

(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

: 08258 - 281039 – 281263, Fax: 08258 – 281265

Report on Mini Project

“Working of Satellite”

**Course Code: 18CS607**

**Course Name: Computer Graphics Lab**

Semester: 6 Section: D

# Submitted To

**Mr. Pradeep Kanchan**

**Asst Prof Gd III, Dept of CSE, NMAMIT**

**Submitted By**

Name: Shreyas K Shetty USN: 4NM18CS181

Name: Shashank V Jogi USN: 4NM18CS166

**Date of submission: 13/05/2021**

**Signature of Course Instructor**

# ABSTRACT:

In this project, we aim to implement an audio-visual model of a Working Of Satellite using OpenGL, which is an open-source cross platform interface for rendering 2D and 3D graphics. A Satellite is an object which has been placed into orbit by human endeavor. Such objects are sometimes called artificial satellites to distinguish them from natural satellites such as the Moon Satellites are used for a large number of purposes. Common types include military and civilian Earth observation satellites, communications satellites, navigation satellites, weather satellites, and research satellites. The components of each of these will be explained in detail in this report.

It will also allow user interactions using mouse and keyboard. It also has background audio and sound effects. Through this, we will be able to understand important OpenGL functions and GLUT libraries. This project helps us familiarize ourselves in the implementation of transformations like translation, rotation and scaling, properties of light, movement etc. on different objects.

# TABLE OF CONTENTS

**TITLE PAGE NO.**

1. INTRODUCTION 4
2. IMPLEMENTATION DETAILS 5-34
3. CONCLUSION 35
4. REFERENCES 36
5. APPENDIX 37-44

**INTRODUCTION:**

Computer Graphics involves technology that is vital to access the functions of a computer. The Process transforms and presents information in a visual form. Computer graphics plays an inevitable role in functioning of a computer by enabling the user interface visually. In today’s life, computer graphics has now become a common element in user interfaces, T.V. commercial motion pictures.

OpenGL (short for *Open Graphics Library*) is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit, to achieve hardware- accelerated rendering.

The project contains a full-fledged audio-visual demonstration of a Working of satellite. The components can be broadly classified into an external view of a satellite, an internal view and the launching view of the satellite.

Firstly, the title screen with all the menu options is displayed. On a “H” key click, the user is taken to the history of the satellite, which is the first component. Which include history of the first Indian satellite that is Aryabhata. On a “S” key click, the user is taken into the static satellite, which is the static view of the satellite.

On a “I” key click, the user is taken into the Inside view of the satellite, which include the inside view of the satellite showing all the internal components.

On the next key click “L”, the user is taken into the Launching state of the satellite, where the countdown will start form 10-0, after that the satellite is seen moving. The satellite will travel to the space. Once the satellite reaches outer space, the particular stage of thrusters will be ejected. And once it reaches to the final stage the satellite will be ejected and the satellite will starts moving around the planet. The satellite will revolve around the satellite and collect the data. All the view screens can be switched back using a menu that appears on Key click “B”.

This project is developed in C/C++ in Code Blocks. Keyboard and Mouse are the two input devices used which will also allow the user to interact with the application and also controls transition between the two views, external, launch and working view of the satellite.

# IMPLEMENTATION DETAILS:

The project “Working of Satellite” is created by using a set of functions for the respective components of the satellite.

1. **void display()** function displays the introductory window:

void display()

{

if (flag == 0)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

page();

glutSwapBuffers();

}

glFlush();

}

1. **void page()** contains the code design of home screen.

void page()

{

glColor3f(1, 1, 1);

glLineWidth(5);

glBegin(GL\_LINE\_LOOP);

glVertex2d(75, 430);

glVertex2d(375, 430);

glVertex2d(375, 300);

glVertex2d(75, 300);

glEnd();

drawstring(200, 470, "WORKING OF SATELLITE");

drawstring(100, 410, "SUBMITTED BY:");

drawstring(100, 380, "NAME : Shashank v Jogi");

drawstring(100, 360, "USN : 4NM18CS166");

drawstring(100, 340, "NAME : Shreyas K Shetty ");

drawstring(100, 320, "USN : 4NM18CS181");

glBegin(GL\_LINE\_LOOP);

glVertex2d(75, 190);

glVertex2d(375, 190);

glVertex2d(375, 280);

glVertex2d(75, 280);

glEnd();

drawstring(100, 260, "SUBMITTED TO:");

drawstring(100, 230, "Mr.Pradeep Kanchan");

drawstring(100, 210, "Asst Prof Gd III,Dept of CSE,NMAMIT");

glBegin(GL\_LINE\_LOOP);

glVertex2d(75, 15);

glVertex2d(375,15);

glVertex2d(375, 170);

glVertex2d(75, 170);

glEnd();

drawstring(100, 150, "INSTRUCTIONS:");

drawstring(100, 100, "Press S :To START");

drawstring(100, 120, "Press H :To Know History Of The First Indian Satellite");

drawstring(100, 80, "Press I :To Show Inside The Rocket");

drawstring(100, 60, "Press L :To Launch ");

//drawstring(100, 100, "Press B :To Back");

drawstring(100, 40, "Press Q :To Quit");

glFlush();

}

drawstring(100, 260, "SUBMITTED TO:");

drawstring(100, 230, "Mr.Pradeep Kanchan");

drawstring(100, 210, "Asst Prof Gd III,Dept of CSE,NMAMIT");

glBegin(GL\_LINE\_LOOP);

glVertex2d(75, 15);

glVertex2d(375,15);

glVertex2d(375, 170);

glVertex2d(75, 170);

glEnd();

drawstring(100, 150, "INSTRUCTIONS:");

drawstring(100, 100, "Press S :To START");

drawstring(100, 120, "Press H :To Know History Of The First Indian Satellite");

drawstring(100, 80, "Press I :To Show Inside The Rocket");

drawstring(100, 60, "Press L :To Launch ");

drawstring(100, 40, "Press Q :To Quit");

glFlush();

}

1. **void keyboard(**unsigned char key, int x, int y**)** has the code for manual launch of satellite using keyboard. It has call to functions static\_rocket(), Inside\_Rocket(), history(), static\_rocket\_to\_launch(),page().

void keyboard(unsigned char key, int x, int y)

{

if (key == 'S' || key == 's')

{

flag =1;

for(int i=0;i<200;i++)

static\_rocket();

}

if (key == 'I' || key == 'i')

{

flag =1;

for(int i=0;i<100;i++)

Inside\_Rocket();

}

if (key == 'H' || key == 'h')

{

flag =1;

for(int i=0;i<100;i++)

history();

}

if (key == 'L' || key == 'l')

{

flag =1;

countNum=0;

for(i=0;i<=1200;i++)

static\_rocket\_to\_launch();

for(int i=0;i<2;i++)

rocket\_to\_cam\_pos();

for(int i=0;i<12;i++)

rocket\_in\_motion();

}

if(key=='B' || key=='b')

{

flag =0;

page();

}

if (key == 'Q' || key == 'q')

exit(0);

}

1. **void drawstring**(int x, int y, char \*s) function display the text content.

void drawstring(int x, int y, char \*s)

{

char \*c;

glRasterPos2i(x, y);

for (c = s; \*c != '\0'; \*c++)

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, \*c);

}

1. **void semicircle**(float radius, float u, float v) has the code to display semicircle.

void semicircle(float radius,float u,float v)

{

glColor3f(1.0 ,1.0 ,1.0);

glBegin(GL\_POLYGON);

for (int i=135; i<=315; i++)

{

float degInRad = i\*DEG2RAD;

glVertex2f(u+cos(degInRad)\*radius,v+(sin(degInRad))\*radius

}

glEnd();

}

1. **void stars()** function has the code to display the stars.

void stars()

{

glColor3f(1.0,1.0,1.0);

glPointSize(1.37);

glBegin(GL\_POINTS);

glVertex2i(10,20);

glVertex2i(20,100);

glVertex2i(30,10);

glVertex2i(15,150);

glVertex2i(17,80);

glVertex2i(200,200);

glVertex2i(55,33);

glVertex2i(400,300);

glVertex2i(330,110);

glVertex2i(125,63);

glVertex2i(63,125);

glVertex2i(20,10);

glVertex2i(110,330);

glVertex2i(440,430);

glVertex2i(32,65);

glVertex2i(110,440);

glVertex2i(210,230);

glVertex2i(390,490);

glVertex2i(12,90);

glVertex2i(400,322);

glVertex2i(420,366);

glVertex2i(455,400);

glVertex2i(20,20);

glVertex2i(111,120);

glVertex2i(401,200);

glVertex2i(230,30);

glVertex2i(220,20);

glVertex2i(122,378);

glVertex2i(133,340);

glVertex2i(345,420);

glVertex2i(130,360);

glVertex2i(333,120);

glVertex2i(250,22);

glVertex2i(242,11);

glVertex2i(280,332);

glVertex2i(233,40);

glVertex2i(210,418);

glVertex2i(256,12);

glVertex2i(288,232);

glVertex2i(247,36);

glVertex2i(229,342);

glVertex2i(257,47);

glVertex2i(290,63);

glVertex2i(232,72);

glVertex2i(243,143);

glVertex2i(100,200);

glVertex2i(90,250);

glVertex2i(80,225);

glVertex2i(50,333);

glVertex2i(60,350);

glVertex2i(243,143);

glVertex2i(243,143);

glEnd();

}

1. **void stars1()** function has the code to display the stars in space around the planet.

void stars1()

{

int l;

glColor3f(1.0,1.0,1.0);

glPointSize(1.0);

glBegin(GL\_POINTS);

glVertex2i(50,20);

glVertex2i(70,100);

glVertex2i(80,10);

glVertex2i(65,150);

glVertex2i(67,80);

glVertex2i(105,33);

glVertex2i(450,300);

glVertex2i(380,110);

glVertex2i(175,63);

glVertex2i(113,125);

glVertex2i(70,10);

glVertex2i(160,330);

glVertex2i(490,430);

glVertex2i(82,65);

glVertex2i(160,440);

glVertex2i(440,490);

glVertex2i(62,90);

glVertex2i(450,322);

glVertex2i(420,366);

glVertex2i(455,400);

glVertex2i(60,20);

glVertex2i(111,120);

glVertex2i(451,200);

glVertex2i(280,30);

glVertex2i(220,20);

glVertex2i(132,378);

glVertex2i(173,340);

glVertex2i(325,420);

glVertex2i(180,360);

glVertex2i(383,120);

glVertex2i(200,22);

glVertex2i(342,11);

glVertex2i(330,332);

glVertex2i(283,40);

glVertex2i(210,418);

glVertex2i(256,12);

glVertex2i(288,232);

glVertex2i(247,36);

glVertex2i(229,342);

glVertex2i(257,47);

glVertex2i(290,63);

glVertex2i(232,72);

glVertex2i(243,143);

glVertex2i(100,200);

glVertex2i(90,250);

glVertex2i(80,225);

glVertex2i(50,333);

glVertex2i(60,350);

glVertex2i(243,143);

glVertex2i(243,143);

glEnd();

for(l=0;l<=10000;l++)

;

}

1. **void static\_rocket()** has the code which displays the static rocket view upon clicking the key.

void static\_rocket()

{

glClearColor(0.196078 ,0.6 ,0.8,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glColor3f(0.0,0.3,0.0);

glBegin(GL\_POLYGON);//green ground

glVertex2f(0.0,0.0);

glVertex2f(0.0,250.0);

glVertex2f(270.0,250.0);

glVertex2f(500.0,50.0);

glVertex2f(500.0,0.0);

glEnd();

glBegin(GL\_POLYGON);//green ground

glVertex2f(280.0,250.0);

glVertex2f(500.0,250.0);

glVertex2f(500.0,60.0);

glEnd();

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON);//road

glVertex2f(260.0,250.0);

glVertex2f(290.0,250.0);

glVertex2f(500.0,70.0);

glVertex2f(500.0,40.0);

glEnd();

glColor3f(0.0,0.0,0.0);

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//house 1

glVertex2f(250.0,250.0);

glVertex2f(300.0,250.0);

glVertex2f(300.0,350.0);

glVertex2f(250.0,350.0);

glEnd();

glColor3f(0.7,0.7,0.7);

glBegin(GL\_POLYGON);//HOUSE A

glVertex2f(255,267.5);

glVertex2f(275.0,267.5);

glVertex2f(275.0,277.5);

glVertex2f(255.0,277.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE B

glVertex2f(255,285.0);

glVertex2f(275.0,285);

glVertex2f(275.0,295);

glVertex2f(255.0,295);

glEnd();

glBegin(GL\_POLYGON);//HOUSE C

glVertex2f(255,302.5);

glVertex2f(275.0,302.5);

glVertex2f(275.0,312.5);

glVertex2f(255.0,312.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE D

glVertex2f(255,320.0);

glVertex2f(275.0,320.0);

glVertex2f(275.0,330.0);

glVertex2f(255.0,330.0);

glEnd();

glBegin(GL\_POLYGON);//HOUSE E

glVertex2f(285,267.5);

glVertex2f(295.0,267.5);

glVertex2f(295.0,277.5);

glVertex2f(285.0,277.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE F

glVertex2f(285,285.0);

glVertex2f(295.0,285);

glVertex2f(295.0,295);

glVertex2f(285.0,295);

glEnd();

glBegin(GL\_POLYGON);//HOUSE G

glVertex2f(285,302.5);

glVertex2f(295.0,302.5);

glVertex2f(295.0,312.5);

glVertex2f(285.0,312.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE H

glVertex2f(285,320.0);

glVertex2f(295.0,320.0);

glVertex2f(295.0,330.0);

glVertex2f(285.0,330.0);

glEnd();

glColor3f(0.647059 ,0.164706 ,0.164706);

glBegin(GL\_POLYGON);//solid cone

glVertex2f(26,250);

glVertex2f(52,250);

glVertex2f(39,290);

glEnd();

semicircle(20.0,50,300);

glColor3f(0.0,0.0 ,0.0);

glBegin(GL\_LINES);//wires

glVertex2f(37,313);

glVertex2f(62,310);

glVertex2f(63,287);

glVertex2f(62,310);

glEnd();

glColor3f(1.0,1.0,1.0);

glEnd();

glPointSize(2.0);

glColor3f(1.0,1.0 ,1.0);

glBegin(GL\_POINTS);//road paint

glVertex2f(497,56);

glVertex2f(488,65);

glVertex2f(479,74);

glVertex2f(470,83);

glVertex2f(460,92);

glVertex2f(450,101);

glVertex2f(439,110);

glVertex2f(428,119);

glVertex2f(418,128);

glVertex2f(408,137);

glVertex2f(398,146);

glVertex2f(388,155);

glVertex2f(378,164);

glVertex2f(366,173);

glVertex2f(356,182);

glVertex2f(346,191);

glVertex2f(336,200);

glVertex2f(324,209);

glVertex2f(314,218);

glVertex2f(304,227);

glVertex2f(294,234);

glVertex2f(284,243);

glVertex2f(278,248);

glEnd();

glColor3f(0.0,0.0,0.0);//stand object

glBegin(GL\_POLYGON);

glVertex2f(130,10.0);

glVertex2f(160,10.0);

glVertex2f(160,180.0);

glVertex2f(130,180.0);

glEnd();

glBegin(GL\_LINES);

glVertex2f(130,30.0);

glVertex2f(262,30.0);

glVertex2f(130,130.0);

glVertex2f(260,130.0);

glEnd();

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//core

glVertex2f(237.5,20.0);

glVertex2f(262.5,20.0);

glVertex2f(262.5,120.0);

glVertex2f(237.5,120.0);

glEnd();

glColor3f(1.0,1.0,1.0);//bonnet

glBegin(GL\_POLYGON);//front

glVertex2f(237.5,120.0);

glVertex2f(262.5,120.0);

glVertex2f(250,170.0);

glEnd();

glColor3f(1.0,0.0,0.0);

glBegin(GL\_POLYGON);//left\_side\_top

glVertex2f(237.5,120.0);

glVertex2f(217.5,95.0);

glVertex2f(237.5,95.0);

glEnd();

glBegin(GL\_POLYGON);//left\_side\_bottom

glVertex2f(237.5,20.0);

glVertex2f(217.5,20.0);

glVertex2f(237.5,70.0);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_bottom

glVertex2f(262.5,20.0);

glVertex2f(282.5,20.0);

glVertex2f(262.5,70.0);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_top

glVertex2f(262.5,120.0);

glVertex2f(262.5,95.0);

glVertex2f(282.5,95.0);

glEnd();

glColor3f(1.0,0.45 ,0.0);

glBegin(GL\_POLYGON);//bottom\_1\_exhaust

glVertex2f(237.5,20.0);

glVertex2f(244.5,20.0);

glVertex2f(241,0.0);

glEnd();

glBegin(GL\_POLYGON);//bottom\_2\_exhaust

glVertex2f(246.5,20.0);

glVertex2f(253.5,20.0);

glVertex2f(249.5,0.0);

glEnd();

glBegin(GL\_POLYGON);//bottom\_3\_exhaust

glVertex2f(262.5,20.0);

glVertex2f(255.5,20.0);

glVertex2f(258.5,0.0);

glEnd();

glColor3f(0.556863 ,0.137255 ,0.419608);

glBegin(GL\_POLYGON);//left\_stand\_holder

glVertex2f(182.5,85.0);

glVertex2f(182.5,0.0);

glVertex2f(187.5,0.0);

glVertex2f(187.5,80.0);

glVertex2f(237.5,80.0);

glVertex2f(237.5,85.0);

glVertex2f(182.5,85.0);

glEnd();

glBegin(GL\_POLYGON);

glVertex2f(312.5,85.0);//right\_stand\_holder

glVertex2f(312.5,0.0);

glVertex2f(307.5,0.0);

glVertex2f(307.5,80.0);

glVertex2f(262.5,80.0);

glVertex2f(262.5,85.0);

glVertex2f(312.5,85.0);

glEnd();

glColor3f(1.0,1.0,1.0);

drawstring(380,30,"Press [B] to Go Back");

for(j=0;j<=1000000;j++)

;

glutSwapBuffers();

glutPostRedisplay();

glFlush();

}

1. **void static\_rocket\_to\_launch()** has the code that displays the static rocket then the countdown starts.

void static\_rocket\_to\_launch()

{

countNum++;

glClearColor(0.196078 ,0.6 ,0.8,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glColor3f(0.0,0.3,0.0);

glBegin(GL\_POLYGON);//green ground

glVertex2f(0.0,0.0);

glVertex2f(0.0,250.0);

glVertex2f(270.0,250.0);

glVertex2f(500.0,50.0);

glVertex2f(500.0,0.0);

glEnd();

glBegin(GL\_POLYGON);//green ground

glVertex2f(280.0,250.0);

glVertex2f(500.0,250.0);

glVertex2f(500.0,60.0);

glEnd();

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON);//road

glVertex2f(260.0,250.0);

glVertex2f(290.0,250.0);

glVertex2f(500.0,70.0);

glVertex2f(500.0,40.0);

glEnd();

glColor3f(0.0,0.0,0.0);

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//house 1

glVertex2f(250.0,250.0);

glVertex2f(300.0,250.0);

glVertex2f(300.0,350.0);

glVertex2f(250.0,350.0);

glEnd();

glColor3f(0.7,0.7,0.7);

glBegin(GL\_POLYGON);//HOUSE A

glVertex2f(255,267.5);

glVertex2f(275.0,267.5);

glVertex2f(275.0,277.5);

glVertex2f(255.0,277.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE B

glVertex2f(255,285.0);

glVertex2f(275.0,285);

glVertex2f(275.0,295);

glVertex2f(255.0,295);

glEnd();

glBegin(GL\_POLYGON);//HOUSE C

glVertex2f(255,302.5);

glVertex2f(275.0,302.5);

glVertex2f(275.0,312.5);

glVertex2f(255.0,312.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE D

glVertex2f(255,320.0);

glVertex2f(275.0,320.0);

glVertex2f(275.0,330.0);

glVertex2f(255.0,330.0);

glEnd();

glBegin(GL\_POLYGON);//HOUSE E

glVertex2f(285,267.5);

glVertex2f(295.0,267.5);

glVertex2f(295.0,277.5);

glVertex2f(285.0,277.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE F

glVertex2f(285,285.0);

glVertex2f(295.0,285);

glVertex2f(295.0,295);

glVertex2f(285.0,295);

glEnd();

glBegin(GL\_POLYGON);//HOUSE G

glVertex2f(285,302.5);

glVertex2f(295.0,302.5);

glVertex2f(295.0,312.5);

glVertex2f(285.0,312.5);

glEnd();

glBegin(GL\_POLYGON);//HOUSE H

glVertex2f(285,320.0);

glVertex2f(295.0,320.0);

glVertex2f(295.0,330.0);

glVertex2f(285.0,330.0);

glEnd();

glColor3f(0.647059 ,0.164706 ,0.164706);

glBegin(GL\_POLYGON);//solid cone

glVertex2f(26,250);

glVertex2f(52,250);

glVertex2f(39,290);

glEnd();

semicircle(20.0,50,300);

glColor3f(0.0,0.0 ,0.0);

glBegin(GL\_LINES);//wires

glVertex2f(37,313);

glVertex2f(62,310);

glVertex2f(63,287);

glVertex2f(62,310);

glEnd();

glColor3f(1.0,1.0,1.0);

glEnd();

glPointSize(2.0);

glColor3f(1.0,1.0 ,1.0);

glBegin(GL\_POINTS);//road paint

glVertex2f(497,56);

glVertex2f(488,65);

glVertex2f(479,74);

glVertex2f(470,83);

glVertex2f(460,92);

glVertex2f(450,101);

glVertex2f(439,110);

glVertex2f(428,119);

glVertex2f(418,128);

glVertex2f(408,137);

glVertex2f(398,146);

glVertex2f(388,155);

glVertex2f(378,164);

glVertex2f(366,173);

glVertex2f(356,182);

glVertex2f(346,191);

glVertex2f(336,200);

glVertex2f(324,209);

glVertex2f(314,218);

glVertex2f(304,227);

glVertex2f(294,234);

glVertex2f(284,243);

glVertex2f(278,248);

glEnd();

glColor3f(0.0,0.0,0.0);//stand object

glBegin(GL\_POLYGON);

glVertex2f(130,10.0);

glVertex2f(160,10.0);

glVertex2f(160,180.0);

glVertex2f(130,180.0);

glEnd();

glBegin(GL\_LINES);

glVertex2f(130,30.0);

glVertex2f(262,30.0);

glVertex2f(130,130.0);

glVertex2f(260,130.0);

glEnd();

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//core

glVertex2f(237.5,20.0);

glVertex2f(262.5,20.0);

glVertex2f(262.5,120.0);

glVertex2f(237.5,120.0);

glEnd();

glColor3f(1.0,1.0,1.0);//bonnet

glBegin(GL\_POLYGON);//front

glVertex2f(237.5,120.0);

glVertex2f(262.5,120.0);

glVertex2f(250,170.0);

glEnd();

glColor3f(1.0,0.0,0.0);

glBegin(GL\_POLYGON);//left\_side\_top

glVertex2f(237.5,120.0);

glVertex2f(217.5,95.0);

glVertex2f(237.5,95.0);

glEnd();

glBegin(GL\_POLYGON);//left\_side\_bottom

glVertex2f(237.5,20.0);

glVertex2f(217.5,20.0);

glVertex2f(237.5,70.0);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_bottom

glVertex2f(262.5,20.0);

glVertex2f(282.5,20.0);

glVertex2f(262.5,70.0);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_top

glVertex2f(262.5,120.0);

glVertex2f(262.5,95.0);

glVertex2f(282.5,95.0);

glEnd();

glColor3f(1.0,0.45 ,0.0);

glBegin(GL\_POLYGON);//bottom\_1\_exhaust

glVertex2f(237.5,20.0);

glVertex2f(244.5,20.0);

glVertex2f(241,0.0);

glEnd();

glBegin(GL\_POLYGON);//bottom\_2\_exhaust

glVertex2f(246.5,20.0);

glVertex2f(253.5,20.0);

glVertex2f(249.5,0.0);

glEnd();

glBegin(GL\_POLYGON);//bottom\_3\_exhaust

glVertex2f(262.5,20.0);

glVertex2f(255.5,20.0);

glVertex2f(258.5,0.0);

glEnd();

glColor3f(0.556863 ,0.137255 ,0.419608);

glBegin(GL\_POLYGON);//left\_stand\_holder

glVertex2f(182.5,85.0);

glVertex2f(182.5,0.0);

glVertex2f(187.5,0.0);

glVertex2f(187.5,80.0);

glVertex2f(237.5,80.0);

glVertex2f(237.5,85.0);

glVertex2f(182.5,85.0);

glEnd();

glBegin(GL\_POLYGON);

glVertex2f(312.5,85.0);//right\_stand\_holder

glVertex2f(312.5,0.0);

glVertex2f(307.5,0.0);

glVertex2f(307.5,80.0);

glVertex2f(262.5,80.0);

glVertex2f(262.5,85.0);

glVertex2f(312.5,85.0);

glEnd();

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON);

glVertex2f(350,280);

glVertex2f(470,280);

glVertex2f(470,380);

glVertex2f(350,380);

glEnd();

glColor3f(1.0,1.0,1.0);

for(j=0;j<=100;j++)

{

drawstring(360,360,"READY TO LAUNCH IN");

}

if(countNum<=100)

drawstring(405,320,"10");

if(countNum>100 && countNum<=200)

drawstring(410,320,"9");

if(countNum>200 && countNum<=300)

drawstring(410,320,"8");

if(countNum>300 && countNum<=400)

drawstring(410,320,"7");

if(countNum>400 && countNum<=500)

drawstring(410,320,"6");

if(countNum>500 && countNum<=600)

drawstring(410,320,"5");

if(countNum>600 && countNum<=700)

drawstring(410,320,"4");

if(countNum>700 && countNum<=800)

drawstring(410,320,"3");

if(countNum>800 && countNum<=900)

drawstring(410,320,"2");

if(countNum>900 && countNum<=1000)

drawstring(410,320,"1");

if(countNum>1000 && countNum<=1100)

drawstring(410,320,"0");

if(countNum>1100 && countNum<=1200)

drawstring(405,320,"GO");

glColor3f(1.0,1.0,1.0);

drawstring(380,30,"Press [B] to Go Back");

for(j=0;j<=100000;j++)

;

glutSwapBuffers();

glutPostRedisplay();

glFlush();

}

1. **void Inside\_Rocket()** has the code which displays the inside view of rocket.

void Inside\_Rocket()

{

glClearColor(0.196078 ,0.6 ,0.8,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glColor3f(1.0,1.0,1.0);//bonnet

glBegin(GL\_POLYGON);//front

glVertex2f(180,370.0);

glVertex2f(320,370.0);

glVertex2f(250,499.0);

glEnd();

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//core

glVertex2f(180.0,70.0);

glVertex2f(320.0,70.0);

glVertex2f(320.0,370.0);

glVertex2f(180.0,370.0);

glEnd();

glColor3f(0.0,0.0 ,0.0);

drawstring(230,330,"Fuel Section");

glColor3f(1.0,1.0,1.0);

glBegin(GL\_POLYGON);

glVertex2f(200.0,120.0);

glVertex2f(245.0,120.0);

glVertex2f(245.0,310.0);

glVertex2f(200.0,310.0);

glEnd();

glColor3f(0.0,0.0 ,0.0);

drawstring(202,200,"Fuel Tank");

glColor3f(0.0,1.0,0.0);

glBegin(GL\_POLYGON);

glVertex2f(255.0,120.0);

glVertex2f(300.0,120.0);

glVertex2f(300.0,310.0);

glVertex2f(255.0,310.0);

glEnd();

glColor3f(0.0,0.0 ,0.0);

drawstring(258,200,"Fuel Tank");

glColor3f(1.0,0.0 ,0.0);

glBegin(GL\_POLYGON);//left\_side\_top

glVertex2f(180.0,370.0);

glVertex2f(180.0,310.0);

glVertex2f(120.0,310.0);

glEnd();

glBegin(GL\_POLYGON);//left\_side\_bottom

glVertex2f(180.0,70.0);

glVertex2f(120,70.0);

glVertex2f(180.0,170.0);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_bottom

glVertex2f(320.0,70.0);

glVertex2f(380.0,70.0);

glVertex2f(320.0,170.0);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_top

glVertex2f(320,370.0);

glVertex2f(320.0,310.0);

glVertex2f(380.0,310.0);

glEnd();

glColor3f(1.0,0.45 ,0.0);

glBegin(GL\_POLYGON);//bottom\_1\_exhaust

glVertex2f(190.0,70.0);

glVertex2f(220.0,70.0);

glVertex2f(205,0.0);

glEnd();

glBegin(GL\_POLYGON);//bottom\_2\_exhaust

glVertex2f(240.0,70.0);

glVertex2f(260.0,70.0);

glVertex2f(250.0,0.0);

glEnd();

glBegin(GL\_POLYGON);//bottom\_3\_exhaust

glVertex2f(310.0,70.0);

glVertex2f(280.0,70.0);

glVertex2f(295.0,0.0);

glEnd();

glColor3f(0.8314,0.6863,0.2157);

glBegin(GL\_POLYGON);//core

glVertex2f(220.0,405.0);

glVertex2f(240.0,405.0);

glVertex2f(245.0,430.0);

glVertex2f(235.0,430.0);

glEnd();

glBegin(GL\_POLYGON);//core

glVertex2f(250.0,405.0);

glVertex2f(265.0,405.0);

glVertex2f(280.0,430.0);

glVertex2f(255.0,430.0);

glEnd();

glColor3f(0.0,0.0,0.0);

glBegin(GL\_POLYGON);//core

glVertex2f(220.0,380.0);

glVertex2f(240.0,380.0);

glVertex2f(270.0,440.0);

glVertex2f(250.0,440.0);

glEnd();

glColor3f(0.0,0.0 ,0.0);

drawstring(233,455,"Satellite");

drawstring(380,30,"Press [B] to Go Back");

glutSwapBuffers();

glutPostRedisplay();

glFlush();

}

1. **void history()** function has the code that computes the history of satellite.

void history()

{

glClearColor(0.196078 ,0.6 ,0.8,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glColor3f(1.0,1.0,1.0);

drawstring(210,450,"ARYABHATA");

drawstring(40,400,"Aryabhata, first unmanned  Earth  satellite built by  India. named after the famous Indian astronomer.It was launched");

drawstring(20,380,"on 19 April 1975 from Kapustin Yar, a Russian rocket launch and development site in Astrakhan Oblast using a Kosmos -");

drawstring(20,360,"3M launch vehicle. It was built by the ISRO.");

drawstring(40,330,"It was launched by India on 19 April 1975  from  Kapustin Yar , a  Russian  rocket launch and development site in ");

drawstring(20,310,"Astrakhan Oblast using a  Kosmos - 3M  launch vehicle. It was built by the Indian Space Research Organisation (ISRO).The ");

drawstring(20,290,"launch came from an agreement between India and the Soviet Union directed by  UR Rao  and signed in 1972. It allowed the ");

drawstring(20,270,"USSR to use Indian ports for tracking ships and launching vessels in return for launching various different Indian satellites.");

drawstring(40,240,"On 19 April 1975, the satellite's 96.46 - minute orbit had an  apogee  of 619 kilometres (385 mi) and a perigee of 563");

drawstring(20,220,"kilometres (350 mi),at an inclination of 50.7 degrees. It was built to conduct experiments in X-ray astronomy, aeronomics,");

drawstring(20,200,"and solar physics.The spacecraft was a 26-sided polyhedron 1.4 metres (4.6 ft) in diameter. All faces ( except the top and ");

drawstring(20,180,"bottom) were covered with solar cells. A power failure halted experiments after four days and 60 orbits with all signals from");

drawstring(20,160,"the spacecraft lost after five days of the operation . Spacecraft mainframe remained active till March 1981.Due to orbital");

drawstring(20,140,"decay the satellite entered Earth's atmosphere on 11 February 1992.");

drawstring(380,30,"Press [B] to Go Back");

glutSwapBuffers();

glutPostRedisplay();

glFlush();

}

1. **void rocket\_to\_cam\_pos()** functionhas the code in which the rocket starts to take off.

void rocket\_to\_cam\_pos()

{

count++;

count3++;

for(i=0;i<=200;i++)

{

glClearColor(0.196078 ,0.6 ,0.8,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//core

glVertex2f(237.5,20.0+i);

glVertex2f(262.5,20.0+i);

glVertex2f(262.5,120.0+i);

glVertex2f(237.5,120.0+i);

glEnd();

glColor3f(1.0,1.0,1.0);//bonnet

glBegin(GL\_POLYGON);//front

glVertex2f(237.5,120.0+i);

glVertex2f(262.5,120.0+i);

glVertex2f(250,170.0+i);

glEnd();

glColor3f(1.0,0.0,0.0);

glBegin(GL\_POLYGON);//left\_side\_top

glVertex2f(237.5,120.0+i);

glVertex2f(217.5,95.0+i);

glVertex2f(237.5,95.0+i);

glEnd();

glBegin(GL\_POLYGON);//left\_side\_bottom

glVertex2f(237.5,20.0+i);

glVertex2f(217.5,20.0+i);

glVertex2f(237.5,70.0+i);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_bottom

glVertex2f(262.5,20.0+i);

glVertex2f(282.5,20.0+i);

glVertex2f(262.5,70.0+i);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_top

glVertex2f(262.5,120.0+i);

glVertex2f(262.5,95.0+i);

glVertex2f(282.5,95.0+i);

glEnd();

glColor3f(0.556863 ,0.137255 ,0.419608);

glBegin(GL\_POLYGON);//bottom\_1\_exhaust

glVertex2f(237.5,20.0+i);

glVertex2f(244.5,20.0+i);

glVertex2f(241,0.0+i);

glEnd();

glBegin(GL\_POLYGON);//bottom\_2\_exhaust

glVertex2f(246.5,20.0+i);

glVertex2f(253.5,20.0+i);

glVertex2f(249.5,0.0+i);

glEnd();

glBegin(GL\_POLYGON);//bottom\_3\_exhaust

glVertex2f(262.5,20.0+i);

glVertex2f(255.5,20.0+i);

glVertex2f(258.5,0.0+i);

glEnd();

if((p%2)==0)

glColor3f(1.0,0.25,0.0);

else

glColor3f(1.0,0.816,0.0);

glBegin(GL\_POLYGON);//outer fume

glVertex2f(237.5,20+i);

glVertex2f(234.16,16.66+i);

glVertex2f(230.82,13.32+i);

glVertex2f(227.48,9.98+i);

glVertex2f(224.14,6.64+i);

glVertex2f(220.8,3.3+i);

glVertex2f(217.5,0+i);

glVertex2f(221.56,-5+i);

glVertex2f(225.62,-10+i);

glVertex2f(229.68,-15+i);

glVertex2f(233.74,-20+i);

glVertex2f(237.8,-25+i);

glVertex2f(241.86,-30+i);

glVertex2f(245.92,-35+i);

glVertex2f(250,-40+i);

glVertex2f(254.06,-35+i);

glVertex2f(258.12,-30+i);

glVertex2f(262.18,-25+i);

glVertex2f(266.24,-20+i);

glVertex2f(270.3,-15+i);

glVertex2f(274.36,-10+i);

glVertex2f(278.42,-5+i);

glVertex2f(282.5,0+i);

glVertex2f(278.5,4+i);

glVertex2f(274.5,8+i);

glVertex2f(270.5,12+i);

glVertex2f(266.5,16+i);

glVertex2f(262.5,20+i);//28 points

glEnd();

if((p%2)==0)

glColor3f(1.0,0.816,0.0);

else

glColor3f(1.0,0.25,0.0);

glBegin(GL\_POLYGON);//inner fume

glVertex2f(237.5,20+i);

glVertex2f(236.5,17.5+i);

glVertex2f(235.5,15+i);

glVertex2f(234.5,12.5+i);

glVertex2f(233.5,10+i);

glVertex2f(232.5,7.5+i);

glVertex2f(236,5+i);

glVertex2f(239.5,2.5+i);

glVertex2f(243,0+i);

glVertex2f(246.5,-2.5+i);

glVertex2f(250,-5+i);

glVertex2f(253.5,-2.5+i);

glVertex2f(257,0+i);

glVertex2f(260.5,2.5+i);

glVertex2f(264,5+i);

glVertex2f(267.5,7.5+i);

glVertex2f(266.5,10+i);

glVertex2f(265.5,12.5+i);

glVertex2f(264.5,15+i);

glVertex2f(263.5,17.5+i);

glVertex2f(262.5,20+i);//21 points

glEnd();

p=p+1;

for(j=0;j<=1000000;j++)

;

glutSwapBuffers();

glutPostRedisplay();

glFlush();

}

}

1. **void rocket\_in\_motion()** function has the code for the rocket which rocket removes the engines and revolves around the planet. which calls the function stars(),stars1().

void rocket\_in\_motion()

{

count++;

for(i=50;i<=280;i++)

{

if(count>=2)

{

glClearColor(0.0 ,0.0 ,0.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

if(flag1==0)

{

stars();

flag1=1;

}

else

{

stars1();

flag1=0;

}

}

else

{

glClearColor(0.196078 ,0.6 ,0.8,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

}

if(count<=6){

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//core

glVertex2f(237.5,20.0+i);

glVertex2f(262.5,20.0+i);

glVertex2f(262.5,120.0+i);

glVertex2f(237.5,120.0+i);

glEnd();

glColor3f(1.0 ,0.45 ,0.0);

glBegin(GL\_POLYGON);//bottom\_1\_exhaust

glVertex2f(237.5,20.0+i);

glVertex2f(244.5,20.0+i);

glVertex2f(241,0.0+i);

glEnd();

glBegin(GL\_POLYGON);//bottom\_2\_exhaust

glVertex2f(246.5,20.0+i);

glVertex2f(253.5,20.0+i);

glVertex2f(249.5,0.0+i);

glEnd();

glBegin(GL\_POLYGON);//bottom\_3\_exhaust

glVertex2f(262.5,20.0+i);

glVertex2f(255.5,20.0+i);

glVertex2f(258.5,0.0+i);

glEnd();

}

if(count>6){

counti+=1;

glColor3f(0.8,0.498039 ,0.196078);

glBegin(GL\_POLYGON);//core

glVertex2f(237.5,20.0-counti);

glVertex2f(262.5,20.0-counti);

glVertex2f(262.5,120.0-counti);

glVertex2f(237.5,120.0-counti);

glColor3f(1.0 ,0.45 ,0.0);

glBegin(GL\_POLYGON);//bottom\_1\_exhaust

glVertex2f(237.5,20.0-counti);

glVertex2f(244.5,20.0-counti);

glVertex2f(241,0.0-counti);

glEnd();

glBegin(GL\_POLYGON);//bottom\_2\_exhaust

glVertex2f(246.5,20.0-counti);

glVertex2f(253.5,20.0-counti);

glVertex2f(249.5,0.0-counti);

glEnd();

glBegin(GL\_POLYGON);//bottom\_3\_exhaust

glVertex2f(262.5,20.0-counti);

glVertex2f(255.5,20.0-counti);

glVertex2f(258.5,0.0-counti);

glEnd();

}

if(count==8 || count==10)

{

glColor3f(1.0,1.0,0.0);

mars(100.0);

planet1(40);

earth(20);

int k = i;

glColor3f(0.8314,0.6863,0.2157);

glBegin(GL\_POLYGON);//core

glVertex2f(87.5+k,260.0);

glVertex2f(102.5+k,260.0);

glVertex2f(102.5+k,230.0);

glVertex2f(87.5+k,230.0);

glEnd();

glColor3f(0.0,0.0,0.4);

glBegin(GL\_POLYGON);//side-panels

glVertex2f(87.5+k,250.0);

glVertex2f(80+k,250.0);

glVertex2f(80+k,240.0);

glVertex2f(87.5+k,240.0);

glVertex2f(102.5+k,250.0);

glVertex2f(77.5+k,250.0);

glVertex2f(77.5+k,240.0);

glVertex2f(102.5+k,240.0);

glEnd();

for(j=0;j<=10000000;j++)

;

}

if(count==9 || count==11)

{

int k = i;

glColor3f(0.8314,0.6863,0.2157);

glBegin(GL\_POLYGON);//core

glVertex2f(407.5-k,260.0);

glVertex2f(422.5-k,260.0);

glVertex2f(422.5-k,230.0);

glVertex2f(407.5-k,230.0);

glEnd();

glColor3f(0.0,0.0,0.4);

glBegin(GL\_POLYGON);//side-panels

glVertex2f(407.5-k,250.0);

glVertex2f(400-k,250.0);

glVertex2f(400-k,240.0);

glVertex2f(407.5-k,240.0);

glVertex2f(422.5-k,250.0);

glVertex2f(397.5-k,250.0);

glVertex2f(397.5-k,240.0);

glVertex2f(422.5-k,240.0);

glEnd();

mars(100.0);

planet1(40);

earth(20);

for(j=0;j<=10000000;j++)

;

}

if(count<8)

{

glColor3f(1.0,1.0,1.0);//bonnet

glBegin(GL\_POLYGON);//front

glVertex2f(237.5,120.0+i);

glVertex2f(262.5,120.0+i);

glVertex2f(250,170.0+i);

glEnd();

}

if(count<8 && count>6)

{

glColor3f(1.0,0.55 ,0.0);

glBegin(GL\_POLYGON);//bottom\_1\_exhaust

glVertex2f(237.5,120.0+i);

glVertex2f(244.5,120.0+i);

glVertex2f(241,100.0+i);

glEnd();

glBegin(GL\_POLYGON);//bottom\_2\_exhaust

glVertex2f(246.5,120.0+i);

glVertex2f(253.5,120.0+i);

glVertex2f(249.5,100.0+i);

glEnd();

glBegin(GL\_POLYGON);//bottom\_3\_exhaust

glVertex2f(262.5,120.0+i);

glVertex2f(255.5,120.0+i);

glVertex2f(258.5,100.0+i);

glEnd();

}

if(count<=4)

{

glColor3f(1.0,0.0,0.0);

glBegin(GL\_POLYGON);//left\_side\_top

glVertex2f(237.5,120.0+i);

glVertex2f(217.5,95.0+i);

glVertex2f(237.5,95.0+i);

glEnd();

glBegin(GL\_POLYGON);//left\_side\_bottom

glVertex2f(237.5,20.0+i);

glVertex2f(217.5,20.0+i);

glVertex2f(237.5,70.0+i);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_bottom

glVertex2f(262.5,20.0+i);

glVertex2f(282.5,20.0+i);

glVertex2f(262.5,70.0+i);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_top

glVertex2f(262.5,120.0+i);

glVertex2f(262.5,95.0+i);

glVertex2f(282.5,95.0+i);

glEnd();

}

if(count>4)

{

countside+=1;

glColor3f(1.0,0.0,0.0);

glBegin(GL\_POLYGON);//left\_side\_top

glVertex2f(237.5,120.0-countside);

glVertex2f(217.5,95.0-countside);

glVertex2f(237.5,95.0-countside);

glEnd();

glBegin(GL\_POLYGON);//left\_side\_bottom

glVertex2f(237.5,20.0-countside);

glVertex2f(217.5,20.0-countside);

glVertex2f(237.5,70.0-countside);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_bottom

glVertex2f(262.5,20.0-countside);

glVertex2f(282.5,20.0-countside);

glVertex2f(262.5,70.0-countside);

glEnd();

glBegin(GL\_POLYGON);//right\_side\_top

glVertex2f(262.5,120.0-countside);

glVertex2f(262.5,95.0-countside);

glVertex2f(282.5,95.0-countside);

glEnd();

}

for(j=0;j<=1000000;j++)

;

glutSwapBuffers();

glutPostRedisplay();

glFlush();

}

}

1. **void mars(float radius)** function has the code for displaying planet mars.

void mars(float radius)

{

glBegin(GL\_POLYGON);

for (int i=0; i<=359; i++)

{

float degInRad = i\*DEG2RAD;

glColor3f(0.6,0.6,0.6);

glVertex2f(250+cos(degInRad)\*radius,270+(sin(degInRad))\*radius);

}

glEnd();

}

1. **void earth(float radius)** function has the code for displaying planet mars.

void earth(float radius)

{

glBegin(GL\_POLYGON);

for (int i=0; i<=359; i++)

{

float degInRad = i\*DEG2RAD;

glColor3f(0.196078 ,0.6 ,0.8);

glVertex2f(450+cos(degInRad)\*radius,400+(sin(degInRad))\*radius);

}

glEnd();

}

# CONCLUSION:

In this project we have demonstrated a complete Working of satellite. There is an external view, internal view of the satellite. The satellite launch is shown upon clicking key. We have shown the movement of satellite in the space. And the satellite which is revolving around the planet is also shown. We have also shown the launching countdown timer from 10 to 0.

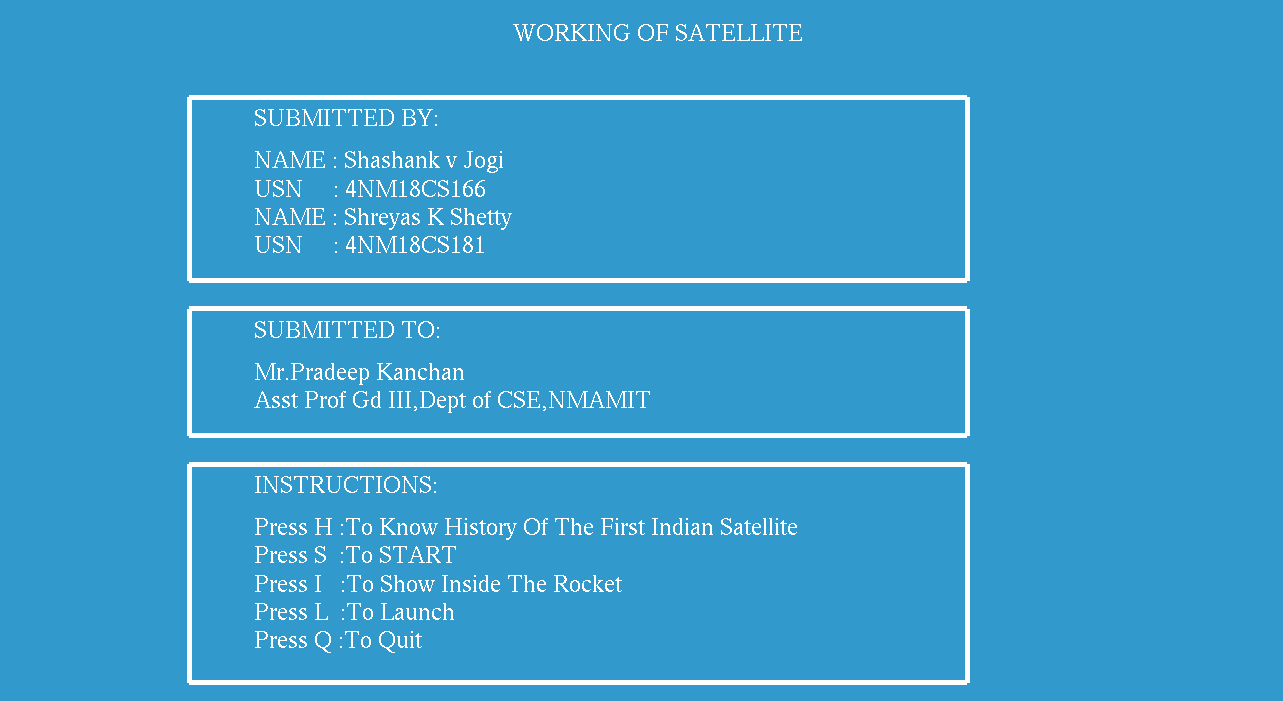
OpenGL supports enormous flexibility in design and the use of OpenGL graphics programs. The Presence of many built in classes methods take care of much functionality and reduce the job of coding as well as makes the implementation simpler. We have implemented the project making it user-friendly and error free

as possible.

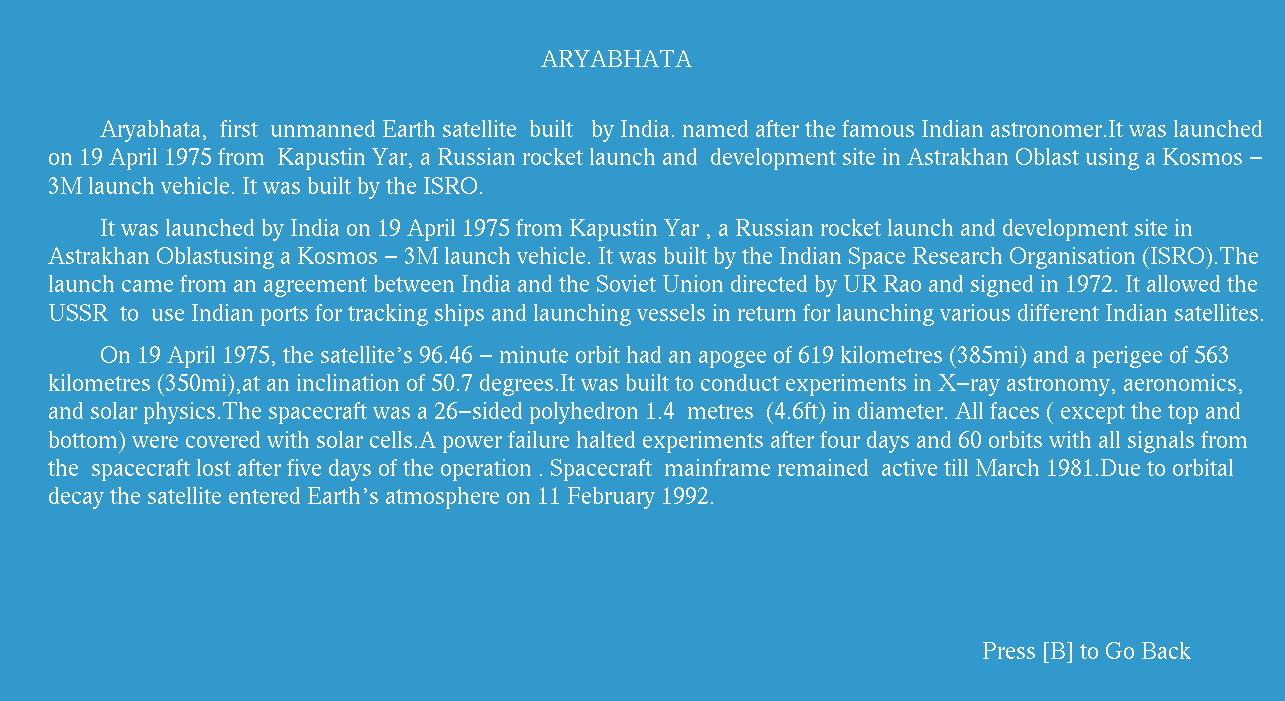
# REFERENCES:

* <https://docs.microsoft.com/en-us/windows/win32/opengl/gl-functions>
* <https://www.opengl.org/resources/libraries/glut/spec3/spec3.html>
* <https://lazyfoo.net/tutorials/OpenGL/index.php>

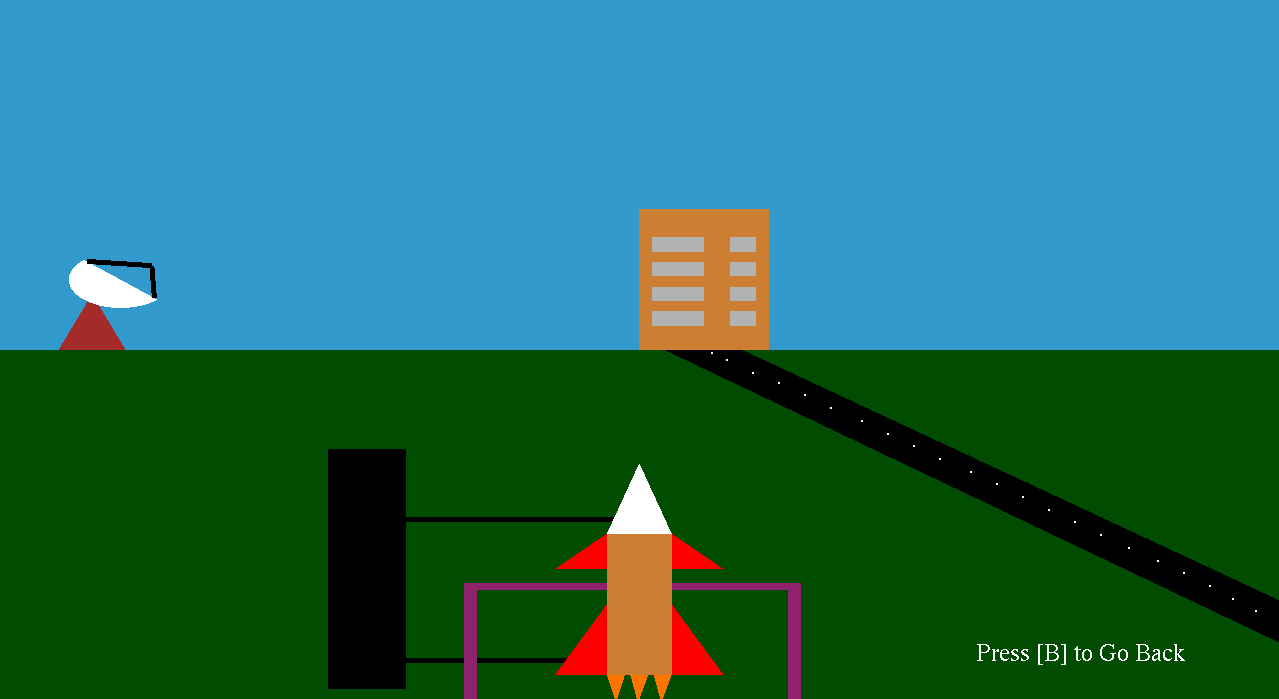
# APPENDIX:



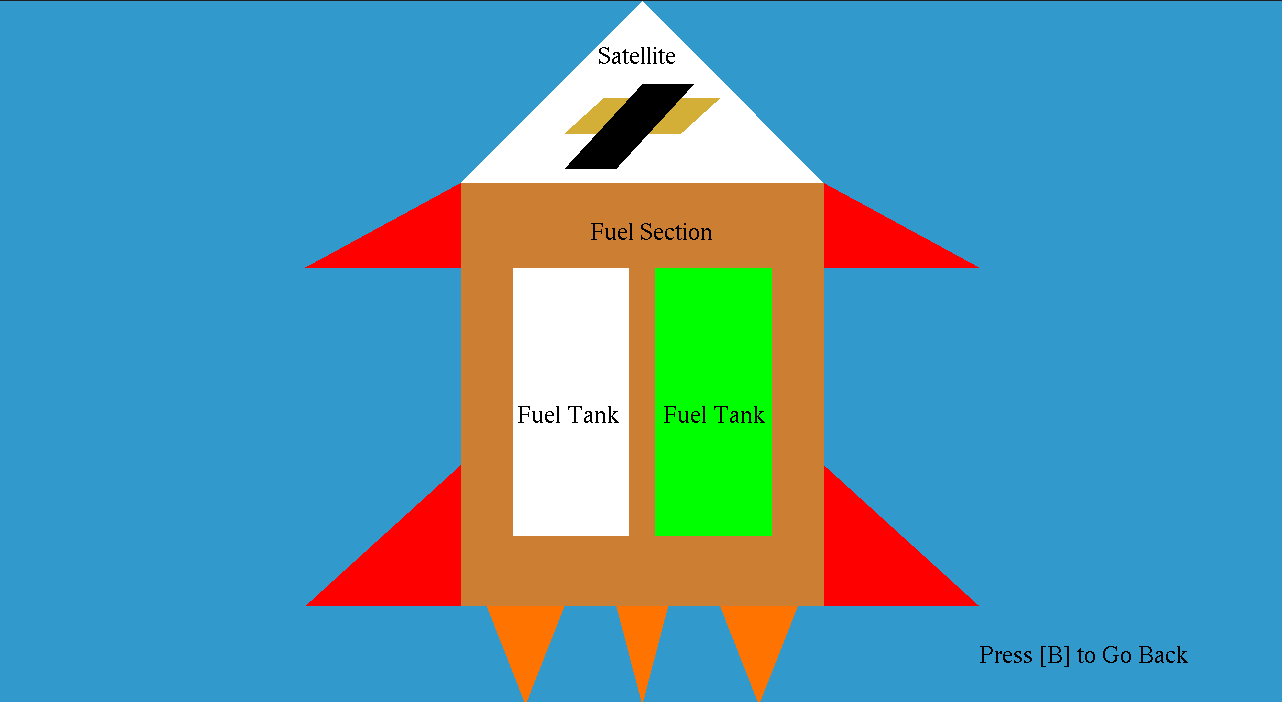
**Figure 1: Introductory page - Name USN**

****

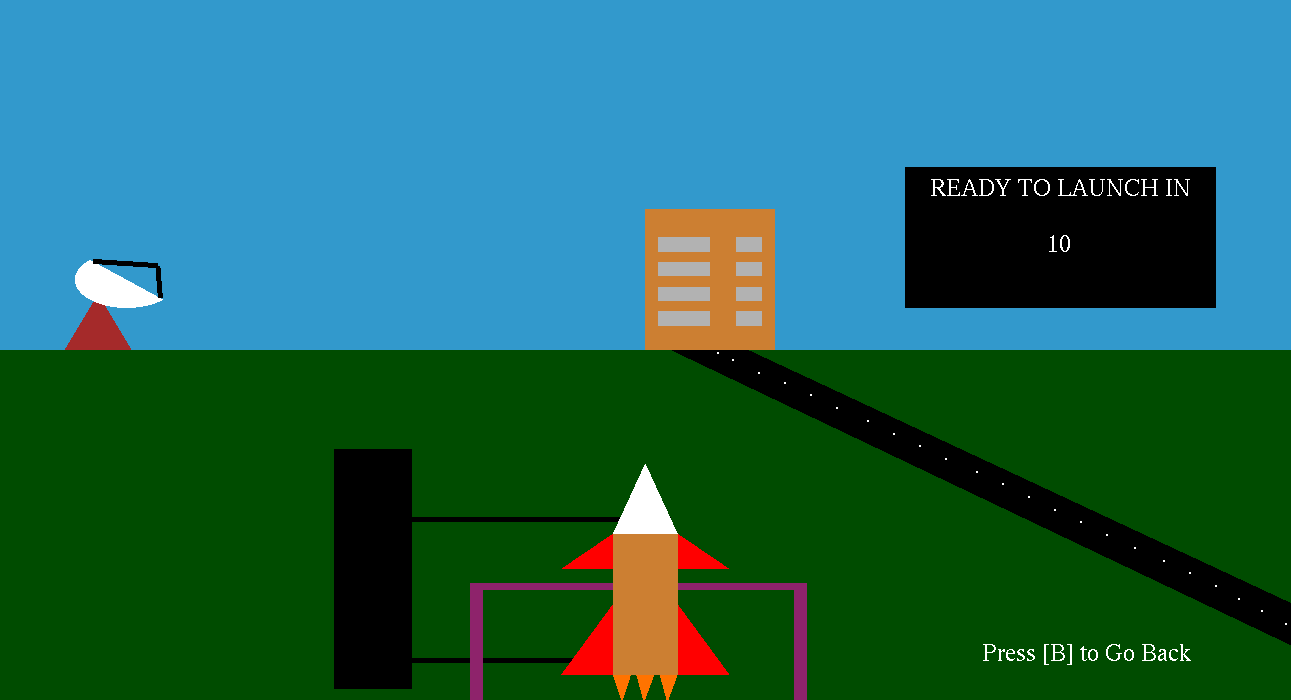
**Figure 2: History of the satellite**



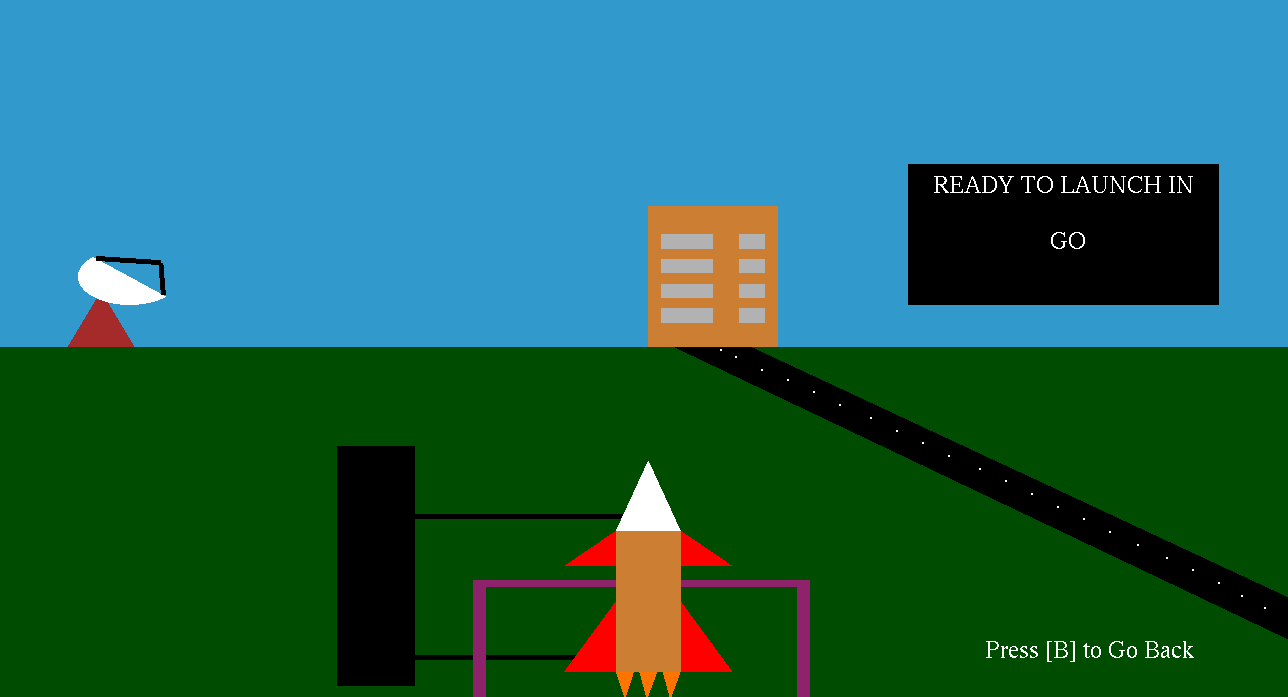
**Figure 3: Static view of the rocket**



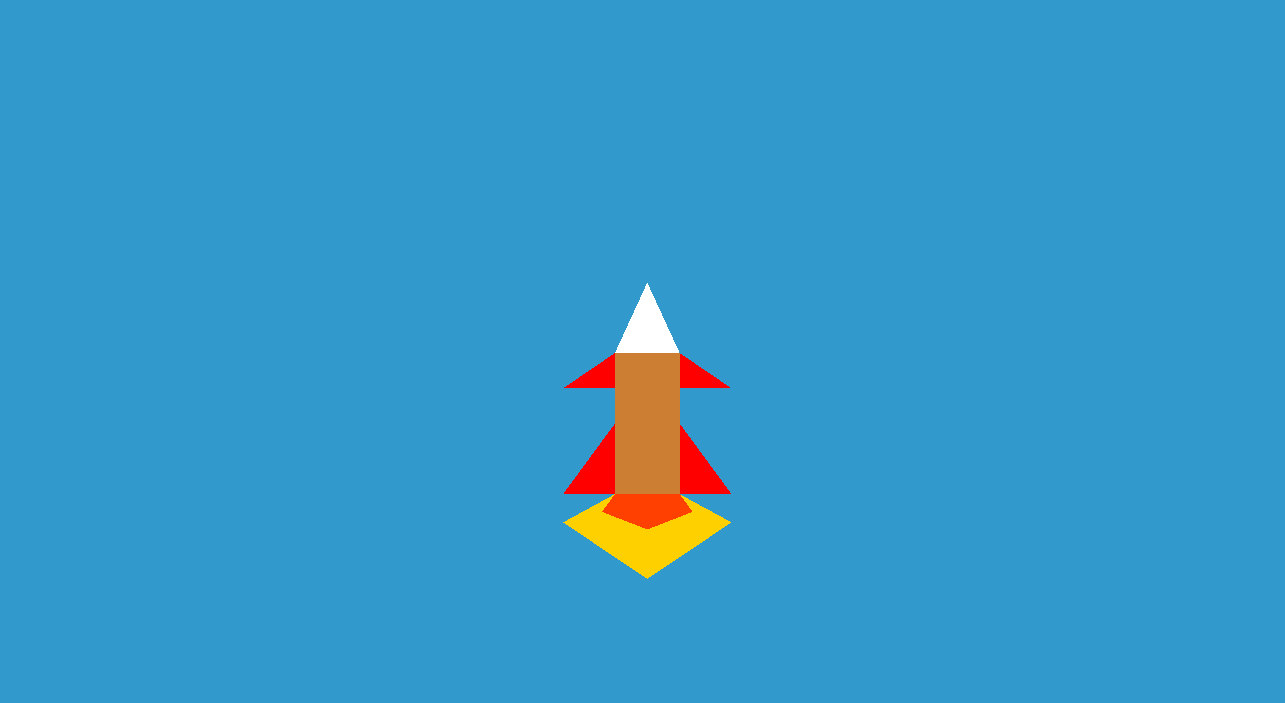
**Figure 4: Inside view of the rocket**

****

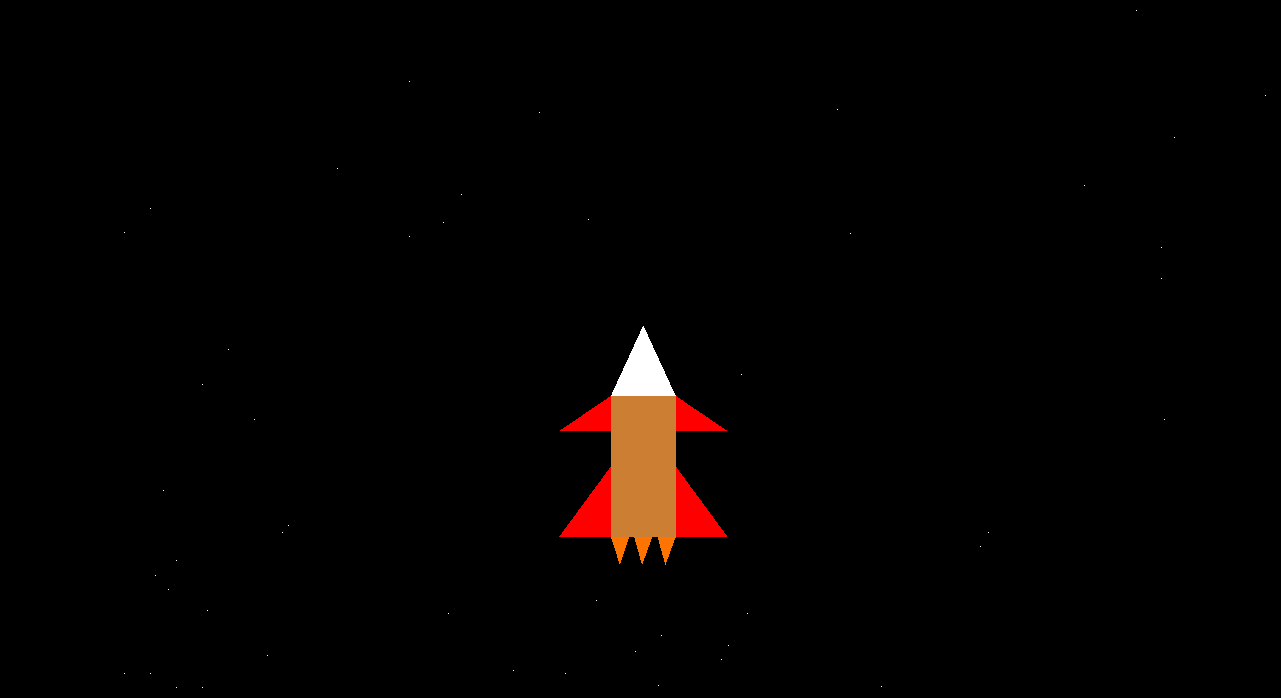
**Figure 5: Launching state of the rocket**



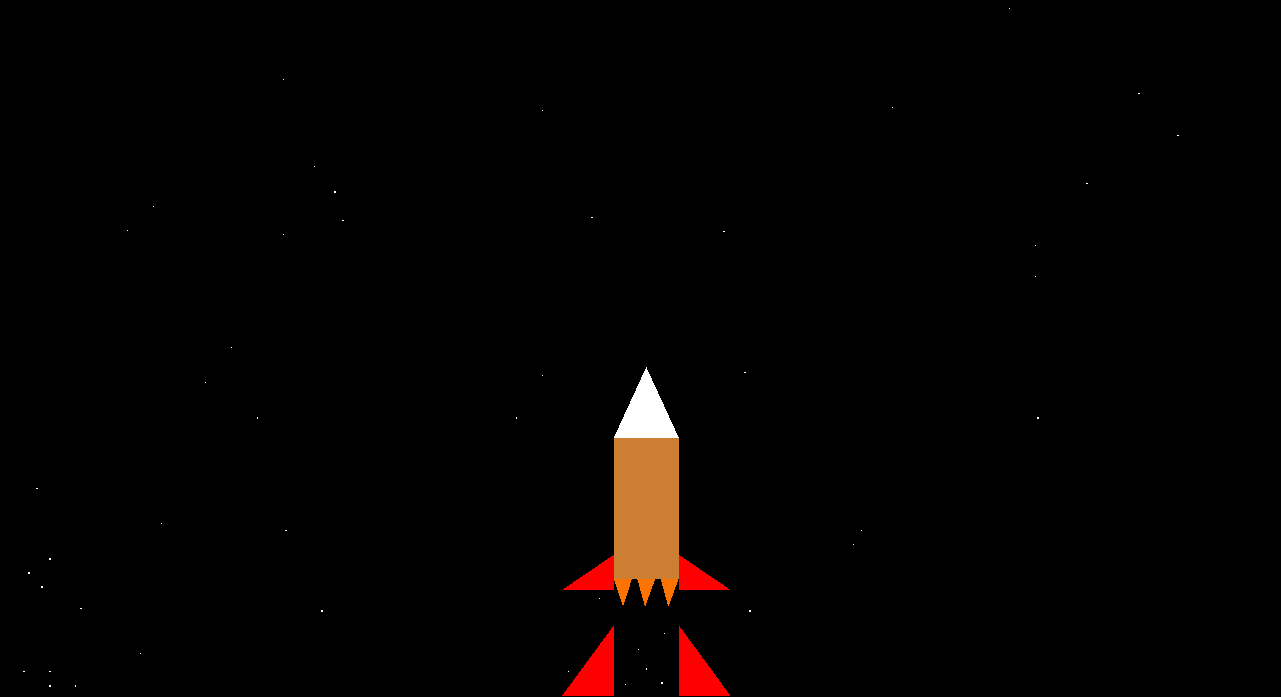
**Figure 6: Ready to launch state of the rocket**

****

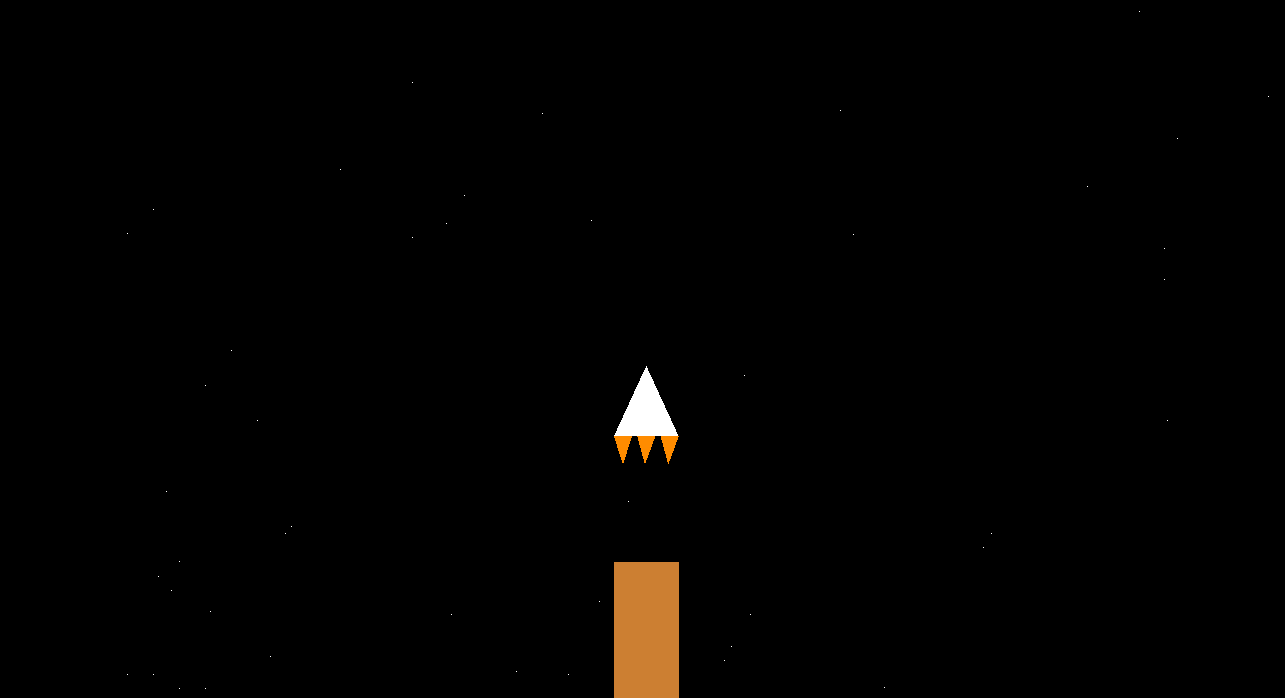
**Figure 7: Takeoff view of the rocket**



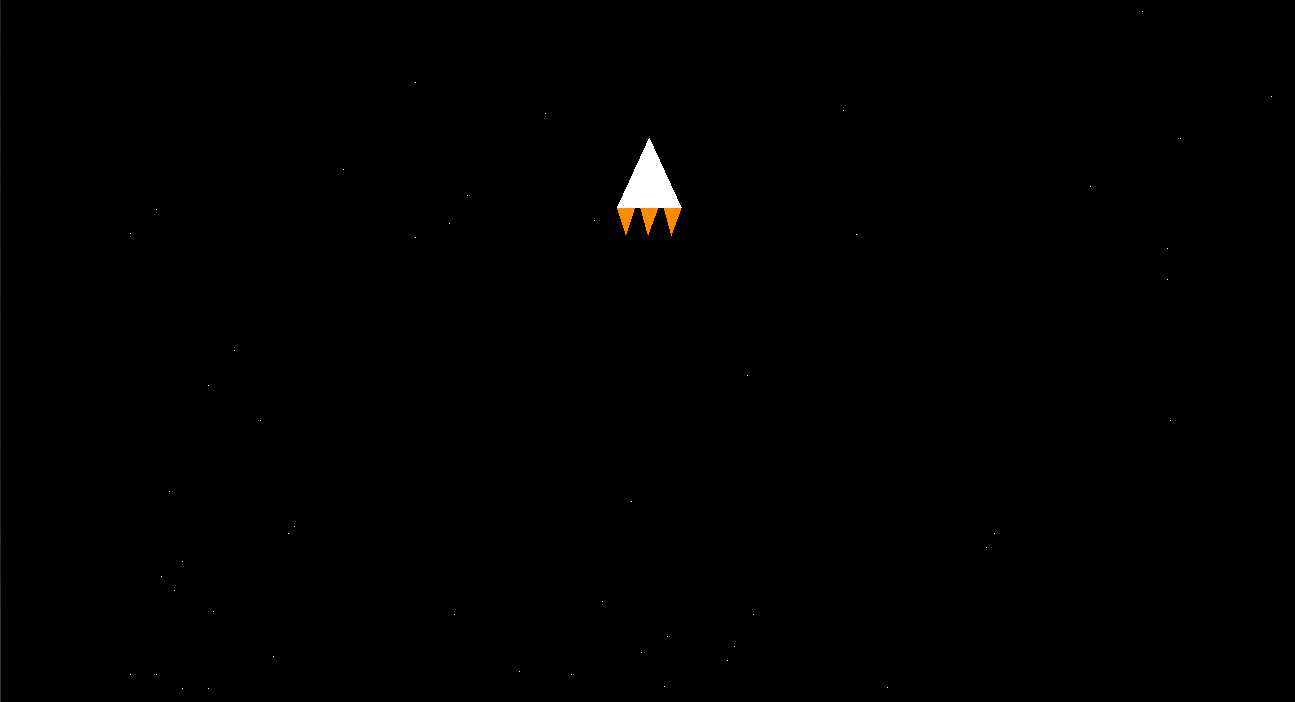
**Figure 8: Rocket travelling in space**

****

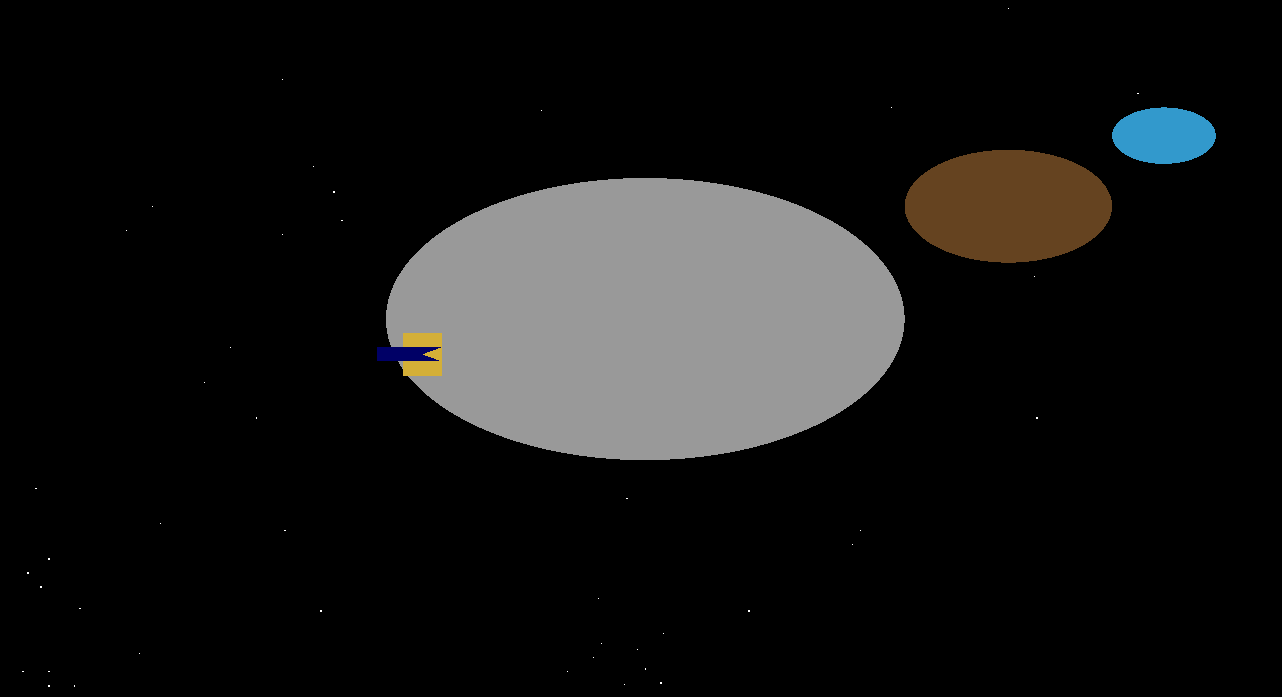
**Figure 9: Ejecting first phase engine**

****

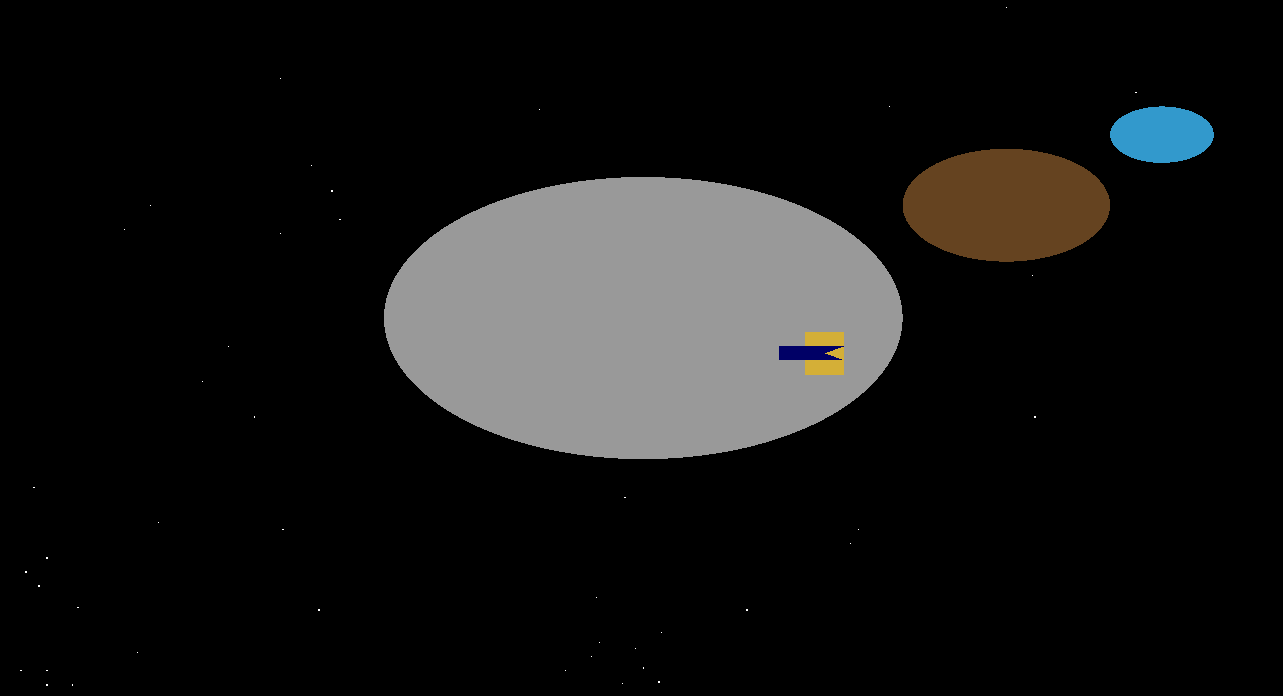
**Figure 10: Ejecting second phase engine**

****

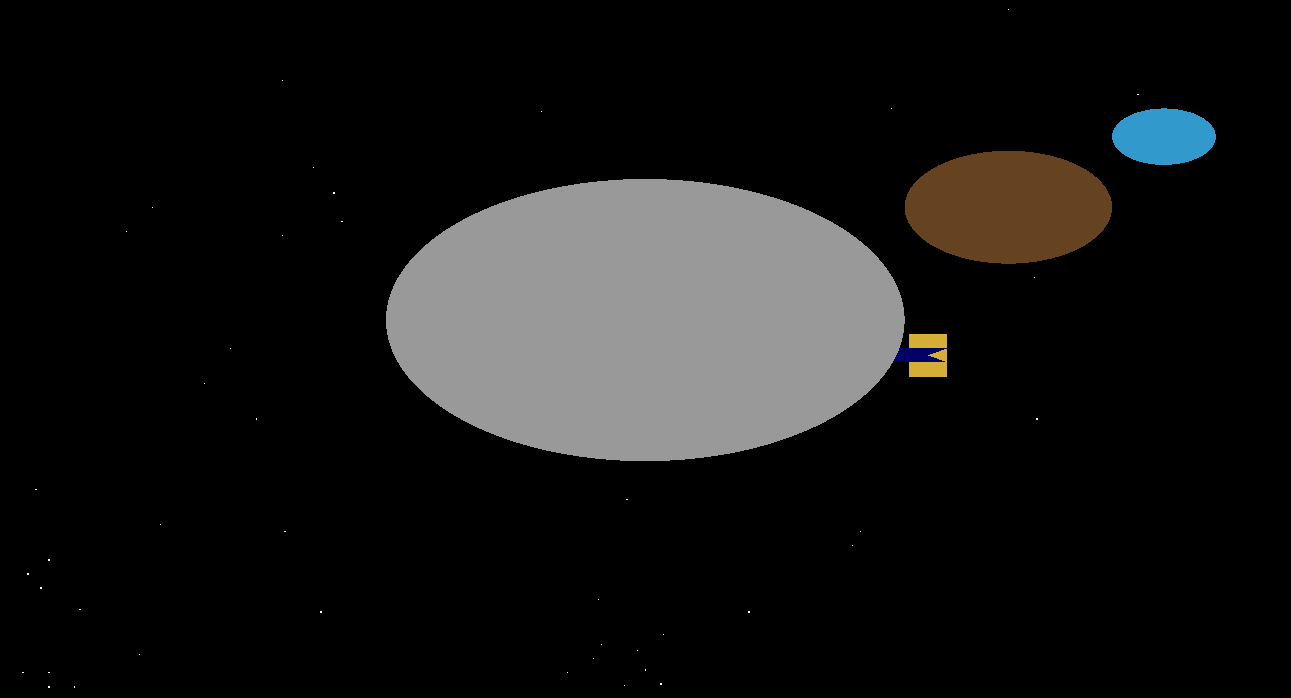
**Figure 11: Satellite carrying vehicle in travelling in space**

****

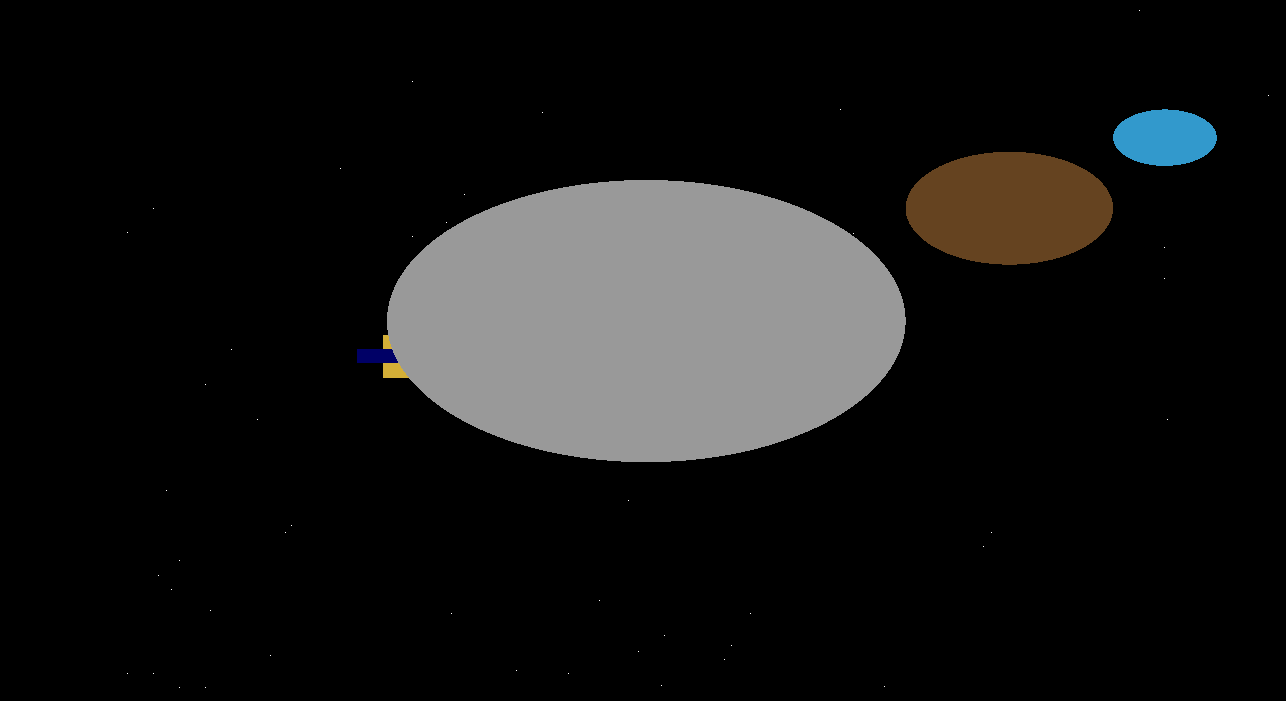
**Figure 12: Satellite revolving around mars**

****

**Figure 13: Satellite revolving around mars**

****

**Figure 14: Satellite revolving around mars and coming in back**

****

**Figure 13: Satellite revolving around mars and coming in back**