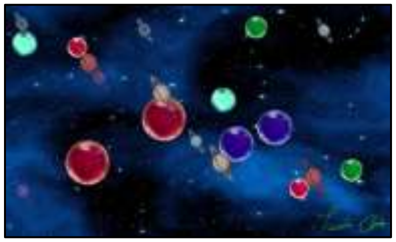
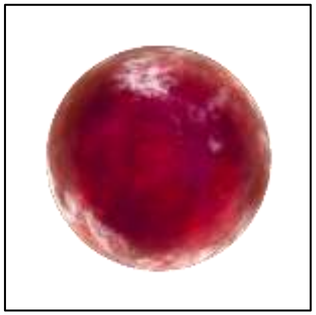
Jacob Clark

Completed



Originals



```
#Jacob Clark 10/20/2
import random
def collage():
  setMediaPath()
  bg=makePicture(getMediaPath("Space BackGround.jpg"))
  canvas2 = makeEmptyPicture(1000,600)
  canvas = scaleDown2(bg,3,5)
  copy(canvas, canvas2, 0,0)
  sig=makePicture(getMediaPath("sig.jpg"))

  planet_list=[]
  planet_list.append(scaleDown(makePicture(getMediaPath("planet.png")),random.randrange(4,10)))
  planet_list.append(scaleDown(makePicture(getMediaPath("planet.png")),random.randrange(4,10)))
  planet_list.append(scaleDown(makePicture(getMediaPath("planet.png")),random.randrange(4,10)))
  planet_list.append(scaleDown(makePicture(getMediaPath("planet.png")),random.randrange(4,10)))
  planet_list.append(scaleDown(makePicture(getMediaPath("planet.png")),random.randrange(4,10)))

  saturn_list=[]
  saturn_list.append(scaleDown(makePicture(getMediaPath("saturn.jpg")),random.randrange(4,10)))
  saturn_list.append(scaleDown(makePicture(getMediaPath("saturn.jpg")),random.randrange(4,10)))
  saturn_list.append(scaleDown(makePicture(getMediaPath("saturn.jpg")),random.randrange(4,10)))
  saturn_list.append(scaleDown(makePicture(getMediaPath("saturn.jpg")),random.randrange(4,10)))
  saturn_list.append(scaleDown(makePicture(getMediaPath("saturn.jpg")),random.randrange(4,10)))

  lighten(planet_list[1])
  swapRedBlue(planet_list[2])
  swapRedGreen(planet_list[3])
  negative(planet_list[4])

  negative(saturn_list[1])
  grayScale(saturn_list[2])
  makeBlue(saturn_list[2])
  grayScale(saturn_list[3])
  makeRed(saturn_list[3])
  swapRedBlue(saturn_list[4])

  number = 2
  #number = requestInteger("How many of each planet")
  while number > 0:
    copyFocus(planet_list[0],canvas,random.randrange(getWidth(canvas)-getWidth(planet_list[0]))
,random.randrange(getHeight(canvas)-getHeight(planet_list[0])),70)
    copyFocus(planet_list[1],canvas,random.randrange(getWidth(canvas)-getWidth(planet_list[1]))
,random.randrange(getHeight(canvas)-getHeight(planet_list[1])),70)
    copyFocus(planet_list[2],canvas,random.randrange(getWidth(canvas)-getWidth(planet_list[2]))
,random.randrange(getHeight(canvas)-getHeight(planet_list[2])),60)
    copyFocus(planet_list[3],canvas,random.randrange(getWidth(canvas)-getWidth(planet_list[3]))
,random.randrange(getHeight(canvas)-getHeight(planet_list[3])),60)
    copyFocusNeg(planet_list[4],canvas,random.randrange(getWidth(canvas)-getWidth(planet_list[4]))
,random.randrange(getHeight(canvas)-getHeight(planet_list[4])),400)

    copyFocus(saturn_list[0],canvas,random.randrange(getWidth(canvas)-getWidth(saturn_list[0]))
,random.randrange(getHeight(canvas)-getHeight(saturn_list[0])),70)
    copyFocusNeg(saturn_list[1],canvas,random.randrange(getWidth(canvas)-getWidth(saturn_list[1]))
,random.randrange(getHeight(canvas)-getHeight(saturn_list[1],)),277)
    copyFocus(saturn_list[2],canvas,random.randrange(getWidth(canvas)-getWidth(saturn_list[2]))
,random.randrange(getHeight(canvas)-getHeight(saturn_list[2])),70)
    copyFocus(saturn_list[3],canvas,random.randrange(getWidth(canvas)-getWidth(saturn_list[3]))
,random.randrange(getHeight(canvas)-getHeight(saturn_list[3])),70)
    copyFocus(saturn_list[4],canvas,random.randrange(getWidth(canvas)-getWidth(saturn_list[4]))
,random.randrange(getHeight(canvas)-getHeight(saturn_list[4])),70)
    number -= 1

  sig=scaleDown(sig, 5)
  chromaSig(sig, canvas, getWidth(canvas)-getWidth(sig), getHeight(canvas)-getHeight(sig))
  show(canvas)
  writePictureTo(canvas, "Collage.jpg")

def copy(source, target, targX, targY):
  targetX = targX
  for sourceX in range(0,getWidth(source)):
    targetY = targY
    for sourceY in range(0,getHeight(source)):
      px = getPixel(source, sourceX, sourceY)
      tx = getPixel(target, targetX, targetY)
      setColor(tx,getColor(px))
      targetY=targetY+1
    targetX=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 scaleDown2(pic, factorX, factorY):
  canvas = makeEmptyPicture(int(getWidth(pic)/factorX), int(getHeight(pic)/factorY))
  scale2(pic, canvas, 1.0 / factorX, 1.0 / factorY)
  return canvas

def scale2(src, canvas, factorX, factorY):
  sourceX = 0
  for targetX in range(0, int(getWidth(src)*factorX)):
    sourceY=0
    for targetY in range(0, int(getHeight(src)*factorY)):
      color=getColor(getPixel(src, int(sourceX), int(sourceY)))
      setColor(getPixel(canvas, targetX, targetY),color)
      sourceY=sourceY + 1.0 / factorY
    sourceX=sourceX + 1.0 / factorX

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

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

def copyFocus(source, target, targetX, targetY, factor):
  for x in range(0, getWidth(source)):
    for y in  range(0, getHeight(source)):
      px = getPixel(source, x, y)
      color= getColor(px)
      targ=getPixel(target, x + targetX, y + targetY)
      if distance(black, color) > factor:
        setColor(targ,color)

def copyFocusNeg(source, target, targetX, targetY, factor):
  for x in range(0, getWidth(source)):
    for y in  range(0, getHeight(source)):
      px = getPixel(source, x, y)
      color= getColor(px)
      targ=getPixel(target, x + targetX, y + targetY)
      if distance(black, color) < factor:
        setColor(targ,color)

def chromaSig(source, target, targetX, targetY):
  for x in range(0, getWidth(source)):
    for y in  range(0, getHeight(source)):
      px = getPixel(source, x, y)
      color= getColor(px)
      targ=getPixel(target, x + targetX, y + targetY)
      if distance(black, color) < 170:
        setColor(targ,green)

def grayScale(picture):
  for px in getPixels(picture):
    newRed=getRed(px)*0.299
    newGreen=getGreen(px)*0.587
    newBlue=getBlue(px)*0.114
    luminance=newRed+newGreen+newBlue
    setColor(px, makeColor(luminance, luminance, luminance))

def makeBlue(picture):
  for p in getPixels(picture):
    value=getRed(p)
    setRed(p, value*0.5)
    value=getGreen(p)
    setGreen(p, value*0.5)


def makeRed(picture):
  for p in getPixels(picture):
    value=getBlue(p)
    setBlue(p, value*0.5)
    value=getGreen(p)
    setGreen(p, value*0.5)

def swapRedBlue(picture):
  for p in getPixels(picture):
    blueValue=getBlue(p)
    redValue=getRed(p)
    setBlue(p, redValue)
    setRed(p, blueValue)


def swapRedGreen(picture):
  for p in getPixels(picture):
    greenValue=getGreen(p)
    redValue=getRed(p)
    setGreen(p, redValue)
    setRed(p, greenValue)
collage()
```