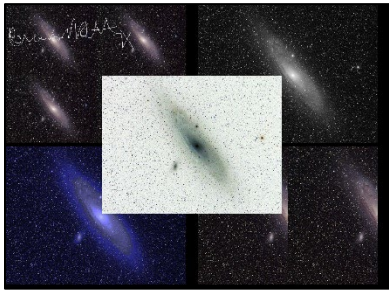


Brennon Mattox

Completed

Original



```
#Brennon Mattox 3/14/2022
```

```
setMediaPath()
```

```
def collage(): #combines all changed images int a collage on an originally blank canvas
    canvas = makeEmptyPicture(1000, 736, black)
    source = makePicture(getMediaPath("galaxy.jpg"))
    signature = makePicture(getMediaPath("signature.jpg"))
    copy(canvas, scale(source, .5), 0, 0) #creates one of the four smaller versions, following lines create rest
    copy(canvas, scale(source, .5), int(getWidth(source) / 2), 0)
    copy(canvas, scale(source, .5), int(getWidth(source) / 2), int(getHeight(source) / 2))
    copy(canvas, scale(source, .5), 0, int(getHeight(source) / 2))
    copy(canvas, cyanotype_toning(source), 0, getHeight(canvas) / 2) #creates bluer image
    copy(canvas, grey_scale(source), getWidth(canvas) / 2, 0) #creates grey image
    copy(canvas, mirror_vertical(source), getWidth(canvas) / 2, getHeight(canvas) / 2) #creates mirrored image
    copy(canvas, invert_colors(source), getWidth(canvas) / 4, getHeight(canvas) / 4) #creates inverted colored image
    add_signature(canvas, invert_colors(signature)) #creates signature
    show(canvas) #shows collage

def copy(canvas, source, startx, starty): #copies changed source image to the canvas
    for x in range(0, getWidth(source)):
        for y in range(0, getHeight(source)):
            canvas_pixel = getPixel(canvas, x + startx, y + starty)
            source_pixel = getPixel(source, x, y)
            color = getColor(source_pixel)
            setColor(canvas_pixel, color)

def scale(picture_in, scale):
    picture_out = makeEmptyPicture(int(getWidth(picture_in) * scale), int(getHeight(picture_in) * scale))
```

```

sourcecx = 0
for targetx in range(0, int(getWidth(picture_in) * (scale))):
    sourcecy = 0
    for targety in range(0, int(getHeight(picture_in) * (scale))):
        color = getColor(getPixel(picture_in, int(sourcecx), int(sourcecy)))
        setColor(getPixel(picture_out, targetx, targety), color)
        sourcecy = sourcecy + (1.0 / scale)
        sourcecx = sourcecx + (1.0 / scale)
return(picture_out)

def blur(picture): #blurs the source image
    target = duplicatePicture(picture)
    for x in range(1, getWidth(picture) - 1):
        for y in range(1, getHeight(picture) - 1):
            top = getPixel(picture, x, y - 1)
            bot = getPixel(picture, x, y + 1)
            left = getPixel(picture, x - 1, y)
            right = getPixel(picture, x + 1, y)
            center = getPixel(picture, x, y)
            new_green = (getGreen(top) + getGreen(bot) + getGreen(left) + getGreen(right) + getGreen(center)) / 5
            new_blue = (getBlue(top) + getBlue(bot) + getBlue(left) + getBlue(right) + getBlue(center)) / 5
            new_red = (getRed(top) + getRed(bot) + getRed(left) + getBlue(right) + getBlue(center)) / 5
            setColor(center, makeColor(new_red, new_green, new_blue))
    return(target)

def grey_scale(source): #turns the source into a grey-scaled image
    picture = duplicatePicture(source)
    for px in getPixels(picture): #creates and changes the pixel into the grey shade
        new_red = getRed(px) * .299
        new_blue = getBlue(px) * .114
        new_green = getGreen(px) * .587
        grey_shade = new_red + new_green + new_blue
        setColor(px, makeColor(grey_shade, grey_shade, grey_shade))
    return(picture)

def cyanotype_toning(source): #makes the picture much more blue
    picture = grey_scale(source)
    for px in getPixels(picture): #blue-ifies the pixels
        r = getRed(px)
        g = getGreen(px)
        b = getBlue(px)
        if b < 63:
            b = b * 2
        if (63 <= b <= 191):
            b = b * 1.3

```

```

    if (b > 191):
        b = b * 1.2
    r = r * .75
    g = g * .75
    setBlue(px, b)
    setRed(px, r)
    setGreen(px, g)
    return (picture)

def mirror_vertical(source): #mirrors source virtically down the middle
    picture = duplicatePicture(source)
    for x in range(0, getWidth(picture) / 2):
        for y in range(0, getHeight(picture)):
            color = getColor(getPixel(picture, x, y))
            setColor(getPixel(picture, x + (getWidth(picture) / 2), y), color)
    return (picture)

def invert_colors(source): #inverts all colors in the source
    picture = duplicatePicture(source)
    for p in getPixels(picture): #grabs all rgb values and inverts them
        setRed(p, 255 - getRed(p))
        setBlue(p, 255 - getBlue(p))
        setGreen(p, 255 - getGreen(p))
    return (picture)

def add_signature(canvas, signature): #adds the signature onto the top left-hand corner of the collage
    for x in range(0, getWidth(signature)):
        for y in range(0, getHeight(signature)):
            canvasp = getPixel(canvas, x, y)
            sigp = getPixel(signature, x, y)
            if getBlue(sigp) > 100: #ensures the only pixels going onto the collage are the signature and not background
                setColor(canvasp, getColor(sigp))

```