

ASSIGNMENT - C1

Problem Statement :

Write a C++/Java program to draw 3-D cube and perform following on it using open GL.

- a) Scaling
- b) Translation
- c) Rotation about one axis.

Objectives :

- To understand and implement open GL function
- To understand and learn the GLUT library.

Outcome :

Learn and understand open GL for animating 3-D objects. Implement a simple cube with faces of different colours using glut library.

Theory :

Open GL (Open Graphics Library) is a standard specification defining a cross language cross platform API for creating application that produce 2D and 3D computer graphics. The interface consists of over 250 different functions calls which can be used to draw complex 3D scenes from simple primitives. Open GL is a low level procedural API requiring the programmer to dictate the exact the steps required to record a scene.

In this basic operation, open GL accepts primitives such as points, lines and polygons and converts them into pixels via a graphics pipeline known as the open GL state machine.

Most Open GL commands either issue primitive to the graphics pipeline or configure how the pipeline processes these primitives.

Commands	Evaluator	Primitive Assembly	Rasterization	Per fragment operation
----------	-----------	--------------------	---------------	------------------------

<div style="border: 1px solid black; padding: 2px; display: inline-block;">Display List</div>	Pixel operation	Texture memory	Frame buffer
---	-----------------	----------------	--------------

The command enter from left and proceed through processing pipeline. Some commands specify geometric objects to the drawing and others control how the objects are handled during various processing stages.

Functions and features of: open GL.

1. Display List.
2. Feedback.
3. Alpha blending.
4. Pixel operation
5. Texture mapping.
6. Colour index mode.

7. Polynomial evaluator -
8. Scaling and rotation.

Algorithm:

```

void DrawAxes() {
    glColor3f(1.0, 1.0, 1.0);
    glBegin(GL_LINES);
    glVertex3fv(org); glVertex3fv(vOP);
    glVertex3fv(org); glVertex3fv(vOP);
    glVertex3fv(org); glVertex3fv(zOP);
    glEnd();
    glRasterPos3f(2, 0.0, 0.0);
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, 'X');
    glRasterPos3f(0.0, 0.0, 2);
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, 'Y');
}

```

```

void display() {
    float int v;
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    glTranslatef(0, 0, -6), DrawAxes();
    glPushMatrix();
    glTranslatef(d[0], d[1], d[2]);
    glScalef(xs, ys, zs);
    glRotatef(x angle, 2, 0, 0);
    Draw_Box();
    glPopMatrix();
    glutSwapBuffer();
}

```

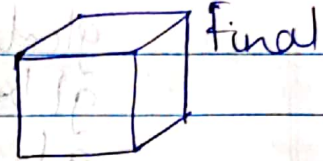
Test Case

1) Translation Scaling

$$T_x, T_y, T_z = 100, 100, 100$$

$$S_x, S_y, S_z = 3, 3, 3$$

Initial

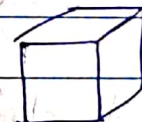


Final

2) Translation

$$T_x, T_y, T_z = 100, 100, 100$$

Initial



Final.

Conclusion

Thus the 3-D cube was implemented in Open GL using glut library -