CPS109 Lab 6

Most of the questions in this lab come from Chapter 5 and Chapter 6 (or earlier chapters) of the course text, Introduction to Computing and Programming in Python, by Guzdial and Ericson. Please put your answers (numbered) in a document and submit it on D2L as a PDF file. Other formats are not accepted.

The **learning objectives** for Chapter 5 include:

- to implement controlled color changes, like redeye removal and sepia
- to use background subtraction and replacement
- to use chromakey for putting in a new background
- to draw borders
- to use conditionals with if, elif, else statements

The **learning objectives** for Chapter 6 include:

- to mirror pictures
- to copy pictures, to make a collage, to scale and rotate
- to use nested loops
- to loop through just part of a two-dimensional array
- to use print statements to help in debugging

To do:

1) Write a function **blueAndGold(picture)**, which puts a blue border on the top and bottom of the picture, and gold borders on the left and right. The border width is 10. Apply your function to a picture of your choosing. Include in your document your code and the picture with the border.
2) Write a function **purpleTeeth(picture)** which starts with a picture of someone you know, and colors the person's teeth purple. You will have to explore the picture first to restrict the range, and then change the color to purple if the color is close enough to white.
3) Write a function **checkLuminance(r, g, b)**, where r, g, b are values of the components red, green and blue of a color. The function computes the luminance as the weighted average of r, g, b (weights: .3, .6, .1, respectively), and then prints a warningmessage if the luminance is in certain ranges:

- If the luminance is less than 10, then you print "That's going to be awfully dark."
- If the luminance is between 50 and 200, then print "Looks like a good range."
- Over 250: "That's nearly white".

4) Write a function **chromakeyBlueAbove(picture, background, skyY)**, which uses the rule

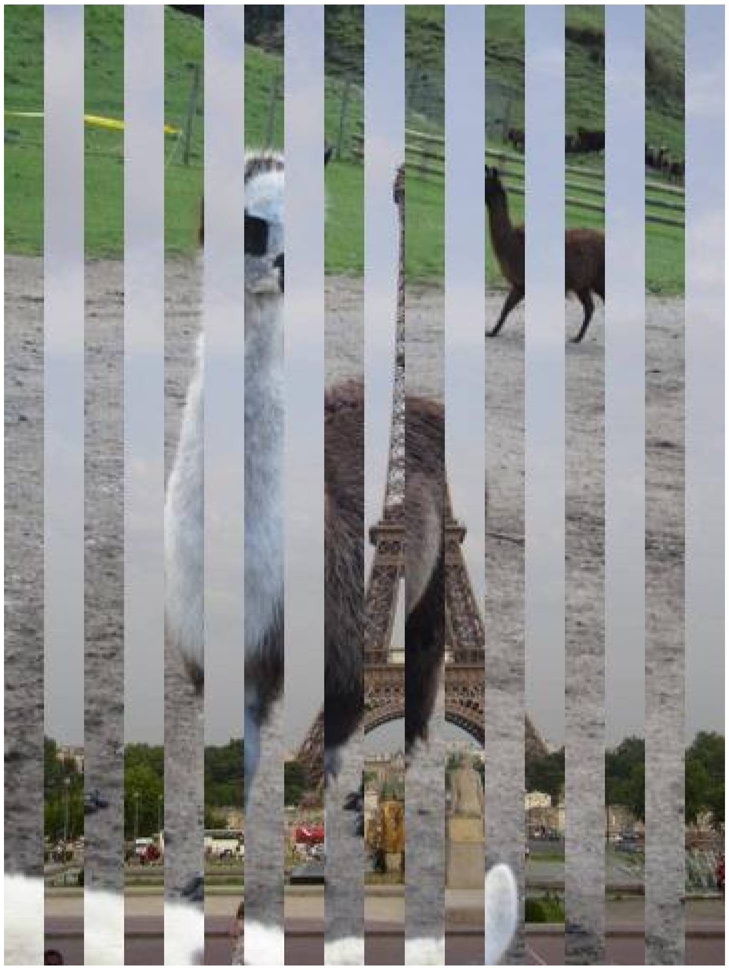
- if getRed(pixel) + getGreen(pixel) < getBlue(pixel) + 100, and
- y < skyY, then swap the pixel color for the background color. Try out your function on statue-tower.jpg and the moon background and with a value for skyY that you choose based on exploring the picture. Include your resulting statue picture and your code in your document.

5) Write a function **interleave(picture1, picture2)**, which interleaves picture1 and picture2 in the following way. Assume that picture1 and picture2 have the same dimensions, with width W and height H. Let **canvas = makeEmptyPicture(W, H)**. Make the color of

- the first 20 pixels of the canvas come from the first 20 pixels of picture1, and

← the second 20 pixels of the canvas come from the second 20 pixels of picture2, and
← the third 20 pixels of the canvas come from the third 20 pixels of picture1, and
← so on

Apply your program to any two pictures that are the same size. Put your code and the resulting picture into your document. For example, try it on eiffel.jpg and llama.jpg, and you should get the following, (but you have to do it on two other pictures for your document):



6) Write a function **getRegion(picture, x1, y1, x2, y2)**, which copies the indicated region, a box with top left corner (x1, y1) and bottom right corner (x2, y2), onto a canvas the same size as picture. Apply your function to extract just the head of the caterpillar. Show your code and the resulting picture in your document.
7) Write a function **getRegionTwice(picture, x1, y1, x2, y2)**, which copies the indicated region, a box with top left corner (x1, y1) and bottom right corner (x2, y2), twice onto a canvas which has size as large as two of those boxes, one sitting on the other. Apply your function to extract two copies of the caterpillar's head. Show your code and the resulting picture in your document.