

LAB REPORT

LAB 5 SECTION C

SUBMITTED BY:

SCOTT VLASIC

SUBMISSION DATE:

FEBRUARY 23, 2017

Problem:

The main problem in this lab was looking at the values the esplora was printing and assigning those values to what we wanted printed. So when I was holding the esplora in a certain orientation, I wanted the program to print either TOP, BOTTOM, RIGHT, LEFT, FRONT, or BACK depending on the orientation of the esplora. Therefore I had to look at the acceleration values and interpret them to what I wanted as the output.

Analysis:

To begin this lab I started by opening lab5.c on Notepad++ so I could edit the code and being solving the problem. I had to look at the source code given and also copy over the magnitude function from lab 4 so I could calculate the proper orientation of the esplora. Then when looking at the values I had to determine if they were within the certain range for each orientation. Then I had to assign the the orientation to the proper values only when the esplora was sitting still. I also had to make it so the program would stop running when I pressed the up button on the esplora.

Design:

To begin solving the problem, I first had to import the code for my magnitude from lab 4 so that the while loop had values to reference. I then had to write a close_to function so that the program could properly read the tolerance, point, and values and then assign them to a position. I did this by creating an if statement that said if value \geq (point-tolerance) and value \leq (point+tolerance), then return true. Else return false. This tested to see which position would be assigned to a value based on the information given. The next step was writing the if statements for each position that the esplora was in. the first if loop looked at a tolerance of 0.10, a point of 1.00, and then the value being the magnitude of(ax, ay, az). Inside of this if statement were 6 more if and if else statements corresponding to each position on the esplora. For example, if the close_to value had a tolerance of 0.20, a point of 1.00, all for the value ay, then the program would print TOP since the parameters were met. To make the function stop when I pressed the up button, I modified the inside of the while loop so that whenever the button was !b2 the loop ran, but as soon as I pressed the button, it stopped. This is similar to a true/false design.

Testing:

To test this lab, I had to ask the TA's which orientation was correct at first since there was a certain side that needed to be facing up. It seemed almost inverted to me so after I asked them I would run the code and see if it matched up to what they wanted. If it wasn't, the fix wasn't too difficult as I just modified which points were in a certain if() statement and that fixed the issue. I had this problem with the left and right positions so I just changed the printf() statement to align with the proper orientation. Here's an output of my code below.

```
svlasic@co1313-33 /cygdrive/u/cpre185/lab5
$ ./explore.exe -p COM5 -t -a -b | ./lab5.exe
FRONT
BOTTOM
BACK
TOP
RIGHT
BACK
RIGHT
BACK
LEFT
BACK
TOP
BACK
BOTTOM
FRONT
```

Comments:

This has been the most difficult lab so far this semester for me, so I spent extra time with the TA's trying to get help on it. In all, I feel much more confident with if() statements as well as writing my own functions. To answer the questions, I approached the design by looking at what values I needed for each position and writing them on paper. I then put them inside of separate if() statements and tested them as I went along to see if they oriented correctly. I read in the data that was the magnitude of the esplora in different positions. I chose to implement the functions of mag and close_to because for one I needed to be able to reference the magnitudes so that I could assign a position value to the esplora. Also I needed close_to so that I could have a range for each position on the esplora. I chose the tolerance values by reading the values the esplora was outputting for the printf() mag function and seeing how when I moved the esplora the values changed.

Implementation:

```
/* Lab 5 Wrapper Program */
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#define TRUE 1
```

```
/* Put your lab 4 functions prototypes here, as well as the prototype for lab 5 */
```

```
double mag(double ax, double ay, double az);
```

```
double close_to(double tolerance, double point, double value);
```

```
int main(void) {
```

```
    int t, b1, b2, b3, b4, b5, s;
```

```
    double ax, ay, az;
```

```
    int steady;
```

```
    while (!b2) {
```

```
        scanf("%d, %lf, %lf, %lf, %d, %d, %d, %d, %d", &t, &ax, &ay, &az, &b1,  
&b2, &b3, &b4, &b5, &s );
```

```
        if(close_to(0.10, 1.00, mag(ax, ay, az))) {
```

```
            if(close_to(0.20, 1.00, ay)) {
```

```
                if(steady != 1) {
```

```
                    steady = 1;
```

```
                    printf("TOP\n");
```

```
                }
```

```
            }
```

```
            else if(close_to(0.20, -1.00, ay)) {
```

```
                if(steady != 2) {
```

```
                    steady = 2;
```

```
                    printf("BOTTOM\n");
```

```
                }
```

```
            }
```

```

        else if(close_to(0.20,1.00,az)){
            if(steady != 3){
                steady = 3;
                printf("FRONT\n");
            }
        }

        else if(close_to(0.20,-1.00,az)){
            if(steady != 4){
                steady = 4;
                printf("BACK\n");
            }
        }

        else if(close_to(0.20,1.00,ax)){
            if(steady != 5){
                steady = 5;
                printf("RIGHT\n");
            }
        }

        else if(close_to(0.20,-1.00,ax)){
            if(steady != 6){
                steady = 6;
                printf("LEFT\n");
            }
        }
    }
}

return 0;
}

/*    CODE SECTION 2 */
    //printf("Echoing output: %d, %lf, %lf, %lf, %d, %d, %d, %d, %d, %d\n", t, ax,
ay, az, b1, b2, b3, b4, b5, s);

/*    CODE SECTION 1 */

```

```
//printf("At %d ms, the acceleration's magnitude was: %lf\n", t, mag(ax, ay, az));
```

```
/* You would be wise (mainly save time) if you copy your code to calculate the  
magnitude from last week  
(lab 4) to this section. You will also need to copy your prototypes and functions  
to the appropriate sections  
in this program. */
```

```
/* Put your lab 4 functions here, as well as your new function close_to */
```

```
double mag(double ax, double ay, double az){  
    return sqrt(ax*ax + ay*ay + az*az);  
}
```

```
double close_to(double tolerance, double point, double value){  
    if(value >= (point-tolerance) && value <= (point+tolerance)){  
        return 1;  
    }  
    else{  
        return 0;  
    }  
}
```