|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Drawing a Quad Shape and a Triangle with OpenGL** |         **SUBMITTED BY**   |  |  |  | | --- | --- | --- | | Name | ID | Section | | Kabid Yeiad | 202-15-14440 | 57\_A |   **SUBMITTED TO**  **Deawan Rakin Ahamed Remal,**  **Lecturer**  **Dept. of CSE**  **Daffodil International University**   |  | | --- | |  |   Submitted on November 1, 2023 |

**Drawing a Quad Shape and a Triangle with OpenGL**

**Code:**

#include <GL/glut.h>

void display() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

// *Quad Shape*

    glColor3f(1.0f, 0.0f, 0.0f);

    glBegin(GL\_QUADS);

        glVertex2f(50.0f, 50.0f);

        glVertex2f(150.0f, 50.0f);

        glVertex2f(150.0f, 150.0f);

        glVertex2f(50.0f, 150.0f);

    glEnd();

// *Triangle*

    glColor3f(0.0f, 0.0f, 1.0f);

    glBegin(GL\_TRIANGLES);

        glVertex2f(200.0f, 50.0f);

        glVertex2f(250.0f, 150.0f);

        glVertex2f(300.0f, 50.0f);

    glEnd();

    glFlush();

}

void init() {

    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(0.0, 400.0, 0.0, 200.0);

}

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

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(400, 200);

    glutCreateWindow("Quad Shape and Triangle");

    init();

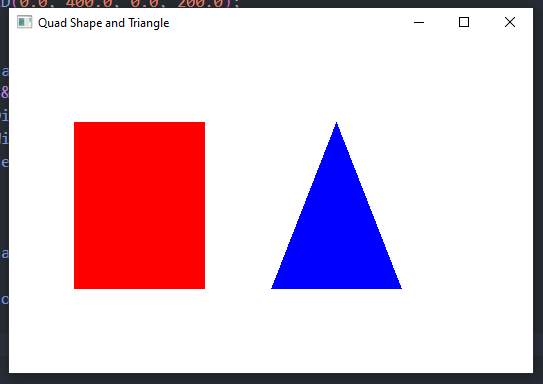
    glutDisplayFunc(display);

    glutMainLoop();

    return 0;

}

Output:



**Graph**:

**Discussion**:

In this experiment, we created an OpenGL program to draw a quad shape and a triangle. The code provided establishes the OpenGL environment, defines the shapes, and renders them on the screen. The quad shape is drawn using a series of vertices to form a four-sided polygon. Similarly, the triangle was drawn using three vertices to create a three-sided polygon. The graph displays the coordinates of the vertices for both the quad shape and the triangle. The x-axis represents the horizontal position, while the y-axis represents the vertical position.

For the quad shape, the vertices are labeled as A, B, C, and D. Their respective coordinates are:

A (50, 50)

B (150, 50)

C (150, 150)

D (50, 150)

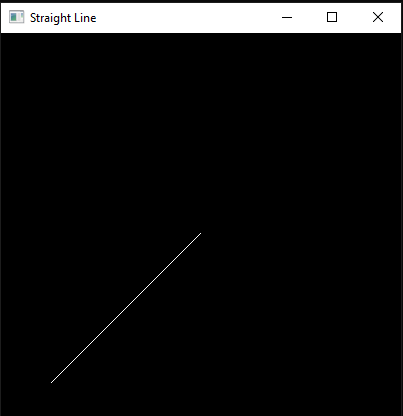
Similarly, for the triangle, the vertices are labeled as P, Q, and R, with coordinates:

P (200, 50)

Q (250, 150)

R (300, 50)

**Output:**



**Graph**:

**Discussion:**

In this experiment, we utilized OpenGL in conjunction with GLUT to draw a straight line. The code provided establishes the OpenGL environment, defines the coordinate system, and renders a line from the point (50, 50) to the point (200, 200). The output of the program clearly demonstrates the line drawn on the window as expected. This verifies that our OpenGL setup and drawing commands are functioning correctly. Additionally, we have included a graph to further illustrate the coordinates and path of the straight line. As seen in the graph, the line originates from (50, 50) and extends to (200, 200), following a linear path.