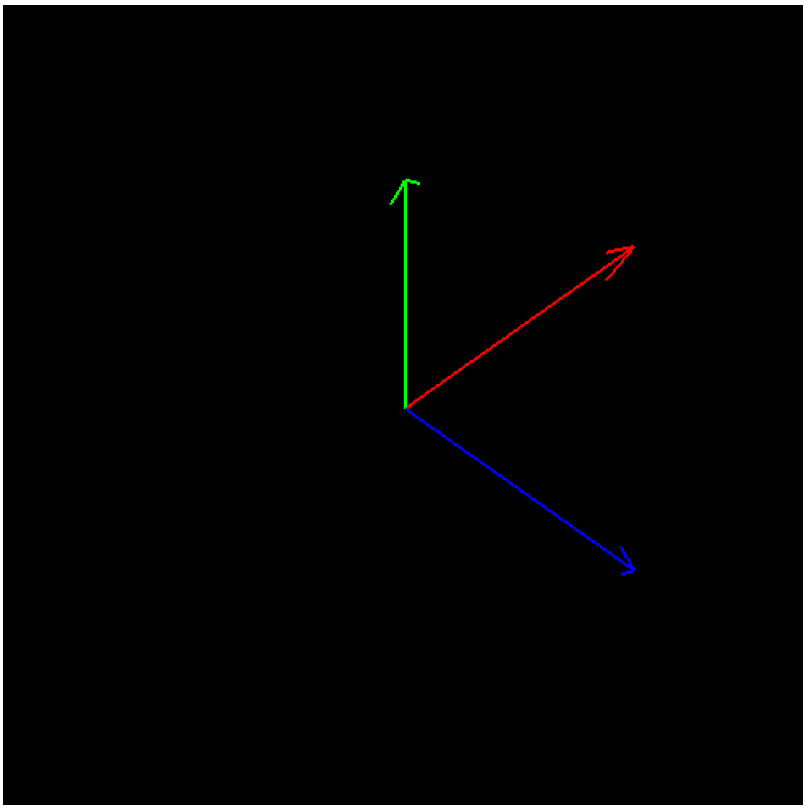
2025-04-19_CG_20_x,y,z 축 화살표 그리기

📁 예제 설명:

📝 목표 출력



📝 조건

📁 해결 코드

📝 핵심 코드

📎 x축 (빨간색)

```
glColor3f(1.0f, 0.0f, 0.0f); // 빨강
glVertex3f(0.0f, 0.0f, 0.0f); // 시작점
glVertex3f(80.0f, 0.0f, 0.0f); // 끝점 (X축 양방향)

glVertex3f(80.0f, 0.0f, 0.0f); // 화살표 좌측 위
glVertex3f(75.0f, 5.0f, 0.0f);
```

```
glVertex3f(80.0f, 0.0f, 0.0f); // 화살표 좌측 아래
glVertex3f(75.0f, -5.0f, 0.0f);
```

y축 (초록색)

```
glColor3f(0.0f, 1.0f, 0.0f); // 초록
glVertex3f(0.0f, 0.0f, 0.0f);
glVertex3f(0.0f, 80.0f, 0.0f);

glVertex3f(0.0f, 80.0f, 0.0f); // 화살표 오른쪽
glVertex3f(5.0f, 75.0f, 0.0f);

glVertex3f(0.0f, 80.0f, 0.0f); // 화살표 왼쪽
glVertex3f(-5.0f, 75.0f, 0.0f);
```

z축 (파란색)

```
glColor3f(0.0f, 0.0f, 1.0f); // 파랑
glVertex3f(0.0f, 0.0f, 0.0f);
glVertex3f(0.0f, 0.0f, 80.0f);

glVertex3f(0.0f, 0.0f, 80.0f); // 화살표 위
glVertex3f(0.0f, 5.0f, 75.0f);

glVertex3f(0.0f, 0.0f, 80.0f); // 화살표 아래
glVertex3f(0.0f, -5.0f, 75.0f);
```

전체 코드

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

#define GL_PI 3.1415f

void RenderScene(void) {

    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0f, 0.0f, 0.0f);
    glLineWidth(2.0f);

    glPushMatrix();
    glRotatef(45, 1.0f, 0.0f, 0.0f);
    glRotatef(45, 0.0f, 1.0f, 0.0f);

    glBegin(GL_LINES);
```

```
    // 1. x축
    glVertex3f(0.0f, 0.0f, 0.0f);
    glVertex3f(80.0f, 0.0f, 0.0f);
    glVertex3f(80.0f, 0.0f, 0.0f);
    glVertex3f(70.0f, 5.0f, 0.0f);
    glVertex3f(80.0f, 0.0f, 0.0f);
    glVertex3f(70.0f, -5.0f, 0.0f);

    // 2. y축
    glColor3f(0.0f, 1.0f, 0.0f);
    glVertex3f(0.0f, 0.0f, 0.0f);
    glVertex3f(0.0f, 80.0f, 0.0f);
    glVertex3f(0.0f, 80.0f, 0.0f);
    glVertex3f(5.0f, 75.0f, 0.0f);
    glVertex3f(0.0f, 80.0f, 0.0f);
    glVertex3f(-5.0f, 75.0f, 0.0f);

    // 3. z축
    glColor3f(0.0f, 0.0f, 1.0f);
    glVertex3f(0.0f, 0.0f, 0.0f);
    glVertex3f(0.0f, 0.0f, 80.0f);
    glVertex3f(0.0f, 0.0f, 80.0f);
    glVertex3f(0.0f, 5.0f, 75.0f);
    glVertex3f(0.0f, 0.0f, 80.0f);
    glVertex3f(0.0f, -5.0f, 75.0f);

    glEnd();

    glPopMatrix();
    glFlush();
}

void ChangeSize(GLsizei w, GLsizei h) {

    GLint wSize = 100.0f;
    GLfloat aspectRatio;

    if (h == 0) h = 1;

    glViewport(0, 0, w, h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    aspectRatio = (GLfloat)w / (GLfloat)h;
    if (aspectRatio >= 1.0f) {
        glOrtho(-wSize*aspectRatio, wSize*aspectRatio, -wSize, wSize, -wSize, wSize);
    }
```

```
    else {
        glOrtho(-wSize, wSize, -wSize/aspectRatio, wSize/aspectRatio, -wSize, wSize);
    }

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void SetupRC(void) {
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
}

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

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);

    glutCreateWindow("simple");

    SetupRC();

    glutDisplayFunc(RenderScene);
    glutReshapeFunc(ChangeSize);

    glutMainLoop();
}
```