

# Brock Hudon

Completed



Original



```
#Project 2 Collage  
#Name: Brock Hudon  
#Date: 10/18/2021
```

```
def main():  
    print "Select sonyCassette.jpg"  
    setMediaPath()  
    cassette = makePicture("sonyCassette.jpg")  
    theWidth = getWidth(cassette)  
    theHeight = getHeight(cassette)  
    print "Select signature.jpg"  
    signature = makePicture("signature.jpg")  
    print "Select volume.jpg"  
    volume = makePicture("volume.jpg")  
    print "Working..."  
    canvas = makeEmptyPicture(theWidth * 3, theHeight * 3)  
  
    #make collage rows 1 and 3  
    yellowPic = makeYellowish(cassette)  
    orangePic = makeOrangeish(cassette)  
    redPic = makeRedish(cassette)  
    bluePic = makeBlueish(cassette)  
    greenPic = makeGreenish(cassette)  
    purplePic = makePurpleish(cassette)  
  
    #make collage row 2  
    flippedPic = doAFlip(cassette)
```

```

mirroredPic = makeAMirror(flippedPic)
accurateGrayPic = grayScale(mirroredPic)
inverseGrayPic = inverseScale(flippedPic)
signaturePic = writeSignature(cassette, signature, volume)

#print collage row 2
copyInto(accurateGrayPic, canvas, ((theWidth - theHeight) / 2), ((theHeight - theWidth) / 2) + theHeight)
copyInto(signaturePic, canvas, theWidth, theHeight)
copyInto(inverseGrayPic, canvas, ((theWidth - theHeight) / 2) + (theWidth * 2), ((theHeight - theWidth) / 2) + theHeight)

#print collage row 1
copyInto(yellowPic, canvas, 0, 0)
copyInto(orangePic, canvas, theWidth, 0)
copyInto(redPic, canvas, theWidth * 2, 0)

#print collage row 3
copyInto(bluePic, canvas, 0, theHeight * 2)
copyInto(greenPic, canvas, theWidth, theHeight * 2)
copyInto(purplePic, canvas, theWidth * 2, theHeight * 2)

show(canvas)
writePictureTo(canvas, r"C:\Users\01y0zhang\Desktop\HudonP2.jpg")

#functions for row 1
def makeYellowish(source):
    picYellow = duplicatePicture(source)
    yellowish = makeColor(240, 183, 54)
    pixels = getPixels(picYellow)

    for px in pixels:
        pxColor = getColor(px)
        if distance(pxColor, white) < 18:
            setColor(px, yellowish)

    for px in pixels:
        if distance(getColor(px), yellowish) != 0:
            #note: this color is taken from a function below
            #colors trade between functions to create the effect
            newRed = ((getRed(px) * 2) + (207 * 3)) / 5
            newGreen = ((getGreen(px) * 2) + (56 * 3)) / 5
            newBlue = ((getBlue(px) * 2) + (45 * 3)) / 5
            newColor = makeColor(newRed, newGreen, newBlue)
            setColor(px, newColor)
    return picYellow

```

```

def makeOrangeish(source):
    picOrange = duplicatePicture(source)
    orangeish = makeColor(224, 130, 48)
    pixels = getPixels(picOrange)

    for px in pixels:
        pxColor = getColor(px)
        if distance(pxColor, white) < 18:
            setColor(px, orangeish)

    for px in pixels:
        if distance(getColor(px), orangeish) != 0:
            newRed = ((getRed(px) * 2) + (138 * 3)) / 5
            newGreen = ((getGreen(px) * 2) + (179 * 3)) / 5
            newBlue = ((getBlue(px) * 2) + (65 * 3)) / 5
            newColor = makeColor(newRed, newGreen, newBlue)
            setColor(px, newColor)
    return picOrange

```

```

def makeRedish(source):
    picRed = duplicatePicture(source)
    redish = makeColor(207, 56, 45)
    pixels = getPixels(picRed)

    for px in pixels:
        pxColor = getColor(px)
        if distance(pxColor, white) < 18:
            setColor(px, redish)

    for px in pixels:
        if distance(getColor(px), redish) != 0:
            newRed = ((getRed(px) * 2) + (240 * 3)) / 5
            newGreen = ((getGreen(px) * 2) + (183 * 3)) / 5
            newBlue = ((getBlue(px) * 2) + (54 * 3)) / 5
            newColor = makeColor(newRed, newGreen, newBlue)
            setColor(px, newColor)
    return picRed

```

```

#functions for row 3
def makeBlueish(source):
    picBlue = duplicatePicture(source)
    blueish = makeColor(71, 137, 195)
    pixels = getPixels(picBlue)

    for px in pixels:

```

```
pxColor = getColor(px)
if distance(pxColor, white) < 18:
    setColor(px, blueish)

for px in pixels:
    if distance(getColor(px), blueish) != 0:
        newRed = ((getRed(px) * 2) + (214 * 3)) / 5
        newGreen = ((getGreen(px) * 2) + (87 * 3)) / 5
        newBlue = ((getBlue(px) * 2) + (138 * 3)) / 5
        newColor = makeColor(newRed, newGreen, newBlue)
        setColor(px, newColor)
return picBlue
```

```
def makeGreenish(source):
    picGreen = duplicatePicture(source)
    greenish = makeColor(138, 180, 65)
    pixels = getPixels(picGreen)

    for px in pixels:
        pxColor = getColor(px)
        if distance(pxColor, white) < 18:
            setColor(px, greenish)

    for px in pixels:
        if distance(getColor(px), greenish) != 0:
            newRed = ((getRed(px) * 2) + (224 * 3)) / 5
            newGreen = ((getGreen(px) * 2) + (130 * 3)) / 5
            newBlue = ((getBlue(px) * 2) + (48 * 3)) / 5
            newColor = makeColor(newRed, newGreen, newBlue)
            setColor(px, newColor)
    return picGreen
```

```
def makePurpleish(source):
    picPurple = duplicatePicture(source)
    purpleish = makeColor(214, 87, 138)
    pixels = getPixels(picPurple)

    for px in pixels:
        pxColor = getColor(px)
        if distance(pxColor, white) < 18:
            setColor(px, purpleish)

    for px in pixels:
        if distance(getColor(px), purpleish) != 0:
            newRed = ((getRed(px) * 2) + (71 * 3)) / 5
```

```
    newGreen = ((getGreen(px) * 2) + (137 * 3)) / 5
    newBlue = ((getBlue(px) * 2) + (195 * 3)) / 5
    newColor = makeColor(newRed, newGreen, newBlue)
    setColor(px, newColor)
return picPurple
```

```
#funtions for row 2
def doAFlip(source):
    theWidth = getWidth(source)
    theHeight = getHeight(source)
    picFlipped = makeEmptyPicture(theHeight, theWidth)

    for x in range(0, theWidth):
        for y in range(0, theHeight):
            px = getPixel(source, x, y)
            pxColor = getColor(px)
            pxFlip = getPixel(picFlipped, y, x)
            setColor(pxFlip, pxColor)
    return picFlipped
```

```
def makeAMirror(source):
    theWidth = getWidth(source)
    theHeight = getHeight(source)
    picMirrored = makeEmptyPicture(theWidth, theHeight)

    for x in range(0, theWidth):
        for y in range(0, theHeight):
            px = getPixel(source, x, y)
            pxColor = getColor(px)
            pxMirror = getPixel(picMirrored, (theWidth - x - 1), (theHeight - y - 1))
            setColor(pxMirror, pxColor)
    return picMirrored
```

```
def grayScale(source):
    picGray = duplicatePicture(source)
    #note: this form of grayscaling looks cooler
    #assigns colors based on the 5 defined ranges
    grayDark = makeColor(24, 24, 24)
    gray1 = makeColor(76, 76, 76)
    gray2 = makeColor(128, 128, 128)
    gray3 = makeColor(180, 180, 180)
    grayLite = makeColor(232, 232, 232)

    for px in getPixels(picGray):
        theRed = getRed(px)
```

```

theGreen = getGreen(px)
theBlue = getBlue(px)
avgColor = (theRed + theGreen + theBlue) / 3

if avgColor <= 51:
    setColor(px, grayDark)
elif avgColor > 51 and avgColor <= 102:
    setColor(px, gray1)
elif avgColor > 102 and avgColor <= 153:
    setColor(px, gray2)
elif avgColor > 153 and avgColor <= 204:
    setColor(px, gray3)
elif avgColor > 204:
    setColor(px, grayLite)
return picGray

```

```

def inverseScale(source):
    picInverse = duplicatePicture(source)
    grayDark = makeColor(24, 24, 24)
    gray1 = makeColor(76, 76, 76)
    gray2 = makeColor(128, 128, 128)
    gray3 = makeColor(180, 180, 180)
    grayLite = makeColor(232, 232, 232)

    for px in getPixels(picInverse):
        theRed = getRed(px)
        theGreen = getGreen(px)
        theBlue = getBlue(px)
        avgColor = (theRed + theGreen + theBlue) / 3

        #inverse of grayScale() function
        if avgColor <= 51:
            setColor(px, grayLite)
        elif avgColor > 51 and avgColor <= 102:
            setColor(px, gray3)
        elif avgColor > 102 and avgColor <= 153:
            setColor(px, gray2)
        elif avgColor > 153 and avgColor <= 204:
            setColor(px, gray1)
        elif avgColor > 204:
            setColor(px, grayDark)
    return picInverse

```

```

def writeSignature(source, signature, volume):
    picSignature = duplicatePicture(source)

```

```
#adds a title to the cassette at the center
for x in range(0, getWidth(volume)):
    for y in range (0, getHeight(volume)):
        px = getPixel(volume, x, y)
        pxColor = getColor(px)
        if distance(pxColor, white) != 0:
            destPx = getPixel(picSignature, x + 110, y + 63)
            setColor(destPx, pxColor)
```

```
#adds my signature in the bottom right corner
```

```
copyInto(signature, picSignature, getWidth(picSignature) - getWidth(signature), getHeight(picSignature) - getHeight(signature) -
```

```
5)
```

```
return picSignature
```