Assignment No. (C1) DATE:

Problem Statement: Write C++ program to draw 3-D cube & perform following transformations on it using openGL A) Scaling B) Translation C) Rotation about one axis.

Learning Objective: To understand and implement open GL function and to learn the glut library.

Learning outcome:
1) Learn and understand OpenGL for animating 3D objects.
2) Implement a simple cube with faces of different colors using glut library.

Requirements: Fedora 20, QT Creator

Theory:
1) OpenGL (Open Graphics Library) is a standard specification defining a cross language, cross platform API for writing appln that produce 2D & 3D Computer graphics. The interface consists of over 250 different fun calls which can be used to draw complex 3D scenes from simple primitives. OpenGL is a low level, procedural API requiring the programmer, to dictate the exact steps required to record a scene.
2) Most openGL commands either issue primitive to the graphics pipeline or configure how the pipeline process these primitives.

Commands → Evaluator → Primitive assembly → Rasterization → Per fragment operation → Framebuffer

Framebuffer ← Pixel operation ← Texture memory ← Display list

Scanned with CamScanner

DATE:

Function and features of openGL:-

1. Display list
2. Feedback
3. Alpha bonding
4. Pixel operation
5. Texture mapping
6. Color index mode
7. Polynomial evaluator
8. Scaling & 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(vop);
    glEND();
    glRasterPos3f(0.0,0.0,0.0);
    glutBitmapCharacter(GLUT_BitMap_HELVETICA_18,'x')
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,'x')
    glRasterPos3f(0.0,0.0,2);
    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18,'x')
}
void display() {
    int r;
    glClear(GL_COLOR_BUFFER_BIT);
    gl(Load,Identity(),
    glTranslate(0,0,-6); DrawAxes();
```

FOR EDUCATIONAL USE

Scanned with CamScanner

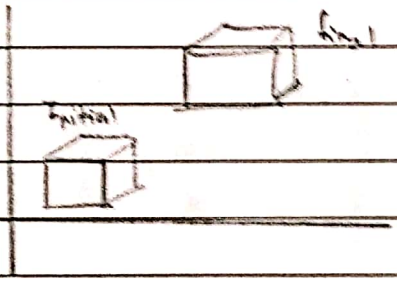
DATE:

```
glPushMatrix()
glTranslate(d[0], d[1], d[2]);
glScale(x^r, y^r, z^r);
glRotate(r angle, 2, 0, 0);
Draw_Box();
glPopMatrix();
glutSwapBuffers();
}
```

Test Cases:

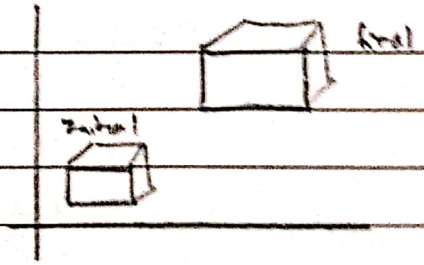
1] Translation

Tx Ty Tz = 100, 100, 100



2] Scaling

Sx Sy Sz = 3, 3, 3



Conclusion: Thus, from this assignment, the 3D cube was implemented in openGL using glut libraries.

FOR EDUCATIONAL USE

Scanned with CamScanner