Title

Draw a rocket using fundamental knowledge of OpenGl and some basic built in functions.

Introduction

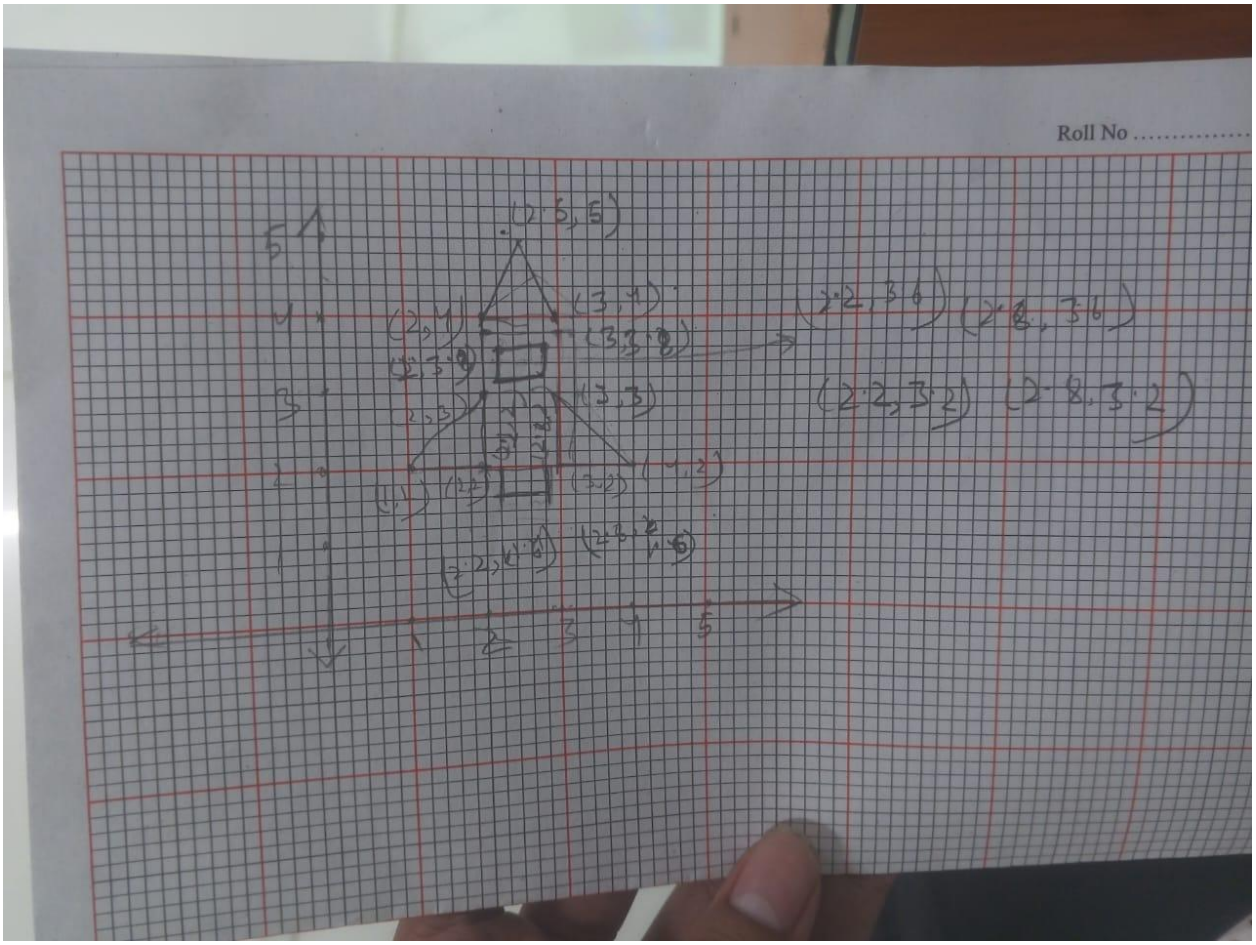
In this lab task, I used OpenGL with GLUT to draw a simple rocket shape using basic geometric primitives. The program demonstrates how to create 2D graphics by combining triangles and polygons with different colors. Through this task, we learned how to initialize an OpenGL window, set background and object colors, and use functions like glBegin(), glVertex3f(), and glFlush() to display objects on the screen. This experiment helps in understanding the fundamentals of computer graphics and how shapes are formed using coordinates in OpenGL.

Contents

In this lab task :

1. Functions used
    - ‘glClear()’ – clears the screen.
    - ‘glColor3f()’ – sets color.
    - ‘glBegin()’/ ‘glEnd()’ – start and end shape drawing.
    - ‘glVertex3f()’ – defines shape corners.
    - ‘glFlush()’ – displays the drawing.
2. Shapes used:
    - Triangles – rocket top and two wings.
    - Polygons (rectangles)– rocket body, window, and blast.

Graph



Code

```
#include <GL/gl.h>
#include <GL/glut.h>
void display(void)
{
/* clear all pixels */
glClear (GL_COLOR_BUFFER_BIT);
/* draw white polygon (rectangle) with corners at
* (0.25, 0.25, 0.0) and (0.75, 0.75, 0.0)
```

```
*/
//upper triangle
    glColor3f (0, 0, 1.0);
    glBegin(GL_TRIANGLES);
    glVertex3f (.25, .5, 0);
    glVertex3f (.2, .4, 0);
    glVertex3f (.3, .4, 0);
    //glVertex3f (0.25, 0.75, 0.0);
    glEnd();

    // main body

    glBegin(GL_POLYGON);
    glVertex3f (.2, .39, 0);
    glVertex3f (.3, .39, 0);
    glVertex3f (.3, .2, 0);
    glVertex3f (.2, .2, 0);
    glEnd();

    // wing 1
     glColor3f(0.0, 1.0, 0.0);

    glBegin(GL_TRIANGLES);
    glVertex3f (.2, .3, 0);
    glVertex3f (.2, .2, 0);
    glVertex3f (.1, .2, 0);
    //glVertex3f (0.25, 0.75, 0.0);
```

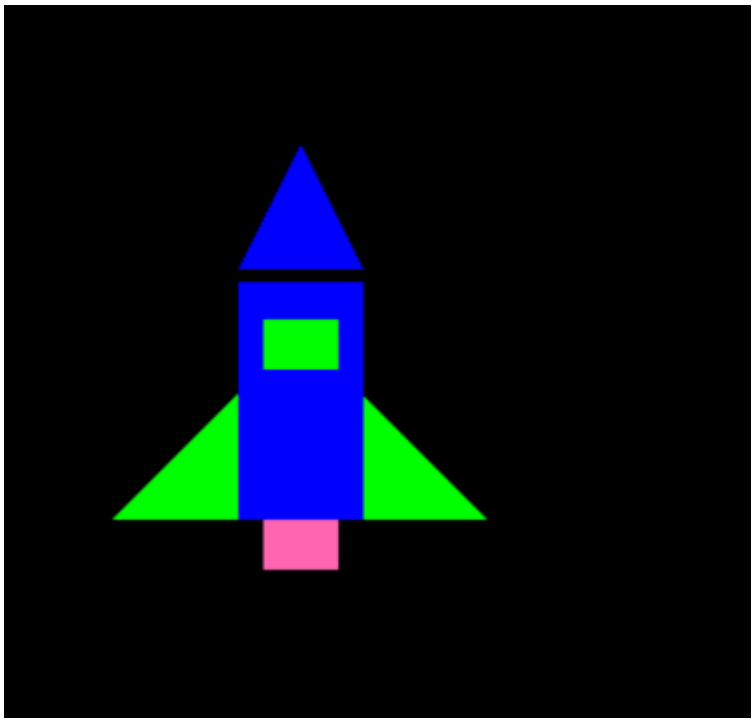
```
    glEnd();
// wing 2
    glBegin(GL_TRIANGLES);
    glVertex3f (.3, .3, 0);
    glVertex3f (.4, .2, 0);
    glVertex3f (.3, .2, 0);
    //glVertex3f (0.25, 0.75, 0.0);
    glEnd();
// blast
    glColor3f(1.0, 0.4, 0.7);
    glBegin(GL_POLYGON);
    glVertex3f (.22, .2, 0);
    glVertex3f (.28, .2, 0);
    glVertex3f (.28, .16, 0);
    glVertex3f (.22, .16, 0);
    glEnd();
//window
    glColor3f(0.0, 1.0, 0.0);

    glBegin(GL_POLYGON);
    glVertex3f (.22, .36, 0);
    glVertex3f (.28, .36, 0);
    glVertex3f (.28, .32, 0);
    glVertex3f (.22, .32, 0);
    glEnd();
```

```
/* don't wait!
* start processing buffered OpenGL routines
*/
    glFlush ();
}
void init (void)
{
/* select clearing (background) color */
    glClearColor (0.0, 0.0, 0.0, 0.0);
    /* initialize viewing values */
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}
/*
* Declare initial window size, position, and display mode
* (single buffer and RGBA). Open window with "hello"
* in its title bar. Call initialization routines.
* Register callback function to display graphics.
* Enter main loop and process events.
*/
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (100, 100);
```

```
    glutCreateWindow ("hello");
    init ();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0; /* ISO C requires main to return int. */
}
```

Output



Discussion

In this lab task, I successfully created a rocket using basic OpenGL functions. The rocket was drawn by combining simple shapes such as triangles for the nose and wings, and polygons for the body, window, and burst. Each part was given a different color using glColor3f() to make the figure more visually clear. The program used glBegin() and glEnd() to define shapes, while glVertex3f() specified their coordinates. Overall, this lab demonstrated how OpenGL can be used to design graphical objects using coordinate geometry & color control.