Daffodil International University

Department of Computer Science and Engineering

FINAL PROJECT REPORT

COURSE NAME: COMPUTER GRAPHICS LAB

COURSE CODE: CSE-422

PROJECT NAME: LAUNCHING SATELLITE THROUGH ROCKET IN OUR SOLAR SYSTEM USING OpenGL IN C++

DATE OF SUBMISSION: 31/5/2023

SUBMITTED BY

NAME: Sayed Nahid

ID: 202-15-3849

SECTION: 56_PC-B1

DEPT. OF CSE

SUBMITTED TO

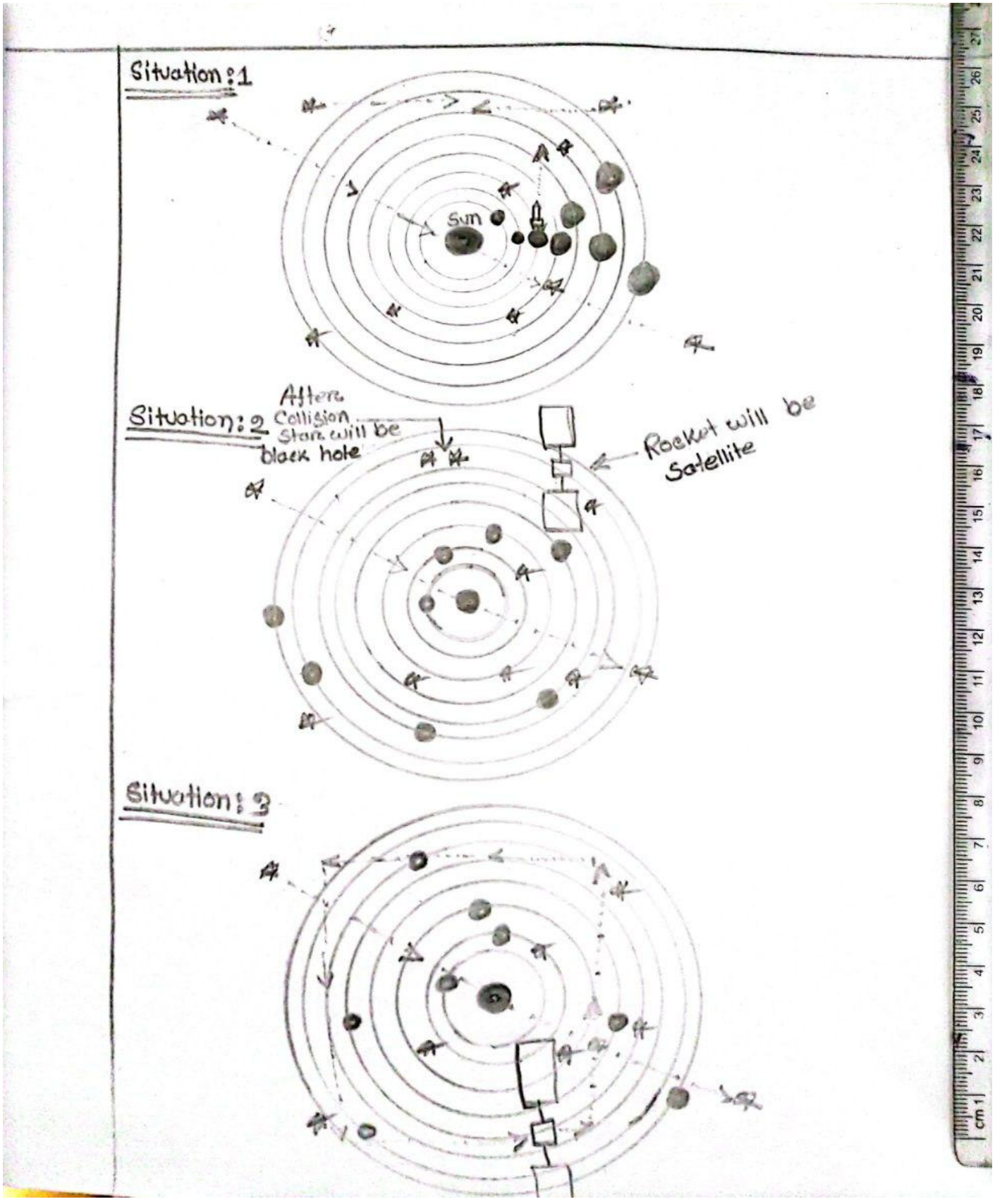
Israt Jahan

Lecturer,

Department of CSE

Daffodil International University

Sketch:


Situation : 1
Sun
Situation : 2
After Collision Star will be black hole
Rocket will be Satellite
Situation : 3

Code:

```
#include <stdio.h>

#include <GL/gl.h>

#include <GL/glut.h>

#include <stdlib.h>

#include <math.h>

#include <cmath>

float starX = -1.0f;

float starY = 0.8f;

float angle = 0.0f;


int a=-1700, b=1700, c1x=-1500, c2x=1500, cb=1300, aa=1000, bb=1200;

int ea=570, eb=0, flag=0, flagl=0, flagb=0, flagr=0, flagt=0, stop=0;


float r, x, y, theta;

float m = 0, v = 0, e = 0, mr = 0, j = 0, s = 0, u = 0, n = 0;

float mercury_x = 200;

float mercury_y = 0;

float venus_x = 400;

float venus_y = 0;

float earth_x = 600;

float earth_y = 0;

float mars_x = 800;

float mars_y = 0;

float jupiter_x = 1000;

float jupiter_y = 0;
```

```
float saturn_x = 1200;
float saturn_y = 0;
float uranus_x = 1400;
float uranus_y = 0;
float neptune_x = 1600;
float neptune_y = 0;
void timer(int);

void display(void)
{
	glClear (GL_COLOR_BUFFER_BIT);
	//////////Moving Star1
	glColor3ub (255, 255, 255);
	glBegin(GL_POLYGON);
	glVertex2d (a, b);//180, 310
	glVertex2d (a+15, b+5);//15  5
	glVertex2d (a+15, b-5);//15 -5
	glEnd();
	glBegin(GL_POLYGON);//4
	glVertex2d (a+15, b+5);//15  5
	glVertex2d (a+20, b+20); //20 20
	glVertex2d (a+25, b+5);//25 5
	glEnd();
	glBegin(GL_POLYGON);
	glVertex2d (a+25, b+5);//15 5
	glVertex2d (a+40, b);//30 0
	glVertex2d (a+25, b-5);//25 -5
	glEnd();
```

```
glBegin(GL_POLYGON);
glVertex2d (a+25, b-5);//25 -5
glVertex2d (a+20, b-20);//20 -20
glVertex2d (a+15, b-5);//15 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (a+15, b+5);//15 5
glVertex2d (a+25, b+5);//25 5
glVertex2d (a+25, b-5);//25 -5
glVertex2d (a+15, b-5);//15 -5
glEnd();
if(b == -1500){
a=-1700;
b=1700;
}
else
{
a++;
b--;
}
//////////Moving Star2
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (aa, bb);//180, 310
glVertex2d (aa+15, bb+5);//15  5
glVertex2d (aa+15, bb-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
```

```
glVertex2d (aa+15, bb+5);//15  5
glVertex2d (aa+20, bb+20); //20 20
glVertex2d (aa+25, bb+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (aa+25, bb+5);//15 5
glVertex2d (aa+40, bb);//30 0
glVertex2d (aa+25, bb-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (aa+25, bb-5);//25 -5
glVertex2d (aa+20, bb-20);//20 -20
glVertex2d (aa+15, bb-5);//15 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (aa+15, bb+5);//15 5
glVertex2d (aa+25, bb+5);//25 5
glVertex2d (aa+25, bb-5);//25 -5
glVertex2d (aa+15, bb-5);//15 -5
glEnd();
if(aa == -1300){
aa=1000;
bb=1200;
}
else
{
aa--;
bb--;
```

```
}
///Collision star
if(stop==0)
{
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (c1x, cb);//180, 310
glVertex2d (c1x+15, cb+5);//15  5
glVertex2d (c1x+15, cb-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
glVertex2d (c1x+15, cb+5);//15  5
glVertex2d (c1x+20, cb+20); //20 20
glVertex2d (c1x+25, cb+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (c1x+25, cb+5);//15 5
glVertex2d (c1x+40, cb);//30 0
glVertex2d (c1x+25, cb-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (c1x+25, cb-5);//25 -5
glVertex2d (c1x+20, cb-20);//20 -20
glVertex2d (c1x+15, cb-5);//15 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (c1x+15, cb+5);//15 5
glVertex2d (c1x+25, cb+5);//25 5
```

```
glVertex2d (c1x+25, cb-5);//25 -5
glVertex2d (c1x+15, cb-5);//15 -5
glEnd();
////
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (c2x, cb);//180, 310
glVertex2d (c2x+15, cb+5);//15  5
glVertex2d (c2x+15, cb-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
glVertex2d (c2x+15, cb+5);//15  5
glVertex2d (c2x+20, cb+20); //20 20
glVertex2d (c2x+25, cb+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (c2x+25, cb+5);//15 5
glVertex2d (c2x+40, cb);//30 0
glVertex2d (c2x+25, cb-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (c2x+25, cb-5);//25 -5
glVertex2d (c2x+20, cb-20);//20 -20
glVertex2d (c2x+15, cb-5);//15 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (c2x+15, cb+5);//15 5
glVertex2d (c2x+25, cb+5);//25 5
```

```
glVertex2d (c2x+25, cb-5);//25 -5
glVertex2d (c2x+15, cb-5);//15 -5
glEnd();
c1x++;
c2x--;
if(c1x == c2x) stop=1;
}
//////fixed stars
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (200, 200);//180, 310
glVertex2d (200+15, 200+5);//15  5
glVertex2d (200+15, 200-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
glVertex2d (200+15, 200+5);//15  5
glVertex2d (200+20, 200+20); //20 20
glVertex2d (200+25, 200+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (200+25, 200+5);//15 5
glVertex2d (200+40, 200);//30 0
glVertex2d (200+25, 200-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (200+25, 200-5);//25 -5
glVertex2d (200+20, 200-20);//20 -20
glVertex2d (200+15, 200-5);//15 -5
```

```
glEnd();
glBegin(GL_POLYGON);
glVertex2d (200+15, 200+5);//15 5
glVertex2d (200+25, 200+5);//25 5
glVertex2d (200+25, 200-5);//25 -5
glVertex2d (200+15, 200-5);//15 -5
glEnd();
///
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (1200, 1200);//180, 310
glVertex2d (1200+15, 1200+5);//15  5
glVertex2d (1200+15, 1200-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
glVertex2d (1200+15, 1200+5);//15  5
glVertex2d (1200+20, 1200+20); //20 20
glVertex2d (1200+25, 1200+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (1200+25, 1200+5);//15 5
glVertex2d (1200+40, 1200);//30 0
glVertex2d (1200+25, 1200-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (1200+25, 1200-5);//25 -5
glVertex2d (1200+20, 1200-20);//20 -20
glVertex2d (1200+15, 1200-5);//15 -5
```

```
glEnd();
glBegin(GL_POLYGON);
glVertex2d (1200+15, 1200+5);//15 5
glVertex2d (1200+25, 1200+5);//25 5
glVertex2d (1200+25, 1200-5);//25 -5
glVertex2d (1200+15, 1200-5);//15 -5
glEnd();
//
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (-1200, -1200);//180, 310
glVertex2d (-1200+15, -1200+5);//15  5
glVertex2d (-1200+15, -1200-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
glVertex2d (-1200+15, -1200+5);//15  5
glVertex2d (-1200+20, -1200+20); //20 20
glVertex2d (-1200+25, -1200+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (-1200+25, -1200+5);//15 5
glVertex2d (-1200+40, -1200);//30 0
glVertex2d (-1200+25, -1200-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (-1200+25, -1200-5);//25 -5
glVertex2d (-1200+20, -1200-20);//20 -20
glVertex2d (-1200+15, -1200-5);//15 -5
```

```
glEnd();
glBegin(GL_POLYGON);
glVertex2d (-1200+15, -1200+5);//15 5
glVertex2d (-1200+25, -1200+5);//25 5
glVertex2d (-1200+25, -1200-5);//25 -5
glVertex2d (-1200+15, -1200-5);//15 -5
glEnd();
//
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (500, -500);//180, 310
glVertex2d (500+15, -500+5);//15  5
glVertex2d (500+15, -500-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
glVertex2d (500+15, -500+5);//15  5
glVertex2d (500+20, -500+20); //20 20
glVertex2d (500+25, -500+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (500+25, -500+5);//15 5
glVertex2d (500+40, -500);//30 0
glVertex2d (500+25, -500-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (500+25, -500-5);//25 -5
glVertex2d (500+20, -500-20);//20 -20
glVertex2d (500+15, -500-5);//15 -5
```

```
glEnd();
glBegin(GL_POLYGON);
glVertex2d (500+15, -500+5);//15 5
glVertex2d (500+25, -500+5);//25 5
glVertex2d (500+25, -500-5);//25 -5
glVertex2d (500+15, -500-5);//15 -5
glEnd();
//
glColor3ub (255, 255, 255);
glBegin(GL_POLYGON);
glVertex2d (-500, -500);//180, 310
glVertex2d (-500+15, -500+5);//15  5
glVertex2d (-500+15, -500-5);//15 -5
glEnd();
glBegin(GL_POLYGON);//4
glVertex2d (-500+15, -500+5);//15  5
glVertex2d (-500+20, -500+20); //20 20
glVertex2d (-500+25, -500+5);//25 5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (-500+25, -500+5);//15 5
glVertex2d (-500+40, -500);//30 0
glVertex2d (-500+25, -500-5);//25 -5
glEnd();
glBegin(GL_POLYGON);
glVertex2d (-500+25, -500-5);//25 -5
glVertex2d (-500+20, -500-20);//20 -20
glVertex2d (-500+15, -500-5);//15 -5
```

```
glEnd();
glBegin(GL_POLYGON);
glVertex2d (-500+15, -500+5);//15 5
glVertex2d (-500+25, -500+5);//25 5
glVertex2d (-500+25, -500-5);//25 -5
glVertex2d (-500+15, -500-5);//15 -5
glEnd();




//glEnd();


//200 random star at each frame
/*for(int z =0; z <= 200; z++)
{
int zx = rand() % 3400 - 1700;
int zy = rand() % 3400 - 1700;
glPointSize(1.5);
glBegin(GL_POINTS);
glColor3ub(255,255,255);
glVertex2f(zx, zy);
}*/
r = 1600;
for(int j=0; j<8; j++)
{
```

```
for(int i=0; i<=360;i++)
{
glPointSize(1.0f);
glBegin(GL_POINTS);
glColor3ub (255, 255, 255);

theta = i*3.142/180;
glVertex2f(r*cos(theta), r*sin(theta));
glEnd();
}
r -=200;
}

//Rocket
if(flag==0)
{
//rocket
//Head of rocket
glColor3ub (255, 0, 0);
glBegin(GL_POLYGON);
glVertex2d (ea+10, eb+145);
glVertex2d (ea+25, eb+180);
glVertex2d (ea+40, eb+145);
glEnd();
//body of rocket
glColor3ub (0, 255, 0);
glBegin(GL_POLYGON);
glVertex2d (ea+10, eb+50);
```

```
glVertex2d (ea+40, eb+50);
glVertex2d (ea+40, eb+150);
glVertex2d (ea+10, eb+150);
glEnd();
//bottom
glColor3ub (181, 197, 213);
glBegin(GL_POLYGON);
glVertex2d (ea, eb);
glVertex2d (ea+50, eb);
glVertex2d (ea+50, eb+50);
glVertex2d (ea, eb+50);
glEnd();
//fire of rocket
glColor3ub (255, 0, 0);
glBegin(GL_POLYGON);
glVertex2d (ea, eb);
glVertex2d (ea+7, eb-50);
glVertex2d (ea+15, eb);
glEnd();
glBegin(GL_POLYGON);
glVertex2d (ea+17, eb);
glVertex2d (ea+23, eb-50);
glVertex2d (ea+32, eb);
glEnd();
glBegin(GL_POLYGON);
glVertex2d (ea+34, eb);
glVertex2d (ea+41, eb-50);
glVertex2d (ea+50, eb);
```

```
glEnd();
eb++;
if(eb == 1400) flag=1;
}
//settelitte
else{
//control panel
glColor3ub (255, 0, 0);
glBegin(GL_POLYGON);
glVertex2d (ea, eb);
glVertex2d (ea+40, eb);
glVertex2d (ea+40, eb+80);
glVertex2d (ea, eb+80);
glEnd();
//connector
glColor3ub (0, 0, 255);
glBegin(GL_LINES);
glVertex2d(ea+20, eb);
glVertex2d(ea+20, eb-50);
glEnd();
glColor3ub (0, 0, 255);
glBegin(GL_LINES);
glVertex2d(ea+20, eb+80);
glVertex2d(ea+20, eb+130);
glEnd();
//Wings of settlelitte
//wings 1
glColor3ub (0, 255, 0);
```

```
glBegin(GL_POLYGON);
glVertex2d(ea-30, eb-50);
glVertex2d(ea+70, eb-50);
glVertex2d(ea+70, eb-200);
glVertex2d(ea-30, eb-200);
glEnd();
glBegin(GL_POLYGON);
glVertex2d(ea-30, eb+130);
glVertex2d(ea+70, eb+130);
glVertex2d(ea+70, eb+280);
glVertex2d(ea-30, eb+280);
glEnd();
if(flagl == 0)
{
ea--;
if(ea == -1200){ flagl=1; flagb=0; flagr=1; flagt=1;}
}
else if(flagb == 0)
{
eb--;
if(eb == -1200) { flagl=1; flagb=1; flagr=0; flagt=1;}
}
else if(flagr==0)
{
ea++;
if(ea==1200) { flagl=1; flagb=1; flagr=1; flagt=0;}
}
else if(flagt==0)
```

```
{
eb++;
if(eb==1200) { flagl=0; flagb=1; flagr=1; flagt=1;}
}
}
```

```
//sun
glBegin(GL_POLYGON);
glColor3ub(254,204,25);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(80*cos(theta), 80*sin(theta));
}
glEnd();

//mercury
glBegin(GL_POLYGON);
glColor3ub(204,126,56);
for(int i = 0; i <= 360; i++)
{
```

```
theta = i*3.142/180;
glVertex2f(50*cos(theta) + mercury_x, 50*sin(theta) + mercury_y);
}
glEnd();

//venus
glBegin(GL_POLYGON);
glColor3ub(215,122,98);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(80*cos(theta) + venus_x, 80*sin(theta) + venus_y);
}
glEnd();

//earth
glBegin(GL_POLYGON);
glColor3ub(70,248,202);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(80*cos(theta) +earth_x, 80*sin(theta) +earth_y);
}
glEnd();

glBegin(GL_POLYGON);
glColor3ub(125,223,63);
for(int i = 200; i <= 260; i++)
```

```
{
theta = i*3.142/180;
glVertex2f(80*cos(theta) +earth_x, 80*sin(theta) +earth_y);
}
glVertex2f(-40+earth_x, -10+earth_y);
glVertex2f(-20+earth_x, -20+earth_y);
glEnd();

glBegin(GL_POLYGON);
glColor3ub(125,223,63);
for(int i = 280; i <= 350; i++)
{
theta = i*3.142/180;
glVertex2f(80*cos(theta) +earth_x, 80*sin(theta) +earth_y);
}
glVertex2f(40+earth_x, -10+earth_y);
glVertex2f(20+earth_x, -20+earth_y);
glEnd();

glBegin(GL_POLYGON);
glColor3ub(125,223,63);
for(int i = 70; i <= 130; i++)
{
theta = i*3.142/180;
glVertex2f(80*cos(theta) +earth_x, 80*sin(theta) +earth_y);
}
glVertex2f(-40+earth_x, 10+earth_y);
glVertex2f(20+earth_x, 20+earth_y);
```

```
glEnd();
//mars
glBegin(GL_POLYGON);
glColor3ub(198,62,60);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(70*cos(theta) +mars_x, 70*sin(theta) +mars_y);
}
glEnd();

//jupiter
glBegin(GL_POLYGON);
glColor3ub(214,206,158);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(120*cos(theta) +jupiter_x, 120*sin(theta) + jupiter_y);
}
glEnd();

//saturn
glBegin(GL_POLYGON);
glColor3ub(231,203,191);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(130*cos(theta) + saturn_x, 60*sin(theta) +saturn_y);
```

```
}
glEnd();
glBegin(GL_POLYGON);
glColor3ub(37,9,50);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(110*cos(theta) + saturn_x, 50*sin(theta) +saturn_y);
}
glEnd();
glBegin(GL_POLYGON);
glColor3ub(227,197,101);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(90*cos(theta) +saturn_x, 90*sin(theta) +saturn_y);
}
glEnd();

//uranus
glBegin(GL_POLYGON);
glColor3ub(36,97,253);
for(int i = 0; i <= 360; i++)
{
theta = i*3.142/180;
glVertex2f(80*cos(theta) +uranus_x, 80*sin(theta) +uranus_y);
}
glEnd();
```

```
	//neptune
	glBegin(GL_POLYGON);
	glColor3ub(153,223,254);
	for(int i = 0; i <= 360; i++)
	{
	theta = i*3.142/180;
	glVertex2f(80*cos(theta) + neptune_x, 80*sin(theta) + neptune_y);
	}
	glEnd();




	glutSwapBuffers();
}

void init (void)
{
	glClearColor (0.0, 0.0, 0.0, 0.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(-1700, 1700, -1700, 1700);

}



int main(int argc, char** argv)
{
```

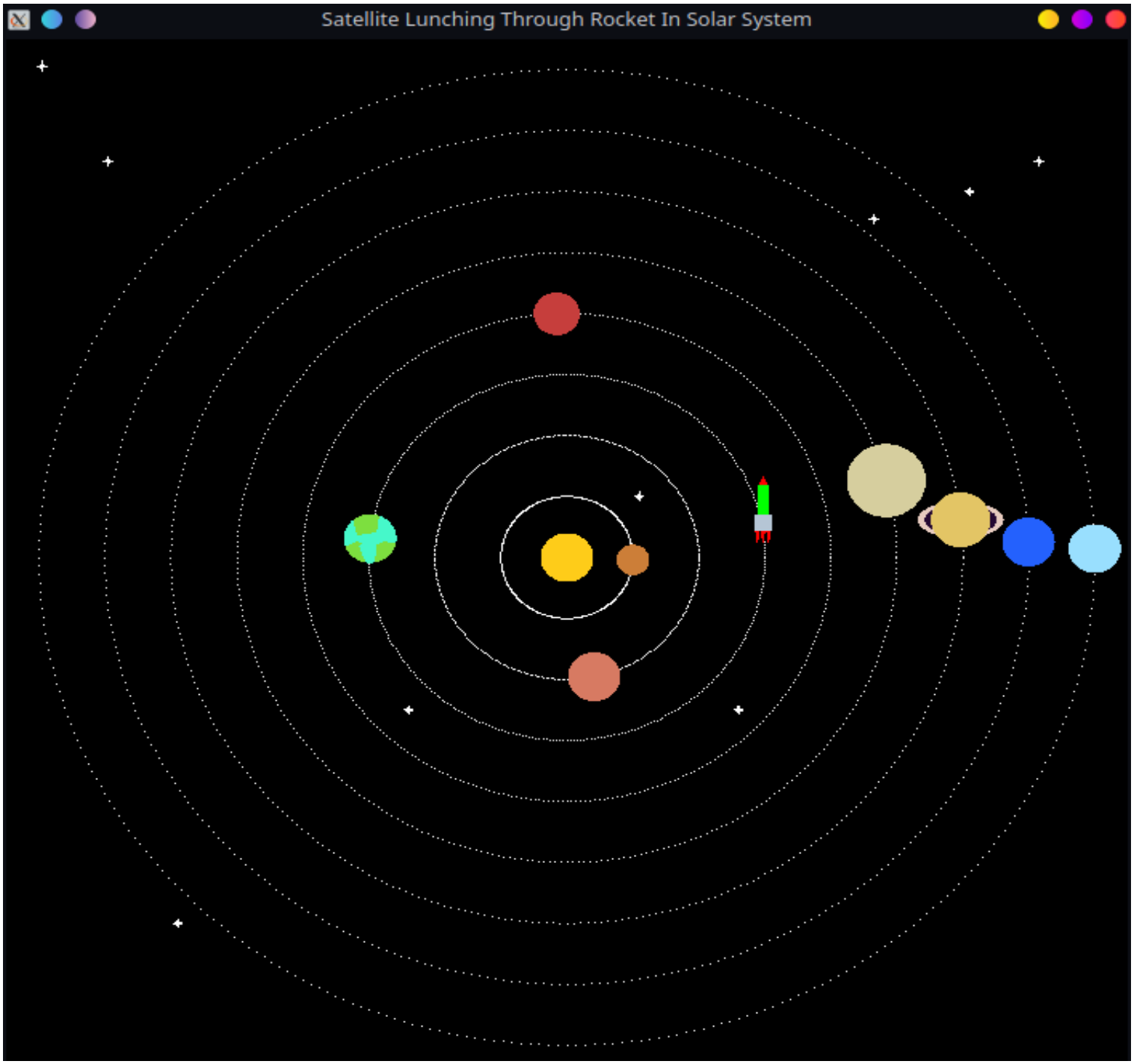
```
	glutInit(&argc, argv);
	glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
	glutInitWindowSize (700, 700);
	//glutInitWindowPosition (100, 0);
	glutCreateWindow ("Solar System Model");
	init ();
	glutDisplayFunc(display);
	glutTimerFunc(1000, timer, 0);
	glutMainLoop();
	return 0;
}

void timer(int)
{
	glutPostRedisplay();
	glutTimerFunc(1000/60, timer, 0);
	if(n < 360)
	{
	theta = n*3.142/180;
	neptune_x = 1600*cos(theta);
	neptune_y = 1600*sin(theta);
	}
	if(n >= 360)
	{
	n = 0;
	}
	if(u < 360)
```

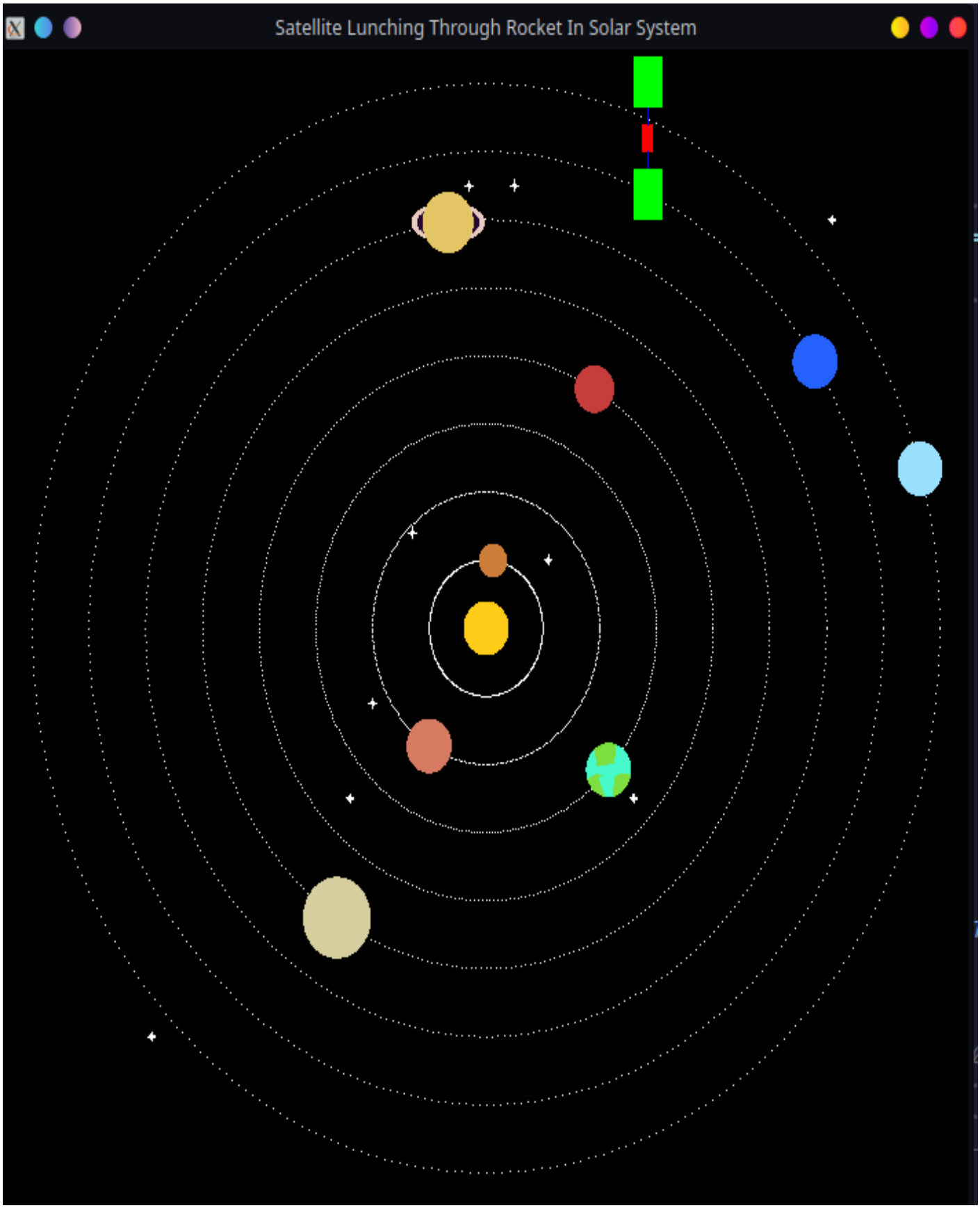
```
{
theta = u*3.142/180;
uranus_x = 1400*cos(theta);
uranus_y = 1400*sin(theta);
}
if(u >= 360)
{
u = 0;
}
if(s < 360)
{
theta = s*3.142/180;
saturn_x = 1200*cos(theta);
saturn_y = 1200*sin(theta);
}
if(s >= 360)
{
s = 0;
}
if(j < 360)
{
theta = j*3.142/180;
jupiter_x = 1000*cos(theta);
jupiter_y = 1000*sin(theta);
}
if(j >= 360)
{
j = 0;
```

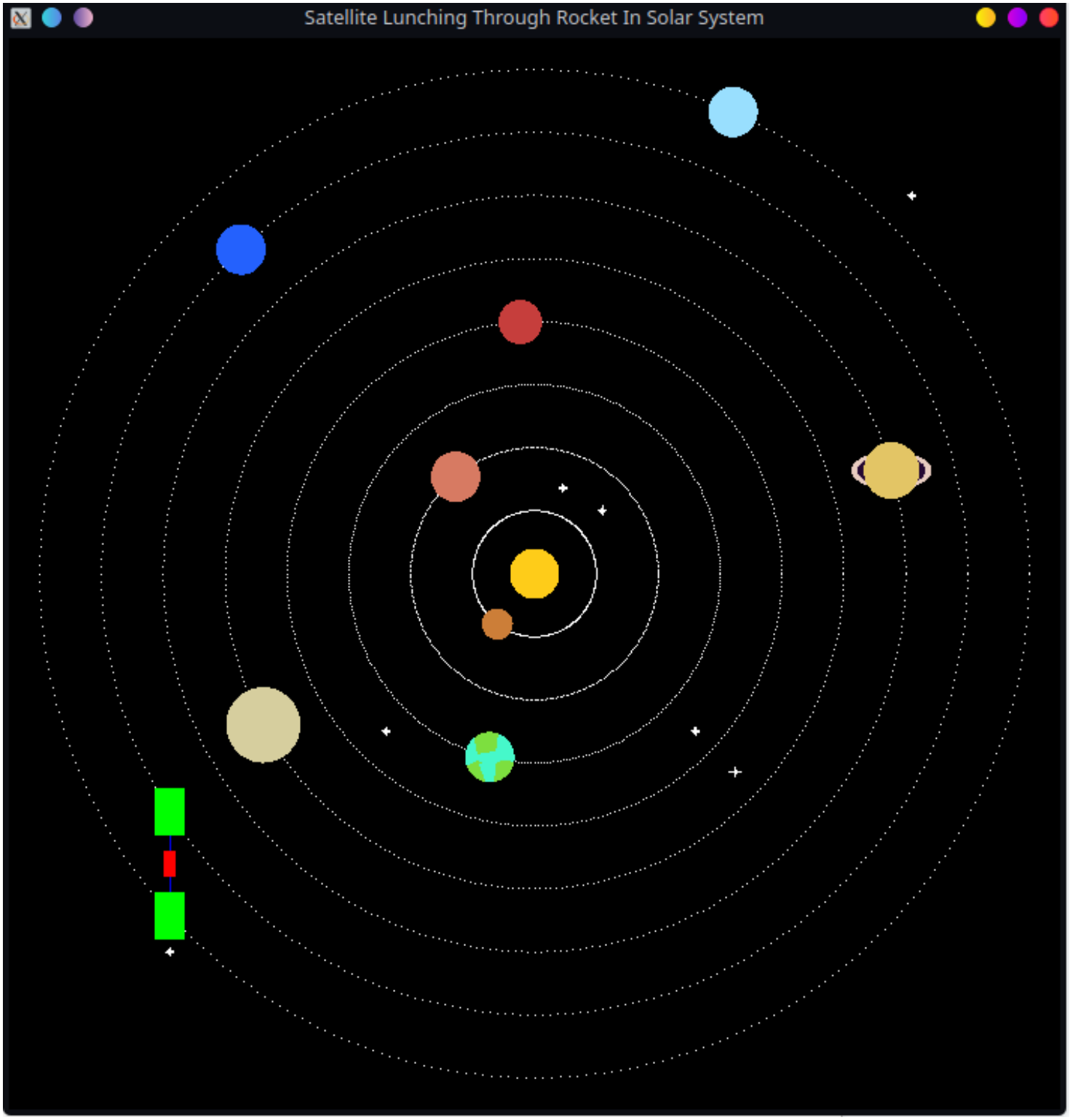
```
}

if(mr < 360)
{
theta = mr*3.142/180;
mars_x = 800*cos(theta);
mars_y = 800*sin(theta);
}
if(mr >= 360)
{
mr = 0;
}

if(e < 360)
{
theta = e*3.142/180;
earth_x = 600*cos(theta);
earth_y = 600*sin(theta);
}
if(e >= 360)
{
e = 0;
}

if(v < 360)
{
theta = v*3.142/180;
venus_x = 400*cos(theta);
```

```
venus_y = 400*sin(theta);
}
if(v >= 360)
{
v = 0;
}

if(m < 360)
{
theta = m*3.142/180;
mercury_x = 200*cos(theta);
mercury_y = 200*sin(theta);
}
if(m >= 360)
{
m = 0;
}
n += (0.006*2);
u += (0.012*2);
s += (0.034*2);
j += (0.084*2);
mr += (0.53*2);
e += (1*2);
v += (1.62*2);
m += (4.16*2);
}
```

Output:




Satellite Lunching Through Rocket In Solar System


Satellite Lunching Through Rocket In Solar System