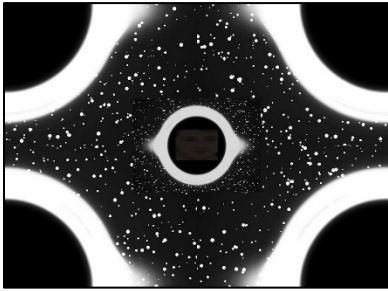
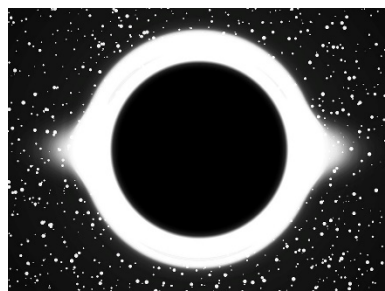


# Patrick Noble

Completed



Original



```
#Patrick Andrew Noble
#March 9th, 2020
#Work Name: In the End Lies the Beginning

#I started by defining pictures with specific variables so I could get the
#exact width and height of them
def collage():
    pic1 = makePicture(getMediaPath("blackhole.jpg"))
    pic2 = makePicture(getMediaPath("cryforhelp.jpg"))
    width1 = getWidth(pic1)
    height1 = getHeight(pic1)
    width2 = getWidth(pic2)
    height2 = getHeight(pic2)
    #Empty backgrounds for each picture to be modified on
    canvas1 = makeEmptyPicture(width1,height1)
    canvas2 = makeEmptyPicture(width1/3,height1/3,black)
    canvas3 = makeEmptyPicture(width2/9,height2/9,black)
    canvas4 = makeEmptyPicture(width2/3,height2/3,black)
    #The Split function I have set utilizes the corner function to set a
    #background for the collage by splitting up the original picture and setting
    #them in a different order
    split(pic1,canvas1)
    #I set a scale down function for each middle pictures seperately so I could
    #specifically set their sizes
    scaleDown(pic1,canvas2)
    scaleDownMore(pic2,canvas3)
    width3 = getWidth(canvas3)
    height3 = getHeight(canvas3)
    #This added modified canvas 3 and 4 together so they can be blended correctly
    #with canvasblend
    canvasfinal(canvas3, canvas4, 0 , width3 , 0, height3 , width3 , height3)
    canvasblend = blend(canvas2, canvas4)
    width = getWidth(canvasblend)
    height = getHeight(canvasblend)
    #This last program tossed together this cacophony of a collage into one while
    #keeping correct placement of each picture
    canvasfinal(canvasblend, canvas1, 0 , width , 0, height , width , height)
    show(pic1)
    show(pic2)
    show(canvas1)
```

```

def split(pic,canvas):
    width = getWidth(pic)
    height = getHeight(pic)
    corner(pic, canvas, 0 , width/2 , 0, height/2 , width/2 , height/2)
    corner(pic, canvas, width/2, width, 0, height/2, 0, height/2)
    corner(pic, canvas, 0, width/2, height/2, height, width/2, 0)
    corner(pic, canvas, width/2, width, height/2 ,height, 0, 0)
    return canvas

```

```

def corner(pic, canvas, start, end, start1, end1,new,old):
    width = getWidth(pic)
    height = getHeight(pic)
    targetX = new
    for sourceX in range(start,end):
        targetY = old
        for sourceY in range(start1,end1):
            color = getColor(getPixel(pic,sourceX,sourceY))
            setColor(getPixel(canvas,targetX,targetY),color)
            targetY = targetY + 1
        targetX = targetX + 1

```

```

def scaleDown(pic,canvas):
    width = getWidth(pic)
    height = getHeight(pic)
    sourceX = 0
    for targetX in range(0,int(width/3)):
        sourceY = 0
        for targetY in range(0,int(height/3)):
            color = getColor(getPixel(pic,sourceX,sourceY))
            setColor(getPixel(canvas,targetX,targetY),color)
            sourceY = sourceY + 3
        sourceX = sourceX + 3
    return canvas

```

```

def scaleDownMore(pic,canvas):
    width = getWidth(pic)
    height = getHeight(pic)
    width1 = getWidth(canvas)
    height1 = getHeight(canvas)
    sourceX = 0
    for targetX in range(0,int(width/9)):
        sourceY = 0
        for targetY in range(0,int(height/9)):
            color = getColor(getPixel(pic,sourceX,sourceY))
            setColor(getPixel(canvas,targetX,targetY),color)
            sourceY = sourceY + 9
        sourceX = sourceX + 9
    return canvas

```

```

def canvasfinal(pic, canvas, start, end, start1, end1,new,old):
    width = getWidth(pic)
    height = getHeight(pic)
    targetX = new
    for sourceX in range(start,end):
        targetY = old
        for sourceY in range(start1,end1):

```

```

    color = getColor(getPixel(pic, sourceX, sourceY))
    setColor(getPixel(canvas, targetX, targetY), color)
    targetY = targetY + 1
targetX = targetX + 1

```

```

def blend(be, bl):
    canvas = makeEmptyPicture(1000/3, 736/3, black)
    sourceX = 0
    for targetX in range(0, 1000/3):
        sourceY = 0
        for targetY in range(0, getHeight(be)):
            color = getColor(getPixel(be, sourceX, sourceY))
            setColor(getPixel(canvas, targetX, targetY), color)
            sourceY = sourceY + 1
        sourceX = sourceX + 1
    overlap = getWidth(be)
    sourceX = 0
    for targetX in range(0, getWidth(bl)):
        sourceY = 0
        for targetY in range(0, getHeight(bl)):
            bPixel = getPixel(be, sourceX, sourceY)
            lPixel = getPixel(bl, sourceX, sourceY)
            #This modifies the opacity of each picture when blending
            newRed = 0.85*getRed(bPixel)+0.15*getRed(lPixel)
            newGreen = 0.85*getGreen(bPixel)+0.15*getGreen(lPixel)
            newBlue = 0.85*getBlue(bPixel)+0.15*getBlue(lPixel)
            color = makeColor(newRed, newGreen, newBlue)
            setColor(getPixel(canvas, targetX, targetY), color)
            sourceY = sourceY + 1
        sourceX = sourceX + 1
    sourceX = overlap
    for targetX in range(overlap, getWidth(bl)):
        sourceY = 0
        for targetY in range(0, getHeight(bl)):
            color = getColor(getPixel(bl, sourceX, sourceY))
            setColor(getPixel(canvas, targetX, targetY), color)
            sourceY = sourceY + 1
        sourceX = sourceX + 1
    return canvas

```