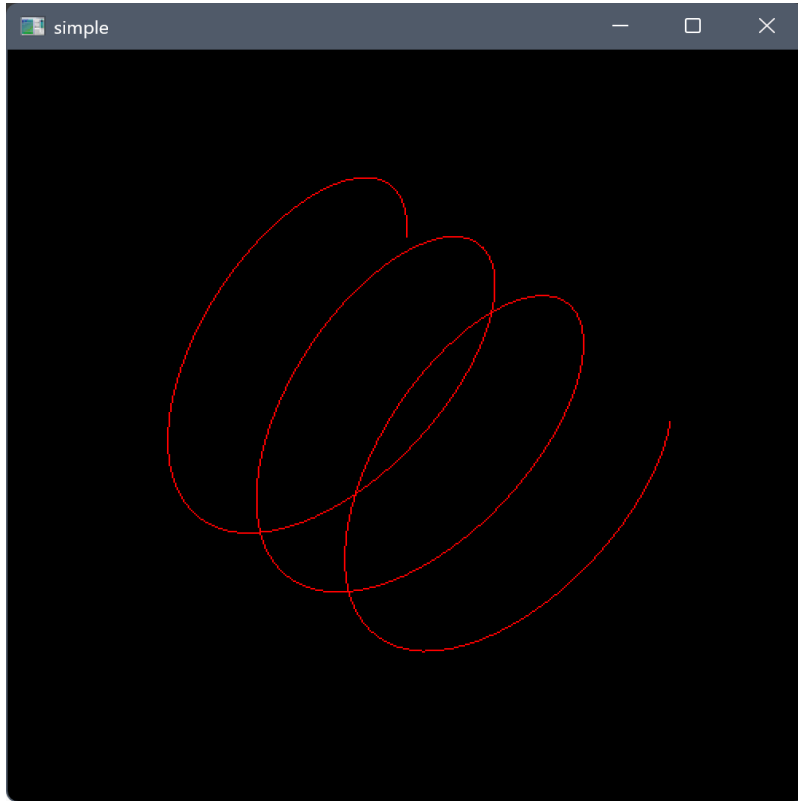


| 2025-04-19_CG_14_나선_06_GL_LINE_STRIP 나선

| 📁 예제 설명: 점이 아닌 선으로 나선 그리기

| 📄 목표 출력



| 📄 조건

- `glBegin(GL_LINE_STRIP)` 사용하기.

| 📁 해결 코드

| 📄 핵심 코드

```
glBegin(GL_LINE_STRIP); // 📌  
z = -50.0f;  
for (angle = 0.0f; angle < 2.0f * GL_PI * 3.0f; angle += 0.1f) {  
    x = 50.0f * cos(angle);  
    y = 50.0f * sin(angle);  
  
    glVertex3f(x, y, z);  
}
```

```
    z += 0.5;
}
glEnd();
```

I 전체 코드

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

#define GL_PI 3.1415f

void RenderScene(void) {

    GLfloat x, y, z, angle;

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

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

    glBegin(GL_LINE_STRIP);
    z = -50.0f;
    for (angle = 0.0f; angle < 2.0f * GL_PI * 3.0f; angle += 0.1f) {
        x = 50.0f * cos(angle);
        y = 50.0f * sin(angle);

        glVertex3f(x, y, z);

        z += 0.5;
    }
    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();
}

```

추가 설명

GL_LINES와 GL_LINE_STRIP의 차이?

- `GL_LINE_STRIP` 과 `GL_LINES` 는 모두 선을 그리는 기본 도형 타입
- 점들 간 연결 방식은 완전히 다르다.

GL_LINE_STRIP

- 연속된 점들을 하나의 선으로 연결한다.

점1 - 점2 - 점3 - 점4 - 점5

I GL_LINES

- 각 두 개의 정점이 한 쌍이 되어 하나의 선분을 만든다.

점1 - 점2 점3 - 점4 점5 - 점6