TITLE

LAB 9

SECTION C

Kenneth A. Jacobson

SUBMISSION DATE:

17.11.2017

Problem

The problem in this lab was to determine based on inputs from an Esplora and determine how far the unit fell by using time. The second part required us to include an estimation of air resistance into our program.

Analysis

The equation for the first part given to us ended up simply being a function of time only requiring only a single input however the second part required we make a second input and generate velocity values as the unit falls making things more difficult to calculate.

Design

We had to design the program to take in data and output understandable words to tell the operator what is happening to the unit. I used some while loops to determine when and for how long the unit was in a particular state and then print lines accordingly.

Testing

I used not only the sample data provided but also some data I gathered myself to test my code and see if my output made sense considering the circumstances..

Comments

N/A

Source Code

Lab9-1:

```
//Author: Kenneth A. Jacobson
//Name: Lab 9
//Description: Code responses for an Esplora Unit dropped
//3.11.2017

#include <stdio.h>
#include <math.h>

#define G 9.8   //Gravitational Acceleration on Earth
#define TOL .6 //tolerance of mag at a standstill
#define STL 1   //Value of mag when at a standstill ish
#define K 10    //Value of interval
double magnitude(double, double, double);
double position(double);

void main(void){

        int t, ti, tf;
        double ax, ay, az, mag, time;
        int cont = 1;

        //printf("How large should the Interval be? ");
        //      scanf("%d", &interval);

        printf("**Recieving Data**\n");
        printf("\tI'm Waiting");
        fflush(stdout);

        while(cont){
                scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az); //read in data from esplora
                mag = TOL+ magnitude(ax, ay, az);
                fflush(stdout);
                if(STL < mag){
                        printf("\n\t\tHelp! I'm falling");
                        ti=t;
                        cont = 0;

                        while(STL < mag){

                                if(t%K == 0){
                                        printf("!");
                }

                scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az);
                mag = TOL+ magnitude(ax, ay, az);
            }

            tf=t;
            printf("\n");

        }
        else if(t%K == 0){
            printf(".");
        }
    }

    time = (tf-ti)/1000.0;  //convert time in ms to time in s
    printf("Ouch! I fell %.3lf meters in %.3lf seconds.", position(time), time);



}

double magnitude(double ax, double ay, double az){
    return sqrt(pow(ax, 2.0)+pow(ay, 2.0)+pow(az, 2.0));
}

double position(double time){
    return (.5*G*pow(time, 2.0));
}
```

Lab9-2:

```
//Author: Kenneth A. Jacobson
//Name: Lab 9
//Description: Code responses for an Esplora Unit dropped
//3.11.2017

#include <stdio.h>
#include <math.h>

#define G 9.8   //Gravitational Acceleration on Earth
#define TOL .6 //tolerance of mag at a standstill
#define STL 1   //Value of mag when at a standstill ish
#define K 10    //Value of interval
double magnitude(double, double, double);
double position(double);

void main(void){

        int t, ti, tf;
        double ax, ay, az, mag, time;
        int cont = 1;

        //printf("How large should the Interval be? ");
        //        scanf("%d", &interval);

        printf("**Recieving Data**\n");
        printf("\tI'm Waiting");
        fflush(stdout);

        while(cont){
                scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az); //read in data from esplora
                mag = TOL+ magnitude(ax, ay, az);
                fflush(stdout);
                if(STL < mag){
                        printf("\n\t\tHelp! I'm falling");
                        ti=t;
                        cont = 0;

                        while(STL < mag){

                                if(t%K == 0){
                                        printf("!");
                                }
                scanf("%d,%lf,%lf,%lf", &t, &ax, &ay, &az);
                mag = TOL+ magnitude(ax, ay, az);
            }

            tf=t;
            printf("\n");

        }
        else if(t%K == 0){
            printf(".");
        }
    }

    time = (tf-ti)/1000.0;  //convert time in ms to time in s
    printf("Ouch! I fell %.3lf meters in %.3lf seconds.", positionB(time), time);
    posDiv(time);


}

double magnitude(double ax, double ay, double az){
    return sqrt(pow(ax, 2.0)+pow(ay, 2.0)+pow(az, 2.0));
}

double positionA(double time){
    return (.5*G*pow(time, 2.0));
}

double positionB(double time){
    return (G*time)+(.5*G*pow(time, 2.0));
}

void posDiv(double time){
    double a = positionA(time);
    double b = positionB(time);


    if (a > b){
        double c = (1 - (b/a))*100;
        printf("That was %.0lf less than previously calculated.");
    }
    else if(a < b){
        double c = (1 - (a/b))*100;
        printf("That was %.0lf more than previously calculated.");
    }
    else{
        printf("That was the same as last time");
    }

}
```

Screen Shots