

Lab Report 2

Name: Arnob Das Shacha

ID: 221-15-5277

Section: 61_D1

Course Code: CSE 422

Experiment Name: Drawing a Robot using different OpenGL functions.

Submitted To:

Sayed Tasnim Alvi

Senior Lecturer

Dept of CSE

Daffodil International University

Title: Drawing a Robot using different OpenGL functions.

Introduction:

In this project, I have designed and drawn a colorful **robot** using the fundamental functions of **OpenGL**. The robot is constructed using a combination of geometric shapes such as polygons and triangles. The main goal of this work was to understand how to use **OpenGL primitives** and color functions to form complex graphical objects from simple shapes. Each part of the robot including the head, body, arms, and legs is drawn by defining specific vertex coordinates and applying different colors for visual distinction.

Contents:

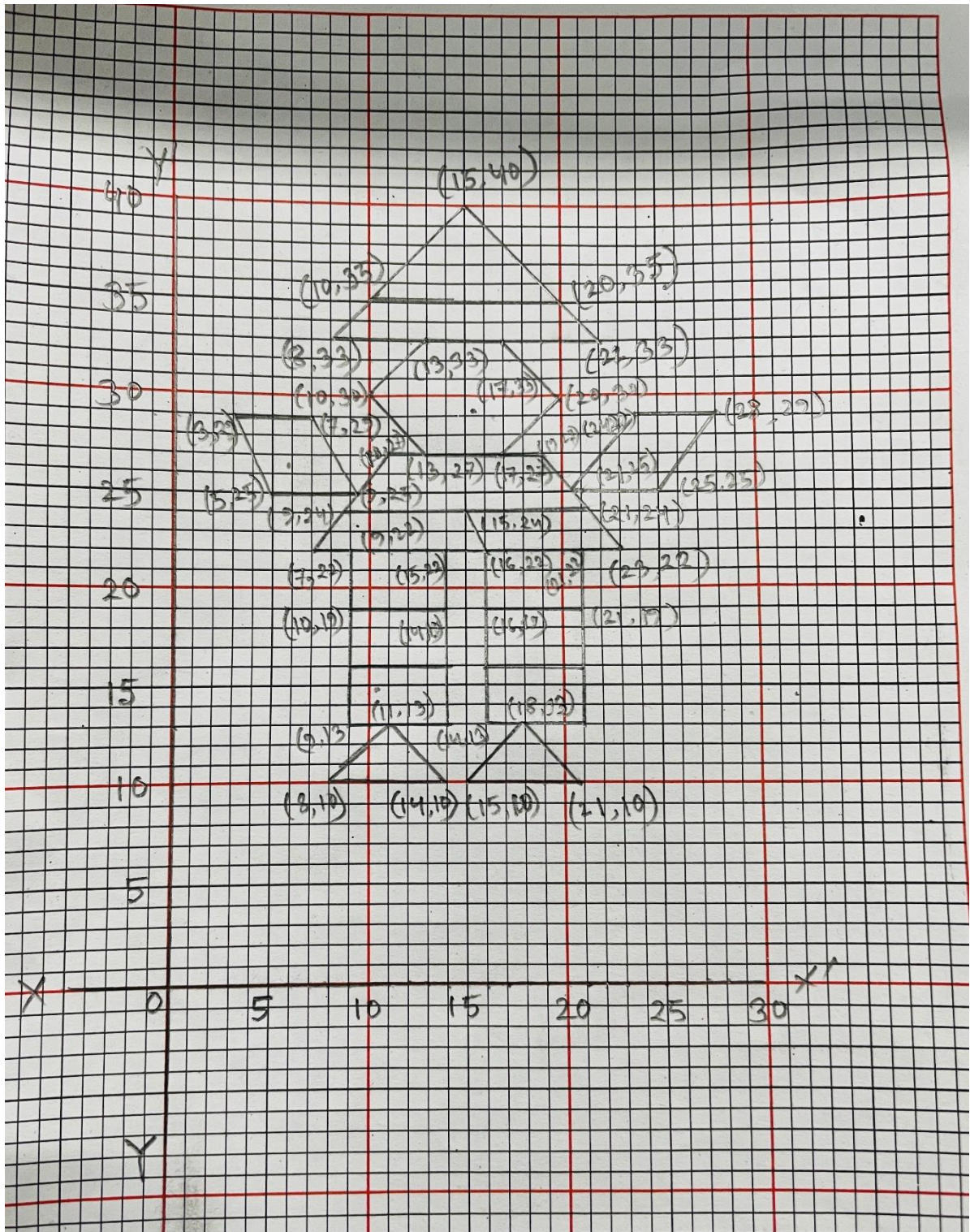
In this project, I have used the following OpenGL functions and shapes:

- **glBegin()** and **glEnd()** – Used to define the start and end of each shape drawing block.
- **glVertex2f()** – Used to specify the coordinates of each vertex.
- **glColor3f()** – Used to set different colors for each shape using RGB values.
- **glClear()** and **glFlush()** – Used to clear the color buffer and display the final drawing on the screen.
- **glOrtho()** – Used to set up a 2D coordinate system for drawing.

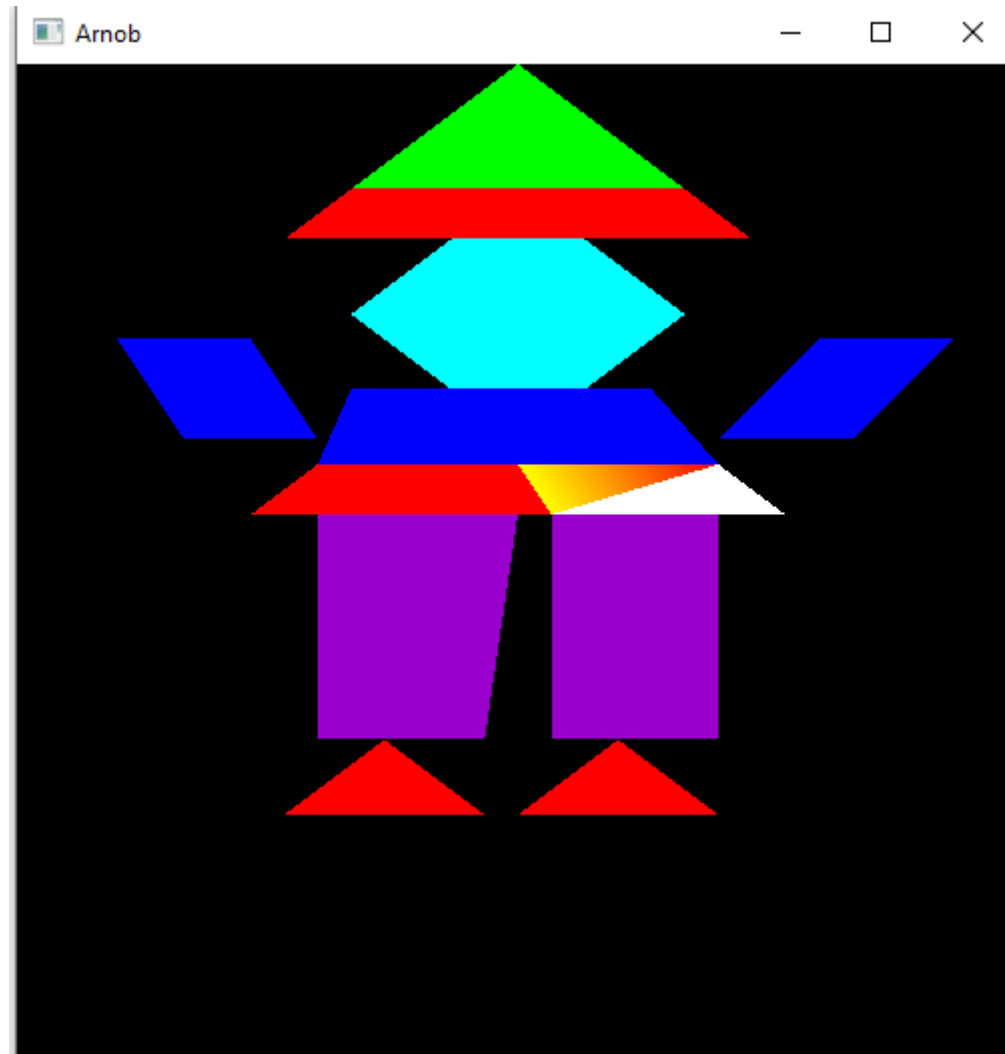
The robot is made using multiple polygons and triangles for its body parts:

- **Triangle** – for the head top (roof).
- **Polygons (Squares/Rectangles)** – for the body, chest, and arms.
- **Trapezoids and Triangles** – for the legs and lower portions.

Graph:



Output:



Code:

```
#include <GL/glut.h>
```

```
void display(void)
```

```
{
```

```
    glClear(GL_COLOR_BUFFER_BIT);
```

```
    // ▲ Triangle (roof)
```

```
glColor3f(0.0, 1.0, 0.0);  
glBegin(GL_POLYGON);  
    glVertex2f(15, 40);  
    glVertex2f(20, 35);  
    glVertex2f(10, 35);  
glEnd();
```

```
// □ Top small square  
glColor3f(1.0, 0.0, 0.0);  
glBegin(GL_POLYGON);  
    glVertex2f(20, 35);  
    glVertex2f(22, 33);  
    glVertex2f(8, 33);  
    glVertex2f(10, 35);  
glEnd();
```

```
// □ Center big 5 square  
glColor3f(0.0, 1.0, 1.0);  
glBegin(GL_POLYGON);  
    glVertex2f(13, 33);  
    glVertex2f(17, 33);  
    glVertex2f(20, 30);  
    glVertex2f(17, 27);  
    glVertex2f(13, 27);  
    glVertex2f(10, 30);  
glEnd();
```

```
// □ big square
```

```
glColor3f(0.0, 0.0, 1.0);  
glBegin(GL_POLYGON);  
glVertex2f(10, 27);  
glVertex2f(19, 27);  
glVertex2f(21, 24);  
glVertex2f(9, 24);
```

```
glEnd();
```

```
// □ below square  
glColor3f(1.0, 0.0, 0.0);  
glBegin(GL_POLYGON);  
glVertex2f(21, 24);  
glVertex2f(23, 22);  
glVertex2f(7, 22);  
glVertex2f(9, 24);
```

```
glColor3f(1.0, 1.0, 0.0);  
glBegin(GL_POLYGON);  
glVertex2f(21, 24);  
glVertex2f(23, 22);  
glVertex2f(16, 22);  
glVertex2f(15, 24);  
glEnd();
```

```
// right wing
```

```
glColor3f(0.0, 0.0, 1.0);
glBegin(GL_POLYGON);
    glVertex2f(3, 29);
    glVertex2f(7, 29);
    glVertex2f(9, 25);
    glVertex2f(5, 25);
glEnd();
```

```
// left wing
glColor3f(0.0, 0.0, 1.0);
glBegin(GL_POLYGON);
    glVertex2f(24, 29);
    glVertex2f(28, 29);
    glVertex2f(25, 25);
    glVertex2f(21, 25);
glEnd();
```

```
// □ left leg
glColor3f(0.6, 0.0, 0.8);
glBegin(GL_POLYGON);
    glVertex2f(9, 22);
    glVertex2f(15, 22);
    glVertex2f(14, 13);
    glVertex2f(9, 13);
glEnd();
```

```
// □ left leg tri
glColor3f(1.0, 0.0, 0.0);
```

```
glBegin(GL_POLYGON);  
    glVertex2f(11, 13);  
    glVertex2f(14, 10);  
    glVertex2f(8, 10);  
glEnd();
```

```
glColor3f(1.0, 1.0, 1.0);  
glBegin(GL_POLYGON);  
    glVertex2f(16, 22);  
    glVertex2f(21, 24);  
    glVertex2f(23, 22);  
glEnd();
```

```
// □ right leg  
glColor3f(0.6, 0.0, 0.8);  
glBegin(GL_POLYGON);  
    glVertex2f(16, 22);  
    glVertex2f(21, 22);  
    glVertex2f(21, 13);  
    glVertex2f(16, 13);  
  
glEnd();
```

```
// □ right leg tri  
glColor3f(1.0, 0.0, 0.0);  
glBegin(GL_POLYGON);  
    glVertex2f(18, 13);  
    glVertex2f(21, 10);
```



```
    glVertex2f(15, 10);  
    glEnd();
```

```
    glFlush();  
}
```

```
void init(void)  
{  
    glClearColor(0.0, 0.0, 0.0, 0.0); // Black background  
    glMatrixMode(GL_PROJECTION);  
    glLoadIdentity();  
    glOrtho(0.0, 30.0, 0.0, 40.0, -1.0, 1.0);  
}
```

```
int main(int argc, char** argv)  
{  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);  
    glutInitWindowSize(500, 500);  
    glutInitWindowPosition(100, 100);  
    glutCreateWindow("Arnob");  
    init();  
    glutDisplayFunc(display);  
    glutMainLoop();  
    return 0;  
}
```

Discussion:

In this project, a robot figure was successfully drawn using basic OpenGL 2D primitives. The structure was created by combining multiple polygons and triangles to represent various robot parts like the head, body, wings, and legs. Each section was assigned a unique color using the `glColor3f()` function to give the robot a more vivid and realistic appearance. The `glOrtho()` function helped establish a 2D viewing coordinate system for accurate shape positioning. Through this task, I learned how simple geometric coordinates can be combined to form a complete structured figure in computer graphics, demonstrating the fundamental concepts of shape modeling and color handling in OpenGL.