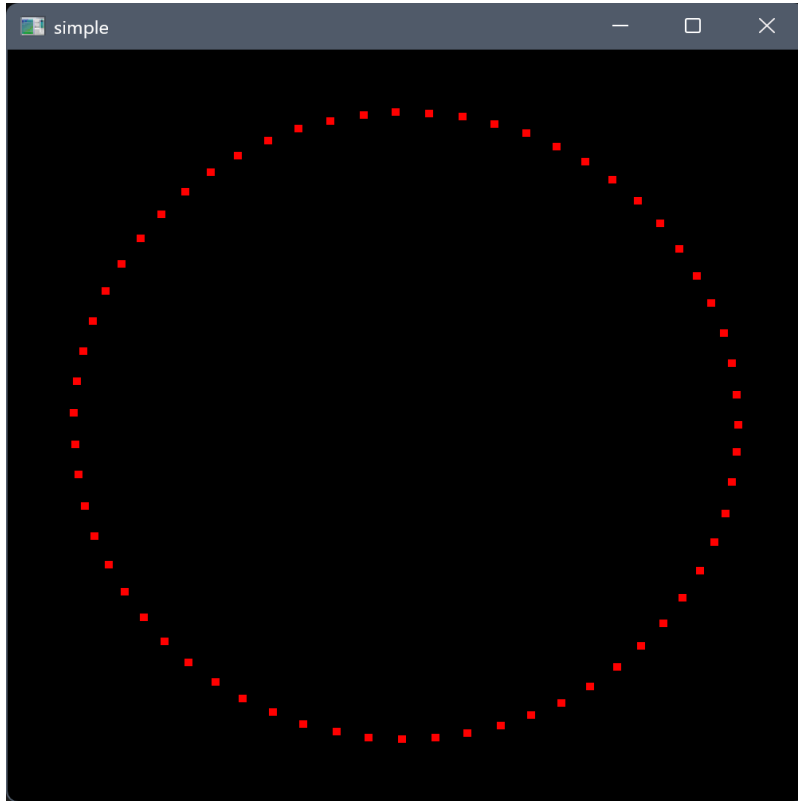


| 2025-04-19_CG_07_점으로 원 그리기

| 📁 예제 설명: 점으로 원 그리기

| 📄 목표 출력



| 📁 해결 코드

| 📄 핵심 코드

```
glBegin(GL_POINTS);  
for (angle = 0.0f; angle <= (2.0f * GL_PI); angle += 0.1f) {  
    x = 50.0f * cos(angle);  
    y = 50.0f * sin(angle);  
    glVertex3f(x, y, 0.0f);  
}  
glEnd();
```

| 📄 전체 코드

```

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

#define GL_PI 3.1415f

void RenderScene(void) {
    std::cout << "RenderScene" << std::endl;

    GLfloat x, y, z, angle;

    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(1.0f, 0.0f, 0.0f);
    glPointSize(5.0f);

    // ✖
    glBegin(GL_POINTS);
    for (angle = 0.0f; angle <= (2.0f * GL_PI); angle += 0.1f) {
        x = 50.0f * cos(angle);
        y = 50.0f * sin(angle);
        glVertex3f(x, y, 0.0f);
    }
    glEnd();

    glFlush();
}

void ChangeSize(GLsizei w, GLsizei h) {
    std::cout << "ChangeSize" << std::endl;

    GLint wSize = 60.0f;
    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, -1, 1);
    }
    else {
        glOrtho(-wSize, wSize, -wSize/aspectRatio, wSize/aspectRatio, -1, 1);
    }
}

```

```

    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(400, 400);

    glutCreateWindow("simple");

    SetupRC();

    glutDisplayFunc(RenderScene);
    glutReshapeFunc(ChangeSize);

    glutMainLoop();
}

```

📁 설명

- 원의 중심이 (0, 0)이고 반지름이 r 일 때, 모든 점은 다음 공식을 만족한다:

$$x = r \cdot \cos(\theta), y = r \cdot \sin(\theta)$$

- 이를 코드로 나타내면:

```

x = 50.0f * cos(angle);
y = 50.0f * sin(angle);

```

- $\text{angle}(\theta)$ 은 0 부터 2π 까지 조금씩 증가 시킨다.

```

for (angle = 0.0f; angle <= (2.0f * GL_PI); angle += 0.1f) { }

```

- 따라서, 0 부터 2π 까지 angle을 증가시키며, 삼각함수를 사용하여 x, y 좌표를 구하고 점까지 찍는 코드는 아래와 같다.
 - 점들을 화면에 찍기 위해, OpenGL의 `GL_POINTS` 모드를 사용한다.

```
// 반지름 50짜리 원의 외곽에 점을 찍는다.  
glBegin(GL_POINTS);  
for (angle = 0.0f; angle <= (2.0f * GL_PI); angle += 0.1f) {  
    x = 50.0f * cos(angle);  
    y = 50.0f * sin(angle);  
    glVertex3f(x, y, 0.0f);  
}  
glEnd();
```

I 📁 추가 설명

I glBegin(), glEnd()

- glBegin(GL_모드)
 - 도형을 그리기 시작한다는 신호.
 - GL_POINTS : 점
 - GL_LINES : 선
 - GL_TRIANGLES : 삼각형
 - GL_QUADS : 사각형
 - GL_POLYGON : 다각형 등
- glEnd()
 - 그리기 명령의 끝을 나타낸다.
 - "지금까지 입력한 꼭짓점들로 도형을 그려줘"라고 명령하는 것과 같다.