A
Project Report
On
"Solar System"

Prepared by:
Kashyap Nirmal(16IT059)
Akshay Pandya(16IT061)

Under the supervision of
Prof. Ayesha Sheikh

A Report Submitted to
Charotar University of Science and Technology
for Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology
in Information Technology
IT244 Software Group Project-I (3rd sem)

Submitted at

DEPARTMENT OF INFORMATION TECHNOLOGY
Chandubhai S. Patel Institute of Technology
At: Changa, Dist: Anand – 388421
December 2017

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**Solar System"** is a bonafied work carried out by **Mr. Kashyap Nirmal (16IT059) and Mr. Akshay Pandya(16IT061)** under the guidance and supervision of **Prof. Ayesha Sheikh** for the subject **Software Group Project-I(IT244)** of **3rd** Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate herself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Ayesha Sheikh
Assistant Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Prof. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

Chandubhai S Patel Institute of Technology

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We are extremely thankful to our internal guide **Prof. Ayesha Shaikh** whose constant support and helping attitude towards the students has always provided us the strength to complete the task efficiently and rapidly.

We sincerely thank Mam for his expert guidance and encouraging us throughout this project work.

Now I would like to forward my thanking to Prof. Parth Shah, Head of Information Technology Department, Charotar institute of Technology.

Kashyap Nirmal(16IT059)

Akshay Pandya(16IT061)

ABSTRACT

The idea behind the project is to develop an animation in C++, which displays the model of our Solar System (i.e. on the basis of Kepler's Law).

This model assists us to visualize the planetary motions in our Solar System. The real life statistics area also displayed, so we can have an overview about the model.

CHAPTER- 1: INTRODUCTION

1.1: Project Definition

Our project includes an animation in C++ using the traditional Graphics library. And clearly visualizing theories that we have studied can be very much helpful and convincing for us. And, as we all know the visual memories are very strong so it can be helpful to memorize the basics of our Solar System.

So it is basic and simple animation with simple 2D graphics. The project can be used to make the user familiar with the planetary motion of our Solar System. It also displays the real life statistics are also displayed.

1.2: Description

First of all, a choice menu is given to the user. The choices are namely for viewing the model. And, the second one for viewing the data. So the user needs to enter the choice as per his/her requirements.

The first choice i.e. viewing the model basically displays the 2D simple animation of the planetary motion. The second displays the theory of the Solar System model. It also displays the statistical data of few very important parameters of the planets.

Major functionality of our project is the animation i.e. motion of planets in an elliptical orbit and it also contains some basic parameters of all planets.

Ultimately, in our project major functionality is logic about deciding the angles for the motion of planets. And deciding various other important parameters.

CHAPTER- 2: SYSTEM REQUIEMENTS

2.1: User characteristics:

It is a single user project. Mostly faculties i.e. school teachers and, the students of science stream can use it. And others who seems to be interested in recalling and viewing the basic Solar System model can use the project.

2.2: Tools and languages used (For development):

- **Software Requirements:**

1. Windows XP and above

2. C++ compiler i.e. Codeblocks or TurboC++

3.Graphics library

- **Hardware Requirements:**

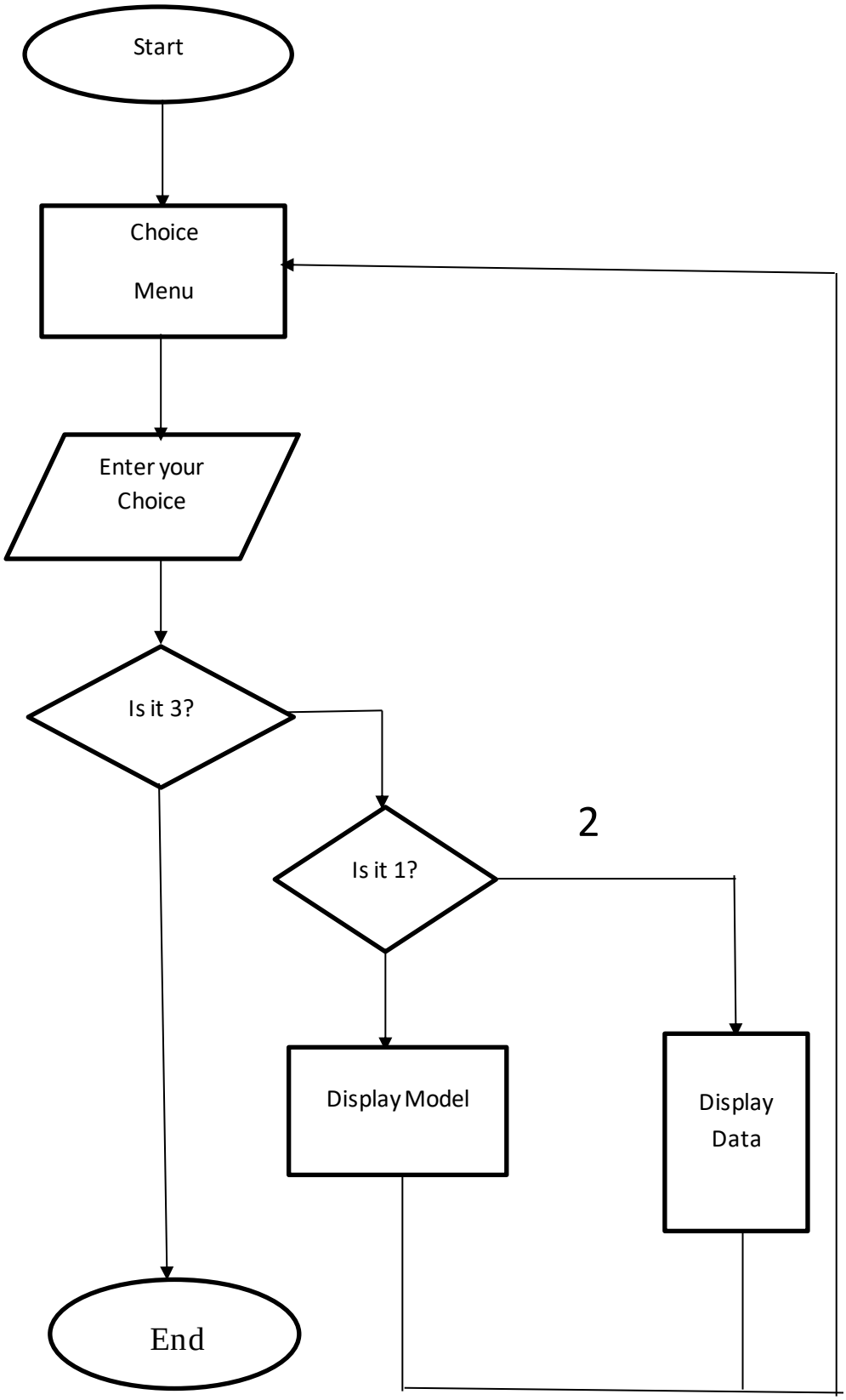
A computer with minimum of 256MB RAM and with very less storage of 500MB.

2.3: Assumptions and dependencies:

We have assumed that user has the basic knowledge of Solar System models. And our project is based on Kepler's Law of Planetary motion i.e. Sun is at one of the foci of the ellipse.

CHAPTER-3: SYSTEM DESIGN

3.1: Project Flow:



CHAPTER- 4: IMPLEMENTATION PLANNING

4.1: Implementation environment:

It is based on simple graphics i.e. 2D graphics using traditional "Graphics" library in C++.So it can be executed on any machine with can Graphics library and C++ compiler.

4.2: Coding Standards:

```
/* This is a program for the animation of Solar System i.e showing the planetary motions. */
#include<graphics.h>
#include<iostream>
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<dos.h>
#include<iomanip>
#include<conio.h>
using namespace std;
double j=40.625,k=82.5,f=44.35,pi=22/7,m=pi/180;
double xi=400.2,yi=412,x_inc=80,y_inc=50;
int i=0,z=1;
int gd=DETECT,gm,c1=0;
   class Solar_System
   {
      public:
      string name;
      double d,md,e,rot_p1,rot_p2,rev_p1,rev_p2;
      Solar_System(){}
      Solar_System(string name1,double d1,double md1,double e1,double rot_p11,double
rot_p22,double rev_p11,double rev_p22)
      {
```

```
        name=name1;
        d=d1;
        e=e1;
        md=md1;
        rot_p1=rot_p11;
        rot_p2=rot_p22;
        rev_p1=rev_p11;
        rev_p2=rev_p22;
    }
    void static ellipticalorbit();
    void static sun();
    void static planets();
    void static label();
    void static animation();
    void static menu();
    void static first();
};

int main()
{
    /*Objects created for passing the data of the planets."*/
    Solar_System o[6];
    Solar_System g("Mercury",4880,58,0.206,59,0,0,88);
    Solar_System h("Venus",12100,108.2,0.007,243,0,0,224);
    Solar_System a("Earth",12756,149.6,0.017,24,0,1,0);
    Solar_System b("Mars",6794,227.9,0.093,24,37,1,325);
    Solar_System c("Jupiter",142800,778.3,0.048,9,55,11,314);
```

```
Solar_System d("Saturn",120660,1427,0.056,10,40,29,168);
Solar_System e("Uranus",51810,2870,0.047,16,48,84,0);
Solar_System f("Neptune",49528,4497,0.009,16,11,165,0);
Solar_System j("Pluto",2290,5900,0.206,0,0,0,0);
o[0]=a;
o[1]=b;
o[2]=c;
o[3]=d;
o[4]=e;
o[5]=f;
Solar_System::first();
A:
switch(c1)
{
   case '2':
      for(i=0;i<6;i++)
      {
         if(i==0)
         {
            cout<<"There are many models of Solar System available.The one displayed here is based on \"Keplers law.\""<<endl;
            cout<<"The real life statistics of few important properties of the Sun and all the planets are as displayed below:"<<endl;
            cout<<endl<<endl;
            /* Sun */
            cout<<setiosflags(ios::left)<<setw(29)<<"Name"<<": Sun"<<endl;
            cout<<setw(29)<<"Diameter"<<": 1.39x10^6 Km"<<endl;
            cout<<setw(29)<<"Rotation Period"<<": 24 hr"<<endl;
```

```
                cout<<setw(29)<<"Surface Temperature of Sun"<<": 5.778 K"<<endl;
                cout<<setw(29)<<"Age of Sun"<<": 4.6 Billion Years"<<endl;
                cout<<endl<<endl;
                /* Mercury */
                cout<<setw(29)<<"Name"<<": "<<g.name<<endl;
                cout<<setw(29)<<"Diameter of Planet"<<": "<<g.d<<" Km"<<endl;
                cout<<setw(29)<<"Mean distance from Sun"<<": "<<g.md<<" x 10^6
Km"<<endl;
                cout<<setw(29)<<"Rotation Period"<<": "<<g.rot_p1<<" days"<<endl;
                cout<<setw(29)<<"Revolution Period"<<": "<<g.rev_p2<<" days"<<endl;
                cout<<endl<<endl;
                /* Venus */
                cout<<setw(29)<<"Name"<<": "<<h.name<<endl;
                cout<<setw(29)<<"Diameter of Planet"<<": "<<h.d<<" Km"<<endl;
                cout<<setw(29)<<"Mean distance from Sun"<<": "<<h.md<<" x 10^6
Km"<<endl;
                cout<<setw(29)<<"Rotation Period"<<": "<<h.rot_p1<<" days"<<endl;
                cout<<setw(29)<<"Revolution Period"<<": "<<h.rev_p2<<" days"<<endl;
                cout<<endl<<endl;
            }
            /* Earth  to Neptune*/
            cout<<setw(29)<<"Name"<<": "<<o[i].name<<endl;
            cout<<setw(29)<<"Diameter of Planet"<<": "<<o[i].d<<" Km"<<endl;
            cout<<setw(29)<<"Mean distance from Sun"<<": "<<o[i].md<<" x 10^6
Km"<<endl;
            cout<<setw(29)<<"Rotation Period"<<": "<<o[i].rot_p1<<"hr
"<<o[i].rot_p2<<"min"<<endl;
            cout<<setw(29)<<"Revolution Period"<<": "<<o[i].rev_p1<<"yrs
"<<o[i].rev_p2<<"days"<<endl;
```

```
            cout<<endl<<endl;
            if(i==5)
            {
               /* Pluto */
               cout<<setw(29)<<"Name"<<": "<<j.name<<endl;
               cout<<setw(29)<<"Diameter of Planet"<<": "<<j.d<<" Km"<<endl;
               cout<<setw(29)<<"Mean distance from Sun"<<": "<<j.md<<" x 10^6 Km"<<endl;
               cout<<setw(29)<<"Rotation Period"<<": 6 days 9 hr 18 min"<<endl;
               cout<<setw(29)<<"Revolution Period"<<": 248 yr"<<endl;
            }
         }
         Solar_System::menu();
         break;
      case '3':
         cout<<"\nTHANK YOU";
         return 1;
      default :
         cout<<"!! Invalid choice. !!"<<endl;
         Solar_System::menu();
         break;
      case '1':
         setcolor(WHITE);
         Solar_System::ellipticalorbit();
         Solar_System::sun();
         Solar_System::planets();
         Solar_System::label();
         getch();
```

```
            cleardevice();
            Solar_System::sun();
            Solar_System::ellipticalorbit();
            Solar_System::animation();
            Solar_System::menu();
            break;
        }
    return 0;
}

void Solar_System::ellipticalorbit()
{
    for(i=0;i<9;i++)
            ellipse(xi+(f*i),yi,360,0,(k*i)+x_inc,(j*i)+y_inc);
}

void Solar_System::first()
{
    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
    settextstyle(1,HORIZ_DIR,8);
    outtextxy(600,326,"SOLAR");
    outtextxy(550,410,"SYSTEM");
    settextstyle(1,HORIZ_DIR,2);
    getch();
    cleardevice();
    Solar_System::menu();
}
```

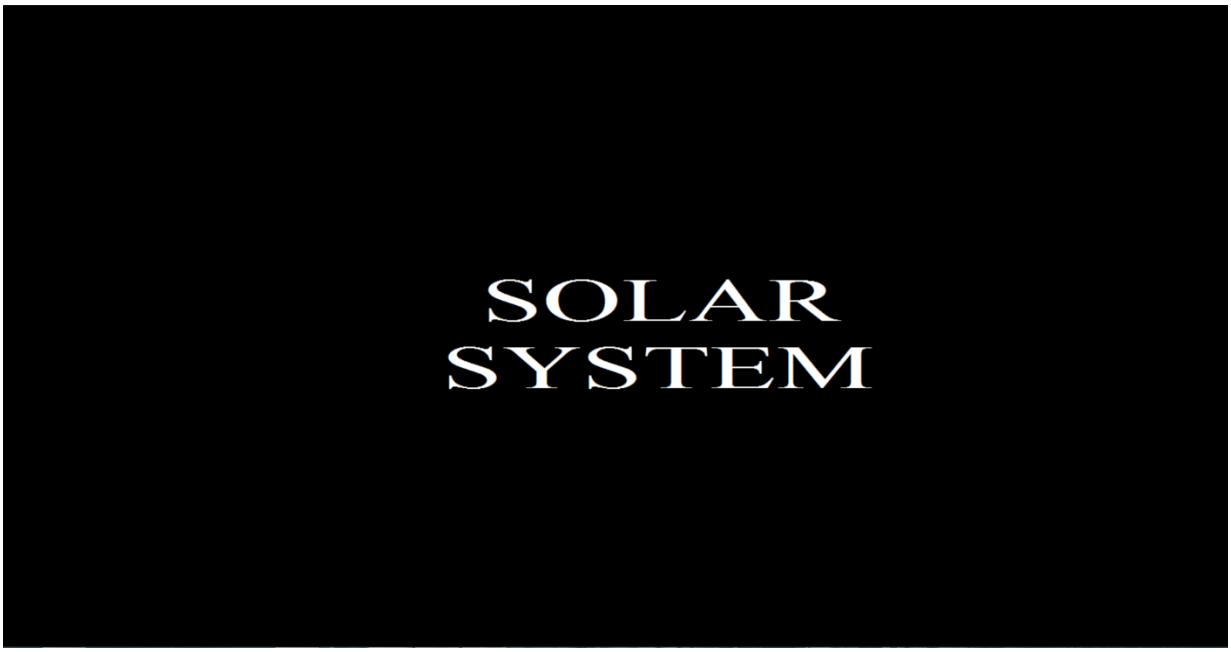
```
void Solar_System::menu()
{
   setbkcolor(BLACK);
   settextstyle(1,HORIZ_DIR,2);
   setcolor(BLUE);
   outtextxy(10,5,"Enter your choice :");
   setcolor(WHITE);
   outtextxy(10,40,"1. View the model for Solar System.");
   outtextxy(10,75,"2. View the data of planets.");
   setcolor(RED);
   outtextxy(10,110,"3. Exit");
   setcolor(BLUE);
   fflush(stdin);
   c1=getch();
   cleardevice();
}

void Solar_System::sun()
{
   setcolor(YELLOW);
   circle(385.2,yi,40);
   setcolor(YELLOW);
   setfillstyle(1,14);
   floodfill(350.2,yi,14);
   settextstyle(1,HORIZ_DIR,1);
   setcolor(WHITE);
```

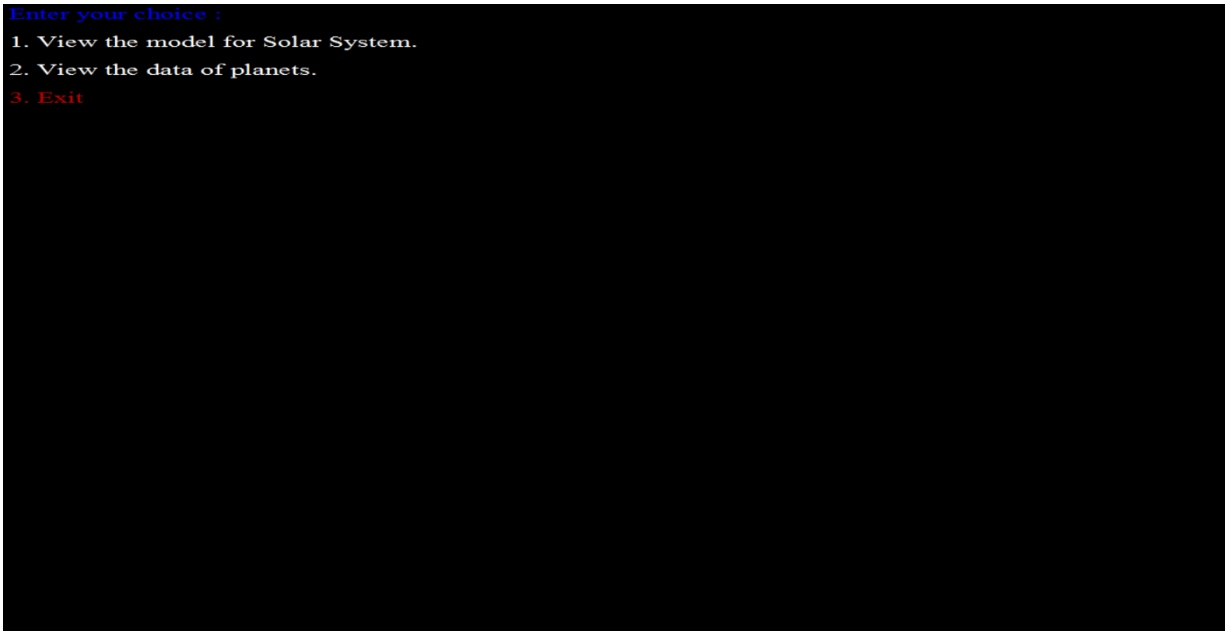
```
    outtextxy(352.69,403,"SUN");
  }

  void Solar_System::planets()
  {
    z=1;
    for(i=0;i<9;i++)
      {
        if(i<3)                              /*Mercury to Earth */
          circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),6*(i+1));
        else if(i==3)                        /* Mars */
          circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),6*(3)/2);
        else if(i==4 || i==5)                /* Jupiter and Saturn */
          circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),6*(9-i));
        else if(i==6)                        /*Uranus */
          circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),18);
        else if(i==7)                        /* Neptune */
          circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),21);
        else if(i==8)                        /* Pluto */
          circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),5);

      }
  }
```

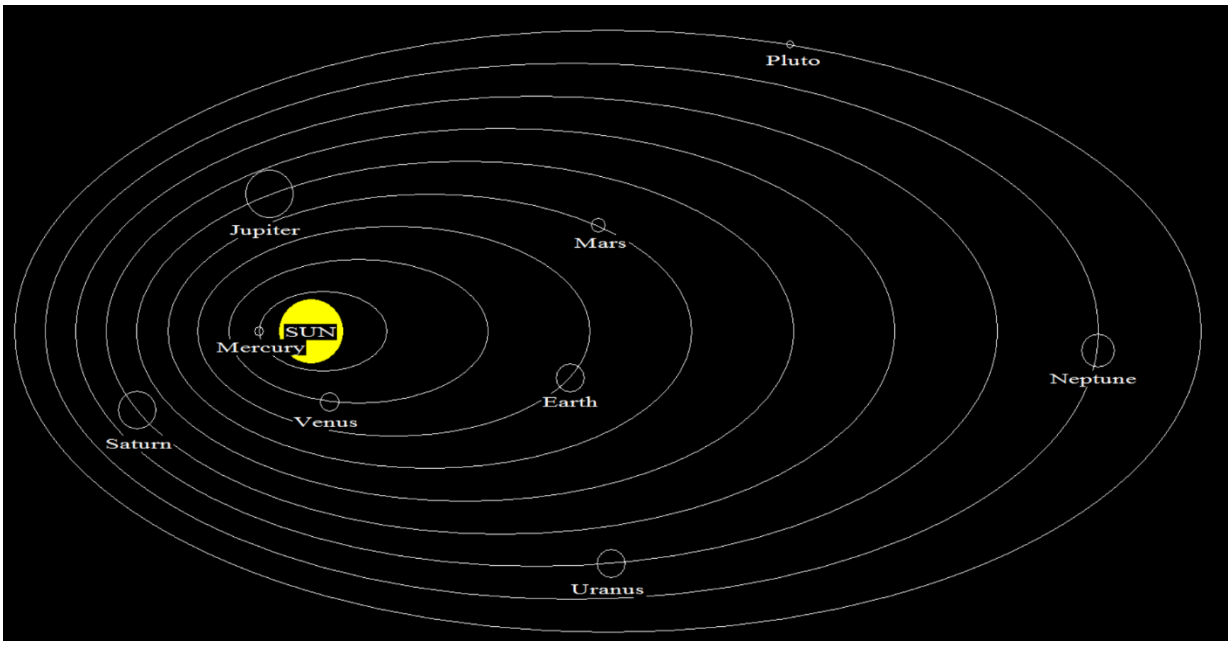
```
void Solar_System::label()
{
   setcolor(WHITE);
   outtextxy((xi+(f*0))-(((k*0)+x_inc)*(cos((-x_inc*0-1)*m)))-53,yi-
((((j*0)+y_inc))*(sin((-x_inc*0-1)*m)))+10,"Mercury");
   outtextxy((xi+(f*1))-(((k*1)+x_inc)*(cos((-x_inc*1-1)*m)))-45,yi-
((((j*1)+y_inc))*(sin((-x_inc*1-1)*m)))+15,"Venus");
   outtextxy((xi+(f*2))-(((k*2)+x_inc)*(cos((-x_inc*2-1)*m)))-35,yi-
((((j*2)+y_inc))*(sin((-x_inc*2-1)*m)))+20,"Earth");
   outtextxy((xi+(f*3))-(((k*3)+x_inc)*(cos((-x_inc*3-1)*m)))-30,yi-
((((j*3)+y_inc))*(sin((-x_inc*3-1)*m)))+12,"Mars");
   outtextxy((xi+(f*4))-(((k*4)+x_inc)*(cos((-x_inc*4-1)*m)))-50,yi-
((((j*4)+y_inc))*(sin((-x_inc*4-1)*m)))+35,"Jupiter");
   outtextxy((xi+(f*5))-(((k*5)+x_inc)*(cos((-x_inc*5-1)*m)))-40,yi-
((((j*5)+y_inc))*(sin((-x_inc*5-1)*m)))+32,"Saturn");
   outtextxy((xi+(f*6))-(((k*6)+x_inc)*(cos((-x_inc*6-1)*m)))-50,yi-
((((j*6)+y_inc))*(sin((-x_inc*6-1)*m)))+22,"Uranus");
   outtextxy((xi+(f*7))-(((k*7)+x_inc)*(cos((-x_inc*7-1)*m)))-60,yi-
((((j*7)+y_inc))*(sin((-x_inc*7-1)*m)))+25,"Neptune");
   outtextxy((xi+(f*8))-(((k*8)+x_inc)*(cos((-x_inc*8-1)*m)))-30,yi-
((((j*8)+y_inc))*(sin((-x_inc*8-1)*m)))+10,"Pluto");
}

void Solar_System::animation()
{
   for(z=1;!kbhit();z+=5)
   {
      for(i=0;i<9;i++)
      {
         if(i<3)                          /*Mercury to Earth */
```

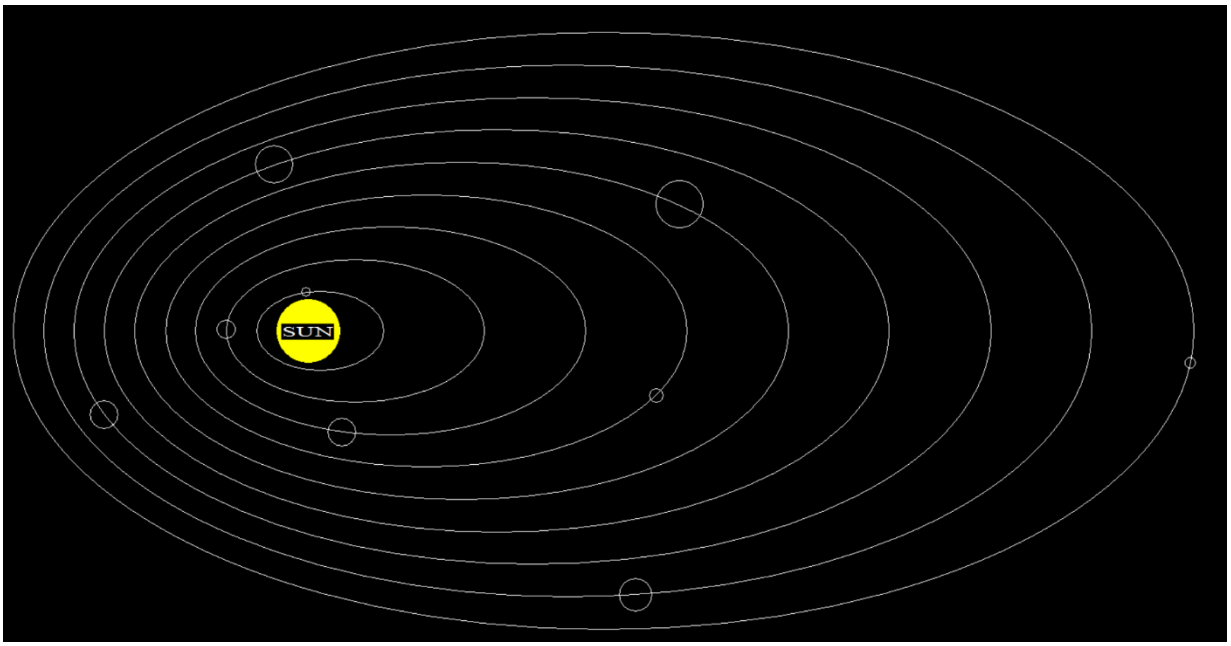
```
            circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),6*(i+1));

        else if(i==3)                    /* Mars */

            circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),6*(3)/2);

        else if(i==4 || i==5)             /* Jupiter and Saturn */

            circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),6*(9-i));

        else if(i==6)                    /*Uranus */

            circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),18);

        else if(i==7)                    /* Neptune */

            circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),21);

        else if(i==8)                    /* Pluto */

            circle((xi+(f*i))-((((k*i)+x_inc))*(cos((-x_inc*i-z)*m))),yi-
((((j*i)+y_inc))*(sin((-x_inc*i-z)*m))),7);


      }

      system("cls");

      cleardevice();

      sun();

      ellipticalorbit();

      delay(10);

    }

    Solar_System::menu();

  }
```

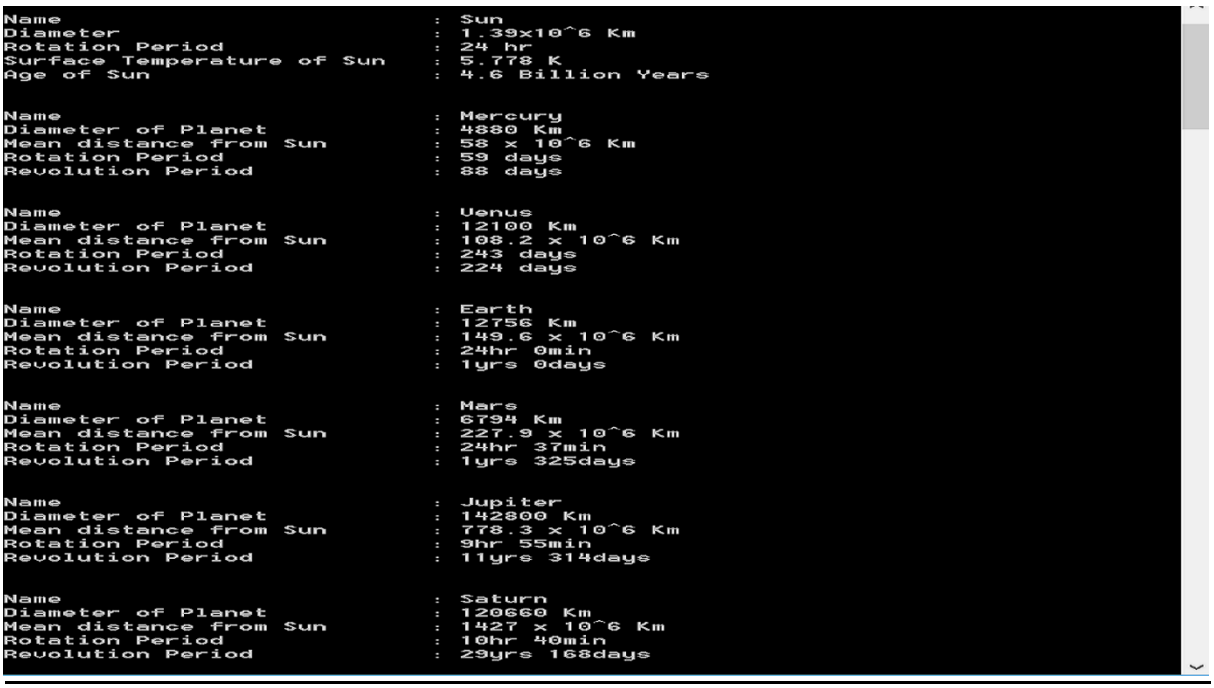
4.3: Snapshots of project:






Pluto
Jupiter
Mars
SUN
Mercury
Neptune
Earth
Venus
Saturn
Uranus


SUN


Name : Sun
Diameter : 1.39x10^6 Km
Rotation Period : 24 hr
Surface Temperature of Sun : 5.778 K
Age of Sun : 4.6 Billion Years
Name : Mercury
Diameter of Planet : 4880 Km
Mean distance from Sun : 58 x 10^6 Km
Rotation Period : 59 days
Revolution Period : 88 days
Name : Venus
Diameter of Planet : 12100 Km
Mean distance from Sun : 108.2 x 10^6 Km
Rotation Period : 243 days
Revolution Period : 224 days
Name : Earth
Diameter of Planet : 12756 Km
Mean distance from Sun : 149.6 x 10^6 Km
Rotation Period : 24hr 0min
Revolution Period : 1yrs 0days
Name : Mars
Diameter of Planet : 6794 Km
Mean distance from Sun : 227.9 x 10^6 Km
Rotation Period : 24hr 37min
Revolution Period : 1yrs 325days
Name : Jupiter
Diameter of Planet : 142800 Km
Mean distance from Sun : 778.3 x 10^6 Km
Rotation Period : 9hr 55min
Revolution Period : 11yrs 314days
Name : Saturn
Diameter of Planet : 120660 Km
Mean distance from Sun : 1427 x 10^6 Km
Rotation Period : 10hr 40min
Revolution Period : 29yrs 168days

CHAPTER-5: LIMITATIONS AND FUTURE ENHANCEMENT

5.1: Limitations

- The main limitation of the project is it has 2D graphics and animations.
- Using C++, the code becomes very long.
- All functions will be developed in CodeBlocks.
- Linux operating system is not supported.

5.2: Future Enhancements

- In future, we would like to add graphical user interface and attracting features which are different from these used.
- We would like to use mouse event for this project with new and some exciting animation effects.

CHAPTER 6-CONCLUSION

6.1: Conclusion

I. Our project helps visualize the basic theories we have studied so far.
II. It helps visualize the model of our Solar System
III. It shows Animation of planets around its orbits.
IV. It also shows some information of planets.

6.2: References

- http://en.wikipedia.org/wiki/Solar System
- http://en.wikipedia.org/wiki/planetsinformation
- http://www.cplusplus.com
- http://forums.codeblocks.org/index.php?topic=14828.0:wap2
- http://www.cplusplus.com/forum/beginner/29936
- Object Oriented Programming with C++- E Balagurusamy