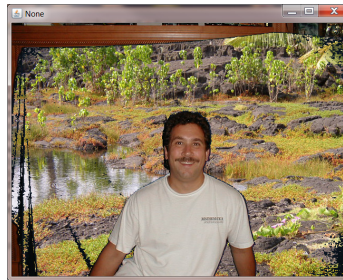


Today we'll have a hands-on lab to further familiarizing you with using nested for-loops to loop over a partial picture, so **pairs of students** should get a laptop from the cart.

Download to the Desktop and extract the files needed for the lab from:

<http://www.cs.uni.edu/~fienup/cs1120s15/sessions/s14/lec14.zip>

Part A. *Background subtraction* and *Chromakey* both replace a known background with an image. For example, the below image of Mark Guzdial in front of a blue sheet is Chromakeyed into the jungle or Eiffel tower pictures.



The textbook function that performs the Chromakey background addition is (and downloaded in a `lec14/blueChromakey.py`):

```
def chromakeyPictures(foregroundPict, newBackgroundPict):
    width = min(getWidth(foregroundPict), getWidth(newBackgroundPict))
    height = min(getHeight(foregroundPict), getHeight(newBackgroundPict))
    target = makeEmptyPicture(width, height)

    for x in range(getWidth(target)):
        for y in range(getHeight(target)):
            targetPixel = getPixel(target, x, y)
            foregroundPixel = getPixel(foregroundPict, x, y)
            newBackgroundPixel = getPixel(newBackgroundPict, x, y)
            if getRed(foregroundPixel) + getGreen(foregroundPixel) < getBlue(foregroundPixel):
                setColor(targetPixel, getColor(newBackgroundPixel))
            else:
                setColor(targetPixel, getColor(foregroundPixel))

    return target    # returns the Chromakey picture
```

For this lab, I want you to generalize this code so it works with any background color (i.e., not just blue) in the `foregroundPict` picture. You can do this by passing additional parameters `backgroundColor` and `tolerance` as in

```
def chromakeyPictures(foregroundPict, newBackgroundPict, backgroundColor, tolerance):
```

In the `if`-statement you'll want to use the built-in JES `distance` function to compute the "closeness" between the `foregroundPict` pixel's color and the `backgroundColor` parameter. If the two colors are close enough (less than or equal to the `tolerance` parameter), then set the `targetPixel` to the `newBackgroundPict` pixel's color.

NOTE: The blue background in the `blue-mark.jpg` picture is approximately (R = 10, G = 32, B = 56) and a tolerance of about 30 should work.

Partial code can be found in the downloaded `lec14/generalChromakey.py`

After you complete Part A, raise your hand and demonstrate your program on the pictures `lec14/blue-mark.jpg` and `lec14/jungle.jpg`