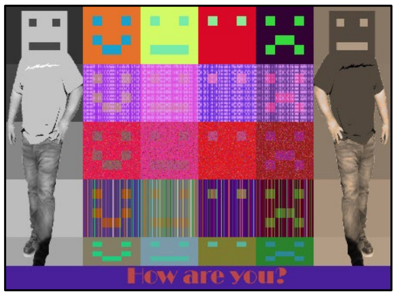
Beethoven Meginnis

Completed



Original

```
#Beethoven Meginnis
#3/8/2022
def collage():
  #randomly generated colors for later use
  color1 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color2 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color3 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color4 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color5 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color6 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color7 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color8 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color9 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color10 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color11 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color12 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color13 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color14 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color15 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color16 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color17 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color18 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color19 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color20 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color21 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color22 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color23 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))
  color24 = makeColor(random.randint(0,255), random.randint(0,255), random.randint(0,255))

  #initial picture canvas and overall picture canvas
  initialPic = makeEmptyPicture(600,150)
  collageCanvas = makeEmptyPicture(1000,736)
  #empty pictures the 4 faces of intial picture
  picture1 = makeEmptyPicture(6, 6, color1)
  picture2 = makeEmptyPicture(6, 6, color3)
  picture3 = makeEmptyPicture(6, 6, color5)
  picture4 = makeEmptyPicture(6, 6, color7)

  #creating 4 faces and copying into initial picture canvas
  happyFace(picture1, color2)
  output = applyScaling(picture1, 25)
  copy(output,initialPic,0,0)

  flatFace(picture2, color4)
  output = applyScaling(picture2, 25)
  copy(output,initialPic, 150, 0)
```

```
blankFace(picture3, color6)
output = applyScaling(picture3, 25)
copy(output,initialPic, 300, 0)

frownFace(picture4, color8)
output = applyScaling(picture4, 25)
copy(output,initialPic, 450, 0)

#side banners (angryguy.png + 2 new faces)
picture5 = makeEmptyPicture(6, 6, color9)
picture6 = makeEmptyPicture(6, 6, color10)
sideBanner = makePicture(getMediaPath("angryguy.png"))
copy(sideBanner,collageCanvas,0,0)
flip(sideBanner)
copy(sideBanner,collageCanvas,800,0)

replaceColor(0,199,0,149,collageCanvas,color13,white,50)
replaceColor(0,199,150,299,collageCanvas,color14,white,50)
replaceColor(0,199,300,449,collageCanvas,color15,white,50)
replaceColor(0,199,450,599,collageCanvas,color16,white,50)
replaceColor(0,199,600,674,collageCanvas,color17,white,50)

replaceColor(800,999,0,149,collageCanvas,color18,white,50)
replaceColor(800,999,150,299,collageCanvas,color19,white,50)
replaceColor(800,999,300,449,collageCanvas,color20,white,50)
replaceColor(800,999,450,599,collageCanvas,color21,white,50)
replaceColor(800,999,600,674,collageCanvas,color22,white,50)

replaceColor(0,199,0,357,collageCanvas,color9,blue,165)
replaceColor(800,999,0,357,collageCanvas,color10,blue,165)

flatFace(picture5, color11)
output = applyScaling(picture5, 20)
copy(output,collageCanvas,40,12)

flatFace(picture6, color12)
output = applyScaling(picture6, 20)
copy(output,collageCanvas, 840, 12)

tint(collageCanvas, 0, 199, 0, 674)
tint(collageCanvas, 800, 999, 0, 674)

#bottom banner
bottomBanner = makePicture(getMediaPath("howareyou.png"))
copy(bottomBanner,collageCanvas,0,675)

replaceColor(0,999,675,735,collageCanvas,color23,white,50)
replaceColor(0,999,675,735,collageCanvas,color24,black,10)

#filling canvas with different versions of the intial 4-face picture
copy(initialPic,collageCanvas, 200, 0)
copyRandomCheckered(initialPic,collageCanvas, 200, 150)
copyRandomColorDistortion(initialPic,collageCanvas, 200, 300)
copy(initialPic,collageCanvas, 200, 450)
copyHalf(initialPic,collageCanvas, 200, 600)

#changing colors of 4 additional copies after applying copy functions
randomColorAverageAndSwap(200,799,150,299,collageCanvas)
randomColorAverageAndSwap(200,799,300,449,collageCanvas)
randomColorAverageAndSwap(200,799,450,599,collageCanvas)
randomColorAverageAndSwap(200,799,600,674,collageCanvas)
```

```
  #modifying the 3rd copy
  colorMod(collageCanvas,200,800,450,600)

  #adding signature as a label to each t-shirt
  signature = makePicture(getMediaPath("signature.png"))
  copySig(signature,collageCanvas,861,145,black)
  copySig(signature,collageCanvas,43,145,white)

  #opens the finalized picture and saves it in the project2 folder.
  #if a "good" picture is generated, change its name so that it is not overwritten.
  explore(collageCanvas)
  writePictureTo(collageCanvas, getMediaPath("picture.png"))

#8 input parameters, replaces one color with another one.
def replaceColor(startX,endX,startY,endY,pic,endColor,replaceColor,threshold):
    for px in getPixels(pic):
      x = getX(px)
      y = getY(px)
      if (startX <= x <= endX) and (startY <= y <= endY):
        if (distance(replaceColor,getColor(px)) < threshold):
          setColor(px,endColor)

#5 input parameters, adds just the signature and allows it to be changed to a set color
def copySig(picture_in,picture_out,target_x,target_y,inputColor):
    for x in range(0,getWidth(picture_in)):
      for y in range(0,getHeight(picture_in)):
        px = getPixel(picture_in,x,y)
        if (distance(white,getColor(px)) > 10):
          color=getColor(px)
          newPx=(getPixel(picture_out,target_x+x,target_y+y))
          setColor(newPx,inputColor)

#4 input parameters, default copy function
def copy(picture_in,picture_out,target_x,target_y):
  for x in range(0,getWidth(picture_in)):
    for y in range(0,getHeight(picture_in)):
      px = getPixel(picture_in,x,y)
      color=getColor(px)
      newPx=(getPixel(picture_out,target_x+x,target_y+y))
      setColor(newPx,color)

#2 input parameters, semi-modified from another activity to serve the purposes of this
project
def applyScaling(picture,scaleFactor):
  upScaled = makeEmptyPicture(getWidth(picture)*scaleFactor, ↵
      getHeight(picture)*scaleFactor)
  scale(picture,upScaled,1/float(scaleFactor))
  return upScaled

#3 input parameters, default scale function
def scale(picture_in,picture_out,scaleFactor):
  sourceX = 0
  for targetX in range(0,getWidth(picture_out)):
    sourceY = 0
    for targetY in range(0,getHeight(picture_out)):
      color = getColor(getPixel(picture_in,int(sourceX),int(sourceY)))
      setColor(getPixel(picture_out,targetX,targetY),color)
      sourceY = sourceY + float(scaleFactor)
    sourceX = sourceX + float(scaleFactor)

#1 input parameter, flips a picture entirely, instead of mirring around an axis
```

```
def flip(picture):
  width = getWidth(picture)
  height = getHeight(picture)
  for x in range(0, width/2):
    for y in range(0, height):
      pixel = getPixel(picture, x, y)
      newPixel = getPixel(picture, width-1-x, y)
      color = getColor(pixel)
      newColor = getColor(newPixel)
      setColor(pixel, newColor)
      setColor(newPixel, color)

#5 input parameters, adds random color bands to the picture (random colors+random
increments)
def colorMod(picture,startX,endX,startY,endY):
  for i in range(0,3):
    increment = random.randint(5,11)
    for x in range(startX,endX,increment):
      color1 = makeColor(random.randint(0,255), random.randint(0,255), ↵
             random.randint(0,255))
      for y in range(startY,endY):
        px = getPixel(picture,x,y)
        color=getColor(px)
        setColor(px,color1)

#5 input parameters, taken from the book to create a sepia-like effect
def tint(picture, startX, endX, startY, endY):
  grayScale(picture)
  for p in getPixels(picture):
    red = getRed(p)
    green = getGreen(p)
    blue = getBlue(p)
    x = getX(p)
    y = getY(p)
    if (startX <= x <= endX) and (startY <= y <= endY):
      if (red < 63):
        red = red*1.1
        blue = blue*0.9

      if (red > 62 and red <192):
        red = red*1.15
        blue = blue*0.85

      if (red > 191):
        red = red*1.08
        if (red >255):
          red = 255
        blue = blue*0.93

      setBlue(p, blue)
      setRed(p, red)

#1 input parameter, taken from the book to produce a grayscale effect
def grayScale(picture):
  for pixel in getPixels(picture):
        intensity = (getRed(pixel)+getGreen(pixel)+getBlue(pixel))/3
        setColor(pixel,makeColor(intensity,intensity,intensity))

#4 input parameters, shrink image in half, vertically
def copyHalf(picture_in,picture_out,target_x,target_y):
  for x in range(0,getWidth(picture_in)):
    for y in range(0,getHeight(picture_in)):
      px = getPixel(picture_in,x,y)
      color=getColor(px)
      newPx=(getPixel(picture_out,target_x+x,target_y + y/2))
      setColor(newPx,color)

#5 input parameters, randomly averages colors and randomly swaps colors
def randomColorAverageAndSwap(xstart, xend, ystart, yend, picture):
      value1 = random.randint(0,255)
      value2 = random.randint(0,255)
      value3 = random.randint(0,255)
      colorValue = random.randint(0,3)
      for pixel in getPixels(picture):
        x = getX(pixel)
        y = getY(pixel)
        if xstart <= x <= xend:
          if ystart <= y <= yend:
            red = getRed(pixel)
            green = getGreen(pixel)
            blue = getBlue(pixel)
            redAverage = (red + value1)/2
            greenAverage = (green + value2)/2
            blueAverage = (blue + value3)/2
            if colorValue < 1:
              setColor(pixel,makeColor(greenAverage,blueAverage,redAverage))
            elif colorValue >1:
              setColor(pixel,makeColor(redAverage,blueAverage,greenAverage))
            else:
              setColor(pixel,makeColor(blueAverage,greenAverage,redAverage))

#4 input parameters, randomly adds pixels of random colors to the picture
def copyRandomColorDistortion(picture_in,picture_out,target_x,target_y):
  for x in range(0,getWidth(picture_in)):
    for y in range(0,getHeight(picture_in)):
      px = getPixel(picture_in,x,y)
      pixelcolorrand =  random.randint(0,2)
      if pixelcolorrand > 0:
        color = getColor(px)
      else:
        color = makeColor(random.randint(0,255), random.randint(0,255), ↵
             random.randint(0,255))
      newPx=(getPixel(picture_out,target_x+x,target_y+y))
      setColor(newPx,color)

#4 input parameters, creates a random checker-like pattern
def copyRandomCheckered(picture_in,picture_out,target_x,target_y):
  for x in range(0,getWidth(picture_in)):
    for y in range(0,getHeight(picture_in), random.randint(1,3)):
      px = getPixel(picture_in,x,y)
      color=getColor(px)
      newPx=(getPixel(picture_out,target_x+x,target_y+y))
      setColor(newPx,color)

#2 input parameters, creates a smiling face
def happyFace(pic,color):
  #eyes
  for x in range(1,2):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(4,5):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  #mouth
  for x in range(1,2):
    for y in range(3,4):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(4,5):
    for y in range(3,4):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(2,4):
    for y in range(4,5):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)

#2 input parameters, creates a frowning face
def frownFace(pic,color):
  #eyes
  for x in range(1,2):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(4,5):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  #mouth
  for x in range(1,2):
    for y in range(4,5):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(4,5):
    for y in range(4,5):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(2,4):
    for y in range(3,4):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)

#2 input parameters, creates a flat face
def flatFace(pic,color):
  #eyes
  for x in range(1,2):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(4,5):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  #mouth
  for x in range(1,5):
    for y in range(4,5):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)

#2 input parameters, creates a blank face
def blankFace(pic,color):
  #eyes
  for x in range(1,2):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  for x in range(4,5):
    for y in range(1,2):
      pixel = getPixel(pic, x, y)
      setColor(pixel, color)
  #no mouth
```

↵ *means that the statement is continued on the next line.*