Beth Martin

Completed

Originals





```
from jes4py import *
from random import randint
#Made by Beth Martin, 4/1/25

def signatureSize(signature, factor):
    signatureBack = makeEmptyPicture(getWidth(signature) * factor, →
      getHeight(signature) * factor)
    picX = 0
    for newX in range(0, getWidth(signatureBack)):
        picY = 0
        for newY in range(0, getHeight(signatureBack)):
            color = getColor(getPixel(signature, int(picX), int(picY)))
            setColor(getPixel(signatureBack, newX, newY), color)
            picY = picY + (1 / factor)
        picX = picX + (1 / factor)
    return signatureBack

def waterBackground(water, factor):
    waterBack = makeEmptyPicture(getWidth(water) * factor, getHeight(water) * factor)
    picX = 0
    for newX in range(0, getWidth(waterBack)):
        picY = 0
        for newY in range(0, getHeight(waterBack)):
            color = getColor(getPixel(water, int(picX), int(picY)))
            setColor(getPixel(waterBack, newX, newY), color)
            picY = picY + (1 / factor)
        picX = picX + (1 / factor)
    return waterBack

def Planet2(planet2, factor):
    planetBack = makeEmptyPicture(getWidth(planet2) * factor, →
      getHeight(planet2) * factor)
    picX = 0
    for newX in range(0, getWidth(planetBack)):
        picY = 0
        for newY in range(0, getHeight(planetBack)):
            color = getColor(getPixel(planet2, int(picX), int(picY)))
            setColor(getPixel(planetBack, newX, newY), color)
            picY = picY + (1 / factor)
        picX = picX + (1 / factor)
    return planetBack

def mountainYellow(MountainBack, SpaceBack):
    for source_pixel in getPixels(MountainBack):
        x = getX(source_pixel)
        y = getY(source_pixel)
        if (getRed(source_pixel) > 153 and getGreen(source_pixel) > 112 and →
           getBlue(source_pixel) < 4):
```

→ at the end of a line of code means that it is continued on the next line.

```
            background_pixel = getPixel(SpaceBack, x, y)
            background_color = getColor(background_pixel)
            setColor(source_pixel, background_color)
    return MountainBack

def waterOnSpace(BothPhoto, WaterBack):
    height = getHeight(BothPhoto)//2
    for x in range(0, getWidth(BothPhoto), 2):
        for y in range(height, getHeight(BothPhoto), 2):
            pixel = getPixel(WaterBack, x, y)
            color = getColor(pixel)
            newPixel = getPixel(BothPhoto, x, y)
            setColor(newPixel, color)
    for x in range(1, getWidth(BothPhoto), 2):
        for y in range(height+1, getHeight(BothPhoto), 2):
            pixel = getPixel(WaterBack, x, y)
            color = getColor(pixel)
            newPixel = getPixel(BothPhoto, x, y)
            setColor(newPixel, color)
    return BothPhoto

def mountainYellowTop(MountainBackTop, WaterOnSpace):
    for source_pixel in getPixels(MountainBackTop):
        x = getX(source_pixel)
        y = getY(source_pixel)
        if (getRed(source_pixel) > 153 and getGreen(source_pixel) > 112 and →
             getBlue(source_pixel) < 4):
            background_pixel = getPixel(WaterOnSpace, x, y)
            background_color = getColor(background_pixel)
            setColor(source_pixel, background_color)
    return MountainBackTop

def Planet2RemoveNPlace(Photo1, Photo2, x, y):
    modPhoto2 = duplicatePicture(Photo2)
    for source_pixel in getPixels(modPhoto2):
        x_source = getX(source_pixel)
        y_source = getY(source_pixel)
        if (getRed(source_pixel) <= 20 and getGreen(source_pixel) <= 20 and →
           getBlue(source_pixel) <= 20):
            # Offset by (x, y)
            background_pixel = getPixel(Photo1, x_source + x, y_source + y)
            background_color = getColor(background_pixel)
            setColor(source_pixel, background_color)
    for small_x in range(getWidth(modPhoto2)):
        for small_y in range(getHeight(modPhoto2)):
            color = getColor(getPixel(modPhoto2, small_x, small_y))
            setColor(getPixel(Photo1, x + small_x, y + small_y), color)

def blur(picture):
    blur_picture = duplicatePicture(picture)
    for x in range(1, getWidth(picture) - 1):
        for y in range(1, getHeight(picture) - 1):
            center = getPixel(blur_picture, x, y)
            left = getPixel(picture, x -1, y)
            right = getPixel(picture, x +1, y)
            top = getPixel(picture, x, y -1)
            bottom = getPixel(picture, x, y + 1)
            new_red = (getRed(center) + getRed(left) + getRed(right) +  →
                 getRed(top)+ getRed(bottom)) // 10
```

→ at the end of a line of code means that it is continued on the next line.

```
            new_green = (getGreen(center) + getGreen(left) + getGreen(right) + →
                 getGreen(top) + getGreen(bottom)) // 10
            new_blue = (getBlue(center) + getBlue(left) + getBlue(right) + →
                 getBlue(top) + getBlue(bottom)) // 10
            setColor(center, makeColor(new_red, new_green, new_blue))
    return blur_picture

def lightenPic(picture):
    for x in range(getWidth(picture)):
        for y in range(getHeight(picture)):
            pixel = getPixel(picture, x, y)
            color = getColor(pixel)
            for i in range(0, 2):
                color = makeLighter(color)
            setColor(pixel, color)
    return picture

def darkenPic(picture):
    for pixel in getPixels(picture):
        getWidth(picture)
        color = getColor(pixel)
        for i in range(0, 1):
            color = makeDarker(color)
        setColor(pixel, color)

def addBorderRed(picture):
    border_thickness = 12
    bottom = getHeight(picture) - border_thickness
    side = getWidth(picture) - border_thickness
    for pixel in getPixels(picture):
        y = getY(pixel)
        if y < border_thickness:
            setColor(pixel, red)
        elif y >= bottom:
            setColor(pixel, red)
    for pixel in getPixels(picture):
        x = getX(pixel)
        if x < border_thickness:
            setColor(pixel, red)
        elif x >= side:
            setColor(pixel, red)
    return picture

def addBorderBlack(picture):
    border_thickness = 10
    bottom = getHeight(picture) - border_thickness
    side = getWidth(picture) - border_thickness
    for pixel in getPixels(picture):
        y = getY(pixel)
        if y < border_thickness:
            setColor(pixel, black)
        elif y >= bottom:
            setColor(pixel, black)
    for pixel in getPixels(picture):
        x = getX(pixel)
        if x < border_thickness:
            setColor(pixel, black)
        elif x >= side:
            setColor(pixel, black)
```

→ at the end of a line of code means that it is continued on the next line.

```
    return picture

def addGlitchBorderBottom(picture):
    border_thickness = 25
    bottom = getHeight(picture) - border_thickness
    side = getWidth(picture) - border_thickness
    for pixel in getPixels(picture):
        x = getX(pixel)
        y = getY(pixel)
        if y < border_thickness or y >= bottom:
            random_color = makeColor(random.randint(0, 255), →
                 random.randint(0, 255), random.randint(0, 255))
            setColor(pixel, random_color)
        elif x < border_thickness or x >= side:
            random_color = makeColor(random.randint(0, 255), →
                 random.randint(0, 255), random.randint(0, 255))
            setColor(pixel, random_color)
    return picture

def addGlitchBorderTop(picture):
    border_thickness = 4
    bottom = getHeight(picture) - border_thickness
    side = getWidth(picture) - border_thickness
    for pixel in getPixels(picture):
        x = getX(pixel)
        y = getY(pixel)
        if y < border_thickness or y >= bottom:
            random_color = makeColor(random.randint(0, 255), →
                 random.randint(0, 255), random.randint(0, 255))
            setColor(pixel, random_color)
        elif x < border_thickness or x >= side:
            random_color = makeColor(random.randint(0, 255), →
                 random.randint(0, 255), random.randint(0, 255))
            setColor(pixel, random_color)
    return picture

def signaturePlace(Photo1, Photo2, x, y):
    modPhoto2 = duplicatePicture(Photo2)
    for source_pixel in getPixels(modPhoto2):
        x_source = getX(source_pixel)
        y_source = getY(source_pixel)
        if (getRed(source_pixel) <= 20 and getGreen(source_pixel) <= 20 and →
           getBlue(source_pixel) <= 20):
            # Offset by (x, y)
            background_pixel = getPixel(Photo1, x_source + x, y_source + y)
            background_color = getColor(background_pixel)
            setColor(source_pixel, background_color)
    for small_x in range(getWidth(modPhoto2)):
        for small_y in range(getHeight(modPhoto2)):
            color = getColor(getPixel(modPhoto2, small_x, small_y))
            setColor(getPixel(Photo1, x + small_x, y + small_y), color)
    return Photo1

def blueTint(picture):
    for pixel in getPixels(picture):
        red_value = getRed(pixel)
        blue_value = getBlue(pixel)
        if red_value < 63:
            red_value = red_value * 0.8
```

→ at the end of a line of code means that it is continued on the next line.

```
            blue_value = blue_value * 1.2
        elif red_value < 192:
            red_value = red_value * 0.75
            blue_value = blue_value * 1.3
        else:
            red_value = red_value * 0.85
            if red_value > 255:
                red_value = 255
            blue_value = blue_value * 1.15
        setRed(pixel, red_value)
        setBlue(pixel, blue_value)

def purpleTint(picture):
    for pixel in getPixels(picture):
        red_value = getRed(pixel)
        blue_value = getBlue(pixel)
        if red_value < 63:
            red_value = red_value * 1.1
            blue_value = blue_value * 1.1
        elif red_value < 192:
            red_value = red_value * 1.15
            blue_value = blue_value * 1.15
        else:
            red_value = red_value * 1.08
            if red_value > 255:
                red_value = 255
            blue_value = blue_value * 1.1
        setRed(pixel, red_value)
        setBlue(pixel, blue_value)

def grayScale(picture):
    for pixel in getPixels(picture):
        intensity = (getRed(pixel) + getGreen(pixel) + getBlue(pixel)) / 3
        setColor(pixel, makeColor(intensity, intensity, intensity))

def spaceBackground(space, new_width, new_height):
    spaceBack = makeEmptyPicture(new_width, new_height)
    scale_x = getWidth(space) / new_width
    scale_y = getHeight(space) / new_height
    picX = 0
    for newX in range(0, new_width):
        picY = 0
        for newY in range(0, new_height):
            origX = int(picX * scale_x)
            origY = int(picY * scale_y)
            color = getColor(getPixel(space, origX, origY))
            setColor(getPixel(spaceBack, newX, newY), color)
            picY += 1
        picX += 1
    height = getHeight(spaceBack)
    mirror_point = height // 2
    for y in range(0, mirror_point):
        for x in range(0, getWidth(spaceBack)):
            top_pixel = getPixel(spaceBack, x, y)
            bottom_pixel = getPixel(spaceBack, x, height - y - 1)
            color = getColor(top_pixel)
            setColor(bottom_pixel, color)
    return spaceBack
```

→ at the end of a line of code means that it is continued on the next line.

```
def mountainSize(mountain, new_width, new_height):
    mountainBack = makeEmptyPicture(new_width, new_height)
    scale_x = getWidth(mountain) / new_width
    scale_y = getHeight(mountain) / new_height
    picX = 0
    for newX in range(0, new_width):
        picY = 0
        for newY in range(0, new_height):
            origX = int(picX * scale_x)
            origY = int(picY * scale_y)
            color = getColor(getPixel(mountain, origX, origY))
            setColor(getPixel(mountainBack, newX, newY), color)
            picY += 1
        picX += 1
    height = getHeight(mountainBack)
    mirror_point = height // 2
    for y in range(0, mirror_point):
        for x in range(0, getWidth(mountainBack)):
            top_pixel = getPixel(mountainBack, x, y)
            bottom_pixel = getPixel(mountainBack, x, height - y - 1)
            color = getColor(top_pixel)
            setColor(bottom_pixel, color)
    return mountainBack

def mountainSizeTop(mountainTop, new_width, new_height):
    mountainBack = makeEmptyPicture(new_width, new_height)
    scale_x = getWidth(mountainTop) / new_width
    scale_y = getHeight(mountainTop) / new_height
    picX = 0
    for newX in range(0, new_width):
        picY = 0
        for newY in range(0, new_height):
            origX = int(picX * scale_x)
            origY = int(picY * scale_y)
            color = getColor(getPixel(mountainTop, origX, origY))
            setColor(getPixel(mountainBack, newX, newY), color)
            picY += 1
        picX += 1
    height = getHeight(mountainBack)
    mirror_point = height // 2
    for y in range(0, mirror_point):
        for x in range(0, getWidth(mountainBack)):
            top_pixel = getPixel(mountainBack, x, y)
            bottom_pixel = getPixel(mountainBack, x, height - y - 1)
            color = getColor(top_pixel)
            setColor(bottom_pixel, color)
    return mountainBack

#-------------------------------------------------------------------------------------
# This has a combonation of things including GreyScale, Scale, 2 Tints, Lighten and
# Darken to the planet photos.
# The Signature was a black background instead of white but it worked...
# Placment of signature and planets along with the glitch code are heavily modified
# and worked on.

def collage():
    setMediaPath()

    #Photos used
```

→ at the end of a line of code means that it is continued on the next line.

```
space = makePicture(getMediaPath("Space.jpg"))
mountain = makePicture(getMediaPath("MountainYellow.png"))
water = makePicture(getMediaPath("Water.jpg"))
mountainTop = makePicture(getMediaPath("MountainYellow2.png"))
planet2 = makePicture(getMediaPath("Planet2.jpg"))
planet = makePicture(getMediaPath("Planet2.jpg"))
signature = makePicture(getMediaPath("Signature.png"))

#Greyscales back plannets to make the purple pop
grayScale(planet2)

#Scales the photos UP or DOWN
SpaceBack = spaceBackground(space, 1050, 750)
Signature = signatureSize(signature, 0.03)
MountainBack = mountainSize(mountain, 1000, 730)
MountainBackTop = mountainSizeTop(mountainTop, 1000, 730)
WaterBack = waterBackground(water, 2)
Planet2Size = Planet2(planet2, 0.25)
Planet2Size2 = Planet2(planet2, 0.20)
Planet2Size3 = Planet2(planet2, 0.15)
Planet2Size4 = Planet2(planet2, 0.10)
TwoPlanet2Size = Planet2(planet, 0.25)
TwoPlanet2Size2 = Planet2(planet, 0.20)
TwoPlanet2Size3 = Planet2(planet, 0.15)
TwoPlanet2Size4 = Planet2(planet, 0.10)

#Places planet on background (left-right, up-down)
Planet2RemoveNPlace(SpaceBack, Planet2Size, 375, 0)
Planet2RemoveNPlace(SpaceBack, Planet2Size2, 125, 150)
Planet2RemoveNPlace(SpaceBack, Planet2Size3, 575, 175)
Planet2RemoveNPlace(SpaceBack, Planet2Size4, 75, 75)
Planet2RemoveNPlace(SpaceBack, Planet2Size4, 350, 125)
Planet2RemoveNPlace(SpaceBack, Planet2Size4, 800, 50)

#Darkens and tints the photo so far
darkenPic(SpaceBack)
purpleTint(SpaceBack)

#Takes Mountain photo and places on Space photo
BothPhoto = mountainYellow(MountainBack, SpaceBack)

#Darkens the photo so far
darkenPic(BothPhoto)

#Places Water photo on Space Mountain photo
WaterOnSpace = waterOnSpace(BothPhoto, WaterBack)

#Darkens the photo so far
darkenPic(WaterOnSpace)

#Places Mountain on top of the Water
MountainTopPhoto = mountainYellowTop(MountainBackTop, WaterOnSpace)

#Places planet on top (left-right, up-down)
Planet2RemoveNPlace(MountainTopPhoto, TwoPlanet2Size, 0, 225)
Planet2RemoveNPlace(MountainTopPhoto, TwoPlanet2Size2, 500, 400)
Planet2RemoveNPlace(MountainTopPhoto, TwoPlanet2Size3, 700, 350)
Planet2RemoveNPlace(MountainTopPhoto, TwoPlanet2Size4, 450, 275)
Planet2RemoveNPlace(MountainTopPhoto, TwoPlanet2Size4, 275, 500)
```

→ *at the end of a line of code means that it is continued on the next line.*

```
#Blurs and lightens the photo so it looks better
Blur = blur(MountainTopPhoto)
Lighten = lightenPic(Blur)

#Places the signature and tints the whole thing
SignaturePlace = signaturePlace(Blur, Signature, 50, 575)
blueTint(SignaturePlace)

#Adds the borders to the photo
Glitch = addGlitchBorderBottom(SignaturePlace)
BorderRed = addBorderRed(Glitch)
BorderBlack = addBorderBlack(BorderRed)
Finish = addGlitchBorderTop(BorderBlack)
explore(Finish)
```

→ at the end of a line of code means that it is continued on the next line.