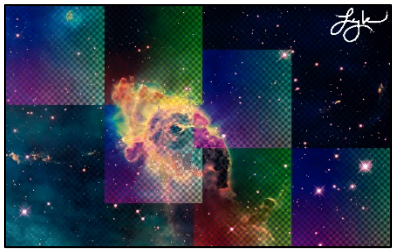
Lyle Dotson

Completed



Original

```
#Lyle Dotson --- Collage Contest 2020

'''Endeavored Space'''
#---------------------#

from jes import *
def collage():

    ''' Defining Imported Images '''

    the_cosmos_picture = "pictures/cosmos.png"
    cosmos = makePicture(the_cosmos_picture)

    the_Cosmos_picture = "pictures/cosmos.png"
    cosmosMask = makePicture(the_Cosmos_picture)

    the_low_value_RGB = "pictures/[000, 000, 000].png"
    xy000 = makePicture(the_low_value_RGB)

    the_high_value_RGB = "pictures/[256, 256, 256].png"
    xy256 = makePicture(the_high_value_RGB)

    the_BumpMap_picture = "pictures/BumpMap.png"
    bumpmap = makePicture(the_BumpMap_picture)

    my_signiture_picture = "pictures/Signiture.png"
    signiture = makePicture(my_signiture_picture)
```

```
#
    ''' Rendering The Picture '''
#
    canvas = makeEmptyPicture(getWidth(cosmos), getHeight(cosmos))

    layer1 = multiply(xy256, bumpmap)
    layer2 = multiply(xy000, bumpmap)

    xy00 = copy(xy256, canvas, 0, 0)
    xy488396 = copy(xy000, canvas, 488, 369)
    xy744369 = copy(xy256, canvas, 744, 369)
    xy256256 = copy(xy256, canvas, 256, 256)
    xy488113 = copy(xy256, canvas, 488, 113)
    xy2560 = copy(xy000, canvas, 256, 0)

    multiply(canvas, cosmos)

    Background(canvas, cosmos)

    GreyScale(cosmosMask)

    Overlay(cosmosMask, cosmos)

    Foreground(cosmosMask, canvas)

    blend(cosmos, cosmosMask) # cannot tell if I like blend

    chromaSig(signiture, cosmosMask, 830, -10)

    show(cosmosMask)

    writePictureTo(cosmosMask,"Endeavored Space.png")

#
    ''' Defining Layer Enhancements '''
#

def Overlay(source,bg):
    for px in getPixels(source):
        x = getX(px)
        y = getY(px)
        if (getRed(px) and getGreen(px) and getBlue(px)) >= 100:
```

```
            inputPixel = getPixel(source, x, y)
            overlayPixel = getPixel(bg, x, y)

            newColor = getOverlayColorValues(inputPixel, overlayPixel)

            setColor(px, newColor)

def getOverlayColorValues(source, overlayer):

    sourceRed = getRed(source)
    overlayedRed = getRed(overlayer)
    sourceGreen = getGreen(source)
    overlayedGreen = getGreen(overlayer)
    sourceBlue = getBlue(source)
    overlayedBlue = getBlue(overlayer)

    newR =   overlayedRed
    newG =   overlayedGreen
    newB =   overlayedBlue

    newC = makeColor(newR, newG, newB)

    return newC

def multiply(source, multiplyer):
    for x in range(0, getWidth(source)):
        for y in range(0, getHeight(source)):

            inputPixel = getPixel(source, x, y)
            multipliedPixel = getPixel(multiplyer, x, y)

            newColor = getMultipliedColorValues(inputPixel, multipliedPixel)

            px = getPixel(source, x, y)
            setColor(px, newColor)

    return source

def getMultipliedColorValues(source, multiplyer):

    sourceRed = getRed(source)
    multipliedRed = getRed(multiplyer)
    sourceGreen = getGreen(source)
```

```
    multipliedGreen = getGreen(multiplyer)
    sourceBlue = getBlue(source)
    multipliedBlue = getBlue(multiplyer)

    newR = 255 - sourceRed + multipliedRed/2
    newG = 255 - sourceGreen + multipliedGreen/2
    newB = 255 - sourceBlue + multipliedBlue/2

    newC = makeColor(newR, newG, newB)

    return newC

def copy(source, target, targX, targY):
    for sourceX in range (0, getWidth(source)):
        for sourceY in range (0, getHeight(source)):
            px = getPixel(source, sourceX, sourceY)
            tx = getPixel(target, targX + sourceX, targY +sourceY)
            setColor(tx, getColor(px))

def Background(source,bg):
    for px in getPixels(source):
        x = getX(px)
        y = getY(px)
        if (getRed(px) and getGreen(px) and getBlue(px)) >= 255:
            bgpx = getPixel(bg,x,y)
            bgcol = getColor(bgpx)
            setColor(px,bgcol)

def GreyScale(picture):
    for each_picture in getPixels(picture):
        intensity = (getRed(each_picture) + getGreen(each_picture) + getBlue(each_picture))/2.5
        setColor(each_picture, makeColor(intensity, intensity, intensity))

def Foreground(source,bg):
    for px in getPixels(source):
        x = getX(px)
        y = getY(px)
        if (getRed(px) and getGreen(px) and getBlue(px)) <= 98:

            inputPixel = getPixel(source, x, y)
            overlayPixel = getPixel(bg, x, y)

            newColor = getOverlayColorValues2(inputPixel, overlayPixel)
```

```
            setColor(px, newColor)

def getOverlayColorValues2(source, overlayer):

    sourceRed = getRed(source)
    overlayedRed = getRed(overlayer)
    sourceGreen = getGreen(source)
    overlayedGreen = getGreen(overlayer)
    sourceBlue = getBlue(source)
    overlayedBlue = getBlue(overlayer)

    newR =  (sourceRed + overlayedRed - 60 )
    newG =  (sourceGreen + overlayedGreen - 60 )
    newB =  (sourceBlue + overlayedBlue - 60 )

    newC = makeColor(newR, newG, newB)

    return newC

def chromaSig(source, target, targetX, targetY):
    for x in range(0, getWidth(source)):
        for y in range(0, getHeight(source)):
            px = getPixel(source, x, y)
            color = getColor(px)
            targ = getPixel(target, x + targetX, y + targetY)
            if distance (black, color) < 200:
                setColor(targ, white)

def blend(source, target):
    for start in range(2):
        for x in range(start, getWidth(source), 2):
            for y in range(start, getHeight(source), 2):
                sourcePixel = getPixel(source, x, y)
                blendPixel = getPixel(target, x, y)
                setColor(blendPixel, getColor(sourcePixel))

collage()
```