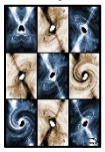
Brady Feeney

Completed



Original



```
# Name: Brady Feeney
# Date: 10/21/2026

from mediaComp import *
import math

GRID_COLUMNS = 3
GRID_ROWS = 3
CELL_WIDTH = 220
CELL_HEIGHT = 330
BAR_SIZE = 5
CANVAS_WIDTH = GRID_COLUMNS * CELL_WIDTH + (GRID_COLUMNS - 1) * BAR_SIZE
CANVAS_HEIGHT = GRID_ROWS * CELL_HEIGHT + (GRID_ROWS - 1) * BAR_SIZE
SIGNATURE_WIDTH = 80


def makeScaledCopy(source, newWidth, newHeight):

    result = makeEmptyPicture(newWidth, newHeight)
    for x in range(newWidth):
        for y in range(newHeight):
            sourceX = int(x * getWidth(source) / newWidth)
            sourceY = int(y * getHeight(source) / newHeight)
            sourcePixel = getPixelAt(source, sourceX, sourceY)
            setColor(getPixelAt(result, x, y), getColor(sourcePixel))
    return result


def copyPicture(source):

    result = makeEmptyPicture(getWidth(source), getHeight(source))
    for x in range(getWidth(source)):
        for y in range(getHeight(source)):
            setColor(getPixelAt(result, x, y), getColor(getPixelAt(source, x, y)))
    return result


def brightness(pixel):

    return (getRed(pixel) + getGreen(pixel) + getBlue(pixel)) / 3


def makeNegative(source):

    result = copyPicture(source)
    for x in range(getWidth(result)):
        for y in range(getHeight(result)):
            pixel = getPixelAt(result, x, y)
            setColor(pixel, makeColor(255 - getRed(pixel),
```

```
                                        255 - getGreen(pixel),
                                        255 - getBlue(pixel)))
    return result


def makeMirror(source):
    width = getWidth(source)
    result = copyPicture(source)
    for x in range(width // 2):
        for y in range(getHeight(source)):
            sourcePixel = getPixelAt(source, x, y)
            setColor(getPixelAt(result, width - 1 - x, y), getColor(sourcePixel))
    return result


def makeFlipVertical(source):

    height = getHeight(source)
    result = makeEmptyPicture(getWidth(source), height)
    for x in range(getWidth(source)):
        for y in range(height):
            sourcePixel = getPixelAt(source, x, height - 1 - y)
            setColor(getPixelAt(result, x, y), getColor(sourcePixel))
    return result


def makeRotate180(source):

    width = getWidth(source)
    height = getHeight(source)
    result = makeEmptyPicture(width, height)
    for x in range(width):
        for y in range(height):
            sourcePixel = getPixelAt(source, width - 1 - x, height - 1 - y)
            setColor(getPixelAt(result, x, y), getColor(sourcePixel))
    return result


def swirl(source, strength):

    width = getWidth(source)
    height = getHeight(source)
    centerX = width / 2
    centerY = height / 2
    maxDistance = min(centerX, centerY)

    result = makeEmptyPicture(width, height)
    for x in range(width):
        for y in range(height):
            deltaX = x - centerX
            deltaY = y - centerY
            distance = math.sqrt(deltaX * deltaX + deltaY * deltaY)
            angle = math.atan2(deltaY, deltaX)

            if distance < maxDistance:
                angle = angle + strength * (1 - distance / maxDistance)

            sourceX = int(centerX + distance * math.cos(angle))
            sourceY = int(centerY + distance * math.sin(angle))
            sourceX = max(0, min(width - 1, sourceX))
```

```
            sourceY = max(0, min(height - 1, sourceY))

            setColor(getPixelAt(result, x, y),
                     getColor(getPixelAt(source, sourceX, sourceY)))
    return result


def fillBlack(picture):

    for x in range(getWidth(picture)):
        for y in range(getHeight(picture)):
            setColor(getPixelAt(picture, x, y), black)


def placePicture(canvas, picture, startX, startY):
    for x in range(getWidth(picture)):
        for y in range(getHeight(picture)):
            setColor(getPixelAt(canvas, startX + x, startY + y),
                     getColor(getPixelAt(picture, x, y)))


def buildGrid(pictures):
    canvas = makeEmptyPicture(CANVAS_WIDTH, CANVAS_HEIGHT)
    fillBlack(canvas)
    for index in range(len(pictures)):
        row = index // GRID_COLUMNS
        column = index % GRID_COLUMNS
        placePicture(canvas, pictures[index],
                     column * (CELL_WIDTH + BAR_SIZE),
                     row * (CELL_HEIGHT + BAR_SIZE))
    return canvas


def isInk(pixel, background):
    difference = (abs(getRed(pixel) - getRed(background)) +
                  abs(getGreen(pixel) - getGreen(background)) +
                  abs(getBlue(pixel) - getBlue(background)))
    return difference > 120


def shrinkSignature(signature, newWidth):
    background = getPixelAt(signature, 0, 0)
    scale = getWidth(signature) / newWidth
    newHeight = max(1, int(getHeight(signature) / scale))
    result = makeEmptyPicture(newWidth, newHeight)

    for x in range(newWidth):
        for y in range(newHeight):
            startX = int(x * scale)
            startY = int(y * scale)
            endX = min(getWidth(signature), max(startX + 1, int((x + 1) * scale)))
            endY = min(getHeight(signature), max(startY + 1, int((y + 1) * scale)))

            foundInk = False
            for sourceX in range(startX, endX):
                for sourceY in range(startY, endY):
                    if isInk(getPixelAt(signature, sourceX, sourceY), background):
                        foundInk = True

            if foundInk:
```

```
                setColor(getPixelAt(result, x, y), black)
            else:
                setColor(getPixelAt(result, x, y), white)
    return result


def signCollage(canvas, signature):
    smallSignature = shrinkSignature(signature, SIGNATURE_WIDTH)
    offsetX = CANVAS_WIDTH - getWidth(smallSignature) - 8
    offsetY = CANVAS_HEIGHT - getHeight(smallSignature) - 8

    for x in range(getWidth(smallSignature)):
        for y in range(getHeight(smallSignature)):
            if getRed(getPixelAt(smallSignature, x, y)) < 128:
                target = getPixelAt(canvas, offsetX + x, offsetY + y)
                if brightness(target) < 128:
                    setColor(target, white)
                else:
                    setColor(target, black)


def collage():
    blackHole = makePicture(getMediaFolder("Blackhole.jpg"))
    signature = makePicture(getMediaFolder("Signature.png"))

    original = makeScaledCopy(blackHole, CELL_WIDTH, CELL_HEIGHT)

    pictures = [makeMirror(original),
                makeNegative(original),
                makeFlipVertical(original),
                makeNegative(makeRotate180(original)),
                original,
                makeNegative(swirl(original, 4.0)),
                swirl(original, 4.0),
                makeNegative(makeFlipVertical(original)),
                makeRotate180(original)]

    finalCollage = buildGrid(pictures)
    signCollage(finalCollage, signature)
    show(finalCollage)
```