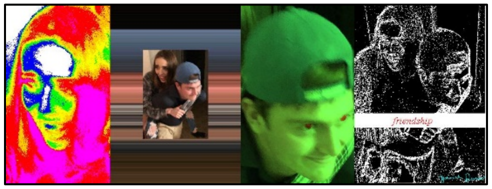
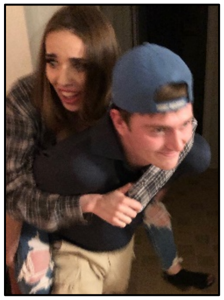
Hannah Dunco

Completed



Original



```
#Hannah Dunco's Project 2 for CS120, Spring 2020

#The final resulting image of this program is a collage made up of four panels. Since various helper
#functions are required to create the unique image in each of these panels, I have grouped my code so that
#the helper functions for each panel are together and the helper functions for the final collage function
#(the ones that create the panel images themselves) are together as well.

#general helper functions

def scale(source, ScaleFactor):
  canvas=makeEmptyPicture(int(getWidth(source)*ScaleFactor),int(getHeight(source)*ScaleFactor))
  sourceX=0
  for targetX in range(0, ScaleFactor*getWidth(source)):
    sourceY=0
    for targetY in range(0, ScaleFactor*getHeight(source)):
      px_in= getPixel(source, int(sourceX),int(sourceY))
      color=getColor(px_in)
      setColor(getPixel(canvas,targetX, targetY), color)
      sourceY=sourceY+1.00/ScaleFactor
    sourceX=sourceX+1.00/ScaleFactor
  return canvas

def complexCopy(source,canvas,targetX):
  targetX=int(targetX)
  for x in range(0,getWidth(source)):
    targetY=0
    for y in range(0,getHeight(source)):
      color=getColor(getPixel(source,x,y))
      setColor(getPixel(canvas,targetX,targetY),color)
      targetY=targetY+1
    targetX=targetX+1
  return canvas

#helper functions for first picture

#this function performs posterization, but with the goal of the resulting image having all of the
colors of the rainbow in it
def rainbowPost(source):
  for px in getPixels(source):
    r=getRed(px)
    g=getGreen(px)
    b=getBlue(px)
    luminance=(r+g+b)/6
    if luminance <12:
      setColor(px,magenta)
    elif 12<=luminance<=24:
      setColor(px,red)
    elif 24<luminance<36:
      setColor(px,orange)
    elif 36<=luminance<48:
      setColor(px,yellow)
    elif 48<=luminance<60:
      setColor(px,green)
    elif 60<=luminance<72:
      setColor(px,blue)
    else:
      setColor(px,white)
  return source

#helper functions for second picture

#this function creates a "smeared" effect by taking the colors of the pixels in a specific column
#(where x=1275) and applying them to every column in the photo
def smearLeft2Right(source, canvas, targetX, target_Y):
  for targetX in range(0,getWidth(canvas)):
    targetY=target_Y
    for SourceY in range(0,getHeight(source)):
      color=getColor(getPixel(source, 1275, SourceY))
      setColor(getPixel(canvas, targetX, targetY), color)
      targetY=targetY +1

def smudgeBackground(pic):
  canvas=makeEmptyPicture(getWidth(pic),getHeight(pic))
  smearLeft2Right(pic,canvas,0,0)
  return(canvas)

#this function shrinks our original photo down and copies it so that it is centered in our smeared
#background/frame
def copy2Frame(source,canvas,ScaleFactor):
  targetX=int(getWidth(canvas)*.5*ScaleFactor)
  for x in range(0,getWidth(source)):
    targetY=int(getHeight(canvas)*.5*ScaleFactor)
    for y in range(0,getHeight(source)):
      color=getColor(getPixel(source,x,y))
      setColor(getPixel(canvas,targetX,targetY),color)
      targetY=targetY+1
    targetX=targetX+1
  return  canvas

#helper functions for third picture

#the goal for the third panel was to make my friend look like an alien. this function achieves this
#and turns his face green by increasing the value of the green component in each pixel and decreasing
#the value of the red and blue components in each as well.
def makeGreen(pic):
  for x in range(0,getWidth(pic)):
    for y in range(0,getHeight(pic)):
      px=getPixel(pic,x,y)
      red=getRed(px)
      green=getGreen(px)
      blue=getBlue(px)
      setColor(px,makeColor(red*.5,green*1.5,blue*.5))
  return pic

#this function is a bit similar to the last, however I am tinting it red this time, and I added input
#parameters so that I could apply the effect to a select portion of the photo. I wanted just the irises
#and pupils of my friends eyes to turn red, so I found some conditions that would apply to those pixels
#only, however because of the lighting, I had to create separate functions for each eye since the different
#colors required different constraints.
def makeRedRight(pic,x1,x2,y1,y2):
  for x in range(x1,x2):
    for y in range(y1,y2):
      px=getPixel(pic,x,y)
      red=getRed(px)
      green=getGreen(px)
      blue=getBlue(px)
      if blue<63 and red<57:
        setColor(px,makeColor(red*1.8,green*.5,blue*.5))
  return pic

def makeRedLeft(pic,x1,x2,y1,y2):
  for x in range(x1,x2):
    for y in range(y1,y2):
      px=getPixel(pic,x,y)
      red=getRed(px)
      green=getGreen(px)
      blue=getBlue(px)
      if blue<63 and red<100:
        setColor(px,makeColor(red*1.8,green*.5,blue*.5))
  return pic


def copyPortion(pic,canvas,x1,x2,y1,y2):
  targetX=0
  for SourceX in range(x1,x2):
    targetY=0
    for SourceY in range(y1,y2):
      color=getColor(getPixel(pic,SourceX,SourceY))
      setColor(getPixel(canvas,targetX,targetY),color)
      targetY=targetY+1
    targetX=targetX+1
  return canvas

#helper functions for fourth picture

#edge detection is the main effect in my fourth panel, and luminance is a calculation that is necessary
#for that effect
def luminance(pixel):
  red=getRed(pixel)
  green=getGreen(pixel)
  blue=getBlue(pixel)
  return(red+green+blue)/3

def edgeDetect(source,threshold):
  for px in getPixels(source):
    x=getX(px)
    y=getY(px)
    if y<getHeight(source)-1 and x<getWidth(source)-1:
      botrt=getPixel(source,x+1,y+1)
      thislum=luminance(px)
      brlum=luminance(botrt)
      if abs(brlum-thislum)>threshold:
        setColor(px,white)
      if abs(brlum-thislum)<=threshold:
        setColor(px,black)
  return source

#helper function to add my signature

def signatureBottomLeft(background):
  signature=makePicture(getMediaPath("signature.jpg"))
  signature=scale(signature,1.000000000000/35)
  for px in getPixels(signature):
    x=getX(px)
    y=getY(px)
    if (getRed(px)<120 and getGreen(px)<120 and getBlue(px)<120):
      bgpx=getPixel(background,getWidth(background)-getWidth(signature)+x, ↵
                getHeight(background)-getHeight(signature)+y)
      setColor(bgpx,cyan)
  return background

#main function for creation of first picture

def pic1(pic):
  pic1canvas=makeEmptyPicture(675,1150)
  targetX=0
  for SourceX in range(250,925):
    targetY=0
    for SourceY in range(0,1150):
      color=getColor(getPixel(pic,SourceX,SourceY))
      setColor(getPixel(pic1canvas,targetX,targetY),color)
      targetY=targetY+1
    targetX=targetX+1
  #My original picture is rather large, so I calculated the scaling factor below for each panel to be
  #able to size them down appropriately: scalingFactor=[1000/(sum of widths of all four panels prior
  #to scaling=6862)]*[(height of original photo=2545)/(height of cropped photo for pictures 1&3,
  #2545 for 2&4)]
  picture1=scale(pic1canvas,0.322507065)
  picture1=rainbowPost(picture1)
  return picture1

#main function for creation  of second picture

def pic2(pic):
  background=smudgeBackground(pic)
  background=scale(background,0.145730108)
  original=scale(pic,.5)
  original=scale(original,0.145730108)
  pic2=copy2Frame(original,background,.5)
  return pic2

#main function for creation of third picture

def pic3(pic):
  pic3canvas=makeEmptyPicture(733,1145)
  #The x1,x2,y1,y2 values below represent the coordinates of the portion of the original picture that
  #only contain my friend's face. The line above this creates a canvas of the dimensions of that specific
  #portion so that we can copy it over, effectively "cropping" the  photo.
  pic3=copyPortion(pic,pic3canvas,915,1648,320,1465)
  pic3=scale(pic3,0.323915392)
  pic3=makeGreen(pic3)
  #The x and y ranges on the two lines below represent the coordinates containing each of my friend's eyes.
  pic3=makeRedRight(pic3,132,150,246,260)
  pic3=makeRedLeft(pic3,206,222,227,241)
  return(pic3)

#main function for creation of fourth picture
def pic4(pic):
  pic4=scale(pic,0.145730108)
  pic4=edgeDetect(pic4,10)
  addRectFilled(pic4, 0, 224, getWidth(pic)-40, 30, white)
  addTextWithStyle(pic4, 80, 246,"friendship", makeStyle(serif, italic, 20),red)
  return pic4

#main function for creation of final picture collage
def collage():
  pic=makePicture(getMediaPath("brad_is_stupid.jpg"))
  canvas=makeEmptyPicture(1000,370)
  picture1=pic1(pic)
  picture2=pic2(pic)
  picture3=pic3(pic)
  picture4=pic4(pic)
  complexCopy(picture1,canvas,0)
  complexCopy(picture2,canvas,getWidth(picture1))
  complexCopy(picture3,canvas,getWidth(picture1)+getWidth(picture2))
  complexCopy(picture4,canvas,getWidth(picture1)+getWidth(picture2)+getWidth(picture3))
  signatureBottomLeft(canvas)
  explore(canvas)
```

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