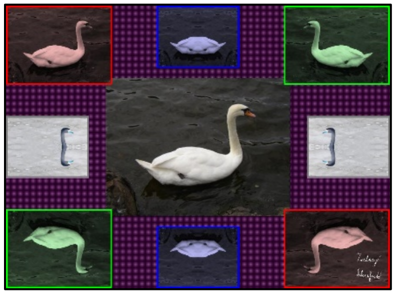
Zachary Shenefield

Completed



Original

```
#written by: Zachary Shenefield
#3/9/2020

#places an image at a certain location on the canvas
def copyImage(canvas, image, xStart, yStart):
  for x in range(getWidth(image)):
    newX = x + xStart
    for y in range(getHeight(image)):
      newY = y + yStart
      pixel = getPixel(image, x, y)
      color = getColor(pixel)
      newPixel = getPixel(canvas, newX, newY)
      setColor(newPixel, color)
  return canvas

#makes the backgorund have pixels of alternating colors
def grid(picture, color):
  height = getHeight(picture)
  width = getWidth(picture)
  for y in range(1, height, 2):
    for x in range(1, width, 2):
      p = getPixel(picture, x, y)
      setColor(p, color)
  return picture
```

```
#allows the scaling of an image
def scaleAny(source,n):
  canvas = makeEmptyPicture(int(getWidth(source)*n),int(getHeight(source)*n))
  sourceX = 0
  for targetX in range(0, getWidth(source)*n):
    sourceY = 0
    for targetY in range(0, getHeight(source)*n):
      color = getColor(getPixel(source,int(sourceX),int(sourceY)))
      setColor(getPixel(canvas,targetX,targetY), color)
      sourceY = sourceY + 1.0/n
    sourceX = sourceX + 1.0/n
  return canvas

#creates a colored border around an image
def colorBorder(borderColor, source):
  bottom = getHeight(source)-5
  end = getWidth(source)-5
  for pixels in getPixels(source):
    y = getY(pixels)
    if y < 5:
      setColor(pixels, borderColor)
    if y > bottom:
      setColor(pixels, borderColor)
    x = getX(pixels)
    if x < 5:
      setColor(pixels, borderColor)
    if x > end:
      setColor(pixels, borderColor)
  return source

#makes a picture cyanotype / blue hued
def makeBlue(picture):
  canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
  for p in getPixels(picture):
    intensity = (getRed(p)+getGreen(p)+getBlue(p))/3
    newPx = getPixel(canvas, getX(p), getY(p))
    setColor(newPx, makeColor(intensity, intensity, intensity))

  for newPx in getPixels(canvas):
    red = getRed(newPx)
    blue = getBlue(newPx)
    green = getGreen(newPx)
    if (blue < 63):
      blue = blue*2.0

    if (red > 62 and red < 192):
      blue = blue*1.3

    if (blue > 191):
      blue = blue*1.2

    setBlue(p, blue)
    red = red*0.75
    green = green*0.75
    setRed(newPx, red)
    setGreen(newPx, green)
  return canvas

#makes a picture red hued
def makeRed(picture):
  canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
  for p in getPixels(picture):
    intensity = (getRed(p)+getGreen(p)+getBlue(p))/3
    newPx = getPixel(canvas, getX(p), getY(p))
    setColor(newPx, makeColor(intensity, intensity, intensity))

  for newPx in getPixels(canvas):
    red = getRed(newPx)
    blue = getBlue(newPx)
    green = getGreen(newPx)

    if (red < 63):
      red = red*2.0

    if (blue > 62 and blue < 192):
      red = red*1.3

    if (red > 191):
      red = red*1.2

    setRed(p, red)
    blue = blue*0.75
    green = green*0.75
    setBlue(newPx, blue)
    setGreen(newPx, green)
  return canvas

#makes a picture green hued
def makeGreen(picture):
  canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
  for p in getPixels(picture):
    intensity = (getRed(p)+getGreen(p)+getBlue(p))/3
    newPx = getPixel(canvas, getX(p), getY(p))
    setColor(newPx, makeColor(intensity, intensity, intensity))

  for newPx in getPixels(canvas):
    red = getRed(newPx)
    blue = getBlue(newPx)
    green = getGreen(newPx)

    if (green < 63):
      green = green*2.0

    if (red > 62 and red < 192):
      green = green*1.3

    if (green > 191):
      green = green*1.2

    setGreen(p, green)
    red = red*0.75
    blue = blue*0.75
    setRed(newPx, red)
    setBlue(newPx, blue)
  return canvas

#makes a picture negative
def negative(picture):
  canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
  for px in getPixels(picture):
    red=getRed(px)
    green=getGreen(px)
    blue=getBlue(px)
    negColor=makeColor( 255-red, 255-green, 255-blue)
    newPx = getPixel(canvas, getX(px), getY(px))
    setColor(newPx,negColor)
  return canvas
```

```
#mirrors a picture horizontally
def mirrorHorizontal(picture, mirrorPoint):
  canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
  width = getWidth(picture)

  for x in range(0, getWidth(picture)):
    for y in range(0,getHeight(picture)):
      px = getPixel(picture, x, y)
      newPx = getPixel(canvas, x, y)
      color = getColor(px)
      setColor(newPx, color)

  for y in range(0, getHeight(picture)):
    for x in range(0, mirrorPoint):
      leftPixel = getPixel(picture, x, y)
      rightPixel = getPixel(canvas, width-x-1, y)
      color = getColor(leftPixel)
      setColor(rightPixel,color)

  return canvas

#mirrors a picture vertically
def mirrorVertical(picture, mirrorPoint):
  canvas = makeEmptyPicture(getWidth(picture), getHeight(picture))
  height = getHeight(picture)

  for x in range(0, getWidth(picture)):
    for y in range(0,getHeight(picture)):
      px = getPixel(picture, x, y)
      newPx = getPixel(canvas, x, y)
      color = getColor(px)
      setColor(newPx, color)

  for x in range(0, getWidth(picture)):
    for y in range(0, mirrorPoint):
      topPixel = getPixel(picture, x, y)
      bottomPixel = getPixel(canvas, x, height-y-1)
      color = getColor(topPixel)
      setColor(bottomPixel,color)

  return canvas
```

```
#takes non-white sections from an image and pastes them on a backgorund at specific coordinates,
#as a specified color
def chromakey(source,bg,startX,startY,color):
  for px  in getPixels(source):
    x=getX(px)
    y=getY(px)
    if (getRed(px) < 150 and getGreen(px) < 150 and getBlue(px) < 150):
      newPx = getPixel(bg, x + startX, y + startY)
      setColor(newPx,color)

#final program that creates a collage
def collage():

  canvas = makeEmptyPicture(1000,736,black)
  swan = makePicture(getMediaPath("swan.jpg"))
  signature = makePicture(getMediaPath("signature.jpg"))

  canvas = grid(canvas, magenta)

  picture = copyImage(canvas, swan, (getWidth(canvas)/2) - (getWidth(swan)/2), ↵
           (getHeight(canvas)/2) - (getHeight(swan)/2))

  smallSwan = scaleAny(swan, 0.45)
  smallSwan2 = scaleAny(swan, 0.575)

  redSwan = makeRed(smallSwan2)
  blueSwan = makeBlue(smallSwan)
  greenSwan = makeGreen(smallSwan2)
  negativeSwan = negative(smallSwan)

  redSwan = colorBorder(red, redSwan)
  blueSwan = colorBorder(blue, blueSwan)
  greenSwan = colorBorder(green, greenSwan)
  negativeSwan = colorBorder(gray, negativeSwan)

  redSwanMirror = mirrorVertical(redSwan, getHeight(redSwan))
  redSwanMirror = mirrorHorizontal(redSwanMirror, getWidth(redSwanMirror))

  greenSwanMirror = mirrorHorizontal(greenSwan, getWidth(greenSwan))
  greenSwanMirror2 = mirrorVertical(greenSwan, getHeight(greenSwan))

  blueSwanMirror = mirrorHorizontal(blueSwan, getWidth(blueSwan)/2)
  blueSwanMirror2 = mirrorVertical(blueSwanMirror, getHeight(blueSwan))
```

```
negativeSwanMirror = mirrorVertical(negativeSwan, getHeight(negativeSwan)/2)
negativeSwanMirror2 = mirrorHorizontal(negativeSwanMirror, getWidth(negativeSwan))

picture = copyImage(picture, redSwan, 0, 0)
picture = copyImage(picture, blueSwanMirror, (getWidth(canvas)/2) - (getWidth(blueSwanMirror)/2), 0)
picture = copyImage(picture, greenSwanMirror, (getWidth(canvas)) - (getWidth(greenSwanMirror)), 0)

picture = copyImage(picture, negativeSwanMirror, 0, ↲
         (getHeight(canvas)/2) - (getHeight(negativeSwanMirror)/2))
picture = copyImage(picture, negativeSwanMirror2, (getWidth(canvas) - getWidth(negativeSwanMirror2)), ↲
         (getHeight(canvas)/2) - (getHeight(negativeSwanMirror2)/2))

picture = copyImage(picture, greenSwanMirror2, 0, (getHeight(canvas)-getHeight(greenSwanMirror2)))
picture = copyImage(picture, blueSwanMirror2, (getWidth(canvas)/2) - (getWidth(blueSwanMirror2)/2), ↲
         (getHeight(canvas)-getHeight(blueSwanMirror2)))
picture = copyImage(picture, redSwanMirror, (getWidth(canvas)) - (getWidth(redSwanMirror)), ↲
         (getHeight(canvas)-getHeight(redSwanMirror)))

signature = scaleAny(signature,0.12)
chromakey(signature,picture,900,640,white)

show(picture)
```

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