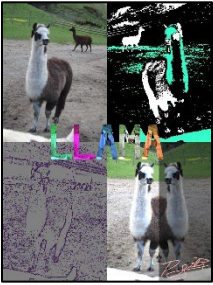


Pierson Silver

Completed

Originals



```
#Written 2/23 by Pierson Silver
def collage():
    picture = makePicture(getMediaPath("llama.jpg"))
    canvas = makeEmptyPicture(getWidth(picture)*2, getHeight(picture)*2)
    show(canvas)
#pictureTopLeft(original)
    copy(picture, canvas, 0, 0)
#posterize (and flip)TopRight(edit1)
    posterizedPicture = posterize(picture)
    flippedPosterized = flip(posterizedPicture)
    copy(flippedPosterized, canvas, getWidth(picture), 0)
#edgeDetectBottomLeft(edit2)
    edgeDetectedPicture = edgeDetect(picture, 15)
    copy(edgeDetectedPicture, canvas, 0, getHeight(picture))
#mirror (and lighten/darken each half)BottomRight(edit3)
    mirroredPicture = mirrorLightDark(picture)
    copy(mirroredPicture, canvas, getWidth(picture), getHeight(picture))
#spell LLAMA with Random Tinted Versions of Picture(edit4)
    spell(picture, canvas, getWidth(picture) - (getWidth(picture)/8),
    getHeight(picture) - (getHeight(picture)/8))
#signature
    sig = makePicture(getMediaPath("signature.jpg"))
    signature = scale(sig, .5)
    sigChroma(signature, canvas, getWidth(canvas) - getWidth(signature),
    getHeight(canvas) - getHeight(signature))
```

#Posterize and Flip Functions

```
def posterize(picture):
    result = makeEmptyPicture(getWidth(picture), getHeight(picture))
    for y in range(0, getHeight(picture)):
        for x in range(0, getWidth(picture)):
            picturePixel = getPixel(picture, x, y)
            resultPixel = getPixel(result, x, y)
            r = getRed(picturePixel)
            g = getGreen(picturePixel)
            b = getBlue(picturePixel)
            luminance = (r+g+b)/3
            if luminance < 65:
                setColor(resultPixel, white)
```

← means the line continues on the next line.

```

    elif luminance <= 165:
        setColor(resultPixel, black)
    else:
        setColor(resultPixel, makeColor(51, 255, 204))
return(result)
def flip(picture):
    result = makeEmptyPicture(getWidth(picture), getHeight(picture))
    for y in range(0, getHeight(picture)):
        for x in range(0, getWidth(picture)):
            picturePixel = getPixel(picture, x, y)
            resultPixel = getPixel(result, getWidth(picture) - 1 - x, y)
            color = getColor(picturePixel)
            setColor(resultPixel, color)
    return(result)

#End Posterize and Flip

def mirrorLightDark(picture):
    result = makeEmptyPicture(getWidth(picture), getHeight(picture))
    width = getWidth(picture)
    mirrorPoint = width/2
    for y in range(0, getHeight(picture)):
        for x in range(0, mirrorPoint):
            leftPixel = getPixel(picture, x, y)
            rightPixel = getPixel(result, width - 1 - x, y)
            color = getColor(leftPixel)
            setColor(getPixel(result, x, y), makeLighter(color))
            setColor(rightPixel, makeDarker(color))
    return(result)

#Helper + main for edgeDetect()

def luminance(pixel):
    r = getRed(pixel)
    g = getGreen(pixel)
    b = getBlue(pixel)
    return (r+g+b)/3
def edgeDetect(picture, threshold):
    result = makeEmptyPicture(getWidth(picture), getHeight(picture), gray)
    for px in getPixels(picture):
        x = getX(px)
        y = getY(px)
        if y < getHeight(picture)-1 and x < getWidth(picture)-1:
            bottomRight = getPixel(picture,x+1,y+1)
            pixelLum = luminance(px)
            brLum = luminance(bottomRight)
            if abs(brLum-pixelLum) > threshold:
                setColor(getPixel(result, x, y), makeColor(60, 30, 80))
            if abs(brLum-pixelLum) <= threshold:
                setColor(getPixel(result, x, y), gray)
    return(result)

#End edgeDetect()

```

← means the line continues on the next line.

```

def scale(picture, scale):
    canvas = makeEmptyPicture(int(float(getWidth(picture))*scale),
int(float(getHeight(picture))*scale))
    picY = 0
    for y in range(0, getHeight(canvas)):
        picX = 0
        for x in range(0, getWidth(canvas)):
            picturePixel = getPixel(picture, int(picX), int(picY))
            canvasPixel = getPixel(canvas, x, y)
            color = getColor(picturePixel)
            setColor(canvasPixel, color)
            picX = picX + 1.0/scale
        picY = picY + 1.0/scale
    return(canvas)

```

```

def copy(picture, canvas, startX, startY):
    for y in range(0, getHeight(picture)):
        for x in range(0, getWidth(picture)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x + startX, y + startY)
            color = getColor(picturePixel)
            setColor(canvasPixel, color)
        repaint(canvas)

```

#Helpers for spelling randomly tinted LLAMA

#Scale (also used in signature)

```

def scale(picture, scale):
    canvas = makeEmptyPicture(int(float(getWidth(picture))*scale),
int(float(getHeight(picture))*scale))
    picY = 0
    for y in range(0, getHeight(canvas)):
        picX = 0
        for x in range(0, getWidth(canvas)):
            picturePixel = getPixel(picture, int(picX), int(picY))
            canvasPixel = getPixel(canvas, x, y)
            color = getColor(picturePixel)
            setColor(canvasPixel, color)
            picX = picX + 1.0/scale
        picY = picY + 1.0/scale
    return(canvas)

```

#RNGtint

```

def randomTint(picture):
    canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
    randomRed = 2 * random.random()
    randomGreen = 2 * random.random()
    randomBlue = 2 * random.random()
    for y in range(0, getHeight(picture)):
        for x in range(0, getWidth(picture)):
            p = getPixel(picture, x, y)
            c = getPixel(canvas, x, y)
            newRed = getRed(p) * randomRed
            if newRed > 254:
                newRed = 254
            newGreen = getGreen(p) * randomGreen
            if newGreen > 254:

```

← means the line continues on the next line.

```

    newGreen = 254
    newBlue = getBlue(p) * randomBlue
    if newBlue > 254:
        newBlue = 254
    setColor(c, makeColor(newRed, newGreen, newBlue))
return(canvas)

```

#Letters

```

def L(picture):
    canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
    for y in range(0, getHeight(picture)):
        for x in range(int(getWidth(picture) * .1), int(getWidth(picture)*.3)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x, y)
            setColor(canvasPixel, getColor(picturePixel))
    for y in range(int(float(getHeight(picture)) * .85), getHeight(picture)):
        for x in range(getWidth(picture)/5, int(getWidth(picture)*.9)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x, y)
            setColor(canvasPixel, getColor(picturePixel))
    return(canvas)

```

```

def A(picture):
    canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
    midDiagR = int(getWidth(picture) *.45)
    midDiagL = int(getWidth(picture) *.55)
    leftDiag = int(getWidth(picture)*.35)
    rightDiag = int(getWidth(picture) *.65)
    for y in range(0, getHeight(picture)):
        for x in range(int(midDiagR), int(rightDiag)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x, y)
            setColor(canvasPixel, getColor(picturePixel))
        midDiagR = midDiagR + .2
        rightDiag = rightDiag + .2
        if rightDiag > getWidth(picture):
            rightDiag = getWidth(picture)
    for y in range(0, getHeight(picture)):
        for x in range(int(leftDiag), int(midDiagL)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x, y)
            setColor(canvasPixel, getColor(picturePixel))
        midDiagL = midDiagL - .2
        leftDiag = leftDiag - .2
        if leftDiag < 0:
            leftDiag = 0
    for y in range(int(getHeight(picture) * .45), int(getHeight(picture) * .6)):
        for x in range(getWidth(picture)/4, int(getWidth(picture) * .75)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x, y)
            setColor(canvasPixel, getColor(picturePixel))
    return(canvas)

```

```

def M(picture):
    canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
    for y in range(0, getHeight(picture)):
        for x in range(int(getWidth(picture)*.1), int(getWidth(picture)*.3)):
            picturePixel = getPixel(picture, x, y)

```

← means the line continues on the next line.

```

    canvasPixel = getPixel(canvas, x, y)
    setColor(canvasPixel, getColor(picturePixel))
for y in range(0, getHeight(picture)):
    for x in range(int(getWidth(picture)*.7), int(getWidth(picture)*.9)):
        picturePixel = getPixel(picture, x, y)
        canvasPixel = getPixel(canvas, x, y)
        setColor(canvasPixel, getColor(picturePixel))
midDiagL = int(getWidth(picture)*.3)
midDiagR = int(getWidth(picture)*.7)
leftDiag = int(getWidth(picture)*.1)
rightDiag = int(getWidth(picture)*.9)
for y in range(0, int(getHeight(picture)*.55)):
    for x in range(int(leftDiag), int(midDiagL)):
        if x > 200:
            x = 200
        picturePixel = getPixel(picture, x, y)
        canvasPixel = getPixel(canvas, x, y)
        setColor(canvasPixel, getColor(picturePixel))
    midDiagL = midDiagL + .45
    leftDiag = leftDiag + .45
for y in range(0, int(getHeight(picture)*.55)):
    for x in range(int(midDiagR), int(rightDiag)):
        if x < 160:
            x = 160
        picturePixel = getPixel(picture, x, y)
        canvasPixel = getPixel(canvas, x, y)
        setColor(canvasPixel, getColor(picturePixel))
    midDiagR = midDiagR - .45
    rightDiag = rightDiag - .45
return(canvas)

```

#End Letters

```

def letterChroma(picture, canvas, startX, startY):
    for y in range(0, getHeight(picture)):
        for x in range(0, getWidth(picture)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x + startX, y + startY)
            color = getColor(picturePixel)
            if (getRed(picturePixel) < 255 and getGreen(picturePixel) < 255 and ←
                getBlue(picturePixel) < 255):
                setColor(canvasPixel, color)
    repaint(canvas)

```

#Main Spell Function

```

def spell(picture, canvas, startX, startY):
    tintL1 = L(randomTint(picture))
    tintL2 = L(randomTint(picture))
    tintA1 = A(randomTint(picture))
    tintM = M(randomTint(picture))
    tintA2 = A(randomTint(picture))
    scaledL1 = scale(tintL1, .25)
    scaledL2 = scale(tintL2, .25)
    scaledA1 = scale(tintA1, .25)
    scaledM = scale(tintM, .25)
    scaledA2 = scale(tintA2, .25)
    width = getWidth(scaledA1)
    letterChroma(scaledL1, canvas, startX - (width*2 - width/4), startY)

```

← means the line continues on the next line.

```

letterChroma(scaledL2, canvas, startX - (width - width/8), startY)
letterChroma(scaledA1, canvas, startX, startY)
letterChroma(scaledM, canvas, startX + (width - width/8), startY)
letterChroma(scaledA2, canvas, startX + (width*2 - width/4), startY)

#End LLAMA Spelling

#Chroma for signature

def sigChroma(picture, canvas, startX, startY):
    for y in range(0, getHeight(picture)):
        for x in range(0, getWidth(picture)):
            picturePixel = getPixel(picture, x, y)
            canvasPixel = getPixel(canvas, x + startX, y + startY)
            color = getColor(picturePixel)
            if (getRed(picturePixel) < 50 and getGreen(picturePixel) < 50 and ←
                getBlue(picturePixel) < 50):
                setColor(canvasPixel, pink)
    repaint(canvas)

```

← means the line continues on the next line.