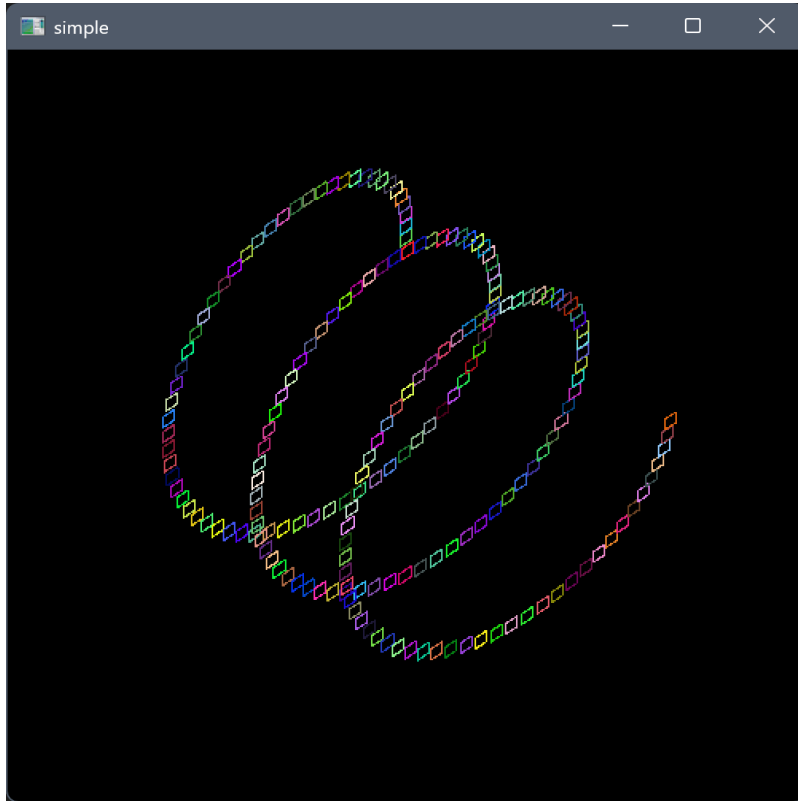


## | 2025-04-19\_CG\_13\_나선\_05\_랜덤 색상

### | 📁 예제 설명:

### | 📄 목표 출력



### | 📁 해결 코드

### | 📄 핵심 코드

```
GLint offset = 2;
glBegin(GL_POINTS);
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);

    GLfloat r = (rand() % 100) / 99.9f;
    GLfloat g = (rand() % 100) / 99.9f;
    GLfloat b = (rand() % 100) / 99.9f;
    glColor3f(r, g, b);
```

```

    for (float i = -offset; i <= offset; i += 0.2) {
        glVertex3f(x + offset, y + i, z);
        glVertex3f(x - offset, y + i, z);
        glVertex3f(x + i, y + offset, z);
        glVertex3f(x + i, y - offset, 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);

    GLint offset = 2;
    glBegin(GL_POINTS);
    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);

        // ✨
        GLfloat r = (rand() % 100) / 99.9f;
        GLfloat g = (rand() % 100) / 99.9f;
        GLfloat b = (rand() % 100) / 99.9f;
        glColor3f(r, g, b);

        for (float i = -offset; i <= offset; i += 0.2) {
            glVertex3f(x + offset, y + i, z);
            glVertex3f(x - offset, y + i, z);
            glVertex3f(x + i, y + offset, z);
            glVertex3f(x + i, y - offset, 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();
}
```

## I 📁 설명

I ? `glPointSize()` 는 `glBegin` 밖에서 호출해야 각 점마다 크기가 달라지는데, `glColor3f()` 는 왜 `glBegin` 안에서도 호출해도 각 점마다 색상이 다르게 적용되는가?

- `glColor3f()` 는 상태(state)를 바꾸고, 버텍스를 지정할 때 그 색 상태가 함께 저장된다.
- 반면 `glPointSize()` 는 하드웨어에 따라 상태 적용 시점이 `glBegin` 이전으로 제한된다.

```
glBegin(GL_POINTS);
glColor3f(1, 0, 0); glVertex3f(...);
glColor3f(0, 1, 0); glVertex3f(...); // 다른 색상 적용됨
glEnd();
```

```
glBegin(GL_POINTS);
glPointSize(5); // 적용되지 않음 ❌
glVertex3f(...);
glEnd();
```

## I 📄 rand()

- `int rand()` 는 0부터 `RAND_MAX` 사이의 정수값을 리턴한다.
- `rand()` 는 아주 큰 수를 리턴하니까, `% 100` 을 하면 0부터 99까지의 정수 중 하나가 된다.
- `/ 99.9f` :
  - `glColor3f(r, g, b)` 함수는 각 색상값을 `0.0 ~ 1.0` 사이로 줘야 한다.
  - 그래서 정수 0~99를 실수 `0.0 ~ 1.0`으로 정규화하려고 `99.9f` 로 나눔.
  - `99.0f` 또는 `100.0f` 로 나눠도 된다.