THE EXQUALIZER

LAB 7

SECTION X

SUBMITTED BY:

NATHAN SHULL

SUBMISSION DATE:

NOVEMBER 3, 2016

Problem

The purpose of this lab, the exqualizer, is to create a program, which reads input, and output commands that will calculate the roll and pitch of the esplora. It will read that amount and divide it into 80 characters ranging from left to right. Depending on the scale it will print equivalent characters on the screen. The point of this lab is to practice output parameters/branching and looping structures.

Analysis

The problem stated that I must print l(for left) or r(for right) or 0 depending on the acceleration values of the esplora. Therefore, I have to create a scale that will divide the value and make it fit the screen. I also had to write pitch and roll functions using the equations $a\sin(a_x)$ and $a\sin(a_y)$.

Design

The problem was to finish writing a skeleton code, so the basic outline of the program was already there for me. I first decided to work on the read_acc function because in my opinion it is the most important, as this is how I got values to come into the program. Looking at the comment, all I Needed to do was add a scanf statement, and an if statement. Following that, pitch and roll are both calculated the same way only pitch uses y values and roll uses x values. Once I read the mag, I had to make sure it didn't go above 1 or less than -1. Then I had to use the asin fuction. To switch between the pitch and roll, I included Boolean, and was able to switch between the two buttons using true and false.

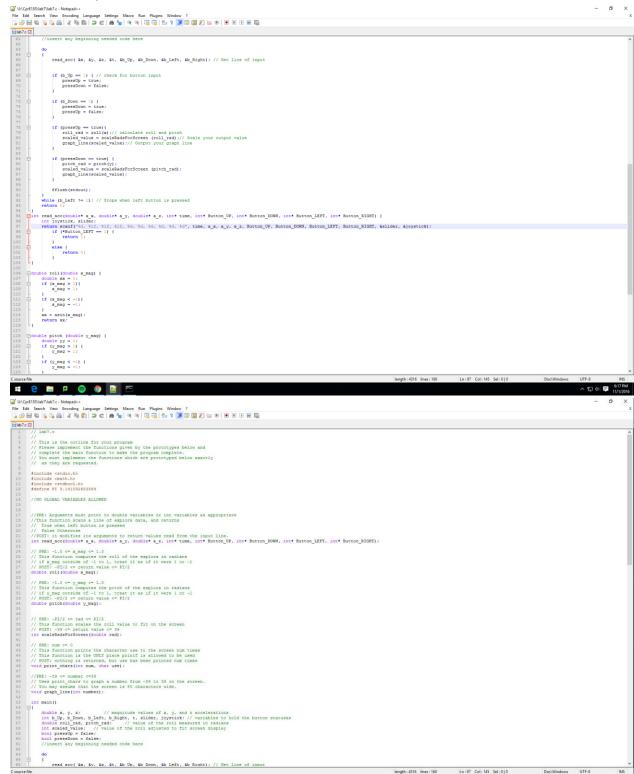
Testing

In order to verify the results of the solution, I had to try the roll/pitch functions separately first and I saw that they both worked. However, the way I switched between them wasn't working correctly. This is when I decided to add Boolean, and went back to lab5 where I also used Boolean to switch between loops. Afterwards, this fixed the problem and the program ran successfully.

Comments

- 1. How did you scale your values? Write an equation and justify it: (rad / (PI/2) * 39;
 - I divided the radians by (PI/2) because that't the maximum value of each side. PI/2 is equal to 90 degrees, which when multiplies by two is 180 degrees, and that covers the whole area. Then I multiplied it by 39. That will return plus or minus 39, +39 being right characters, or -39 being left characters.
- 2. How many degrees does each letter in your graph represent? This is the precision of your graph. As you experiment with roll and pitch, what do you notice about the graph's behavior near the limits of its values.

It's 2.3769231 degrees. I found that by dividing 90 degrees by 39. When the graph is near the limit, it fills the 39 characters.



^ \$\bullet \phi \bullet \quad \bullet \quad \bullet \quad \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \quad \bullet \quad \bullet \quad \quad \quad \bullet \quad \qu

🤚 🗎 P 同 💿 📓 🖭