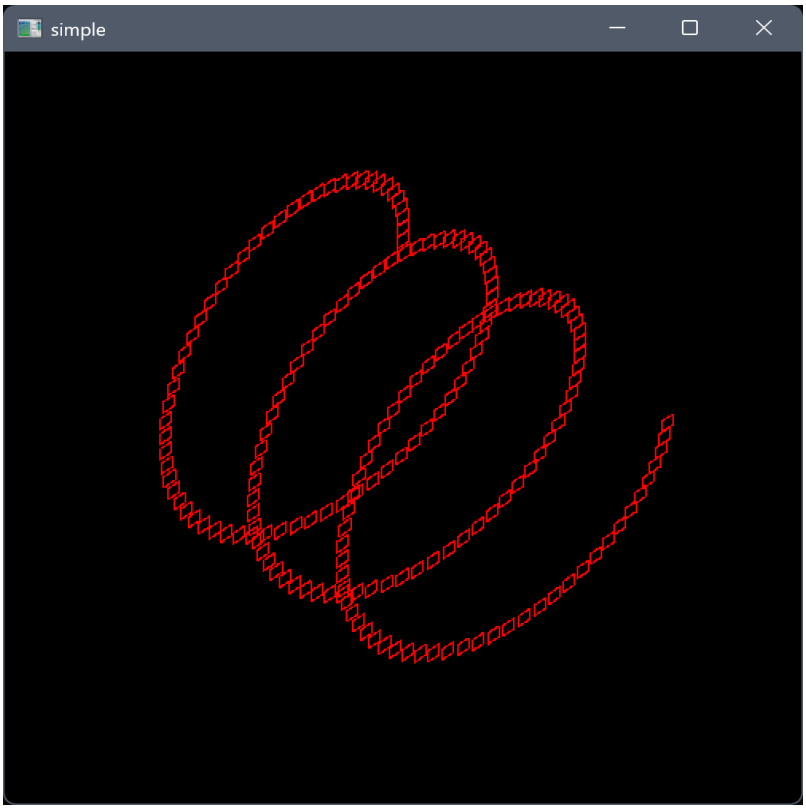
2025-04-19_CG_12_나선_04_점대신 사각형 찍기

📁 예제 설명:

📝 목표 출력

- 

📝 조건

- 점을 찍는 방식으로 구현하여
  - 사각형이 나선으로 찍히도록 보여야 한다.

📁 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);
```

```
    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();
```

📝 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);

        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();
```

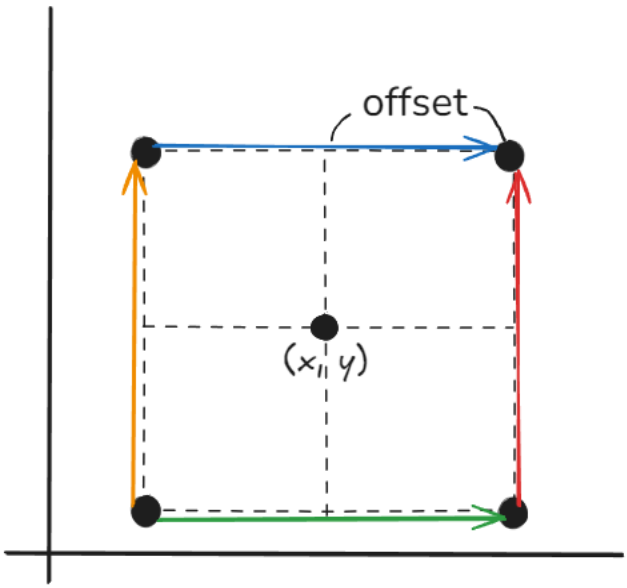
```
    glFlush();
}

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();
}
```

설명

- 기존의 `glVertex3f(x, y, z);` 를 아래와 같이 변경한 것이다.

```
for (float i = -offset; i <= offset; i += 0.2) {
    glVertex3f(x + offset, y + i, z);  // 수직선 우측
    glVertex3f(x - offset, y + i, z);  // 수직선 좌측
    glVertex3f(x + i, y + offset, z);  // 수평선 위
    glVertex3f(x + i, y - offset, z);  // 수평선 아래
}
```

- 그림으로 보면 아래와 같다.



i 는 -offset 부터 offset 까지 증가해가면서 반복

glVertex3f(x + offset, y + i, z);

glVertex3f(x - offset, y + i, z);

glVertex3f(x + i, y + offset, z);

glVertex3f(x + i, y - offset, z);

- 위 사각형이 나선 경로를 따라 그려진다.

추가 예제

```
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);

    for (float theta = 0.0f; theta <= 2.0f * GL_PI; theta += 0.1f) {
        float dx = offset * cos(theta);
        float dy = offset * sin(theta);
        glVertex3f(x + dx, y + dy, z);
    }
```

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