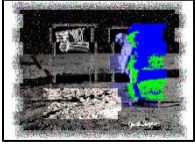


Jacob Shopher

Completed



Original



```
# Jacob Shopher, 10/21/24
# This takes a while to run, please bear with it.
```

```
from jes4py import *
import random

def collage():
    mainPic = makeEmptyPicture(1000, 736, white)
    moonLandingPic = makePicture(getMediaPath("moonlanding.jpg"))
    glitchMain(mainPic, 90, 910, 48, 687, 10)
    moonLanding(mainPic, moonLandingPic)
    glitchPost(mainPic, 90, 910, 48, 687)
    glitchMain(mainPic, 261, 467, 138, 510, 4)
    grayScale(mainPic, moonLandingPic, true)
    glitchMain(mainPic, 569, 802, 152, 565, 4)
    cyanotypeTint(mainPic, moonLandingPic)
    posterize(mainPic, moonLandingPic)
    lighten(mainPic)
    finalPic = offset(mainPic)
    chromakey(finalPic)
    explore(finalPic)

def moonLanding(mainPic, moonLandingPic):
    # Places the moon landing picture in the center of the main picture.
    target_x = 91
    for source_x in range(0, getWidth(moonLandingPic)):
        target_y = 49
        for source_y in range(0, getHeight(moonLandingPic)):
            moonLandingPixel = getPixel(moonLandingPic, source_x, source_y)
            color = getColor(moonLandingPixel)
            mainPicPixel = getPixel(mainPic, target_x, target_y)
            setColor(mainPicPixel, color)
            target_y = target_y + 1
        target_x = target_x + 1
```

```

def glitchMain(mainPic, startX, endX, startY, endY, width):
    # Runs four different layers of randomized pixels layered on top of eachother to create a glitch effect
    # Each layer gets progressively smaller in size depending on the width value
    glitchFirstLayer(mainPic, startX-(width*4), endX+(width*4), startY-(width*4), endY+(width*4))
    glitchSecondLayer(mainPic, startX-(width*3), endX+(width*3), startY-(width*3), endY+(width*3))
    glitchThirdLayer(mainPic, startX-(width*2), endX+(width*2), startY-(width*2), endY+(width*2))
    glitchFourthLayer(mainPic, startX-width, endX+width, startY-width, endY+width)

def glitchFirstLayer(mainPic, startX, endX, startY, endY):
    # Replaces 4/16 pixels randomly
    for pixel in getPixels(mainPic):
        x = getX(pixel)
        y = getY(pixel)
        if startX <= x <= endX:
            if startY <= y <= endY:
                factor = random.randint(1,16)
                if factor == 4:
                    setColor(pixel, green)
                elif factor == 3:
                    setColor(pixel, red)
                elif factor == 2:
                    setColor(pixel, cyan)
                elif factor == 1:
                    setColor(pixel, magenta)

def glitchSecondLayer(mainPic, startX, endX, startY, endY):
    # Replaces 4/12 pixels randomly
    for pixel in getPixels(mainPic):
        x = getX(pixel)
        y = getY(pixel)
        if startX <= x <= endX:
            if startY <= y <= endY:
                factor = random.randint(1,12)
                if factor == 4:
                    setColor(pixel, green)
                elif factor == 3:
                    setColor(pixel, red)
                elif factor == 2:
                    setColor(pixel, cyan)
                elif factor == 1:
                    setColor(pixel, magenta)

```

```

def glitchThirdLayer(mainPic, startX, endX, startY, endY):
    # Replaces 4/8 pixels randomly
    for pixel in getPixels(mainPic):
        x = getX(pixel)
        y = getY(pixel)
        if startX <= x <= endX:
            if startY <= y <= endY:
                factor = random.randint(1,8)
                if factor == 4:
                    setColor(pixel, green)
                elif factor == 3:
                    setColor(pixel, red)
                elif factor == 2:
                    setColor(pixel, cyan)
                elif factor == 1:
                    setColor(pixel, magenta)

def glitchFourthLayer(mainPic, startX, endX, startY, endY):
    # Replaces 4/6 pixels randomly
    for pixel in getPixels(mainPic):
        x = getX(pixel)
        y = getY(pixel)
        if startX <= x <= endX:
            if startY <= y <= endY:
                factor = random.randint(1,6)
                if factor == 4:
                    setColor(pixel, green)
                elif factor == 3:
                    setColor(pixel, red)
                elif factor == 2:
                    setColor(pixel, cyan)
                elif factor == 1:
                    setColor(pixel, magenta)

def glitchPost(mainPic, startX, endX, startY, endY):
    # Replaces 4/96 pixels randomly
    for pixel in getPixels(mainPic):
        x = getX(pixel)
        y = getY(pixel)
        if startX <= x <= endX:
            if startY <= y <= endY:
                factor = random.randint(1,96)
                if factor == 4:
                    setColor(pixel, green)
                elif factor == 3:

```

```

        setColor(pixel, red)
    elif factor == 2:
        setColor(pixel, cyan)
    elif factor == 1:
        setColor(pixel, magenta)

def grayScale(mainPic, moonLandingPic, applyToMain):
    # If applyToMain is true, places the grayscale pixels into the main picture
    # If applyToMain is anything else, returns the moonLandingPhoto grayscaled
    for pixel in getPixels(moonLandingPic):
        new_red = getRed(pixel) * 0.299
        new_green = getGreen(pixel) * 0.587
        new_blue = getBlue(pixel) * 0.114
        intensity = new_red + new_green + new_blue
        setColor(pixel, makeColor(intensity, intensity, intensity))
    if applyToMain == true:
        target_x = 261
        for source_x in range(170, 376):
            target_y = 138
            for source_y in range(89, 461):
                moonLandingPixel = getPixel(moonLandingPic, source_x, source_y)
                color = getColor(moonLandingPixel)
                mainPicPixel = getPixel(mainPic, target_x, target_y)
                setColor(mainPicPixel, color)
                target_y = target_y + 1
            target_x = target_x + 1
    else:
        return moonLandingPic

def cyanotypeTint(mainPic, moonLandingPic):
    grayScale(mainPic, moonLandingPic, false)
    for pixel in getPixels(moonLandingPic):
        blueValue = getBlue(pixel)
        greenValue = getGreen(pixel) * .75
        redValue = getRed(pixel) * .75
        if blueValue < 63:
            blueValue = blueValue * 2
        elif blueValue > 191:
            blueValue = blueValue * 1.2
        else:
            blueValue = blueValue * 1.3
        setRed(pixel, redValue)
        setGreen(pixel, greenValue)
        setBlue(pixel, blueValue)
    target_x = 569

```

```

for source_x in range(478, 711):
    target_y = 152
    for source_y in range(103, 516):
        moonLandingPixel = getPixel(moonLandingPic, source_x, source_y)
        color = getColor(moonLandingPixel)
        mainPicPixel = getPixel(mainPic, target_x, target_y)
        setColor(mainPicPixel, color)
        target_y = target_y + 1
    target_x = target_x + 1

def posterize(mainPic, moonLandingPic):
    for pixel in getPixels(moonLandingPic):
        red_value = getRed(pixel)
        green_value = getGreen(pixel)
        blue_value = getBlue(pixel)
        luminance = (red_value + green_value + blue_value) // 3
        if luminance > 100:
            setColor(pixel, green)
        else:
            setColor(pixel, blue)
    target_x = 663
    for source_x in range(572, 756):
        target_y = 125
        for source_y in range(76, 472):
            moonLandingPixel = getPixel(moonLandingPic, source_x, source_y)
            color = getColor(moonLandingPixel)
            mainPicPixel = getPixel(mainPic, target_x, target_y)
            setColor(mainPicPixel, color)
            target_y = target_y + 1
        target_x = target_x + 1

def lighten(mainPic):
    moonLandingPic = makePicture(getMediaPath("moonlanding.jpg"))
    for pixel in getPixels(moonLandingPic):
        color = getColor(pixel)
        color = makeLighter(color)
        color = makeLighter(color)
        setColor(pixel, color)
    target_x = 219
    for source_x in range(128, 518):
        target_y = 466
        for source_y in range(417, 577):
            moonLandingPixel = getPixel(moonLandingPic, source_x, source_y)
            color = getColor(moonLandingPixel)
            mainPicPixel = getPixel(mainPic, target_x, target_y)

```

```

        setColor(mainPicPixel, color)
        target_y = target_y + 1
    target_x = target_x + 1

def offset(mainPic):
    # Offsets the picture by offsetAmount shifting it either right or left alternating
    finalPic = makeEmptyPicture(1000, 736, white)
    barHeight = 73
    offsetAmount = 10
    for counter in range(0,9,2):
        target_x = 0
        for source_x in range(offsetAmount, 1000):
            target_y = barHeight*counter
            for source_y in range(barHeight*counter, barHeight*(counter+1)):
                mainPicPixel = getPixel(mainPic, source_x, source_y)
                color = getColor(mainPicPixel)
                finalPicPixel = getPixel(finalPic, target_x, target_y)
                setColor(finalPicPixel, color)
                target_y = target_y + 1
            target_x = target_x + 1
    for counter in range(1,10,2):
        target_x = offsetAmount
        for source_x in range(0, 999-offsetAmount):
            target_y = barHeight*counter
            for source_y in range(barHeight*counter, barHeight*(counter+1)):
                mainPicPixel = getPixel(mainPic, source_x, source_y)
                color = getColor(mainPicPixel)
                finalPicPixel = getPixel(finalPic, target_x, target_y)
                setColor(finalPicPixel, color)
                target_y = target_y + 1
            target_x = target_x + 1
    return finalPic

def chromakey(finalPic):
    signaturePic = makePicture(getMediaPath("signature.png"))
    target_x = 631
    for source_x in range(0, getWidth(signaturePic)):
        target_y = 606
        for source_y in range(0, getHeight(signaturePic)):
            signaturePicPixel = getPixel(signaturePic, source_x, source_y)
            totalPixelRGB = getRed(signaturePicPixel) + getBlue(signaturePicPixel) + getGreen(signaturePicPixel)
            if totalPixelRGB > 400:
                color = getColor(signaturePicPixel)
                color = makeLighter(color)
                finalPicPixel = getPixel(finalPic, target_x, target_y)

```

```
        setColor(finalPicPixel, color)
    target_y = target_y + 1
    target_x = target_x + 1
```