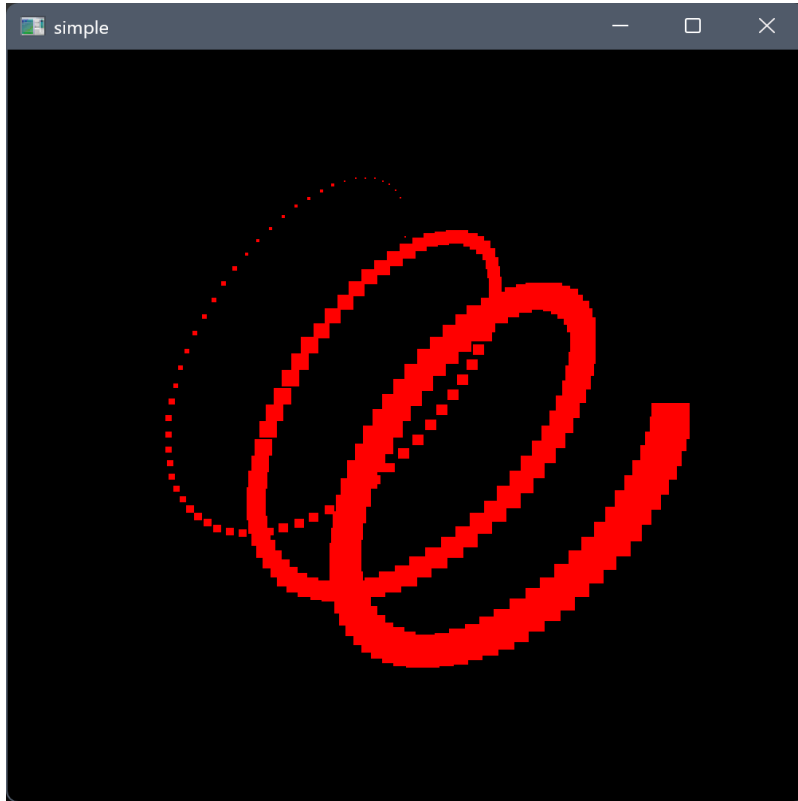


| 2025-04-19_CG_10_나선_03_점점 커지는 나선

| 📁 예제 설명: 점점 커지는 나선

| 📄 목표 출력



| 📁 해결 코드

| 📄 핵심 코드

```
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);

    glPointSize(curSize);

    // ✨
    glBegin(GL_POINTS);
        glVertex3f(x, y, z);
    glEnd();
}
```

```

    z += 0.5f;
    curSize += step;
}

```

전체 코드

```

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

#define GL_PI 3.1415f

void RenderScene(void) {

    GLfloat x, y, z, angle;
    GLfloat sizes[2];      // ✖
    GLfloat step;          // ✖
    GLfloat curSize = 0.0f // ✖

    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);

    glGetFloatv(GL_POINT_SIZE_RANGE, sizes);      // ✖
    glGetFloatv(GL_POINT_SIZE_GRANULARITY, &step); // ✖

    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);

        // ✖
        glPointSize(curSize);

        // ✖
        glBegin(GL_POINTS);
        glVertex3f(x, y, z);
        glEnd();

        z += 0.5f;

        // ✖
        if (curSize < sizes[1]) {

```

```

        curSize += step;
    }
}

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();
}
```

📁 설명

📌 `glBegin()` ~ `glEnd()` 를 왜 for문 안에 위치시키는가?

```
glBegin(GL_POINTS);
for (...) {
    glVertex3f(...);
}
glEnd();
```

- `glPointSize()` 는 `glBegin()` 이전에 설정해야 반영된다.
 - `glPointSize()` 는 상태(state) 기반 함수이므로,
 - `glBegin()` 이전에 호출되어야 그 점의 크기가 적용된다.
 - 만약 `glBegin()` 밖에서 한 번만 호출하면, 모든 점이 동일한 크기로 출력된다.

📌 만약 반대로 구현했다면,

```
glBegin(GL_POINTS);
for (...) {
    glPointSize(curSize); // ❌ 이걸 무시됨!
    glVertex3f(x, y, z); // → 이전 상태의 크기만 반영됨
}
glEnd();
```

- `glBegin()` 안에서는 상태 변경(`glPointSize` 등)이 무시된다.
- 그래서 모든 점이 같은 크기로 찍힌다.

📌 `glColor3f()` 도 `glBegin()` 전에 호출해야 개별 색상이 적용되는가?

- ❌ `glColor3f()` 는 `glBegin()` ~ `glEnd()` 사이에 호출해도 유효하게 작동하며, 그 위치 이후의 정점부터 색이 적용된다.