Lab 4 - What's Up

LAB 2

SECTION C

James Mechikoff

SUBMISSION DATE: 9/25/2018

DATE: 9/24/2018

Problem

The problem this week required inputs to be taken from the dual shock 4 remote controller and run through conditional statements to output a desired output based on the orientation of the Dual Shock 4 controller. It also required three new functions to be made.

Analysis

The problem has two parts, the first one requiring the user to orient the controller in a specific direction and have the program output it based on the conditions it is given. The second part is using three new functions developed by the creator of the programmer.

Design

The design for the first part of the problem is simple if else statements using the close to function that was developed in the second part. It also uses our magnitude function that was created back in lab 3 to tell if it is moving and using the button input to check if the program needs to end. The second part of the problem is making three new functions. The designs used in the lab were the end function to end the program, the close to function to check if as value is close to something or not, and the fun button function just to check if your pressing another button on the controller as a small Easter egg.

Testing

The testing done for this was a bit more extensive than past labs, there was testing for tolerance to find what is an okay value to input for magnitude and for checking how close a number is to the desired orientation. This required a lot of checking and re-reading values. Then for the other functions, it required a lot of just checking where they are. The conditionals had a lot of retrying to be done to make sure they are executing properly.

Comments

I really loved this lab and started to learn some real-world input for the dual shocks. I already knew and have a decent amount of experience with conditional statements, so it was a fun review for those. It is always great to learn more about functions since you start to learn even more real world applications for them.

Source Code

```
/*---------------------------------------------------------------------------
--
-                                       SE 185 Lab 04
-             Developed for 185-Rursch by T.Tran and K.Wang
-     Name:       James Mechikoff
-     Section:    C
-     NetID:            726219551
-     Date:       9/18/2018
-----------------------------------------------------------------------------
*/

/*---------------------------------------------------------------------------
--
-                                 Includes
-----------------------------------------------------------------------------
*/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>


/*---------------------------------------------------------------------------
--
-                                 Defines
-----------------------------------------------------------------------------
*/
#define TRUE 1
#define FALSE 0

/*---------------------------------------------------------------------------
--
-                                 Prototypes
-----------------------------------------------------------------------------
*/
double mag(double x, double y, double z);
int close_to(double tolerance, double point, double value);
int endProgram(int b);
int funBut(int fB);

/*---------------------------------------------------------------------------
--
-                                           Implementation
-----------------------------------------------------------------------------
*/
int main(void) {
    int t, b1, b2, b3, b4, fB;
    double ax, ay, az, gx, gy, gz;
      int facing = -1;


    while (TRUE) {
        scanf("%d, %lf, %lf, %lf, %lf, %lf, %lf, %d, %d, %d, %d", &t, &ax,
&ay, &az, &gx, &gy, &gz, &b1, &b2, &b3, &b4 );
```

```
        /* printf for observing values scanned in from ds4rd.exe, be sure to
comment or remove in final program */
        //printf("Echoing output: %d, %lf, %lf, %lf, %lf, %lf, %lf, %d, %d,
%d, %d \n", t, ax, ay, az, gx, gy, gz, b1, b2, b3, b4);

        /* It would be wise (mainly save time) if you copy your code to
calculate the magnitude from last week
         (lab 3).  You will also need to copy your prototypes and functions
to the appropriate sections
         in this program. */

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

            //printf("gx = %lf\n", gx);

            if(close_to(.05, 0.0, mag(ax, ay, az))){

                  if(close_to(0.15,-1.0, gx) && facing !=0){

                        printf("RIGHT\n");

                        facing = 0;

                  }
                  else if(close_to(0.15,1.0,gx) && facing != 1){

                        printf("LEFT\n");

                        facing = 1;

                  }
                  else if(close_to(0.15,1.0, gy) && facing !=2){

                        printf("TOP\n");

                        facing = 2;

                  }
                  else if(close_to(0.15,-1.0,gy) && facing !=3){

                        printf("BOTTOM\n");

                        facing = 3;

                  }
                  else if(close_to(0.15,1.0, gz) && facing !=4){

                        printf("BACK\n");

                        facing = 4;

                  }
                  else if(close_to(0.15,-1.0,gz) && facing !=5){

                        printf("FRONT\n");
                        facing = 5;

                  }
            }

            if(endProgram(b1)==TRUE){
                  break;
            }

            if(funBut(b3) == 1){

                  printf("Stop pressing X...\n");

            }
}
return 0;
}


/* Put your functions here */

int endProgram(int b){
      if (b == 1){
            return TRUE;
      } else {
            return FALSE;
      }
}
double mag(double x, double y, double z) {
    return sqrt(pow(x,2 ) + pow(y, 2) + pow(z, 2));
}

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

int funBut(int fB){
      if (fB == 1){

            return 1;

      }
}
```