + 25))
        setBlue(p, min(255, avg + 10))
```

```
def colorDistance(pixel1, pixel2):
    r1 = getRed(pixel1)
    g1 = getGreen(pixel1)
    b1 = getBlue(pixel1)

    r2 = getRed(pixel2)
    g2 = getGreen(pixel2)
    b2 = getBlue(pixel2)

    return ((r1 - r2) ** 2 + (g1 - g2) ** 2 + (b1 - b2) ** 2) ** 0.5

def edgeDetect(pic, threshold):
    bw = duplicatePicture(pic)
    grayscale(bw)

    for x in range(getWidth(bw) - 1):
        for y in range(getHeight(bw) - 1):
            p = getPixelAt(bw, x, y)
            rightPixel = getPixelAt(bw, x + 1, y)

            if colorDistance(p, rightPixel) > threshold:
                setColor(getPixelAt(pic, x, y), black)
            else:
                setColor(getPixelAt(pic, x, y), white)

def twoFuturesFilter(pic):
    width = getWidth(pic)

    for x in range(width):
        for y in range(getHeight(pic)):
            p = getPixelAt(pic, x, y)
            r = getRed(p)
            g = getGreen(p)
            b = getBlue(p)

            if x < width / 2:
                setRed(p, min(255, int(r * 1.3)))
                setGreen(p, int(g * 0.55))
                setBlue(p, int(b * 0.55))
            else:
                setRed(p, min(255, int(r * 1.15 + 20)))
                setGreen(p, min(255, int(g * 1.05 + 10)))
                setBlue(p, min(255, int(b * 0.9)))
```

```
    mid = width // 2
    for y in range(getHeight(pic)):
        for offset in range(-2, 3):
            x = mid + offset
            if x >= 0 and x < getWidth(pic):
                p = getPixelAt(pic, x, y)
                setRed(p, min(255, getRed(p) + 40))
                setGreen(p, min(255, getGreen(p) + 40))
                setBlue(p, min(255, getBlue(p) + 40))

def reverseChromakey(signaturePic, collagePic, startX, startY):
    for x in range(getWidth(signaturePic)):
        for y in range(getHeight(signaturePic)):
            sigPixel = getPixelAt(signaturePic, x, y)
            targetX = startX + x
            targetY = startY + y

            if targetX >= 0 and targetX < getWidth(collagePic) and targetY >=0 and targetY <getHeight(collagePic):
                r = getRed(sigPixel)
                g = getGreen(sigPixel)
                b = getBlue(sigPixel)
                avg = (r + g + b) / 3

                if avg < 128:
                    targetPixel = getPixelAt(collagePic, targetX, targetY)
                    setRed(targetPixel, 255)
                    setGreen(targetPixel, 255)
                    setBlue(targetPixel, 255)
```