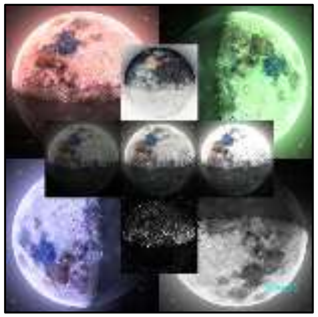
Sean Trzybinski

Completed



Originals

```
#Sean Trzybinski, October 22 2021

def collage():
  setMediaPath()
  origenal = makePicture(getMediaPath("moon.jpg"))
  mod1 = makePicture(getMediaPath("moon.jpg"))
  mod2 = makePicture(getMediaPath("moon.jpg"))
  mod3 = makePicture(getMediaPath("moon.jpg"))
  mod4 = makePicture(getMediaPath("moon.jpg"))
  mod5 = makePicture(getMediaPath("moon.jpg"))
  mod6 = makePicture(getMediaPath("moon.jpg"))
  mod7 = makePicture(getMediaPath("moon.jpg"))
  mod8 = makePicture(getMediaPath("moon.jpg"))
  canvas = makeEmptyPicture(getWidth(origenal) * 2, getHeight(origenal) * 2)
  sig = makePicture(getMediaPath("sig.jpg"))
  sig = scaleDown(sig, 20)

  pic = scaleDown(origenal, 2)

  negative(mod1)
  pic1 = scaleDown(mod1, 2)
  lighten(mod2)
  pic2 = scaleDown(mod2, 2)
  darken(mod3)
  pic3 = scaleDown(mod3, 2)
  edge(mod4)
  pic4 = scaleDown(mod4, 2)
  increaseRed(mod5)
  increaseGreen(mod6)
  pic6a = rotateleft(mod6)
  pic6b = rotateleft(pic6a)
  pic6c = rotateleft(pic6b)
  increaseBlue(mod7)
  pic7 = rotateleft(mod7)
  grayScaleNew(mod8)
  pic8a = rotateleft(mod8)
  pic8b = rotateleft(pic8a)

  copy(mod5, canvas, 0, 0)
  copy(pic6c, canvas, getWidth(mod6), 0)
  copy(pic7, canvas, 0, getHeight(mod7))
  copy(pic8b, canvas, getWidth(mod8), getHeight(mod8))
  copy(pic1, canvas, int(getWidth(canvas) / 2 - getWidth(pic1) / 2), getHeight(pic1)/2)
  copy(pic2, canvas, int(getWidth(canvas) / 2 + getWidth(pic1) / 2), int(getHeight(canvas) / 2 - getHeight(pic1)/2))
  copy(pic3, canvas, int(getWidth(canvas) / 2 - getWidth(pic1)*(3.0/2)), int(getHeight(canvas) / 2-getHeight(pic1)/2))
  copy(pic4, canvas, int(getWidth(canvas) / 2 - getWidth(pic1) / 2), int(getHeight(canvas) / 2 + getHeight(pic1)/2))

  copy(pic, canvas, int(getWidth(canvas) / 2 - getWidth(pic) / 2), int(getHeight(canvas) / 2 - getHeight(pic)/2))
  chromaSig(sig, canvas, 1000, 1050)
  Final = scaleDown(canvas, 1.7)
  show(Final)
  writePictureTo(canvas, "Sean_Trzybinski.jpg")

def rotateleft(src):
  targetX = 0
  width = getWidth(src)
  canvas = makeEmptyPicture(getHeight(src), getWidth(src))
  for sourceX in range(0, getWidth(src)):
    targetY = 0
    for sourceY in range(0, getHeight(src)):
      color = getColor(getPixel(src, sourceX, sourceY))
      setColor(getPixel(canvas, targetY, width - targetX - 1), color)
      targetY += 1
    targetX += 1
  return(canvas)

def copy(pic, target, targX, targY):
  targetX = targX
  for x in range(getWidth(pic)):
    targetY = targY
    for y in range(getHeight(pic)):
      pixel = getPixel(pic, x, y)
      tx = getPixel(target, targetX, targetY)
      setColor(tx, getColor(pixel))
      targetY  += 1
    targetX += 1

def scaleDown(pic, factor):
  canvas = makeEmptyPicture(int(getWidth(pic) / factor), int(getHeight(pic) / factor))
  scale(pic, canvas, 1.0 / factor)
  return canvas

def scale(src, canvas, factor):
  sourceX = 0
  for targetX in range(0, int(getWidth(src) * factor)):
    sourceY = 0
    for targetY in range(0, int(getHeight(src) * factor)):
      color = getColor(getPixel(src, int(sourceX), int(sourceY)))
      setColor(getPixel(canvas, targetX, targetY), color)
      sourceY = sourceY + 1.0 / factor
    sourceX = sourceX + 1.0 / factor


def negative(pic):
  for each_pixel in getPixels(pic):
    r = getRed(each_pixel)
    b = getBlue(each_pixel)
    g = getGreen(each_pixel)
    neg = makeColor(255-r, 255-g, 255-b)
    setColor(each_pixel, neg)

def lighten(pic):
  for each_pixel in getPixels(pic):
    color = getColor(each_pixel)
    color = makeLighter(color)
    setColor(each_pixel, color)

def darken(pic):
  for p in getPixels(pic):
    color = getColor(p)
    color = makeDarker(color)
    color = makeDarker(color)
    setColor(p, color)

def edge(picture):
  for p in getPixels(picture):
    x = getX(p)
    y = getY(p)
    if y < getHeight(picture) - 1 and x < getWidth(picture) - 1:
      sum = getRed(p) + getGreen(p) + getBlue(p)
      botrt = getPixel(picture, x+1, y+1)
      sum2 = getRed(botrt) + getGreen(botrt) + getBlue(botrt)
      diff = abs(sum2-sum)
      newColor = makeColor(diff, diff, diff)
      setColor(p, newColor)

def increaseRed(picture):
  for pixel in getPixels(picture):
    value = getRed(pixel)
    setRed(pixel, value * 1.5)

def increaseGreen(picture):
  for pixel in getPixels(picture):
    value = getGreen(pixel)
    setGreen(pixel, value * 1.5)

def increaseBlue(picture):
  for pixel in getPixels(picture):
    value = getBlue(pixel)
    setBlue(pixel, value * 1.5)

def grayScaleNew(picture):
  for p in getPixels(picture):
    newRed = getRed(p) * .299
    newGreen = getGreen(p) * .587
    newBlue = getBlue(p) * .114
    luminance = newRed + newGreen + newBlue
    setColor(p, makeColor(luminance, luminance, luminance))

def chromaSig(source, target, targetX, targetY):
  for x in range(0, getWidth(source)):
    for y in range(0, getHeight(source)):
      p = getPixel(source, x, y)
      color = getColor(p)
      targ = getPixel(target, x + targetX, y +targetY)
      if distance (black, color) < 190:
        setColor(targ, makeColor(0, 255,255))

collage()
```