

Lab 6: The DS4 Equalizer

Objectives:

- Practice top-down program design, problem solving in C
- Practice writing functions to a specification
- Emphasize the importance of branching and looping structures
- Practice using output parameters

Starting Point:

- lab6.c

Process:

Creating a New Folder

Create a new folder named *lab6* in your *cpre185labs* folder on the U: drive. You will want to copy over ds4rd.exe to the *lab6* folder.

NOTE: You should use the "-g" version of the ds4rd.exe command line from previous labs.

Problem

You are working on using the DualShock 4 as a data source for a game, but you are having trouble visualizing the readings from the DualShock 4 as a function of time. It would be nice to plot a bar graph in real time showing either the roll or pitch of the DualShock 4, but you realize that you are in a hurry and don't want to deal with getting the graphics code right. Suddenly, you realize that you can graph these variables as a horizontal bar chart using characters on the screen and then let the lines scroll down the screen to form a moving bar graph.

For our purposes, **roll** is defined as the angle the DualShock 4 is **tipped left or right while holding it normally**.

Pitch is the angle from level that the DualShock 4 is **tipped forward or backward**.

In this lab, you will use skeleton code as your starting point to design and implement a modular DualShock 4 graphing application. Download the skeleton code lab6.c. You must use the skeleton code's function and basic structure to implement your program. **Carefully read the skeleton program before beginning. It may help you focus your strategy.** Take your time and carefully

write and test individual functions in lab.

Because both roll and pitch may be positive or negative, you will need a bar graph where 0 is halfway across the screen at column 40. In the sample output below, if the value is 0 (or not enough to justify printing a character) output a 0 at column 40. If the value is greater than 0, output r's at column 40 and to the right. The number should be proportional to the value being graphed and such that $\pi/2$ maps to 39 characters. Similarly, the l's should be used when the value is less than 0.

```

                                0
                                rrrrrrr
                                rrrr
                                r
                        |||||
                    |||||
                |||||
            |||||
```

The above output would show a graph of 0, a slight positive trend of 3 positive readings, and 3 increasingly negative readings. When run, the program should graph roll. To switch to roll, the user should press the TRIANGLE button. To switch to pitch, the user should press the X button. To stop the program, the user should press the SQUARE button. **(BONUS:** Advanced option (5 points): toggle between roll and pitch with a single button. This is optional and harder to do reliably than you might think. **To earn the bonus points, you must demonstrate your code before next week's lab.)**

When you have your program working, have your undergraduate TA check off your source code and include a copy of it in your lab report. You must demonstrate your code this week to your TA or Peer Mentor to earn the bonus points.

Extra Criteria

Only one printf and scanf are allowed in this lab. The scanf has to be in the function read_line() and the printf has to be in the function print_chars(). The functions read_line() and print_chars() may be called multiple times.

Questions and Experiments

1. How did you scale your values? Write an equation and justify it.
2. How many degrees does each letter in your graph represent? This is the precision of your graph. As your experiment with the roll and pitch, what do you notice about the graph's behavior near the limits of its values?

Equations:

$$\text{roll} = \text{asin}(g_x)$$

$$\text{pitch} = \text{asin}(g_z)$$

Turn-in:

This is a one week lab. You will submit your lab report by next week and include answers to the all questions in for this week's lab and all your source code that has been checked by your undergraduate TA.