

Jeremy Wagoner

Completed



Original



```
#JeremyWagoner
#CS120
#09/20/2021
#The original photo was taken by Simon Zhu and is free to download on unsplash.com

#Calling my main functions and assigning the picture to different values to modify
def collage():
    setMediaPath()
    original = makePicture(getMediaPath("cyberpunk.jpg"))
    mod1 = makePicture(getMediaPath("cyberpunk.jpg"))
    mod2 = makePicture(getMediaPath("cyberpunk.jpg"))
    mod3 = makePicture(getMediaPath("cyberpunk.jpg"))
    mod4 = makePicture(getMediaPath("cyberpunk.jpg"))
    canvas = makeEmptyPicture(getWidth(original) * 2, getHeight(original) * 2)
    sig = makePicture(getMediaPath("sig.jpg"))
    sig = scaleDown(sig, 2)

#Calling all my functions
    lighten(mod1)
    redtoblue(mod2)
    negative(mod3)
    grayscale(mod4)
    flip_rtb(mod2)
    flip_grayscale(mod4)
    mirror_negative(mod3)
    negative_border(mod3)
    original_border(original)
```

```

mod1_border(mod1)
mod2_border(mod2)
mod4_border(mod4)

#Copying the pictures to the right spots on the canvas
copy(original, canvas, 0, 0)
copy(mod1, canvas, getWidth(mod1), 0)
copy(mod2, canvas, 0, getHeight(mod2))
copy(mod4, canvas, getWidth(mod4), getHeight(mod4))
copy(mod3, canvas, int(getWidth(canvas)/2-getWidth(original)/2), int(getHeight(canvas)/2-getHeight(original)/2))
chromaSig(sig, canvas, 650, 900)
show(canvas)
writePictureTo(canvas, "jeremy_wagoner_project3.jpg")

#The copy function copies each image to their respected positions on the canvas
def copy(pic, target, targX, targY):
    targetX = targX
    for x in range(getWidth(pic)):
        targetY = targY
        for y in range(getHeight(pic)):
            pixel = getPixel(pic, x, y)
            tx = getPixel(target, targetX, targetY)
            setColor(tx, getColor(pixel))
            targetY = targetY + 1
            targetX = targetX + 1

#Lightening the top right picture
def lighten(pic):
    for each_pixel in getPixels(pic):
        color = getColor(each_pixel)
        color = makeLighter(makeLighter(makeLighter(color)))
        setColor(each_pixel, color)

#Changing the red to blue on the bottom left picture
def redtoblue(pic):
    endX = getWidth(pic) - 1
    endY = getHeight(pic) - 1
    startX = 0
    for targetX in range(0, endX):
        startY = 0
        for targetY in range(0, endY):
            red = getBlue(getPixel(pic, startX, (startY)))
            blue = getRed(getPixel(pic, startX, startY))
            green = getGreen(getPixel(pic, startX, startY))

```

```

    color2 = makeColor(red, green, blue)
    setColor(getPixel(pic, targetX, targetY), color2)
    startY = startY + 1
startX = startX + 1

#Getting the negative of the middle picture
def negative(pic):
    for each_pixel in getPixels(pic):
        r = getRed(each_pixel)
        b = getBlue(each_pixel)
        g = getGreen(each_pixel)
        neg = makeColor(255-r, 255-g, 255-b)
        setColor(each_pixel, neg)

#Making the bottom right image black and white
def grayscale(pic):
    for p in getPixels(pic):
        intensity = (getRed(p) + getGreen(p) + getBlue(p)) / 3
        setColor(p, makeColor(intensity, intensity, intensity))

#Grabbing just my signature from the signature picture
def chromaSig(source, target, targetX, targetY):
    for x in range(0, getWidth(source)):
        for y in range(0, getHeight(source)):
            px = getPixel(source, x, y)
            color = getColor(px)
            targ = getPixel(target, x + targetX, y + targetY)
            if distance (black, color) < 200:
                setColor(targ, orange)

#Flipping the redtoblue modification 180 degrees
def flip_rtb(pic):
    width = getWidth(pic)
    height = getHeight(pic)
    for y in range(0, height / 2):
        for x in range(0, width):
            p1 = getPixel(pic, x, y)
            p2 = getPixel(pic, x, height - y - 1)
            color = getColor(p1)
            setColor(p1, getColor(p2))
            setColor(p2, color)

#Flipping the black and white modification 180 degrees
def flip_grayscale(pic):
    width = getWidth(pic)

```

```

height = getHeight(pic)
for y in range(0, height / 2):
    for x in range(0, width):
        p1 = getPixel(pic, x, y)
        p2 = getPixel(pic, x, height - y - 1)
        color = getColor(p1)
        setColor(p1, getColor(p2))
        setColor(p2, color)

#Mirroring the middle picture horizontally
def mirror_negative(pic):
    mirrorPoint = getHeight(pic) / 2
    height = getHeight(pic)
    for x in range(0, getWidth(pic)):
        for y in range(0, mirrorPoint):
            topPixel = getPixel(pic, x, y)
            bottomPixel = getPixel(pic, x, height - y - 1)
            color = getColor(topPixel)
            setColor(bottomPixel, color)

#Adding a black border around the middle picture
def negative_border(pic):
    bottom = getHeight(pic) - 10
    side = getWidth(pic) - 10
    for px in getPixels(pic):
        y = getY(px)
        x = getX(px)
        if y < 10:
            setColor(px, black)
        if y > bottom:
            setColor(px, black)
        if x < 10:
            setColor(px, black)
        if x > side:
            setColor(px, black)

#Adding pink borders around the whole canvas
def original_border(source):
    bottom = getHeight(source) - 10
    sides = getWidth(source) - 10
    for px in getPixels(source):
        y = getY(px)
        x = getX(px)
        if y < 10:
            setColor(px, black)

```

```

    if x < 10:
        setColor(px, black)

def mod1_border(source):
    bottom = getHeight(source) - 10
    sides = getWidth(source) - 10
    for px in getPixels(source):
        y = getY(px)
        x = getX(px)
        if y < 10:
            setColor(px, black)
        if x > sides:
            setColor(px, black)

def mod2_border(source):
    bottom = getHeight(source) - 10
    sides = getWidth(source) - 10
    for px in getPixels(source):
        y = getY(px)
        x = getX(px)
        if y > bottom:
            setColor(px, black)
        if x < 10:
            setColor(px, black)

def mod4_border(source):
    bottom = getHeight(source) - 10
    sides = getWidth(source) - 10
    for px in getPixels(source):
        y = getY(px)
        x = getX(px)
        if y > bottom:
            setColor(px, black)
        if x > sides:
            setColor(px, black)

#Scaling down my signature
def scaleDown(pic, factor):
    canvas = makeEmptyPicture(int(getWidth(pic) / factor), int(getHeight(pic) / factor))
    scale(pic, canvas, 1.0 / factor)
    return canvas

#Scaling the collage (scale function was called in the scaleDown function)
def scale(src, canvas, factor):
    sourceX = 0

```

```
for targetX in range(0, int(getWidth(src) * factor)):
    sourceY = 0
    for targetY in range(0, int(getHeight(src) * factor)):
        color = getColor(getPixel(src, int(sourceX), int(sourceY)))
        setColor(getPixel(canvas, targetX, targetY), color)
        sourceY = sourceY + 1.0 / factor
    sourceX = sourceX + 1.0 / factor

collage()
```