Lab Report 1

Name: Arnob Das Shacha

ID: 221-15-5277

Section: 61_D1

Experiment Name: Draw a rocket using fundamental knowledge of OpenGL and some basic built in functions.

Submitted To:

Syada Tasmia Alvi

Senior Lecturer

Department of Computer Science and Engineering

Daffodil International University

Title: Draw a rocket using fundamental knowledge of OpenGL and some basic built in functions.

Introduction:

OpenGL is a powerful cross-platform graphics library that allows developers to create 2D and 3D graphics using simple geometric primitives. It is widely used in graphics applications, game design, and computer simulations.

In this experiment, we have used the basic OpenGL functions and concepts to draw a simple rocket using polygons. The main objective of this program is to understand how to represent complex objects by combining simple shapes like triangles, rectangles, and polygons. The rocket is designed using basic 2D coordinates and color functions to represent different parts such as the nose (triangle), body (rectangle), window (small square), and fins (wings). This helps beginners learn how to use OpenGL's coordinate system, color settings, and primitive drawing commands effectively.

Contents:

Tools and Libraries Used:

Header Files: <GL/glut.h> — includes necessary OpenGL and GLUT functions

Functions Used:

glBegin() / glEnd() → to start and end drawing shapes

glVertex2f() → to specify coordinates

glColor3f() → to set colors

glFlush() → to execute all OpenGL commands

glClear() → to clear the window before drawing

glOrtho() → to define the 2D viewing area

Rocket Structure and Logic:

Triangle (Top Cone): Represents the rocket's nose. Drawn using GL_POLYGON with 3 vertices.

Big Rectangle: Represents the main body of the rocket.

Small Square: Represents a window or a marking in the center.

Left and Right Wings: Drawn using small triangles on each side to represent fins for stability.
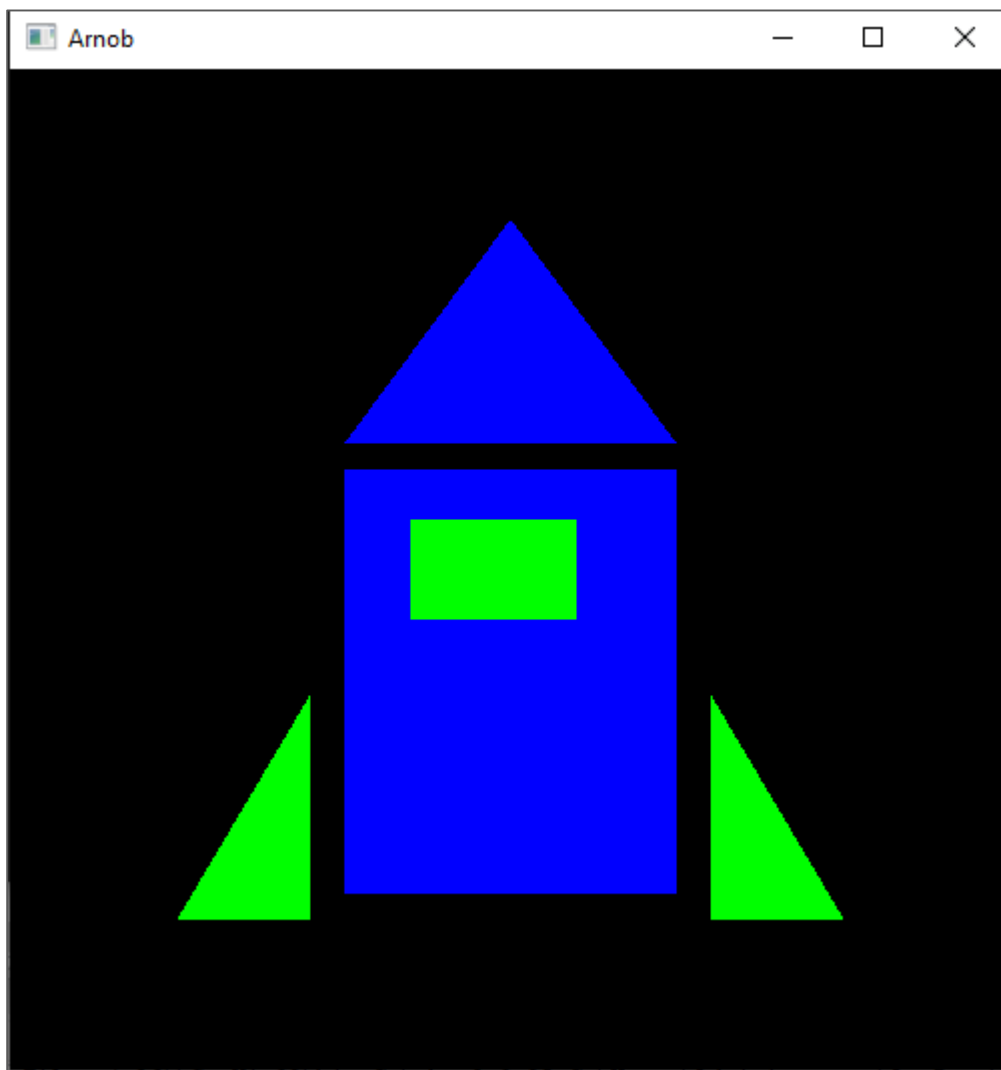
Color Scheme:

Blue for the main body and nose.Green for the wings and window area.Black background for contrast.
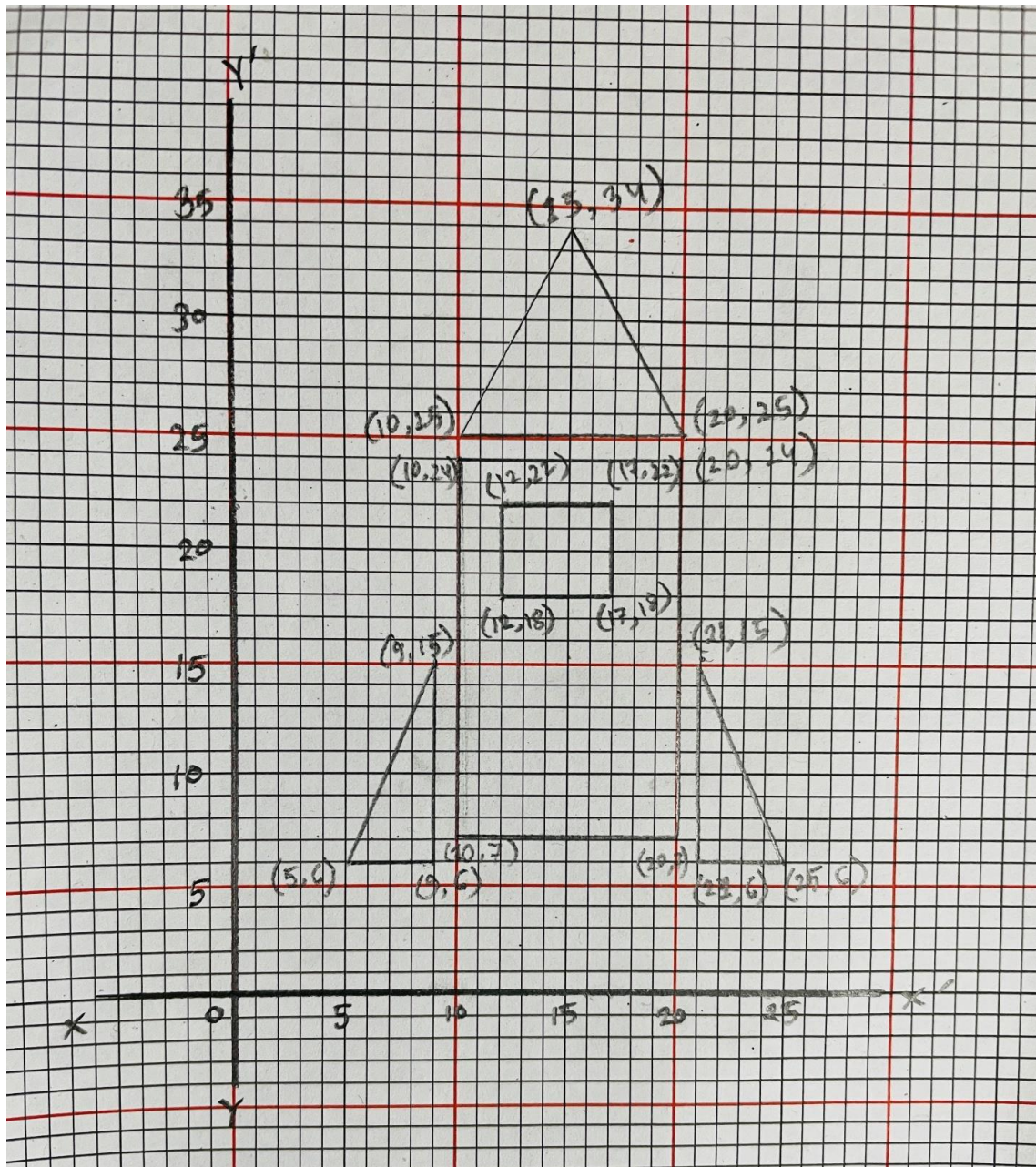
Coordinate System:

The rocket is drawn using a 2D orthographic projection.

The coordinates are within a visible range (glOrtho(0, 30, 0, 40)), ensuring the full rocket appears centered in the window.

Output:

Graph:

Code:

```
void display(void)
{
```

```
glClear(GL_COLOR_BUFFER_BIT);


// Triangle
glColor3f(0.0, 0.0, 1.0); // Blue
glBegin(GL_POLYGON);
    glVertex2f(15, 34);
    glVertex2f(10, 25);
    glVertex2f(20, 25);
glEnd();


// Big square
glColor3f(0.0, 0.0, 1.0); // Blue
glBegin(GL_POLYGON);
    glVertex2f(10, 24);
    glVertex2f(10, 7);
    glVertex2f(20, 7);
    glVertex2f(20, 24);
glEnd();

// small square
glColor3f(0.0, 1.0, 0.0); // Green
glBegin(GL_POLYGON);
    glVertex2f(12, 22);
    glVertex2f(12, 18);
    glVertex2f(17, 18);
    glVertex2f(17, 22);
```

```
    glEnd();


    // Left wing
    glColor3f(0.0, 1.0, 0.0); // Green
    glBegin(GL_POLYGON);
        glVertex2f(9, 15);
        glVertex2f(5, 6);
        glVertex2f(9, 6);
    glEnd();


    // Right wing
    glColor3f(0.0, 1.0, 0.0); // Green
    glBegin(GL_POLYGON);
        glVertex2f(21, 15);
        glVertex2f(21, 6);
        glVertex2f(25, 6);
    glEnd();


    glFlush();
}
```

Discussion:

In this experiment, a simple two-dimensional rocket was drawn using the fundamental drawing functions of OpenGL. By combining a few basic geometric shapes such as triangles, rectangles, and polygons, the complete structure of a rocket was formed. Each part of the rocket the nose, body, window, and fins was represented using simple coordinates within a defined 2D orthographic projection. Through this, we learned how OpenGL uses the coordinate system to position different shapes accurately within a display window. Overall, this practical exercise helped develop a basic understanding of how simple OpenGL commands can be used to create meaningful 2D graphics.