

Assignment 1 for CPS109

Objective:

The purpose of this assignment is to give you practice writing and calling functions in python. The assignment is based on what we have learned from Chapters 1-4 in our course text, Introduction to Computing and Programming with Python, by M Guzdial and B Ericson. To do the assignment, you must use functions from the JES environment, such as `getPixels()`, and you must not use any functions from any other environment nor functions that require imports other than `math`.

To do:

Write twenty (20) new functions for drawing on or modifying a picture. Use all of your functions on one or more images and show the results. The image(s) that you use can come anywhere, including from the book image bank (provided on D2L and in the lab).

What to submit:

Submit on D2L a document (which must be in pdf format) that includes a description of your work, your python code with a short explanation of what each function does, the input picture(s) and the output picture(s). Thus, you are submitting just one file, which is a pdf document.

You must work alone and produce original work, but you are free to discuss with other students regarding what they are doing and how, and you can ask TAs and professors for ideas or help.

Example

Python code

The following contains four or five functions which you can use and count in your 20 required functions.

```
# A start on the required program for assignment a1 in cps109.  
# The object is to write and use 20 functions for modifying a picture.  
# The functions are based on material from Chapters 1-4 of the course text.
```

```
# I am modifying the llama picture from the course resources, but  
# you can use any picture of your own or from the media resources.
```

Function 1

```
# This function puts a red box on the llama  
# First explore to find the coordinates  
def addBox(picture) :  
    pixels = getPixels(picture)  
    for pixel in pixels :  
        x = getX(pixel)  
        y = getY(pixel)  
        if 171 < x < 233 and 152 < y < 188 :  
            setColor(pixel, red)
```

Function 2

Function addBox2 is a parameterization so that we can put a
box of a given color anywhere

```
def addBox2(picture, x1, y1, x2, y2, color) :  
    print 'finish'
```

Function 3

Add a circle of a given radius and color

```
def addCircle(picture, xc, yc, radius, color) :  
    pixels = getPixels(picture)  
    for pixel in pixels :  
        x = getX(pixel)  
        y = getY(pixel)  
        distance = math.sqrt((x - xc)**2 + (y - yc)**2)  
        if distance < radius :  
            setColor(pixel, color)
```

Function 4

Draw a black line on the picture, given two points defining the line

Given two points on a line: (x1, y1), (x2, y2)

the slope is $m = (y2 - y1) / (x2 - x1)$

the intercept is $b = y1 - m * x1$

the line is $y = m * x + b$

```
def addLine(picture, x1, y1, x2, y2) :  
    pixels = getPixels(picture)  
    m = 1.0 * (y2 - y1) / (x2 - x1)  
    b = y1 - m * x1  
    for pixel in pixels :  
        x = getX(pixel)  
        y = getY(pixel)  
        y_line = m * x + b  
        if y == y_line :  
            setColor(pixel, black)
```

Function 5

Draw a line on the picture, given two points defining the line

and a parameter defining the thickness, and a parameter for color

Given two points on a line: (x1, y1), (x2, y2)

the slope is $m = (y2 - y1) / (x2 - x1)$

the intercept is $b = y1 - m * x1$

the line is $y = m * x + b$

```
def addLine2(picture, x1, y1, x2, y2, thickness, color) :  
    pixels = getPixels(picture)  
    m = 1.0 * (y2 - y1) / (x2 - x1)  
    b = y1 - m * x1  
    for pixel in pixels :  
        x = getX(pixel)  
        y = getY(pixel)
```

```
y_line = m * x + b
distance = int(abs(y_line - y) + 0.5)
if distance < thickness / 2 :
    setColor(pixel, color)
```

Below we are calling the functions and saving the picture

```
f_llama = '/home/eharley/1/llama.jpg'
llama = makePicture(f_llama)

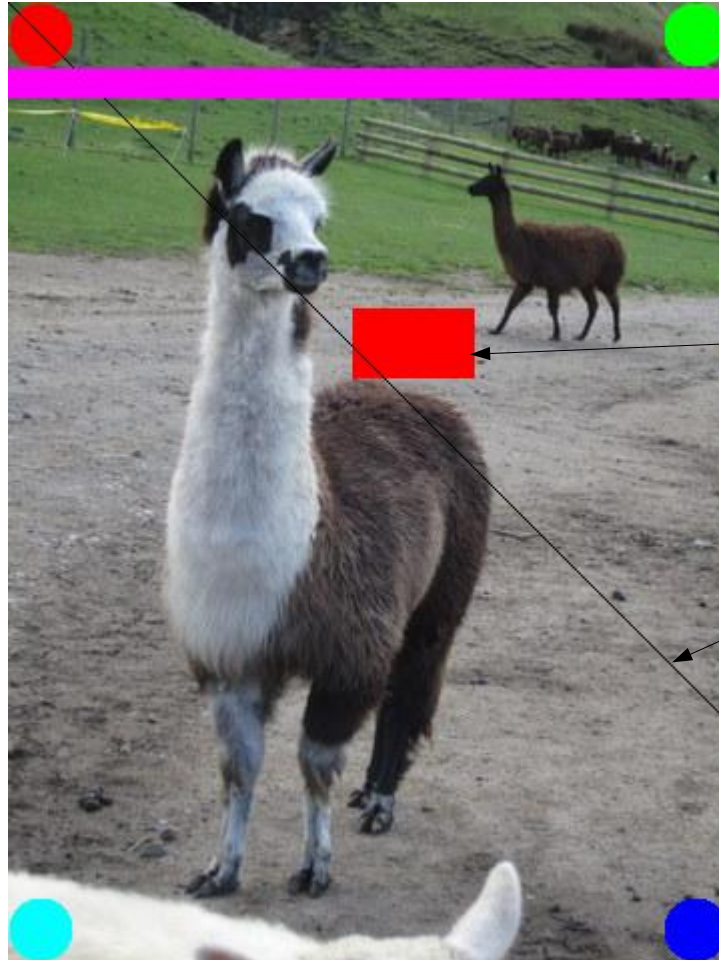
addBox(llama)
addCircle(llama, 16, 16, 16, red)
W = getWidth(llama)
H = getHeight(llama)
addCircle(llama, W-17, 16, 16, green)
addCircle(llama, W-17, H-17, 16, blue)
addCircle(llama, 16, H-17, 16, cyan)
addLine(llama, 0, 0, 10, 10)
addLine2(llama, 0, 40, 10, 40, 16, magenta)
explore(llama)
writePictureTo(llama, '/home/eharley/1/a1.jpg')
```

Input Image (llama.jpg from the book resources)



Output image:

3. addCircle



5. addLine2

1. addBox

4. addLine

Marking rubric (out of 10):

2 points for documentation of your functions, including readability and understandability.

5 points for your functions -- that they are present, appear to work and are original.

3 points for esthetics of your picture -- that it looks like you put thought into the application of your functions. Part of this mark is for showing in your document (as above) arrows to the picture indicating where you used the 20 functions.

Academic Integrity Reminder:

You have to work alone when writing your code and your document. You can discuss ideas with your classmates, but you cannot copy code or develop code together (and then change wording). If you write your own code, then it will appear unique. If your code looks strangely similar to someone else's code, then this will affect your mark negatively. If the code or ideas are so similar that the marker is quite sure there is unfair collaboration, then there might be unfortunate charges of academic misconduct.