# Cameron Granger, March 14, 2021. Title: Echoes of an Old Life

def collage():
    # Assigns picture files
    temple = makePicture(getMediaPath("newTemple.jpg"))
    kratos = makePicture(getMediaPath("kratos.jpg"))
    omega = makePicture(getMediaPath("omega.jpg"))
    signature = makePicture(getMediaPath("signature.jpg"))
    canvas = makeEmptyPicture(1000, 736)

    # Scales the Kratos image down and back up, resulting in a pixelated look
    # The pixel sizes are the same dimensions as the temple picture
    modKratos = scale(kratos, 0.05, 0.063)
    modKratos = scale(modKratos, 20, 16)

    # Converts the temple picture to grayscale, than tiles it.
    grayScale(temple)
    tiledTemple = tile(temple, canvas, 0, 48, 0, 44)
    tint(tiledTemple, modKratos)

    # Tiles the Omega symbol on the x and y axis
    tiledOmegaX = tile(omega, canvas, 0, 1, 0, 46)
    tiledOmegaY = tile(omega, canvas, 0, 50, 0, 1)
# Copies the tiled temple image and tiled omega images into the canvas

```python
copy(tiledTemple, canvas, 20, 16)
copy(tiledOmegaX, canvas, 0, 0)
copy(tiledOmegaX, canvas, 980, 0)
copy(tiledOmegaY, canvas, 0, 0)
copy(tiledOmegaY, canvas, 0, 720)
```

# Darkens an area around the canvas, but after the border

```python
dark(canvas, 20, 980, 16, 32)
dark(canvas, 20, 980, 704, 720)
dark(canvas, 20, 40, 32, 704)
dark(canvas, 960, 980, 32, 704)
```

signatureCopy(signature, canvas, 30, 16)

explore(canvas)

# Copies the picture passed to it.

```python
def copy(picture, canvas, shiftX, shiftY):
    targetX = shiftX
    for x in range(0, getWidth(picture)):
        targetY = shiftY
        for y in range(0, getHeight(picture)):
            color = getColor(getPixel(picture, x, y))
            setColor(getPixel(canvas, targetX, targetY), color)
            targetY = targetY + 1
        targetX = targetX + 1
    return(canvas)
```

# Tiles the copied picture from copy.

```python
def tile(picture, canvas, startX, endX, startY, endY):
    # Creates an empty picture the size of the end X and end Y multiplied by the width and height of the tiles respectively
    tiledPic = makeEmptyPicture(endX * 20, endY * 16)
    for x in range(startX, endX): # The for loops below make a tiling effect
        shiftX = x * 20
        for y in range(startY, endY):
            shiftY = y * 16
            tiled = copy(picture, tiledPic, shiftX, shiftY)
```
return(tiled)

# Converts image to grayscale
def grayScale(picture):
    for px in getPixels(picture):
        newRed = getRed(px) * 0.299
        newGreen = getGreen(px) * 0.587
        newBlue = getBlue(px) * 0.114
        luminance = newRed + newGreen + newBlue
        setColor(px, makeColor(luminance, luminance, luminance))

# Takes a picture and creates a tint mask using the refPic
def tint(picture, refPic):
    for x in range(0, getWidth(picture)):
        for y in range(0, getHeight(picture)):

            # Assignment of RGB color values for the picture and the refPicture
            px = getPixel(picture, x, y)
            px2 = getPixel(refPic, x, y)
            iRed = getRed(px)
            iGreen = getGreen(px)
            iBlue = getBlue(px)
            refRed = getRed(px2)
            refGreen = getGreen(px2)
            refBlue = getBlue(px2)

            if (iRed < 63):
                newColor = makeColor((refRed+iRed)*0.25, (refGreen+iGreen)*0.4, (refBlue+iBlue)*0.4)
                setColor(px, newColor)

            if (iRed > 62 and iRed < 192):
                newColor = makeColor((refRed+iRed)*0.5, (refGreen+iGreen)*0.5, (refBlue+iBlue)*0.5)
                setColor(px, newColor)

            if (iRed > 191):
                newColor = makeColor((refRed+iRed)*0.75, (refGreen+iGreen)*0.6, (refBlue+iBlue)*0.6)
                setColor(px, newColor)

    # Scales the picture up or down depending on the factor
    def scale(picture_in, factor, factor2):
        picture_out = makeEmptyPicture(int(getWidth(picture_in)*factor), int(getHeight(picture_in)*factor2))
sourceX = 0
for targetX in range(0, int(getWidth(picture_in) * factor)):
    sourceY = 0
    for targetY in range(0, int(getHeight(picture_in) * factor2)):

        srcpx = getPixel(picture_in, int(sourceX), int(sourceY))
        color = getColor(srcpx)
        setColor(getPixel(picture_out, targetX, targetY), color)
        sourceY = sourceY + (1 / float(factor2))
        sourceX = sourceX + (1 / float(factor))

    return(picture_out)

# Darkens an area around the canvas, this area is after the border
def dark(canvas, startX, endX, startY, endY):

    for count in range(0, 1):
        for x in range(startX, endX):  # The for loops below make a tiling effect
            for y in range(startY, endY):
                px = getPixel(canvas, x, y)
                color = getColor(px)
                color = makeDarker(color)
                setColor(px, color)

# Chroma keys a signature
def signatureCopy(sig, canvas, shiftX, shiftY):

    sig = scale(sig, 0.5, 0.5)
    targetX = shiftX
    for x in range(0, getWidth(sig)):
        targetY = shiftY
        for y in range(0, getHeight(sig)):

            refRed = getRed(getPixel(sig, x, y))
            refGreen = getGreen(getPixel(sig, x, y))
            refBlue = getBlue(getPixel(sig, x, y))
            color = getColor(getPixel(sig, x, y))

            if refGreen < 50 and refBlue < 50:
                setColor(getPixel(canvas, targetX, targetY), color)
        targetY = targetY + 1
        targetX = targetX + 1