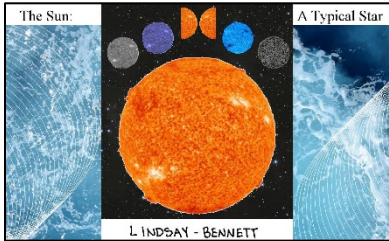


Lindsay Bennett

Completed



Originals



```
#Lindsay Bennett  
#CS 120 Section 3  
#March 15, 2021
```

```
#main function  
def collage():  
    #set pictures  
    bg = makePicture(getMediaPath("galaxy6.jpg"))  
    canvas = makeEmptyPicture(getWidth(bg)/4,getHeight(bg)/4)  
    water = makePicture(getMediaPath("water.jpg"))  
    sun = makePicture(getMediaPath("sun3.jpg"))  
    name = makePicture(getMediaPath("signature.jpg"))  
    new_canvas = makeEmptyPicture(getWidth(canvas),getHeight(canvas))  
  
    #background modifications  
    #galaxy bg  
    scale(bg,canvas,4)  
    #paste water  
    copy_left(water,canvas,0,0)  
    copy_right(water,canvas,(getWidth(canvas)/4)*3,0)  
    #arcs  
    arc(canvas)  
  
    #main sun modifications  
    small_sun = makeEmptyPicture(getWidth(sun)/2,getHeight(sun)/2)
```

```
scale(sun, small_sun, 2)
position(small_sun, new_canvas, getWidth(canvas)/4, getHeight(canvas)-getHeight(small_sun)-30, 5, 30)
chromakey(new_canvas, canvas, 249, 249, 249)
```

```
#tiny suns
tiny_sun = makeEmptyPicture(getWidth(sun)/10, getHeight(sun)/10)
sun_canvas = duplicatePicture(new_canvas)
empty_canvas = duplicatePicture(sun_canvas)
scale(sun, tiny_sun, 10)
gray_sun = duplicatePicture(tiny_sun)
reversed_sun = duplicatePicture(tiny_sun)
negative_sun = duplicatePicture(tiny_sun)
edge_sun = duplicatePicture(tiny_sun)
```

```
#tiny sun modifications
```

```
#gray sun (sun1)
grayscale(gray_sun)
position(gray_sun, sun_canvas, getWidth(canvas)/4+10, 70, 5, 5)
```

```
#blue sun (sun2)
tint(gray_sun)
position(gray_sun, sun_canvas, getWidth(canvas)/4+getWidth(tiny_sun), 35, 5, 5)
chromakey(sun_canvas, empty_canvas, 190, 190, 254)
```

```
#mirrored sun (sun3)
reverse_right(reversed_sun, reversed_sun, 5, 0)
mirror(reversed_sun)
position(reversed_sun, sun_canvas, getWidth(canvas)/4+(2*getWidth(tiny_sun))+5, 0, 5, 5)
```

```
#negative sun (sun4)
negative(negative_sun)
position(negative_sun, sun_canvas, getWidth(canvas)/4+(3*getWidth(tiny_sun))+3, 35, 3, 5)
chromakey_dark(sun_canvas, empty_canvas, 1, 1, 1)
```

```
#edge detect sun (sun5)
edge_detect(edge_sun, 15)
position(edge_sun, sun_canvas, getWidth(canvas)/4+(4*getWidth(tiny_sun)), 70, 7, 5)
```

```
#final chromakey
chromakey(sun_canvas, new_canvas, 249, 249, 249)
```

```

#finishingtouches
#white borders
border(sun_canvas)
#text
text(sun_canvas)
#signature
small_name = makeEmptyPicture(getWidth(name)/6,getHeight(name)/6)
scale(name,small_name,6)
final_image = makeEmptyPicture(getWidth(canvas),getHeight(canvas))
position(small_name,final_image,(getWidth(canvas)/4)+60,getHeight(canvas)-getHeight(small_name),0,0)
chromakey(final_image,sun_canvas,254,254,254)

#final image :)
explore(final_image)

#helper functions

#reusable functions

#scale function
def scale(picture_in,picture_out,scale):
    width = getWidth(picture_in)
    height = getHeight(picture_in)
    srcX = 0
    for targetX in range(0,width/scale):
        srcY = 0
        for targetY in range(0,height/scale):
            color = getColor(getPixel(picture_in,srcX,srcY))
            setColor(getPixel(picture_out,targetX,targetY),color)
            srcY = srcY + scale
            srcX = srcX + scale

#positioning function
def position(src,bg,target_x,target_y,width_adjust,height_adjust):
    targX = target_x
    for srcX in range(width_adjust,getWidth(src)-width_adjust):
        targY = target_y
        for srcY in range(0,getHeight(src)-height_adjust):
            px = getPixel(src,srcX,srcY)
            targpx = getPixel(bg,targX,targY)
            setColor(targpx,getColor(px))
            targY = targY + 1

```

```
targX = targX + 1
```

```
#chromakey function (light)
```

```
def chromakey(source,bg,r,g,b):
```

```
    for px in getPixels(source):
```

```
        x = getX(px)
```

```
        y = getY(px)
```

```
        if (getRed(px)>r and getGreen(px)>g and getBlue(px)>b):
```

```
            bgpx= getPixel(bg,x,y)
```

```
            bgcol=getColor(bgpx)
```

```
            setColor(px,bgcol)
```

```
#chromakey function (dark)
```

```
def chromakey_dark(source,bg,r,g,b):
```

```
    for px in getPixels(source):
```

```
        x = getX(px)
```

```
        y = getY(px)
```

```
        if (getRed(px)<r and getGreen(px)<g and getBlue(px)<b):
```

```
            bgpx= getPixel(bg,x,y)
```

```
            bgcol=getColor(bgpx)
```

```
            setColor(px,bgcol)
```

```
#background modifications
```

```
#copy water background (left side)
```

```
def copy_left(src,bg,target_x,target_y):
```

```
    targX = target_x
```

```
    for srcX in range(0,getWidth(bg)/4):
```

```
        targY = target_y
```

```
        for srcY in range(0,getHeight(bg)):
```

```
            px = getPixel(src,srcX,srcY)
```

```
            targpx = getPixel(bg,targX,targY)
```

```
            setColor(targpx,getColor(px))
```

```
            targY = targY + 1
```

```
        targX = targX + 1
```

```
#copy water background (right side)
```

```
def copy_right(src,bg,target_x,target_y):
```

```
    targX = target_x
```

```
    for srcX in range((getWidth(bg)/4)*3,getWidth(bg)):
```

```
        targY = target_y
```

```
        for srcY in range(0,getHeight(bg)):
```

```

    px = getPixel(src,srcX,srcY)
    targpx = getPixel(bg,targX,targY)
    setColor(targpx,getColor(px))
    targY = targY + 1
    targX = targX + 1

#arc designs
def arc(pic):
    x = 0
    for index in range(24):
        addArc(pic,getWidth(pic)-x,0+x,(getWidth(pic)/4)*3,getHeight(pic),270,360,white)
        addArc(pic,(0-(getWidth(pic)/2))-x,0+x,(getWidth(pic)/4)*3,getHeight(pic),180,270,white)
        x = x +10

#white borders
def border(pic):
    for x in range(0,getWidth(pic)/4):
        for y in range(0,50):
            setColor(getPixel(pic,x,y),white)
    for x in range(getWidth(pic)/4,(getWidth(pic)/4)*3):
        for y in range(getHeight(pic)-50,getHeight(pic)):
            setColor(getPixel(pic,x,y),white)
    for x in range((getWidth(pic)/4)*3,getWidth(pic)):
        for y in range(0,50):
            setColor(getPixel(pic,x,y),white)

#sun text
def text(pic):
    import java.awt.Font as Font
    font = makeStyle("Times New Roman", Font.PLAIN, 35)
    addTextWithStyle(pic,35,40,"The Sun:",font,black)
    addTextWithStyle(pic,((getWidth(pic)/4)*3)+10,40,"A Typical Star",font,black)

#sun modifications

#sun1
#grayscale
def grayscale(pic):
    for px in getPixels(pic):
        intensity = (getRed(px)+getGreen(px)+getBlue(px))/3
        setColor(px,makeColor(intensity,intensity,intensity))

#sun2
#blue tint

```

```

def tint(pic):
    for px in getPixels(pic):
        red = getRed(px)
        blue = getBlue(px)
        green = getGreen(px)
        #tint shadows
        if blue < 63:
            blue = blue*2
        #tint midtones
        if blue > 62 and blue < 191:
            blue = blue*1.3
        #tint highlights
        if blue > 190:
            blue = blue*1.2
        #set new color values
        setBlue(px, blue)
        setRed(px, red*.75)
        setGreen(px, green*.75)

#sun3
#copy right
def reverse_right(pic_in,pic_out,target_x,target_y):
    targX = target_x
    for x in range(getWidth(pic_in)/2,getWidth(pic_in)):
        targY = target_y
        for y in range(0,getHeight(pic_in)):
            px = getPixel(pic_in,x,y)
            targpx = getPixel(pic_in,targX,targY)
            setColor(targpx,getColor(px))
            targY = targY+1
            targX = targX+1

#mirror image
def mirror(pic):
    for x in range(0,getWidth(pic)/2):
        for y in range(0,getHeight(pic)):
            leftpx = getPixel(pic,x,y)
            rightpx = getPixel(pic,getWidth(pic)-x-1,y)
            setColor(rightpx,getColor(leftpx))

#sun4
#negative
def negative(pic):
    for px in getPixels(pic):
        setColor(px ,makeColor(255-getRed(px), 255-getGreen(px), 255-getBlue(px)))

```

```
#sun5
#edge detect
def luminance(px):
    r = getRed(px)
    g = getGreen(px)
    b = getBlue(px)
    return (r+g+b)/3
def edge_detect(src, threshold):
    for px in getPixels(src):
        x = getX(px)
        y = getY(px)
        if y < getHeight(src)-1 and x < getWidth(src)-1:
            botrt = getPixel(src,x+1,y+1)
            thislum = luminance(px)
            brlum = luminance(botrt)
            if abs(brlum-thislum) > threshold:
                setColor(px,gray)
            if abs(brlum-thislum) <= threshold:
                setColor(px,white)
```