

Smoothing

LAB 8

SECTION 2

Gavin Monroe

SUBMISSION DATE:

11/07/2017

Problem

Smooth the axis of x, y, and z after a inputted number from the user. I solved this by using the example given. It took me three tries but I got it figured out by talking to a fellow classmate.

Analysis

We need use arrays to hold data, and then replace and move the array around to make sure we can calculate the new average for each axis. We use 3 different inputs for 3 different axis's.

Design

We use three different functions to solve, calculate and solve the problem that we face. Then too find the max and min I use a simplistic way to find that by using a loop with a set already max and min. For the update buffer I used of some code from a classmate but modified it.

Testing

I had to redo my code three different times until I got it right. And for the demo I didn't have enough time, and when I went to go demo it on Friday all the controllers were gone so I couldn't demo it. I got the code done shortly after class though. I had to solve and redo my code three different times because the lab instructions weren't written that well.

Comments

Please be more clear in the instructions

Source Code

```
// SE 185 Lab 8
#include <stdio.h>

#define MAXPOINTS 10000

// compute the average of the first num_items of buffer
double avg(double buffer[], int num_items);

//update the max and min of the first num_items of array
void maxmin(double array[], int num_items, double* max, double* min);

//shift length-1 elements of the buffer to the left and put the
//new_item on the right.
void updatebuffer(double buffer[], int length, double new_item);

int main(int argc, char* argv[]) {

    /* DO NOT CHANGE THIS PART OF THE CODE */
    double x[MAXPOINTS], y[MAXPOINTS], z[MAXPOINTS];
    int lengthofavg = 0;
    if (argc>1) {
        sscanf(argv[1], "%d", &lengthofavg );
        printf("You entered a buffer length of %d\n", lengthofavg);
    }
    else {
        printf("Enter a length on the command line\n");
        return -1;
    }
    if (lengthofavg <1 || lengthofavg >MAXPOINTS) {
        printf("Invalid length\n");
        return -1;
    }
    /* PUT YOUR CODE HERE */
    int on = 1;
    double gx, gy, gz;
    double _maxX, _maxY, _maxZ;
    double _minX, _minY, _minZ;

    do{
        scanf("%lf, %lf, %lf", &gx, &gy, &gz);
        updatebuffer(x, lengthofavg, gx);
        updatebuffer(y, lengthofavg, gy);
        updatebuffer(z, lengthofavg, gz);
        maxmin(x, lengthofavg, &_amp;maxX, &_amp;minX);
        maxmin(y, lengthofavg, &_amp;maxY, &_amp;minY);
        maxmin(z, lengthofavg, &_amp;maxZ, &_amp;minZ);
        printf("%lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf, %lf\n",
        gx, gy, gz, avg(x, lengthofavg), avg(y, lengthofavg), avg(z,
lengthofavg), _maxX, _minX, _maxY, _minY, _maxZ, _minZ);
```

```

        }while(on == 1);

    }
    double avg(double buffer[], int num_items){
        int i;
        int i2;
        double total = 0;
        for(i=0;i<num_items;i++){
            total += buffer[i];
        }
        return (total / num_items);
    }
    void maxmin(double array[], int num_items, double* _max, double* _min){
        int i;
        *_max=-99000;
        *_min=99000;
        for (i=0; i<num_items; i++){

            if (array[i] > *_max){
                *_max=array[i];
            }
            if (i != 0){
                if (array[i]< *_min){
                    *_min=array[i];
                }
            }
        }

    }

}

void updatebuffer(double buffer[], int length, double new_item){
    int i;
    for(i=1; i<length;i++){
        buffer[i-1] = buffer[i];

    }
    buffer[length - 1] = new_item;
}

```

Screen Shots

0.013001	0.950803	0.190014	0.009414	0.951105	0.190394	0.016053	0.002380	0.959958	0.942868	0.197461	0.184276
0.011292	0.948972	0.190502	0.009414	0.951104	0.190397	0.016053	0.002380	0.959958	0.942868	0.197461	0.184276
0.016175	0.952634	0.183910	0.009420	0.951108	0.190391	0.016175	0.002380	0.959958	0.942868	0.197461	0.183910
0.029360	0.955197	0.181102	0.009439	0.951112	0.190380	0.029360	0.002380	0.959958	0.942868	0.197461	0.181102
0.049869	0.945065	0.195019	0.009478	0.951102	0.190381	0.049869	0.002380	0.959958	0.942868	0.197461	0.181102
0.069890	0.937008	0.209302	0.009539	0.951084	0.190397	0.069890	0.002380	0.959958	0.937008	0.209302	0.181102
0.065251	0.949460	0.210157	0.009597	0.951081	0.190418	0.069890	0.002380	0.959958	0.937008	0.210157	0.181102
0.043399	0.961790	0.212232	0.009629	0.951091	0.190442	0.069890	0.002380	0.961790	0.937008	0.212232	0.181102
0.089544	0.944332	0.201733	0.009705	0.951080	0.190454	0.089544	0.002380	0.961790	0.937008	0.212232	0.181102
0.125191	0.945187	0.201367	0.009820	0.951070	0.190466	0.125191	0.002380	0.961790	0.937008	0.212232	0.181102
0.105902	0.957273	0.204175	0.009917	0.951076	0.190481	0.125191	0.002380	0.961790	0.937008	0.212232	0.181102
0.076726	0.953244	0.199048	0.009987	0.951081	0.190492	0.125191	0.002380	0.961790	0.937008	0.212232	0.181102
0.052921	0.949216	0.196606	0.010034	0.951082	0.190498	0.125191	0.002380	0.961790	0.937008	0.212232	0.181102
0.039980	0.951901	0.187695	0.010063	0.951084	0.190496	0.125191	0.002380	0.961790	0.937008	0.212232	0.181102
0.041445	0.943844	0.178661	0.010096	0.951079	0.190486	0.125191	0.002380	0.961790	0.937008	0.212232	0.178661
0.044864	0.949582	0.179882	0.010133	0.951077	0.190477	0.125191	0.002380	0.961790	0.937008	0.212232	0.178661

X:0.009827, Y:0.944088, Z:0.185131, Avg X:0.009820, Avg Y:0.952144, Avg Z:0.186541, Max X:0.016786, Min X:0.004090, Max Y:0.962522, Min Y:0.944088, Max Z:0.193676, Min Z:0.180004
 X:0.008973, Y:0.953610, Z:0.197206, Avg X:0.009817, Avg Y:0.952144, Avg Z:0.186544, Max X:0.016786, Min X:0.004090, Max Y:0.962522, Min Y:0.944088, Max Z:0.193676, Min Z:0.180004
 X:0.008118, Y:0.949460, Z:0.185131, Avg X:0.009814, Avg Y:0.952142, Avg Z:0.186542, Max X:0.016786, Min X:0.004090, Max Y:0.962522, Min Y:0.944088, Max Z:0.193676, Min Z:0.180004
 X:0.007752, Y:0.959104, Z:0.185619, Avg X:0.009814, Avg Y:0.952153, Avg Z:0.186542, Max X:0.016786, Min X:0.004090, Max Y:0.962522, Min Y:0.944088, Max Z:0.193676, Min Z:0.180004
 X:0.009949, Y:0.963987, Z:0.185009, Avg X:0.009813, Avg Y:0.952165, Avg Z:0.186541, Max X:0.016786, Min X:0.004090, Max Y:0.963987, Min Y:0.944088, Max Z:0.193676, Min Z:0.180004