Avery Walczewski

Completed



Originals










```
from mediaComp import *

from alterations import *
from colors import *

# Avery Walczewski -- 10/23/2026


def placingTuxes(tux, canvas):

    # Tuxes x & y variables defined
    tux0_x = 480
    tux0_y = 30

    tux1_x = 50
    tux1_y = 250

    tux2_x = 290
    tux2_y = 290

    tux3_x = 303
    tux3_y = 523

    tux4_x = 160
    tux4_y = 230

    tux1 = scale(tux, 1 / 2)
    for j in range(2): tux1 = blur(tux1)
    copy(tux1, canvas, tux1_x, tux1_y)

    tux2 = scale(tux, 1 / 2)
    tux2 = mirrorV(tux2, 3)
    tux2 = edgeDetect(tux2, 8)
    copy(tux2, canvas, tux2_x, tux2_y)
```

```
    tux3 = scale(tux, 1 / 2)
    tux3 = colorSwap(tux3)
    reverseChromakey(tux3, canvas, tux3_x, tux3_y, 3, 1, 4)
    copy(tux3, canvas, tux3_x, tux3_y)

    tux4 = scale(tux, 1 / 4); tux4 = scale(tux4, 2) # Basically pixelated the image
    copy(tux4, canvas, tux4_x, tux4_y)

    tux0 = scale(tux, 1 / 2)  # Original
    tux0 = mirrorH(tux0)
    copy(tux0, canvas, tux0_x, tux0_y)

def collage():

    # -- Defined Variables --
    pic = makePicture(getMediaFolder("Subject.jpg"))  # Me :)

    error = makePicture(
        getMediaFolder("error.jpg")
    )  # https://perishablepress.com/wp/wp-content/images/2009/pimp-your-404/404-not-found.png *Stock Image*

    tux = makePicture(
        getMediaFolder("Tux.jpg")
    )  # Tux, the linux mascot, owned by Larry Ewing and Simon Budig *Public Domain*

    flag = makePicture(
        getMediaFolder("flag.jpg")
    )  # I lossed the source :(, though its Public Domain... cuz its the USA flag

    sign = makePicture(
     getMediaFolder("sign.png")
    ) # signature

    canvas = makeEmptyPicture(getWidth(pic), getHeight(pic), black)

    # --- Main Picture ---
    posterize(pic); edgeDetect(pic, 17); mirrorV(pic, 3)
    copy(pic, canvas, 0, 0)

    # ----- ERROR -----
    error = scale(error, 3 / 4)
    for block in range(5):
        block = 5 + block * 32
        copy(error, canvas, block, block)
```

```
    # ----- Tux -----
    placingTuxes(tux, canvas)

    # ----- Flag -----
    flag_x = 400
    flag_y = 224

    flag = scale(flag, 1 / 4)
    flag = colorSwap(flag)
    reverseChromakey(flag, canvas, flag_x, flag_y, 24, 16, 12)
    copy(flag, canvas, flag_x, flag_y)

    # --- Sign ---
    sign_x = 530
    sign_y = 600

    sign = scale(sign, 1/3)
    chromakey(sign, canvas, sign_x, sign_y, 119, 119, 119)
    copy(sign, canvas, sign_x, sign_y)

    # --- Showcase ---
    pictureTool(canvas)
```

```
from mediaComp import *

# -- Functions w/ alteration --

def copy(picture, canvas, value_x, value_y):
    canvas_x = value_x
    for x in range(getWidth(picture)):
        canvas_y = value_y
        for y in range(getHeight(picture)):
            pixels = getPixelAt(picture, x, y)
            color = getColor(pixels)
            canvas_pixels = getPixelAt(canvas, canvas_x, canvas_y)
            setColor(canvas_pixels, color)
            canvas_y = canvas_y + 1
        canvas_x = canvas_x + 1
    return canvas

def blur(picture):
    blur_picture = duplicatePicture(picture)
    for x in range(1, getWidth(picture) - 1):
```

```
        for y in range(1, getHeight(picture) - 1):

            center = getPixelAt(blur_picture, x, y)

            left   = getPixelAt(picture, x - 1, y)
            right  = getPixelAt(picture, x + 1, y)
            top    = getPixelAt(picture, x, y - 1)
            bottom = getPixelAt(picture, x, y + 1)

            new_red   = (getRed(center) + getRed(left) + getRed(right) + getRed(top) + getRed(bottom)) // 5
            new_green = (getGreen(center) + getGreen(left) + getGreen(right) + getGreen(top)+getGreen(bottom))//5
            new_blue  = (getBlue(center) + getBlue(left) + getBlue(right) + getBlue(top) + getBlue(bottom)) // 5

            setColor(center, makeColor(new_red, new_green, new_blue))
    return blur_picture

def scale(source_picture, factor):
    new_width  = getWidth(source_picture) * factor
    new_height = getHeight(source_picture) * factor
    picture    = makeEmptyPicture(new_width, new_height)
    source_x = 0
    for new_x in range(getWidth(picture)):
        source_y = 0
        for new_y in range(getHeight(picture)):
            color = getColor(getPixelAt(source_picture, int(source_x), int(source_y)))
            setColor(getPixelAt(picture, new_x, new_y), color)
            source_y = source_y + 1 / factor
        source_x = source_x + 1 / factor
    return picture

def mirrorV(picture, dif): # amount of mirror splits
    width = getWidth(picture)
    point = width // dif
    for x in range(point, width):
        for y in range(getHeight(picture)):
            pixels = getPixelAt(picture, x, y)
            mirror_pixels = getPixelAt(picture, width - x - 1, y)
            color = getColor(pixels)
            setColor(mirror_pixels, color)
    return picture

def mirrorH(picture):
    height = getHeight(picture)
    for y in range(height):
        for x in range(getWidth(picture)):
```

```
            t_pixel = getPixelAt(picture, x, y)
            b_pixel = getPixelAt(picture, x, height - y - 1)
            color = getColor(t_pixel)
            setColor(b_pixel, color)
    return picture
```

```
from mediaComp import *

# --- Functions relating to changing Colors ---

def luminance(pixel):  # Used in edgeDetect
    red_value   = getRed(pixel)
    green_value = getGreen(pixel)
    blue_value  = getBlue(pixel)
    return (red_value + green_value + blue_value) // 3

def edgeDetect(picture, threshold): # Threshold is how strong the edgeDetect is
    for pixel in getPixels(picture):
        x = getX(pixel)
        y = getY(pixel)
        if y < getHeight(picture) - 1 and x < getWidth(picture) - 1:
            bottom_right_pixel     = getPixelAt(picture, x + 1, y + 1)
            this_luminance         = luminance(pixel)
            bottom_right_luminance = luminance(bottom_right_pixel)
            if abs(bottom_right_luminance - this_luminance) > threshold:
                setColor(pixel, black)
    return picture

def posterize(picture):
    for pixel in getPixels(picture):
        luminanceHere = luminance(pixel)

        if luminanceHere <= 21:
            setColor(pixel, makeColor(2, 50, 69))  # Cobalt

        elif luminanceHere >= 22 and luminanceHere <= 58:
            setColor(pixel, makeColor(173, 175, 228))  # Purple

        elif luminanceHere >= 59 and luminanceHere <= 80:
            setColor(pixel, makeColor(173, 228, 199))  # Blue

        elif luminanceHere > 81 and luminanceHere < 110:
            setColor(pixel, makeColor(228, 173, 202))  # Light Pink
```

```
        elif luminanceHere >= 146:
            setColor(pixel, white)  # Should be obvious

        else:
            setColor(pixel, makeColor(252, 232, 236))  # Pink
    return picture

def lighten(picture):
    for pixel in getPixels(picture):
        color = getColor(pixel)
        color = makeLighter(color)
        setColor(pixel, color)
    return picture

def chromakey(source, background, value_x, value_y, red, green, blue):  # Greater than
    for source_pixel in getPixels(source):
        x = getX(source_pixel)
        y = getY(source_pixel)
        if (getRed(source_pixel) >= red and getGreen(source_pixel) >= green and getBlue(source_pixel) >= blue):
            background_pixel = getPixelAt(background, x + value_x, y + value_y)
            background_color = getColor(background_pixel)
            setColor(source_pixel, background_color)

def reverseChromakey(source, background, value_x, value_y, red, green, blue):  # Less than
    for source_pixel in getPixels(source):
        x = getX(source_pixel)
        y = getY(source_pixel)
        if (getRed(source_pixel) <= red and getGreen(source_pixel) <= green and getBlue(source_pixel) <= blue):
            background_pixel = getPixelAt(background, x + value_x, y + value_y)
            background_color = getColor(background_pixel)
            setColor(source_pixel, background_color)

def colorSwap(picture):  # Flips to complemetry colors
    for pixel in getPixels(picture):
        red   = getRed(pixel)
        blue  = getBlue(pixel)
        green = getGreen(pixel)
        setColor(pixel, makeColor(255 - red, 255 - green, 255 - blue))
    return picture
```