VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI

Mini Project Report on

“Rocket Launch and Deploy”

Submitted in the partial fulfillment for the requirements of Computer Graphics & Visualization Laboratory of 6th semester CSE requirement in the form of the Mini Project work

Submitted By

Nakul S	**USN: 1BY19CS090**
Mananya V Dhagge	**USN: 1BY19CS075**
Prajwal M	**USN: 1BY19CS106**

Under the guidance of

Prof. Muneshwara M S
Assistant Professor
Dept. of CSE

Prof. Chethana C
Assistant Professor
Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

YELAHANKA, BENGALURU - 560064.

2021-2022

BMS INSTITUTE OF TECHNOLOGY &MANAGEMENT
YELAHANKA, BENGALURU – 560064

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled **"Rocket Launch and Deploy"** is a bonafide work carried out by Nakul S (1BY19CS090), Mananya V Dhagge (1BY19CS075), Prajwal M (1BY19CS106) in partial fulfillment for *Mini Project* during the year 2021-2022. It is hereby certified that this project covers the concepts of *Computer Graphics & Visualization*. It is also certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report.

Signature of the Guide
Prof. Muneshwara M S/
Prof. Chethana C
Assistant Professor
CSE, BMSIT&M

Signature of the HOD
Dr. Thippeswamy G
Professor & HOD
CSE, BMSIT&M

Name and Signature of the Examiner

Internal Examiner

External Examiner

INSTITUTE VISION

To emerge as one of the finest technical institutions of higher learning, to develop engineering professionals who are technically competent, ethical and environment friendly for betterment of the society.

INSTITUTE MISSION

Accomplish stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.

DEPARTMENT VISION

To develop technical professionals acquainted with recent trends and technologies of computer science to serve as valuable resource for the nation/society.

DEPARTMENT MISSION

Facilitating and exposing the students to various learning opportunities through dedicated academic teaching, guidance and monitoring.

PROGRAM EDUCATIONAL OBJECTIVES

1. Lead a successful career by designing, analyzing and solving various problems in the field of Computer Science & Engineering.
2. Pursue higher studies for enduring edification.
3. Exhibit professional and team building attitude along with effective communication.
4. Identify and provide solutions for sustainable environmental development.

PROGRAM SPECIFIC OUTCOMES

1. Analyze the problem and identify computing requirements appropriate to its solution.
2. Apply design and development principles in the construction of software systems of varying complexity.

ACKNOWLEDGEMENT

We are happy to present this project after completing it successfully. This project would not have been possible without the guidance, assistance and suggestions of many individuals.

We would like to express our deep sense of gratitude and indebtedness to each and every one who has helped us make this project a success.

We heartily thank our Principal, **Dr. MOHAN BABU G N,** BMS Institute of Technology &Management, for his constant encouragement and inspiration in taking up this project.

We heartily thank our Professor and Head of the Department, **Dr. Thippeswamy G**, Department of Computer Science and Engineering, BMS Institute of Technology and Management, for his constant encouragement and inspiration in taking up this project.

We gracefully thank our Project Guide, **Prof. Muneshwara M S and Prof. Chethana C**, Assistant Professors, Department of Computer Science and Engineering for their intangible support and for being constant backbone for our project.

Special thanks to all the staff members of Computer Science Department for their help and kind co-operation.

Lastly, we thank our parents and friends for the support and encouragement given throughout in completing this precious work successfully.

Nakul S (1BY19CS090)

Mananya V Dhagge (1BY19CS075)

Prajwal M (1BY19CS106)

ABSTRACT

Main aim of this Mini Project is to show the simulation of a rocket launch.

illustrate the concepts and usage of pre-built functions in OpenGL.

When the rocket has completely exhausted is fuel from adjoining tanks, then the extra weight is lost by disconnecting them.

When the final capsule reached the orbiting location, the last part is also disconnected and the satellite is placed in orbit.

We have used input device keyboard to interact with the program

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Brief introduction to the project including the idea behind it is explained in this chapter.

1.1 Background

i) Interactive graphics is the most important means of producing pictures since the invention of photography and television.
ii) We can make pictures of not only the real-world objects but also of abstract objects such as mathematical surfaces on 4D and of data that have no inherent geometry.
iii) A computer graphics system is a computer system with all the components of the general-purpose computer system. There are five major elements in system: input devices, processor, memory, frame buffer, output devices.

1.2 Motivation

i) Computers have become a powerful medium for the rapid and economical production of pictures.
ii) There is virtually no area in which graphical displays cannot be used to some advantage.
iii) Graphics provide a so natural means of communicating with the computer that they have become widespread.

1.3 Problem Statement

System will provide smart way to organize the events. Users will easily get all the products related to their event as per budget. This application will be used over worldwide. Easiest way to manage major college events.

1.4 Proposed System

To achieve three dimensional effects, open GL software is proposed. It is software which provides a graphical interface. It is a interface between application program and graphics hardware. the advantages are:

i) Open GL is designed as a streamlined.
ii) It's a hardware independent interface i.e. it can be implemented on many different hardware platforms.
iii) With OpenGL, we can draw a small set of geometric primitives such as points, lines and polygons etc.
iv) It provides double buffering which is vital in providing transformations.
v) It is event driven software.
vi) It provides call back function.

1.5 Limitations

Existing system for a graphics is the TC++. This system will support only the 2D graphics. 2D graphics package being designed should be easy to use and understand. It should provide various option such as free hand drawing, line drawing, polygon drawing, filled polygons, flood fill, translation, rotation, scaling, clipping etc. Even though these properties were supported, it was difficult to render 2D graphics cannot be. Very difficult to get a 3-Dimensional object. Even the effects like lighting, shading cannot be provided. So, we go for Microsoft Visual Studio software.

1.6 Summary

This chapter summarizes the introduction of our whole project. It includes background, literature survey, motivation, problem statement, objectives, scope and challenges of the project. Thus we have concluded the basic process of rocket launch. The proposed product is easy to use, low-cost and does not need any special training. Our scheme would help to save the time of people and would help the people do deploy satellites efficiently. Few of the advantages of the system make it more robust.

CHAPTER 2

LITERATURE SURVEY

As a software interface for graphics hardware, OpenGL's main purpose is to render two- and three-dimensional objects into a frame buffer.

These objects are described as sequences of vertices or pixels.

OpenGL performs several processing steps on this data to convert it to pixels to form the final desired image in the frame buffer.

OpenGL Fundamentals

This section explains some of the concepts inherent in OpenGL.

Primitives and Commands

OpenGL draws primitives—points, line segments, or polygons—subject to several selectable modes.

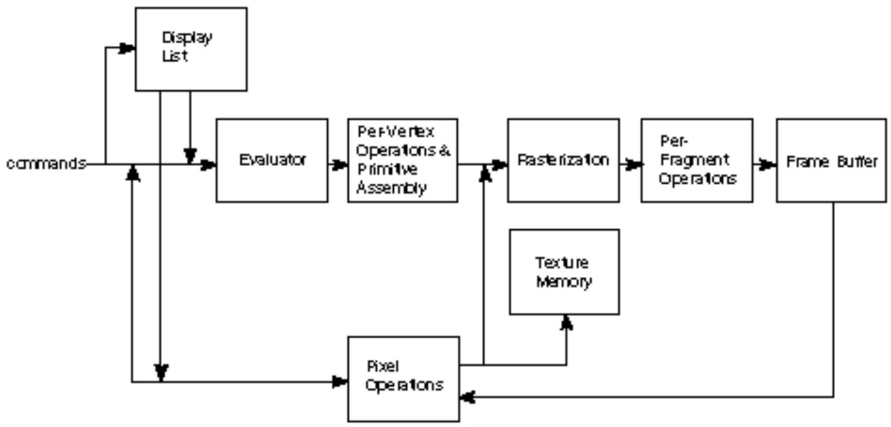
You can control modes independently of each other; that is, setting one mode doesn't affect whether other modes are set. Primitives are specified, modes are set, and other OpenGL operations are described by issuing commands in the form of function calls.

Primitives are defined by a group of one or more vertices. A vertex defines a point, an endpoint of a line, or a corner of a polygon where two edges meet. Data is associated with a vertex, and each vertex and its associated data are processed independently, in order, and in the same way. The type of clipping depends on which primitive the group of vertices represents.

Commands are always processed in the order in which they are received, although there may be an indeterminate delay before a command takes effect.

This means that each primitive is drawn completely before any subsequent command takes effect. It also means that state-querying commands return data that's consistent with complete execution of all previously issued OpenGL commands.

The figure shown below gives an abstract, high-level block diagram of how OpenGL processes data. In the diagram, commands enter from the left and proceed through what can be thought of as a processing pipeline. Some commands specify geometric objects to be drawn, and others control how the objects are handled during the various processing stages.



As shown by the first block in the diagram, rather than having all commands proceed immediately through the pipeline, you can choose to accumulate some of them in a display list for processing at a later time.

Rasterization produces a series of frame buffer addresses and associated values using a two-dimensional description of a point, line segment, or polygon.

Each fragment so produced is fed into the last stage, per-fragment operations, which performs the final operations on the data before it's stored as pixels in the frame buffer. These operations include conditional updates to the frame buffer based on incoming and previously stored z-value s (for z-buffering) and blending of incoming pixel colours with stored colors, as well as masking and other logical operations on pixel values.

All elements of OpenGL state, including the contents of the texture memory and
even of the frame buffer, can be obtained by an OpenGL application.

CHAPTER 3

SYSTEM REQUIREMENT

- **Software Requirements :**
  - Microsoft Visual C++
  - Opengl

- **Hardware Requirement :**
  - Graphics System,
  - Pentium P4 With 256 Of RAM(Min)

CHAPTER 4

SYSTEM ANALYSIS

There are three major launch types for spacecraft: vertical take-off, horizontal take-off, and air launch. All three are discussed below, with a strong emphasis on the air launch method.

A. Vertical Take-off

Vertical Take-off is the traditional and by far the most popular launch mode. Vertical takeoff rockets typically are able to achieve necessary propellant mass fractions with enough margin to permit the use of relatively low technology pressure fed rocket motors. Precise longitudinal control of the CG location is not necessary as it is with the other two launch modes, which makes propellant storage and routing significantly simpler. Additionally, vertically-launched vehicles are exposed to less bending and no twisting moments during launch, unlike vehicles that are launched horizontally or in the air.1 Launch site location is extremely important for vertical launches. Vertically-launched spacecraft can only be directly launched into an orbit with an inclination at least as large as the latitude of the launch site. For example, a vertical launch at Kennedy Space Center in Florida cannot put a satellite into an equatorial orbit because the launch site is at a 28.5-degree latitude, as opposed to the 0.0-degree latitude at the equator. In order to transfer into an orbit with a different inclination, an expensive plane change maneuver must be performed. Additionally, to avoid dropping vehicle parts on populated areas, vertical launches are performed over large bodies of water or deserts. Since spacecraft are almost always launched eastward (to gain a Δv assist from the earth's rotation), these limits launch site locations to East coasts (or deserts) at the lowest possible latitude.

B. Horizontal Take-off

Horizontal take-off vehicles launch from a runway, similar to an aircraft. This eliminates the need for an expensive launch pad. It also allows the vehicle to fly to any latitude before entering space, which eliminates the need for costly inclination change maneuvers. Horizontal launch vehicles may be powered by rocket engines only or by a combination of jet engines and rocket engines.1 5 While horizontal take-off vehicles are similar to aircraft, they need larger empty mass fractions and lower propellant mass fractions than regular aircraft. The vehicles must be sturdy enough to endure intense bending modes, turbulent air, high aerodynamic pressures, and supersonic flight. The propellant mass fractions of horizontally-launched vehicles are less favorable than those of vertical launches and air launches.

C. Air Launch

Air launches typically involve two vehicles: a winged carrier aircraft and an upper stage spacecraft. There are numerous advantages of air launch systems over the other two launch modes. While vertical launches are constrained to a launch azimuth of the launch site's latitude or higher, air launch systems (like horizontal launch systems) can be flown to any latitude before launch. This opens up any orbital inclination to the spacecraft, including equatorial orbits. Since plane changes from American launch sites to an equatorial orbit are very expensive, the capability to launch into any azimuth is a significant advantage of air launch systems.

While the advantages of air launches are plentiful, there are also several disadvantages. Modifying a carrier aircraft to be capable of supporting an upper stage vehicle can be complicated and expensive. Additionally, using a carrier aircraft limits the size of the spacecraft, meaning that smaller payloads are available. However, these payloads are often easier to integrate into the vehicle than those of vertical launches. Depending on the mode of air launch, separation of the spacecraft from the carrier can be dangerous, especially in cases of strong aerodynamic flow.7,8 Propellant boil off is a concern for upper stages with cryogenic propellants (except in the internally carried configuration, as discussed below), since the spacecraft is exposed to radiation heat from the sun and convective heat from the air stream. For example, the X-15 lost 60-80% of its liquid oxygen during its one-hour climb, and the attached B-52 carrier aircraft had to refuel the X-15 before it was released.

CHAPTER 5

SYSTEM IMPLEMENTATION

This program is implemented using various openGL functions which are shown below.

Various functions used in this program.

 glutInit() : interaction between the windowing system and OPENGL is initiated

 glutInitDisplayMode() : used when double buffering is required and depth information is required

 glutCreateWindow() : this opens the OPENGL window and displays the title at top of the window

 glutInitWindowSize() : specifies the size of the window

 glutInitWindowPosition() : specifies the position of the window in screen co-ordinates

 glutKeyboardFunc() : handles normal ascii symbols

 glutSpecialFunc() : handles special keyboard keys

 glutReshapeFunc() : sets up the callback function for reshaping the window

 glutIdleFunc() : this handles the processing of the background

 glutDisplayFunc() : this handles redrawing of the window

 glutMainLoop() : this starts the main loop, it never returns

 glViewport() : used to set up the viewport

 glVertex3fv() : used to set up the points or vertices in three dimensions

 glColor3fv() : used to render color to faces

 glFlush() : used to flush the pipeline

 glutPostRedisplay() : used to trigger an automatic redrawal of the object

 glMatrixMode() : used to set up the required mode of the matrix

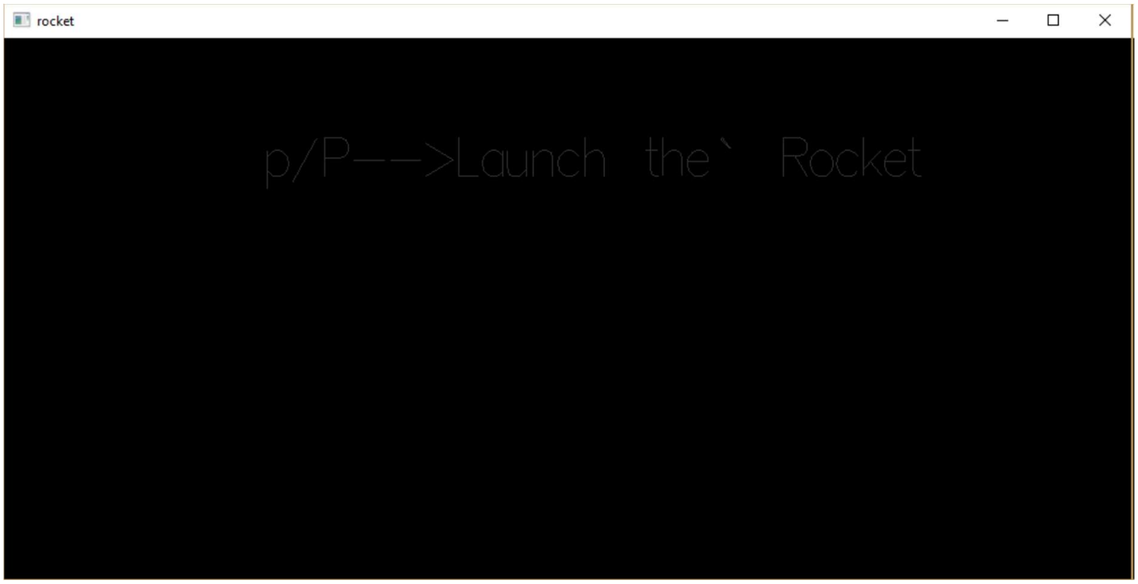
 glLoadIdentity() : used to load or initialize to the identity matrix

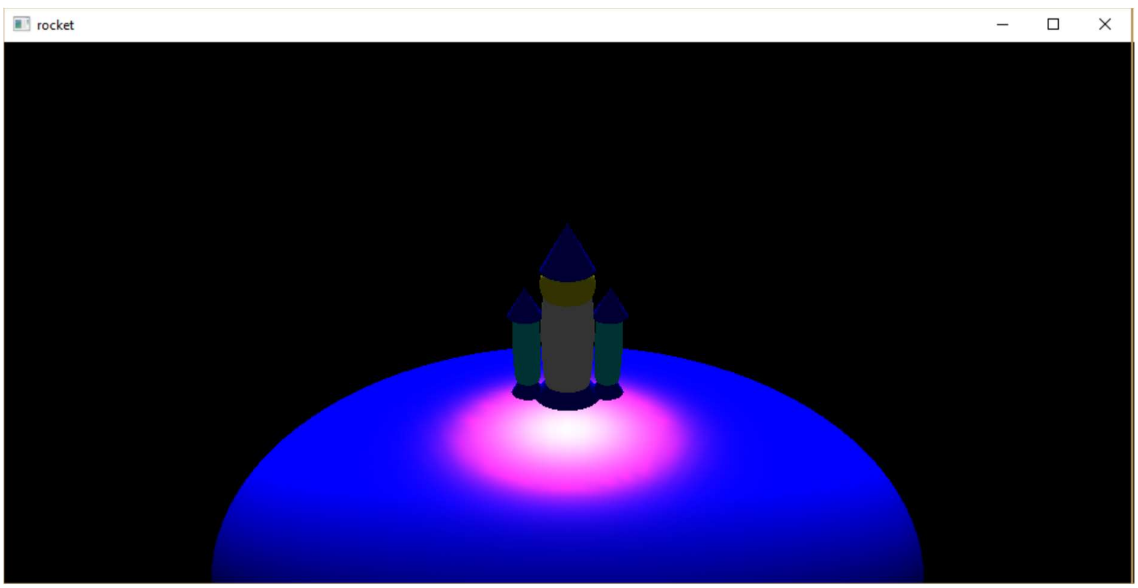
 glTranslatef() : used to translate or move the rotation centre from one point to another in three dimensions

 glRotatef() : used to rotate an object through a specified rotation angle

CHAPTER 6

INTERPRETATION OF RESULTS





CHAPTER 7

CONCLUSION

The project “Rocket Launch” clearly demonstrates the Simulation of a Rocket launch and putting the satellite in orbit.

Finally, we conclude that this program clearly illustrates the Rocket launch by using OpenGL and pre-built objects and has been completed successfully and is ready to be demonstrated.

FUTURE ENHANCEMENTS

- The rocket can be enhanced by providing animation for working of rocket in the space.
- The rocket can be enhanced from 2D model to 3D model.
- We can use this project for developing a real time rocket.
- We can add texture effect by using the instructions available in the OpenGL.
- Changing the environment around it to make more interactive.
- The unmounting of the satellite in space from the rocket can be shown.
- The rotation of the satellite in space on its orbit can be tracked.

BIBLIOGRAPHY

We have obtained information from many resources to design and Implement our project successively. We have acquired most of the Knowledge from related websites. The following are some of the Resources:

TEXT BOOKS:

Interactive computer graphics a top-down approach

-by Edward angel.

COMPUTER GRAPHICS, PRINCIPLES & PRACTICES

- Foley van dam

- Feiner hughes

WEB REFERENCES:

http://jerome.jouvie.free.fr/OpenGl/Lessons/Lesson3.php

http://google.com

http://opengl.org