

Ryland Fields

Completed



Originals



```
# Ryland Fields - 10/16/2021
```

```
def collage():
```

```
    originalPic = makePicture(getMediaPath("building.jpg"))
```

```
    OGWidth = getWidth(originalPic)
```

```
    OGHeight = getHeight(originalPic)
```

```
    sig = makePicture(getMediaPath("signature.jpg"))
```

```
    # the canvas that all modified pictures will end up on; the finished canvas
```

```
    masterCanvas = makeEmptyPicture(1000, 736, getColor(getPixel(originalPic, 0, OGHeight-1)))
```

```
    # now I'm scaling down the original picture from 4032x3024 to 981x736, or about one quarter its original
```

```
    # resolution so it will fit inside the 1000x736 canvas.
```

```
    # it would've been nice to scale it to exacty 1/4th resolution, but that results in a width of 1008 pixels.
```

```
    scaledCenterPic = makeEmptyPicture(int(OGWidth*0.2434), int(OGHeight*0.2434), black)
```

```
    scale(originalPic, scaledCenterPic, 0.2434)
```

```
#SMALLER BORDER PICTURES#
```

```
    tiny1 = tinyPics(originalPic)
```

```
    Hmirror(tiny1)
```

```
    copy(tiny1, masterCanvas, 0, 0)
```

```
    tiny2 = tinyPics(originalPic)
```

```
    posterize(tiny2)
```

```
    copy(tiny2, masterCanvas, 0, getHeight(tiny1))
```

```
    tiny3 = tinyPics(originalPic)
```

```
    tiny3 = shiftQuarters(tiny3)
```

```
    copy(tiny3, masterCanvas, 0, getHeight(tiny1)*2)
```

```
    tiny4 = tinyPics(originalPic)
```

```
    tiny4 = Cyanotype(tiny4)
```

```
    copy(tiny4, masterCanvas, 0, getHeight(tiny1)*3)
```

```

tiny5 = tinyPics(originalPic)
tiny5 = waffle(tiny5)
copy(tiny5, masterCanvas, getWidth(masterCanvas)-getWidth(tiny5), 0)
tiny6 = tinyPics(originalPic)
addRect(tiny6, 0, 0, 244, 182, gray)
addRect(tiny6, 1, 1, 242, 180, gray)
copy(tiny6, masterCanvas, getWidth(masterCanvas)-getWidth(tiny6), getHeight(tiny5))
tiny7 = tinyPics(originalPic)
bigTiny7 = split(tiny7, 5)
tempCanvas = makeEmptyPicture(245, 183, getColor(getPixel(tiny7, 0, getHeight(tiny7)-1)))
Tiny7 = crop2(bigTiny7, tempCanvas, 0, 0)
copy(Tiny7, masterCanvas, getWidth(masterCanvas)-getWidth(tiny7), 183*2)
tiny8 = tinyPics(originalPic)
tiny8 = blur(tiny8)
copy(tiny8, masterCanvas, getWidth(masterCanvas)-getWidth(tiny8), 183*3)

###MAIN PICTURE###
# these next two functions crop down the original image into just the top portion of the main building.
# This will be used to create the four copies seen in the final image
# the topOfBuilding variable is the cropped down portion of the original image.
topOfBuilding = makeEmptyPicture(605, 328, black)
crop(scaledCenterPic, topOfBuilding, 0, 0)
# the leftBuilding variable is the topOfBuilding image but rotated to the left by 90 degrees
leftBuilding = rtL90(topOfBuilding)
# the rightBuilding variable is the topOfBuilding image but rotated to the right by 90 degrees
rightBuilding = rtR90(topOfBuilding)
# in these next two lines I'm copying those rotated pictures to the center of the master canvas next to
# each other.
selectCopy(leftBuilding, masterCanvas, 172, 66)
selectCopy(rightBuilding, masterCanvas, 500, 66)
# the "selectCopy" function is a function I created that utilizes the functionality of the chromakey function
# while allowing you to place the resulting image anywhere on top of another image it's basically the equivalent
# of removing the background of a picture and then being able to move it anywhere over a background. This
# process was essential for creating the final image.
selectCopy(topOfBuilding, masterCanvas, 172, 408)
# because the top upside-down picture is upside down, it must be rotated
topHalf = rt180(topOfBuilding)
# and then selectively copied
selectCopy(topHalf, masterCanvas, 172, 0)
#SIGNATURE#
selectCopy(sig, masterCanvas, 800, 700)
explore(masterCanvas)

```

```

def tinyPics(originalPic):
    tinyPic = makeEmptyPicture(int(getWidth(originalPic)*0.0608), int(getHeight(originalPic)*0.0608), black)
    scale(originalPic, tinyPic, 0.0608)
    return tinyPic

def posterize(source):
    for p in getPixels(source):
        r = getRed(p)
        g = getGreen(p)
        b = getBlue(p)
        luminance = (r+g+b)/3
        if luminance < 63:
            setColor(p, black)
        elif 63 <= luminance < 126:
            setRed(p,102)
            setGreen(p,102)
            setBlue(p,102)
        elif 126 <= luminance < 189:
            setRed(p,204)
            setGreen(p,204)
            setBlue(p,204)
        else:
            setColor(p, white)

def scale(src, canvas, scaleFactor):
    sourceX = 0
    for targetX in range(0, int(getWidth(src)*scaleFactor)):
        sourceY = 0
        for targetY in range(0, int(getHeight(src)*scaleFactor)):
            color = getColor(getPixel(src, int(sourceX), int(sourceY)))
            setColor(getPixel(canvas, targetX, targetY), color)
            sourceY = sourceY + 1.0 / scaleFactor
            sourceX = sourceX + 1.0 / scaleFactor

def copy(source, target, startX, firstY):
    height = getHeight(source)
    width = getWidth(source)
    for x in range(0,width):
        startY = firstY
        for y in range(0,height):
            px = getPixel(source,x,y)
            px2 = getPixel(target, startX, startY)
            setColor(px2,getColor(px))
            startY = startY + 1

```

```

    startX = startX + 1

def crop(source, target, startX, firstY):
    for x in range(214,819):
        startY = firstY
        for y in range(15,343):
            px = getPixel(source,x,y)
            px2 = getPixel(target, startX, startY)
            setColor(px2,getColor(px))
            startY = startY + 1
        startX = startX + 1

def selectCopy(source, target, startX, firstY):
    height = getHeight(source)
    width = getWidth(source)
    for x in range(0,width):
        startY = firstY
        for y in range(0,height):
            px = getPixel(source, x, y)
            if ((getRed(px) + getGreen(px) + getBlue(px))/3 >20):
                px2 = getPixel(target, startX, startY)
                setColor(px2,getColor(px))
            startY = startY + 1
        startX = startX + 1

def chromakey(source):
    bg
    for px in getPixels(source):
        x = getX(px)
        y = getY(px)
        if (getRed(px) > 145 and getGreen(px) > 145 and getBlue(px) < 150):
            bgpx = getPixel(bg,x,y)
            bgcol = getColor(bgpx)
            setColor(px,bgcol)
    return source

def rtL90(source):
    height = getHeight(source)
    width = getWidth(source)
    target = makeEmptyPicture(height, width, black)
    startX = 0
    for x in range(0,width):
        startY = 0
        for y in range(0,height):
            px = getPixel(source,x,y)

```

```

    px2 = getPixel(target, startY, width - startX -1)
    setColor(px2,getColor(px))
    startY = startY + 1
    startX = startX + 1
return target

def rtR90(source):
    height = getHeight(source)
    width = getWidth(source)
    target = makeEmptyPicture(height, width, black)
    startX = 0
    for x in range(0,width):
        startY = 0
        for y in range(0,height):
            px = getPixel(source,x,y)
            px2 = getPixel(target, height - startY -1, startX)
            setColor(px2,getColor(px))
            startY = startY + 1
        startX = startX + 1
    return target

def rt180(source):
    height = getHeight(source)
    width = getWidth(source)
    target = makeEmptyPicture(width, height, black)
    startX = 0
    for x in range(0,width):
        startY = 0
        for y in range(0,height):
            px = getPixel(source,x,y)
            px2 = getPixel(target, width - startX - 1, height - startY -1)
            setColor(px2,getColor(px))
            startY = startY + 1
        startX = startX + 1
    return target

#tiny pic functions#

def Hmirror(source):
    mirrorPoint = getHeight(source)/2
    height = getHeight(source)
    for x in range(0,getWidth(source)):
        for y in range(0,mirrorPoint):
            leftPixel = getPixel(source,x,y)
            rightPixel = getPixel(source,x,height-y-1)

```

```

    color = getColor(leftPixel)
    setColor(rightPixel,color)

def shiftQuarters(source):
    height = getHeight(source)
    width = getWidth(source)
    canvas = makeEmptyPicture(width, height, getColor(getPixel(source, 0, getHeight(source)-1)))
    helper(source, canvas, 0, 0, width/2, 0)
    helper(source, canvas, 0, height/2, 0, 0)
    helper(source, canvas, width/2, height/2, 0, height/2)
    helper(source, canvas, width/2, 0, width/2, height/2)
    return canvas

def helper(pic, canvas, startX, startY, TstartX, TstartY):
    two_x = TstartX
    for x in range(startX, startX+getWidth(pic)/2):
        two_y = TstartY
        for y in range(startY, startY+getHeight(pic)/2):
            px = getPixel(pic,x,y)
            px2 = getPixel(canvas,two_x,two_y)
            setColor(px2, getColor(px))
            two_y = two_y + 1
            two_x = two_x + 1

def Cyanotype(source):
    pic = source
    for pixel in getPixels(pic):
        blue1 = getBlue(pixel)
        if (blue1 < 63):
            blue1 = blue1*2
        elif (63 <= blue1 <= 191):
            blue1 = blue1*1.3
        elif (blue1 > 191):
            blue1 = blue1*1.2
        red1 = getRed(pixel)*0.75
        green1 = getGreen(pixel)*0.75
        setBlue(pixel,blue1)
        setRed(pixel,red1)
        setGreen(pixel,green1)
    return pic

def waffle(source):
    pict = makeEmptyPicture(246, 184, getColor(getPixel(source, 0, getHeight(source)-1)))
    for x in range(0,246,2):
        for y in range(0,184,2):

```

```

    px = getPixel(pict,x,y)
    px2 = getPixel(source,x,y)
    setColor(px, getColor(px2))
return pict

def split(source, sliceNumber):
    pict1 = source
    pict1Width = getWidth(pict1)
    barWidth = pict1Width/sliceNumber
    widthMultiplier = sliceNumber*2-1
    pict2Width = barWidth*widthMultiplier
    height = getHeight(pict1)
    pict2 = makeEmptyPicture(pict2Width,height,getColor(getPixel(source, 0, getHeight(source)-1)))
    for slice in range(0,sliceNumber):
        helper3(pict1, pict2, barWidth*slice, barWidth*slice+barWidth)
    return pict2

def helper3(pict1, pict2, start, stop):
    height = getHeight(pict1)
    newX = start*2
    for x in range(start,stop):
        for y in range(0,height):
            px = getPixel(pict1,x,y)
            px2 = getPixel(pict2, newX, y)
            setColor(px2,getColor(px))
            newX = newX+1

def crop2(source, target, startX, firstY):
    for x in range(97,342):
        startY = firstY
        for y in range(0,getHeight(target)):
            px = getPixel(source,x,y)
            px2 = getPixel(target, startX, startY)
            setColor(px2,getColor(px))
            startY = startY + 1
            startX = startX + 1
    return target

def blur(source):
    target = duplicatePicture(source)
    for x in range(1, getWidth(source)-1):
        for y in range(1, getHeight(source)-1):
            top = getPixel(source,x,y-1)
            left = getPixel(source,x-1,y)
            bottom = getPixel(source,x,y+1)

```

```
right = getPixel(source,x+1,y)
center = getPixel(target,x,y)
newRed = (getRed(top) + getRed(left) + getRed(bottom) + getRed(right) + getRed(center))/5
newGreen = (getGreen(top) + getGreen(left) + getGreen(bottom) + getGreen(right) + getGreen(center))/5
newBlue = (getBlue(top) + getBlue(left) + getBlue(bottom) + getBlue(right) + getBlue(center))/5
setColor(center, makeColor(newRed, newGreen, newBlue))
return target
```