Ethan Andreshak

#10/20/2023 Ethan Andreshak

def collage():
    xnum=0  # where to begin placing on canvas regarding the width
    ynum=0  # where to begin placing on canvas regarding the height
    mfactV=2  # where to mirror the picture regarding the x value (vertical mirroring)
    mfactH=2  # where to mirror the picture regarding the y value (horizontal mirroring)
    pic1=makePicture(getMediaPath("Cuddly.jpg"))
    s1factW=333.3/getWidth(pic1)  # make the scale width of the picture set to exactly 1/3 of the width of the canvas
    s1factH=245.3/getHeight(pic1)  # make the scale height of the picture set to exactly 1/3 of the height of the canvas
    scaledpic1=makeEmptyPicture(int(getWidth(pic1)*s1factW),int(getHeight(pic1)*s1factH))
    pic2=makePicture(getMediaPath("Door.jpg"))
    s2factW=333.3/getWidth(pic2)  # make the scale width of the picture set to exactly 1/3 of the width of the canvas
    s2factH=245.3/getHeight(pic2)  # make the scale height of the picture set to exactly 1/3 of the height of the canvas
    scaledpic2=makeEmptyPicture(int(getWidth(pic2)*s2factW),int(getHeight(pic2)*s2factH))
    canvas=makeEmptyPicture(999,736,black)
    scale(1.0/s1factW,1.0/s1factH,pic1,scaledpic1)  # works, this is to scale the first pic selected
    scale(1.0/s2factW,1.0/s2factH,pic2,scaledpic2)  # works, this is to scale the second pic selected
    TopLeft_Pic=duplicatePicture(scaledpic1)
    TopLeft_PicX=duplicatePicture(TopLeft_Pic)
    TopMiddle_Pic=duplicatePicture(scaledpic2)
    TopMiddle_PicO=duplicatePicture(TopMiddle_Pic)
    TopRight_Pic=duplicatePicture(scaledpic1)
    TopRight_PicO=duplicatePicture(TopRight_Pic)
    MiddleLeft_Pic=duplicatePicture(scaledpic2)
    MiddleLeft_PicO=duplicatePicture(MiddleLeft_Pic)
    MiddleMiddle_Pic=duplicatePicture(scaledpic1)
    MiddleMiddle_PicX=duplicatePicture(MiddleMiddle_Pic)
    MiddleRight_Pic1=duplicatePicture(scaledpic2)
    MiddleRight_Pic2=duplicatePicture(MiddleRight_Pic1)
    BottomLeft_Pic=duplicatePicture(scaledpic1)
    BottomLeft_PicX=duplicatePicture(BottomLeft_Pic)
    BottomMiddle_Pic1=duplicatePicture(scaledpic2)
BottomMiddle_Pic2 = duplicatePicture(BottomMiddle_Pic1)
BottomRight_Pic = duplicatePicture(scaledpic1)
BottomRight_PicX = duplicatePicture(BottomRight_Pic)

sepiatone(TopLeft_Pic) # works
yellowtint(TopLeft_PicX) # works
cyanotype(TopMiddle_PicO) # works
lighter(TopMiddle_Pic) # works
mirrorH(mfactH, TopRight_PicO) # works
mirrorV(mfactV, TopRight_PicO) # works
edgedetect(4, TopRight_Pic, darkGray, red) # works
greentint(MiddleLeft_PicO) # works
darker(MiddleLeft_Pic) # works
edgedetect(4, MiddleMiddle_PicX, orange, lightGray) # works
lighter(MiddleRight_Pic2) # works
greyscale(MiddleRight_Pic1) # works
darker(MiddleRight_Pic1) # works
darker(BottomLeft_PicX) # works
purpletint(BottomLeft_PicX) # works
mirrorH(mfactH, BottomMiddle_Pic2) # works
mirrorV(mfactV, BottomRight_PicX) # works
sepiatone(BottomMiddle_Pic1) # works
darker(BottomMiddle_Pic1) # works
cyanotype(BottomRight_Pic) # works
darker(BottomRight_Pic) # works
crosses(TopLeft_PicX) # works, makes the X's in tic-tac-toe
crosses(MiddleMiddle_PicX)
crosses(BottomLeft_PicX)
crosses(BottomRight_PicX)
circles(TopMiddle_PicO) # works, makes the O's in tic-tac-toe
circles(TopRight_PicO)
circles(MiddleLeft_PicO)
grid(MiddleRight_Pic1, white) # works, puts grid on picture
grid(BottomMiddle_Pic1, white)
chromakey(TopLeft_PicX, TopLeft_Pic, 255, 255, 255) # chromakey magic ;)
chromakey(MiddleMiddle_PicX, MiddleMiddle_Pic, 255, 255, 255)
chromakey(BottomLeft_PicX, BottomLeft_Pic, 255, 255, 255)
chromakey(BottomRight_PicX, BottomRight_Pic, 255, 255, 255)
chromakey(TopMiddle_PicO, TopMiddle_Pic, 255, 255, 255)
chromakey(TopRight_PicO, TopRight_Pic, 255, 255, 255)
chromakey(MiddleLeft_PicO, MiddleLeft_Pic, 255, 255, 255)
chromakey(MiddleRight_Pic1, MiddleRight_Pic2, 255, 255, 255)
chromakey(BottomMiddle_Pic1, BottomMiddle_Pic2, 255, 255, 255)
copy(xnum, ynum, TopLeft_PicX, canvas) # works, this is to set the picture to the top left corner
copy(xnum+getWidth(scaledpic1), ynum, TopMiddle_PicO, canvas) # works, this is to set the picture to the top middle
copy(xnum+getWidth(scaledpic1)*2, ynum, TopRight_PicO, canvas) # works, this is to set the picture to the top right
copy(xnum, ynum+getHeight(scaledpic1), MiddleLeft_Pic0, canvas) # works, this is to set the picture to the middle left
# works, this is to set the picture to the center of the canvas

# works, this is to set the picture to the middle right

# works, this is to set the picture to the bottom left

# works, this is to set the picture to the bottom middle

# works, this is to set the picture to the bottom right

TicTacToe(canvas)
signature1=makePicture(getMediaPath("Signature1.png")) # hopefully the signature is appropriate for you
signature2=makePicture(getMediaPath("Signature2.png"))
canvasSans=duplicatePicture(canvas)
copy(getWidth(canvas)/2-(getWidth(signature1)/2), getHeight(canvas)/3, signature1, canvas)
copy(getWidth(canvas)/2-(getWidth(signature2)/2), getHeight(canvas)/3*2-50, signature2, canvas)
chromakey(canvas, canvasSans, 255, 255, 255)
explore(canvas)

def copy(n1, n2, pic_in, pic_out):# copies the pictures onto the new final canvas
    w=getWidth(pic_in)
    h=getHeight(pic_in)
    tX=n1
    for sX in range(0, w):
        tY=n2
        for sY in range(0, h):
            c=getColor(getPixel(pic_in, sX, sY))
            setColor(getPixel(pic_out, tX, tY), c)
            tY=tY+1
        tX=tX+1
    return(pic_out)

def scale(nW, nH, pic_in, pic_out):# works with any pic and scale
    w=getWidth(pic_out)
    h=getHeight(pic_out)
    sX=0
    for tX in range(0, w):
        sY=0
        for tY in range(0, h):
            c=getColor(getPixel(pic_in, int(sX), int(sY)))
            setColor(getPixel(pic_out, tX, tY), c)
            sY=sY+nH
        sX=sX+nW
    return(pic_out)
def blur(source, trg):  # this does work, even if first glance it doesn't
    for x in range(1, getWidth(source) - 1):
        for y in range(1, getHeight(source) - 1):
            top = getPixel(source, x, y - 1)
            left = getPixel(source, x - 1, y)
            bottom = getPixel(source, x, y + 1)
            right = getPixel(source, x + 1, y)
            center = getPixel(trg, x, y)
            newR = (getRed(top) + getRed(left) + getRed(bottom) + getRed(right) + getRed(center)) / 5
            newG = (getGreen(top) + getGreen(left) + getGreen(bottom) + getGreen(right) + getGreen(center)) / 5
            newB = (getBlue(top) + getBlue(left) + getBlue(bottom) + getBlue(right) + getBlue(center)) / 5
            setColor(center, makeColor(newR, newG, newB))
    return(trg)

def lighter(pic):  # makes the picture lighter
    for x in range(0, getWidth(pic)):
        for y in range(0, getHeight(pic)):
            p = getPixel(pic, x, y)
            c = getColor(p)
            setColor(p, makeLighter(c))
    return(pic)

def darker(pic):  # makes the picture darker
    for x in range(0, getWidth(pic)):
        for y in range(0, getHeight(pic)):
            p = getPixel(pic, x, y)
            c = getColor(p)
            setColor(p, makeDarker(c))
    return(pic)

def mirrorV(n, pic):  # mirrors the picture along the x axis
    mp = getWidth(pic) / n
    w = getWidth(pic)
    for y in range(0, getHeight(pic)):
        for x in range(0, int(mp)):
            lp = getPixel(pic, x, y)
            rp = getPixel(pic, w - x - 1, y)
            c = getColor(lp)
            setColor(rp, c)
    return(pic)

def mirrorH(n, pic):  # mirrors the picture along the y axis
    mp = getHeight(pic) / n
    h = getHeight(pic)
    for y in range(0, getHeight(pic)):
        for x in range(0, int(mp)):
            lp = getPixel(pic, x, y)
            rp = getPixel(pic, x, h - y - 1)
            c = getColor(lp)
            setColor(rp, c)
    return(pic)
for x in range(0, getWidth(pic)):
    for y in range(0, int(mp)):
        tp=getPixel(pic, x, y)
        bp=getPixel(pic, x, h-y-1)
        c=getColor(tp)
        setColor(bp, c)
        return(pic)

def luminance(pixel): #luminance for edgedetect
    r=getRed(pixel)
    g=getGreen(pixel)
    b=getBlue(pixel)
    return (r+g+b)/3

def edgecolor(source, n): #puts a single color along the border of the picture
    c=pickAColor()
    for p in getPixels(source):
        x=getX(p)
        y=getY(p)
        if x<n or y<n:
            setColor(p, c)
        if x>getWidth(source)-n or y>getHeight(source)-n:
            setColor(p, c)
    return(source)

def TicTacToe(source): #makes the hashtag grid for tic-tac-toe
    for y in range(8, getHeight(source)-8):
        for x in range(getWidth(source)/3-10, getWidth(source)/3+10):
            setColor(getPixel(source, x, y), pink)
        for x in range(getWidth(source)/3*2-10, getWidth(source)/3*2+10):
            setColor(getPixel(source, x, y), cyan)
    for x in range(8, getWidth(source)-8):
        for y in range(getHeight(source)/3-10, getHeight(source)/3+10):
            setColor(getPixel(source, x, y), yellow)
        for y in range(getHeight(source)/3*2-10, getHeight(source)/3*2+10):
            setColor(getPixel(source, x, y), orange)
    return(source)

def edgedetect(n, source, c1, c2): #detects certain color values of pixels and decides to set it to one of two colors
    pixels=getPixels(source)
    for p in pixels:
        x=getX(p)
        y=getY(p)
        if y<getHeight(source)-1 and x<getWidth(source)-1:
            botrt=getPixel(source, x+1, y+1)
thislum=luminance(p)
brlum=luminance(botrt)
if abs(brlum-thislum)>n:
      setColor(p,c1)
if abs(brlum-thislum)<=n:
      setColor(p,c2)
return(source)
def purpletint(pic):#tints the picture purple
greyscale(pic)
for p in getPixels(pic):
    r=getRed(p)
    g=getGreen(p)
    b=getBlue(p)
    if (r<63):
        r=(r*1.5)
    elif (63<=r<=191):
        r=(r*1.2)
    elif (r>191):
        r=(r*1.1)
    setGreen(p,g*.3)
    setRed(p,r)
    setBlue(p,b*1.2)
return(pic)
def yellowtint(pic):#tints the picture yellow
greyscale(pic)
for p in getPixels(pic):
    r=getRed(p)
    g=getGreen(p)
    b=getBlue(p)
    if (g<63):
        g=(g*1.5)
    elif (63<=g<=191):
        g=(g*1.2)
    elif (g>191):
        g=(g*1.1)
    setGreen(p,g)
    setRed(p,r*1.2)
    setBlue(p,b*.3)
return(pic)
def greentint(pic):#tints the picture green
greyscale(pic)
for p in getPixels(pic):
def sepiatone(pic): # tints the picture red/brown
greyscale(pic)
for p in getPixels(pic):
    r = getRed(p)
    g = getGreen(p)
    b = getBlue(p)
    if (r < 63):
        r = (r * 1.5)
    elif (63 <= r <= 191):
        r = (r * 1.2)
    elif (r > 191):
        r = (r * 1.1)
    setRed(p, r)
    setBlue(p, b * .8)
    setGreen(p, g * .8)
return(pic)

def cyanotype(pic): # tints the picture blue
greyscale(pic)
for p in getPixels(pic):
    r = getRed(p)
    g = getGreen(p)
    b = getBlue(p)
    if (b < 63):
        b = (b * 2)
    elif (63 <= b <= 191):
        b = (b * 1.3)
    elif (b > 191):
        b = (b * 1.2)
    setBlue(p, b)
    setRed(p, r * .75)
setGreen(p, g*.75)
return(p)

def greyscale(pic):  # tints the picture grey
    for p in getPixels(pic):
        intensity=(getRed(p)+getGreen(p)+getBlue(p))/3
        setColor(p, makeColor(intensity, intensity, intensity))
    return(pic)

# background magic, works like a green/blue screen for the signature and the X's and O's
def chromakey(pic_for, pic_back, rValue, gValue, bValue):
    for p in getPixels(pic_for):
        x = getX(p)
        y = getY(p)
        if (getRed(p) == rValue and getGreen(p) == gValue and getBlue(p) == bValue):
            backp = getPixel(pic_back, x, y)  # background pixel at x, y
            backc = getColor(backp)  # background color at background pixel
            setColor(p, backc)  # sets a pixel to the background color
    return(pic_for)

def crosses(pic):  # this is the X's for tic-tac-toe
    for x in range(getWidth(pic)/5, (int(getWidth(pic)/3.5)+12)):
        a = 0
        for y in range(getHeight(pic)/5, int(getHeight(pic)/1.25)):
            setColor(getPixel(pic, int(x+a), y), white)
            a = a + 1
    for x in range(int(getWidth(pic)/1.55-2), int(getWidth(pic)/1.3-3)):
        a = 0
        for y in range(getHeight(pic)/5, int(getHeight(pic)/1.25)):
            setColor(getPixel(pic, x-a, y), white)
            a = a + 1
    return(pic)

def circles(pic):  # this is the original formula for drawing the circle, it is the O's for tic-tac-toe
    x = getWidth(pic)/2
    y = getHeight(pic)/2
    n = 10
    r = 80-(n/2)
    for count in range(0, 360):  # more x's and y's mean fuller circle
        x1 = x + r*(math.sin(math.radians(count)))
        y1 = y + r*(math.cos(math.radians(count)))
        x2 = (x+1) + r*(math.sin(math.radians(count)))
        y2 = (y+1) + r*(math.cos(math.radians(count)))
        x3 = (x-1) + r*(math.sin(math.radians(count)))
        y3 = (y-1) + r*(math.cos(math.radians(count)))
for thick in range(0,n):
    setColor(getPixel(pic,int(x1)+thick,int(y1)+thick),white)
    setColor(getPixel(pic,int(x1)-thick,int(y1)-thick),white)
    setColor(getPixel(pic,int(x2)+thick,int(y2)+thick),white)
    setColor(getPixel(pic,int(x2)-thick,int(y2)-thick),white)
    setColor(getPixel(pic,int(x3)+thick,int(y3)+thick),white)
    setColor(getPixel(pic,int(x3)-thick,int(y3)-thick),white)
for p in getPixels(pic):
    x=getX(p)
    y=getY(p)
    # this rounds out the circle and fills in spots to make it look less spotty
    if (getRed(p)==255 and getGreen(p)==255 and getBlue(p)==255):
        setColor(getPixel(pic,x+3,y),white)
        setColor(getPixel(pic,x,y+3),white)
        setColor(getPixel(pic,x+6,y),white)
        setColor(getPixel(pic,x,y+6),white)
        setColor(getPixel(pic,x+9,y),white)
        setColor(getPixel(pic,x,y+9),white)
return(pic)

def grid(pic,c):# striped stuff
    for x in range(0,getWidth(pic),4):
        for y in range(0,getHeight(pic),4):
            setColor(getPixel(pic,x,y),c)
    return(pic)