

Destiny Wells

Completed



Originals



#Project Two solution written by Destiny Wells - October 14th 2022

#Main Function

```
def collage():
    BlueHy = makePicture(getMediaPath("hyBlue.jpg"))
    eye = makePicture(getMediaPath("bluEye.jpg"))
    sig = makePicture(getMediaPath("signature.png"))
    newEye = makeEmptyPicture(960, 720)
    scale(eye, newEye, 2)
    finalBlue = makeEmptyPicture(getWidth(BlueHy) / 7, getHeight(BlueHy) / 7)
    scale(BlueHy, finalBlue, 7)
    posBlue(newEye)
    copy(finalBlue, newEye, 354, 317)
    sketch(newEye)
    inv(newEye)
    rainbow(newEye)
    glitch(newEye)
    replaceBG(sig, newEye)
    explore(newEye)
```

#Helper Function: creates and places the inverted flower

```
def inv(canvas):
    inverseHy = makePicture(getMediaPath("hyBlue.jpg"))
    finalInverse = makeEmptyPicture(160, 104)
    scale(inverseHy, finalInverse, 4)
    inverse(finalInverse, 0, 160, 0, 104)
    for i in range(1, 6):
        copy(finalInverse, canvas, 0, 104 * i)
```

#Helper Function: makes the shaded blue hydrangea

```
def posBlue(canvas):
    posterizeHy = makePicture(getMediaPath("hyBlue.jpg"))
    finalPos = makeEmptyPicture(128, 83)
    scale(posterizeHy, finalPos, 5)
    light = makeColor(0, 221, 255)
    dark = makeColor(0, 10, 241)
    for x in range(0, 128):
        for y in range(0, 83):
            pixel = getPixel(finalPos, x, y)
            b = getBlue(pixel)
            if b > 201:
                setColor(pixel, light)
            elif b > 120:
```

```

        setColor(pixel, dark)
    else:
        setColor(pixel, black)
for i in range(0, 6):
    copy(finalPos, canvas, 160 + (128 * i), 104)
    copy(finalPos, canvas, 160 + (128 * i), 533)

def glitch(canvas):
    glitchHy = makePicture(getMediaPath("hyBlue.jpg"))
    finalGlitch = makeEmptyPicture(160,104)
    scale(glitchHy, finalGlitch, 4)
    for x in range(0,160):
        for y in range(0, 104):
            pixel = getPixel(finalGlitch, x, y)
            r = getRed(pixel)
            g = getGreen(pixel)
            b = getBlue(pixel)
            setColor(pixel, makeColor(r/1.55, g * .3, b*.95))
            color = getColor(pixel)
            if (y % 2 == 0):
                setColor(pixel, makeLighter(color))
            else:
                setColor(pixel, makeDarker(color))
    for i in range(0, 6):
        copy(finalGlitch, canvas, 160 * i, 0)

#Helper Function: changes the colors of an image to its inverse color
def inverse(source, x1, x2, y1, y2):
    for x in range(x1,x2):
        for y in range(y1, y2):
            pixel = getPixel(source, x, y)
            r = getRed(pixel)
            g = getGreen(pixel)
            b = getBlue(pixel)
            setColor(pixel, makeColor(255-r, 255-g, b * 2))

#Helper Function: Creates the inverted sketch
def sketch(canvas):
    sketchHy = makePicture(getMediaPath("hyBlue.jpg"))
    finalSketch = makeEmptyPicture(160, 104)
    scale(sketchHy, finalSketch, 4)
    for x in range(0,159):
        for y in range(0, 103):
            pixel = getPixel(finalSketch, x, y)
            sum = getRed(pixel) + getGreen(pixel) + getBlue(pixel)
            botrt = getPixel(finalSketch, x+1, y+1)
            sum2 = getRed(botrt)+getGreen(botrt)+getBlue(botrt)
            diff = abs(sum2-sum)
            setColor(pixel, makeColor(diff, diff, diff))
    for bar in range(1,3):
        inverse(finalSketch, 0, 160, 0, 104)
    for i in range(1, 6):
        copy(finalSketch, canvas, 800 , 104 * i)

#Helper Function: changes pixels to make one color dominate
def filter(source, x1, x2, y1, y2, color):
    for x in range(x1, x2):
        for y in range(y1, y2):
            pixel = getPixel(source, x, y)
            r = getRed(pixel)

```

```

    g = getGreen(pixel)
    b = getBlue(pixel)
    if color == "b":
        setColor(pixel, makeColor(r, g, b *3))
    elif color == "g":
        setColor(pixel, makeColor(r, g*2.5, b))
    else:
        setColor(pixel, makeColor(r*3, g, b))

#Helper: Applies rainbow-ish function
def rainbow(canvas):
    rainbowHy = makePicture(getMediaPath("hyBlue.jpg"))
    finalRain = makeEmptyPicture(160, 104)
    scale(rainbowHy, finalRain, 4)
    for rain in range(0, 52, 5):
        filter(finalRain, rain, rain + 10, 0, 104, "b")
        filter(finalRain, rain *2, rain*2+10, 0, 104, "g")
        filter(finalRain, rain*3, rain*3+10, 0, 104, "r")
    for i in range(0, 6):
        copy(finalRain, canvas, 160 * i, 616)

#Helper Function: scales picture down based on input
def scale(picture_in, picture_out, factor):
    sourceX = 0
    for targetX in range(0, getWidth(picture_out)):
        sourceY = 0
        for targetY in range(0, getHeight(picture_out)):
            color = getColor(getPixel(picture_in, int(sourceX), int(sourceY)))
            setColor(getPixel(picture_out, targetX, targetY), color)
            sourceY = sourceY + factor
            sourceX = sourceX + factor
    return picture_out

#Helper Function: displays image on canvas
def copy(source,target, tx, ty):
    targetX = tx
    for sourceX in range(0, getWidth(source)):
        targetY = ty
        for sourceY in range(0, getHeight(source)):
            px = getPixel(source, sourceX, sourceY)
            dx = getPixel(target, targetX, targetY)
            setColor(dx, getColor(px))
            targetY = targetY + 1
            targetX = targetX + 1

#Helper Function: removes background of photo and places it on canvas
def replaceBG(source, bg):
    sourceX = 0
    for x in range(568, 768):
        sourceY = 0
        for y in range(430, 541):
            pixel = getPixel(source, sourceX, sourceY)
            if (getRed(pixel) < 255 and getGreen(pixel)< 255 and getBlue(pixel) < 255):
                color = getColor(pixel)
                setColor(getPixel(bg, x, y), color)
            sourceY = sourceY + 1
        sourceX = sourceX + 1

```