CS 112 Lab Assignment

Instructor: Dan Fleck
Lab: Turtle Graphics Function Library

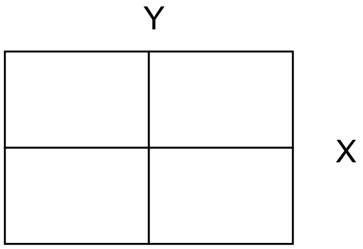
Overview

This lab will familiarize you with creating functions in Python that take parameters. In addition you will learn a little about the Turtle graphics library.

Turtle graphics was used in the 1960s primarily in the Logo language. It is a very simple and clear graphics programming language. Today I do not think it is used for anything except teaching, but will provide a good basis to write functions as part of a program.

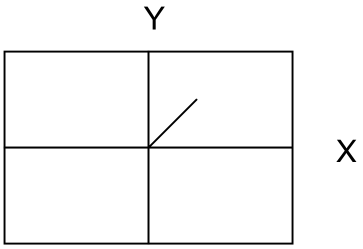
The fundamental idea is that you are controlling a turtle walking around a screen. The turtle can turn, go forward, put the pen down (draw while moving) and pick the pen up (do not draw). Python supports turtle graphics using the turtle library. See the library reference for detailed information about this library you will need to complete this lab.

Think of the screen as a grid like this:



The middle is (0,0). Moving right is positive X, left is negative. Up is positive Y, down is negative. Each increment is a single dot on the screen (usually your screen has at least 1024x768 "dots" (normally called pixels). Your window will be smaller than that, but it's still pretty big.

Telling the turtle who starts at position 0,0 to go to position 50,50 would result in something like this:



Using the turtle module, you can tell the turtle where to go. If the turtle has the pen down, then while the turtle walks it will draw a line. If the pen is up no line will be drawn while the turtle walks.
Additionally you can set the color of the lines using turtle.color(r, g, b). Where "R" is the amount of red to draw from 0-1. B is Blue, G is green. You can make any color using percentages of these. For example, "Yellow" is (1, 1, 0).

You can also look up the turtle documentation in the Python Global Module index.

More information about the origins of Turtle Graphics can be seen at http://en.wikipedia.org/wiki/Turtle_graphics.

In this lab you will be given a test program that calls functions. The functions must be defined in a module called turtle_shapes.py. This is the module you will write. The test program (frequently called a driver) will execute your program and you can then verify the results.

There is also some sample code that will help show you how to do the various drawings. You need to use functions though!!

Helper code is at:
http://cs.gmu.edu/~dfleck/classes/cs112/spring08/labs/turtle/turtle_sample.py

Assignment

You will write a Python module called turtle_shapes.py that has at least the following functions (you can add more if you see a need, but these are required):

# Draw a line from points (x1, y1) to (x2, y2) using the current settings.
def drawLine(x1, y1, x2, y2):

# Draw a house with the bottom left corner at x, y
# If x and y are -1, -1, then assume the user wants the
# current location, and do not move the turtle
# Color can be either "red", "green" or "blue"
# Width is the total width of the house and is the height of the "body" of the house.
# The roof should go above the house 0.5*width
#
# For example, if the width is 10, the house is a 10x10 square, with a triangle on top that is 5 pixels high.
def drawHouse(x, y, width, color):

```
# This function draws a stick person with a specific size at location x, y.
# If x,y are both 'c' then the stick figure is drawn at the 'c'urrent location
# x, y, is the location of the bottom left "foot".
# The width from foot to foot is 50
# The height from foot to top of leg is 30
# The torso length is 20
# The arms are each 15 long
# The head is a circle with radius 10
# x, y is the bottom left location (left foot)
def drawStickPerson(x, y, color):
```

Other Requirements

In ALL functions, if the color is not valid you should return ‘invalid color’ from the function and not draw anything. Our valid colors are only the strings ‘red’, ‘green’, and ‘blue’

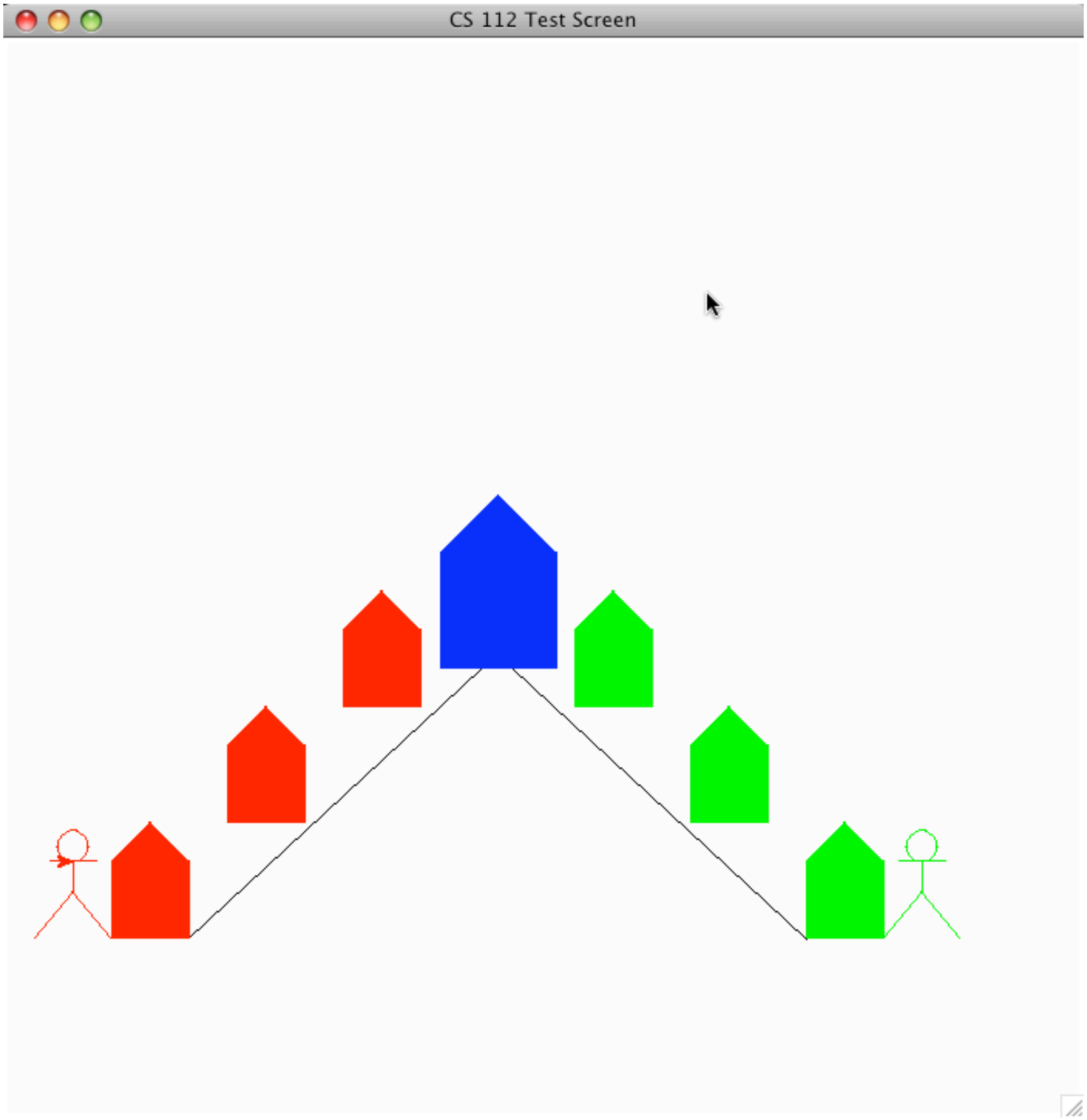
In ALL functions that have ‘c’urrent as an option for the location, if the user has ‘c’ for one value, but not the other, you should return ‘invalid location’ from the function and not draw anything. So you need an if statement to check:

```
if both are ‘c’
        do not move the turtle
else if either are ‘c’
        that means one must be and the other is not… return an error
else
        neither are c, so move the turtle (without drawing) to the correct location
```

Turtle Driver is located at
http://cs.gmu.edu/~dfleck/classes/cs112/spring08/labs/turtle/turtle_driver.py

Output from turtle_driver

```
Error in drawHouse : invalid location
Error in drawStickPerson : orange is not a color
```



What to turn in:

1. Python source code