

# Derrick Williams

Completed



Originals



```
# Derrick
# CS120
# Project 2
```

```
from mediaComp import *
```

```
def scaleToWidth(pic, newWidth):
    oldWidth = getWidth(pic)
    oldHeight = getHeight(pic)
    scale = newWidth / oldWidth
    newHeight = int(oldHeight * scale)
    newPic = makeEmptyPicture(newWidth, newHeight)
    for x in range(newWidth):
        for y in range(newHeight):
            oldX = int(x / scale)
            oldY = int(y / scale)
            color = getColor(getPixelAt(pic, oldX, oldY))
            setColor(getPixelAt(newPic, x, y), color)
    return newPic

def darken(pic):
    for p in getPixels(pic):
        r = int(getRed(p) * 0.6)
        g = int(getGreen(p) * 0.6)
        b = int(getBlue(p) * 0.6)
        setColor(p, makeColor(r, g, b))
    return pic

def lighten(pic):
    for p in getPixels(pic):
        r = int(min(255, getRed(p) * 1.4))
        g = int(min(255, getGreen(p) * 1.4))
        b = int(min(255, getBlue(p) * 1.4))
        setColor(p, makeColor(r, g, b))
    return pic

def mirrorVertical(pic):
    width = getWidth(pic)
    height = getHeight(pic)
    newPic = makeEmptyPicture(width, height)
    for x in range(width):
        for y in range(height):
            sourcePixel = getPixelAt(pic, x, height - 1 - y)
            setColor(getPixelAt(newPic, x, y), getColor(sourcePixel))
    return newPic

def posterize(pic, levels=4):
    for p in getPixels(pic):
        r = getRed(p) // (256 // levels) * (256 // levels)
```

```

    g = getGreen(p) // (256 // levels) * (256 // levels)
    b = getBlue(p) // (256 // levels) * (256 // levels)
    setColor(p, makeColor(r, g, b))
return pic

def paste(pic, canvas, startX, startY):
    for x in range(getWidth(pic)):
        for y in range(getHeight(pic)):
            canvasX = startX + x
            canvasY = startY + y
            if canvasX >= getWidth(canvas) or canvasY >= getHeight(canvas):
                continue
            color = getColor(getPixelAt(pic, x, y))
            setColor(getPixelAt(canvas, canvasX, canvasY), color)

def pasteSkipWhite(pic, canvas, startX, startY):
    for x in range(getWidth(pic)):
        for y in range(getHeight(pic)):
            canvasX = startX + x
            canvasY = startY + y
            if canvasX >= getWidth(canvas) or canvasY >= getHeight(canvas):
                continue
            pixel = getPixelAt(pic, x, y)
            if getRed(pixel) > 240 and getGreen(pixel) > 240 and getBlue(pixel)>240:
                continue
            setColor(getPixelAt(canvas, canvasX, canvasY), getColor(pixel))

def drawBorder(canvas, thickness):
    w = getWidth(canvas)
    h = getHeight(canvas)

    for y in range(thickness):
        for x in range(w):
            setColor(getPixelAt(canvas, x, y), white)
            setColor(getPixelAt(canvas, x, h - 1 - y), white)

    for x in range(thickness):
        for y in range(h):
            setColor(getPixelAt(canvas, x, y), white)
            setColor(getPixelAt(canvas, w - 1 - x, y), white)

def collage():
    canvas = makeEmptyPicture(1000, 736)
    BORDER = 20
    SIDE_ROWS = 7

    original = makePicture(getMediaFolder() + "tre.jpg")

    big = scaleToWidth(original, 380)
    mirror = mirrorVertical(big)
    mirror = posterize(mirror, levels=4)
    midX = (1000 - getWidth(big)) // 2
    paste(big, canvas, midX, BORDER)
    paste(mirror, canvas, midX, BORDER + getHeight(big))

    availableHeight = 736 - 2 * BORDER
    sideHeight = availableHeight // SIDE_ROWS
    tempSide = scaleToWidth(original, 90)
    scaleFactor = sideHeight / getHeight(tempSide)
    sideWidth = int(getWidth(tempSide) * scaleFactor)

```

```
dark = darken(scaleToWidth(original, sideWidth))
light = lighten(scaleToWidth(original, sideWidth))

y = BORDER
for i in range(SIDE_ROWS):
    img = dark if i % 2 == 0 else light
    paste(img, canvas, BORDER, y)
    y += getHeight(img)

y = BORDER
for i in range(SIDE_ROWS):
    img = light if i % 2 == 0 else dark
    paste(img, canvas, 1000 - sideWidth - BORDER, y)
    y += getHeight(img)

sig = makePicture(getMediaFolder() + "sig.png")
sigSmall = scaleToWidth(sig, 150)
sigX = midX + (getWidth(big) - getWidth(sigSmall)) // 2
sigY = BORDER + getHeight(big) + getHeight(mirror)
pasteSkipWhite(sigSmall, canvas, sigX, sigY)

drawBorder(canvas, BORDER)

show(canvas)
```