Conditionals Lab

LAB # 5
SECTION #2

FULL NAME : Jaden Burke

SUBMISSION DATE: 10/3/2022

DATE 10/3/2022

Problem

I must create a program in C that will take outputted data from a DualShock 4 controller and use it to do things. Specifically, I need to output the controllers orientation based on it's gyroscopic data without repeating orientation and stop running the program if the triangle button is pressed.

Analysis

My first thought was to inspect the excel sheets and the graphs that were created during lab 3 to get an idea of what values I needed to check against to see how the controller is orientated. After that I need to make a tolerance function since the values outputted by the controller are hardly static. This can be done using math functions from the math library as well as using the < operator in an if statement. Next up is the output of the orientation which is pretty simple in that I just need to output UP, DOWN, etc depending on the value of the gyroscopic x, y, and z. Since I need to ensure that I don't keep printing the same orientation, I need to create a holder for the previous orientation to check against. The last step is making the triangle button stop the function, which will just require an if statement and a break.

Design

After finding that the gyroscopic x, y , or z need to be equal to 1 or -1 in order for it to be one of the required orientations, all I had to do was make a function that outputted which orientation it is given the x,y, and z. Before I could do that though, I designed a tolerance function. This was done just by taking three values, the tolerance, point, and value, and returning true if the absolute value of the value-point was less than the tolerance. Next was just returning the orientation depending on if x,y, or z was equal to 1 or -1 with tolerance. After that I created a simple holder variable to hold the previous orientation in order to make sure I don't reprint the same orientation a ton of times. This was used in an if statement to check whether or not the orientation needed to be reprinted. Finally was just having what I now realize to be an overcomplicated way of just stopping the while loop when the triangle button is pressed. This was done by assigning a variable to the while loop condition, and then constantly assigning it to the opposite of the value of the triangle button state variable.

Testing

When I initially tested this I ran into quite a few issues. My first issue was that I mistakenly used abs() instead of fabs() when comparing doubles which caused the wrong thing to be outputted in my tolerance function. After fixing that, my next issue came from trying to not reprint the same orientation. Originally, the function that I used to check the orientation returned a string, however I ran into issues with that. This was due to the fact that when it was moving, it is not any of the orientations, but the function still needed to return something. This meant that the holder variable I was using would be filled with an empty string and would cause the orientation to be reprinted depending on how stable I held the controller in some orientations. This was fixed by changing the function to return an integer, and then having a set of if

statements in the while loop to print its orientation. The triangle button step ran as expected though, which was nice.

Comments

Question 1: Answered in the design header above

Question 2: the data that I had to read in my function was the gyroscopic data from the DualShock Controller

Question 3: I chose to implement an orientation checker function, a tolerance function, and a stop function. I chose to have an orientation function in order to easily recheck what the orientation was given the gyroscopic values that were read from the controller. The tolerance function was created in order to check tolerance in a quick and easy manner since it would need to be called several times. I also created a stop function which in hindsight was unnecessary, however it was mainly made to make the code that was in main more compact. It was used to read whether the triangle button was pressed and then stop the while loop.

Question 4: I only ever chose to use one tolerance value which was .05. It was kind of an arbitrary decision, but I knew that I wanted a value lower than .1 due to wanting pretty high precision in calculations as to whether something was oriented in a certain way or not. With too big or a tolerance value, the function would have returned an orientation even if it was orientated all the way correctly.

Screen Shots