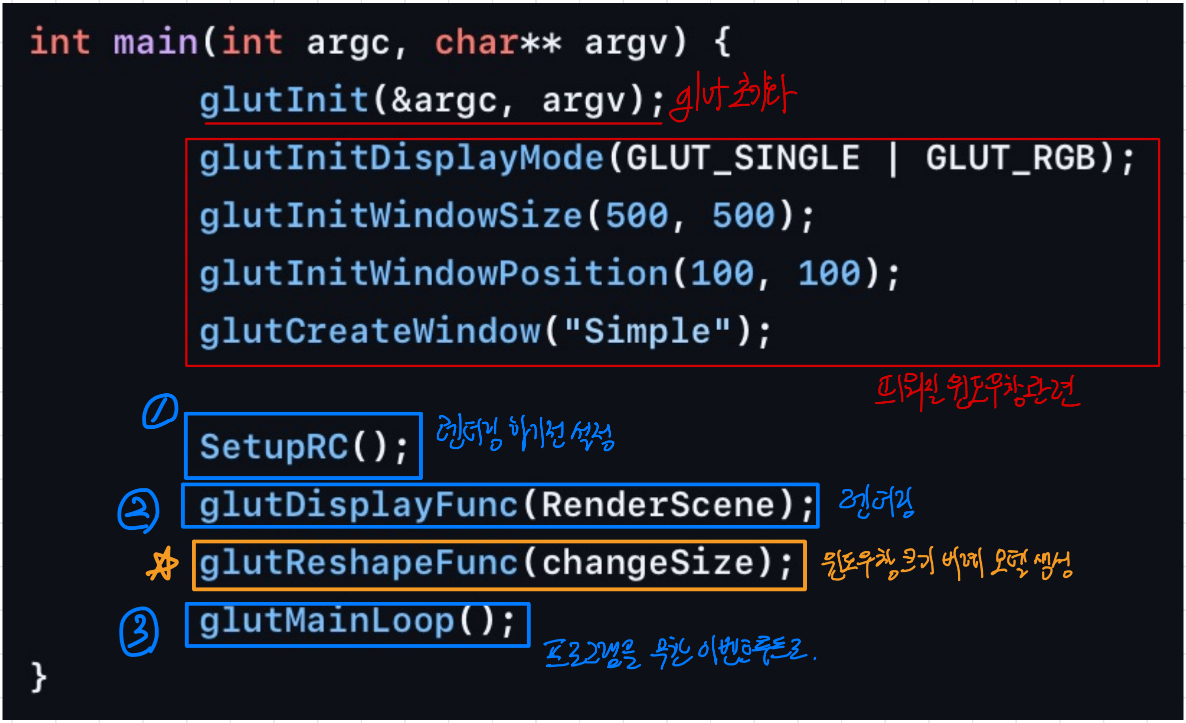


```
int main(int argc, char** argv) {
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(500, 500);
        glutInitWindowPosition(100, 100);
        glutCreateWindow("Simple");

        SetupRC();
        glutDisplayFunc(RenderScene);
        glutReshapeFunc(changeSize);
        glutMainLoop();
}
```

glutInit(&argc, argv); glut 초기화

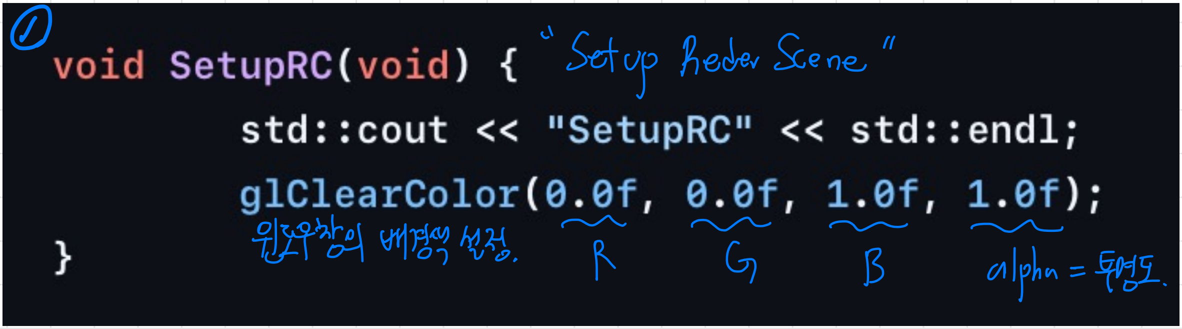
띄워질 윈도우창 관련

① SetupRC(); 렌더링 하기전 설정

② glutDisplayFunc(RenderScene); 렌더링

☆ glutReshapeFunc(changeSize); 윈도우창 크기 비례 오류 생성

③ glutMainLoop(); 프로그램을 무한 이벤트루프로.

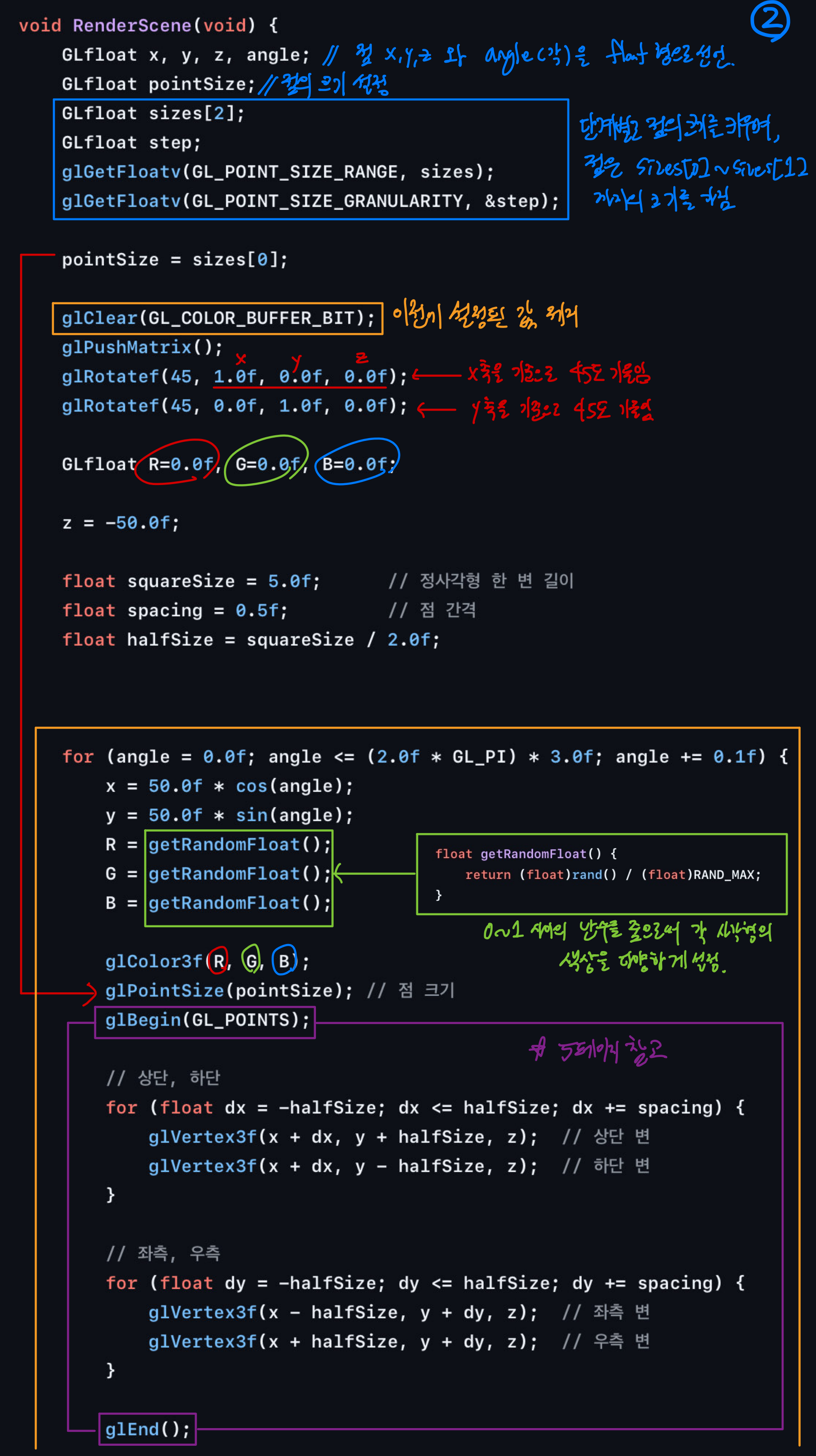


```
void SetupRC(void) {
        std::cout << "SetupRC" << std::endl;
        glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
}
```

① void SetupRC(void) { " Setup Reder Scene "

윈도우창의 배경색 설정.

R G B alpha = 투명도.



②

```
void RenderScene(void) {
    GLfloat x, y, z, angle; // 점 x,y,z 와 angle(각)을 float 형으로 선언.
    GLfloat pointSize; //점의 크기 설정
    GLfloat sizes[2];
    GLfloat step;
    glGetFloatv(GL_POINT_SIZE_RANGE, sizes);
    glGetFloatv(GL_POINT_SIZE_GRANULARITY, &step);
```

단계별로 점의 크기를 키우며, 점은 sizes[0]~sizes[1] 까지의 크기를 가짐

```
    pointSize = sizes[0];

    glClear(GL_COLOR_BUFFER_BIT);
```

이전에 설정된 값 제거

```
    glPushMatrix();
                   x     y     z
    glRotatef(45, 1.0f, 0.0f, 0.0f);
```

x축을 기준으로 45도 기울임

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

y축을 기준으로 45도 기울임

```
    GLfloat R=0.0f, G=0.0f, B=0.0f;

    z = -50.0f;

    float squareSize = 5.0f;       // 정사각형 한 변 길이
    float spacing = 0.5f;          // 점 간격
    float halfSize = squareSize / 2.0f;



    for (angle = 0.0f; angle <= (2.0f * GL_PI) * 3.0f; angle += 0.1f) {
        x = 50.0f * cos(angle);
        y = 50.0f * sin(angle);
        R = getRandomFloat();
        G = getRandomFloat();
        B = getRandomFloat();
```

```
float getRandomFloat() {
    return (float)rand() / (float)RAND_MAX;
}
```

0~1 사이의 난수를 줌으로써 각 사각형의 색상을 다양하게 설정.

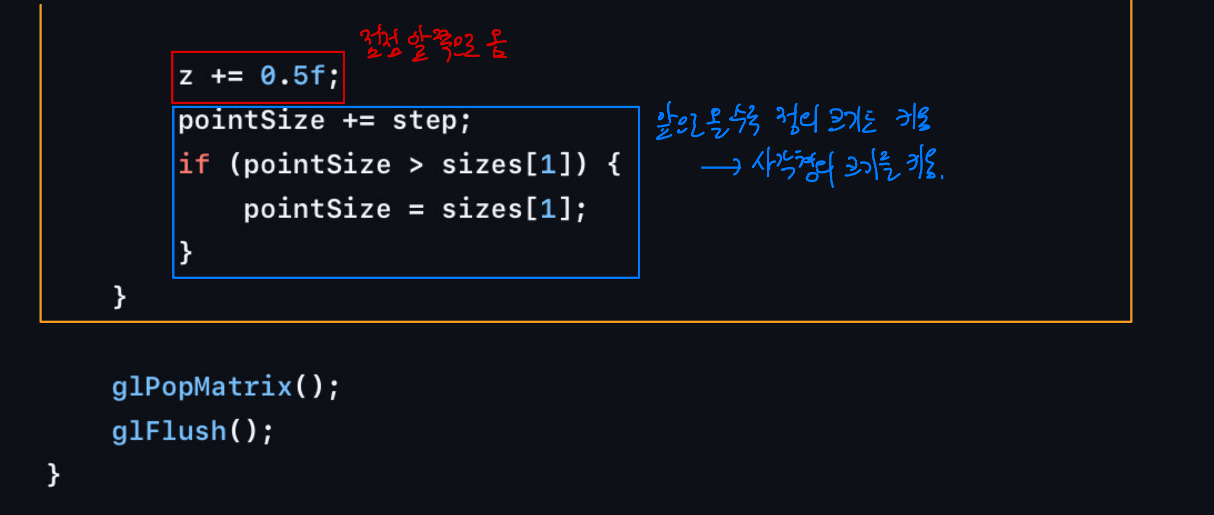
```
        glColor3f(R, G, B);
        glPointSize(pointSize); // 점 크기
        glBegin(GL_POINTS);
```

5페이지 참고

```
        // 상단, 하단
        for (float dx = -halfSize; dx <= halfSize; dx += spacing) {
            glVertex3f(x + dx, y + halfSize, z);  // 상단 변
            glVertex3f(x + dx, y - halfSize, z);  // 하단 변
        }

        // 좌측, 우측
        for (float dy = -halfSize; dy <= halfSize; dy += spacing) {
            glVertex3f(x - halfSize, y + dy, z);  // 좌측 변
            glVertex3f(x + halfSize, y + dy, z);  // 우측 변
        }

        glEnd();
```



```
        z += 0.5f;
        pointSize += step;
        if (pointSize > sizes[1]) {
            pointSize = sizes[1];
        }
    }

    glPopMatrix();
    glFlush();
}
```

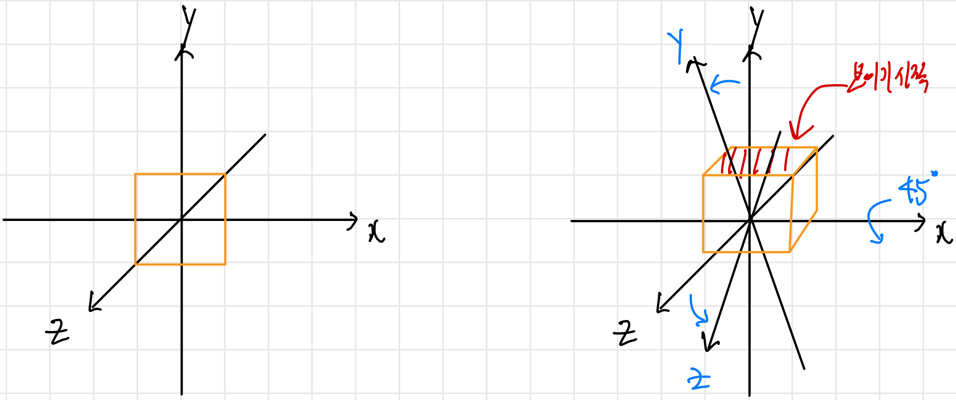
점점 앞쪽으로 옴

앞으로 올수록 점의 크기는 키움

→ 사각형의 크기를 키움.

glRotatef (angle, x, y, z).

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

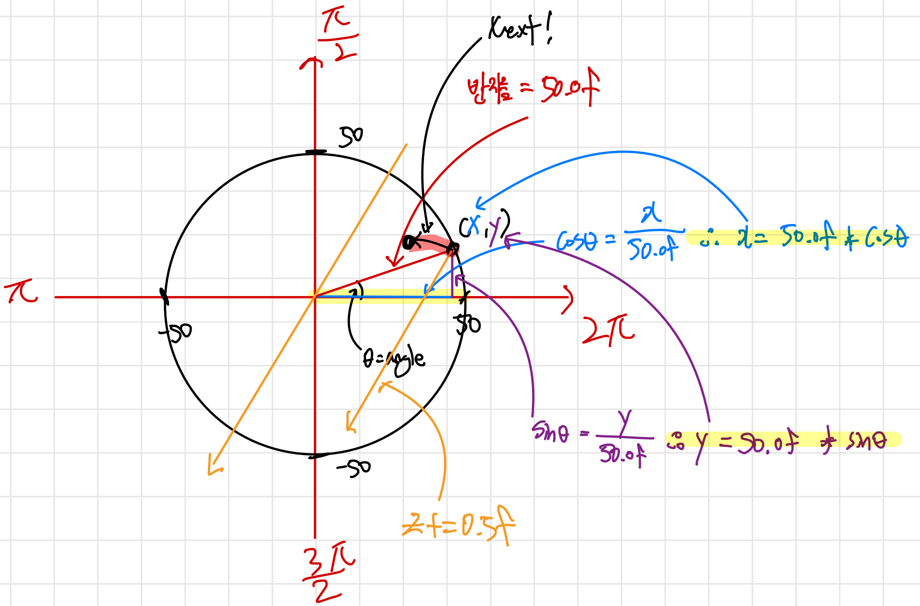


glRotatef (0, 0.0f, 0.0f, 0.0f);

---

for (angle = 0.0f ; angle <= (2.0f * π) * 3.0f ; angle += 0.1f)

* angle은 라디안! 즉 2π가 한바퀴임.   ∴ 3바퀴를 그리겠다는 뜻



$\cos\theta = \frac{x}{50.0f}$ ∴ x = 50.0f * cosθ

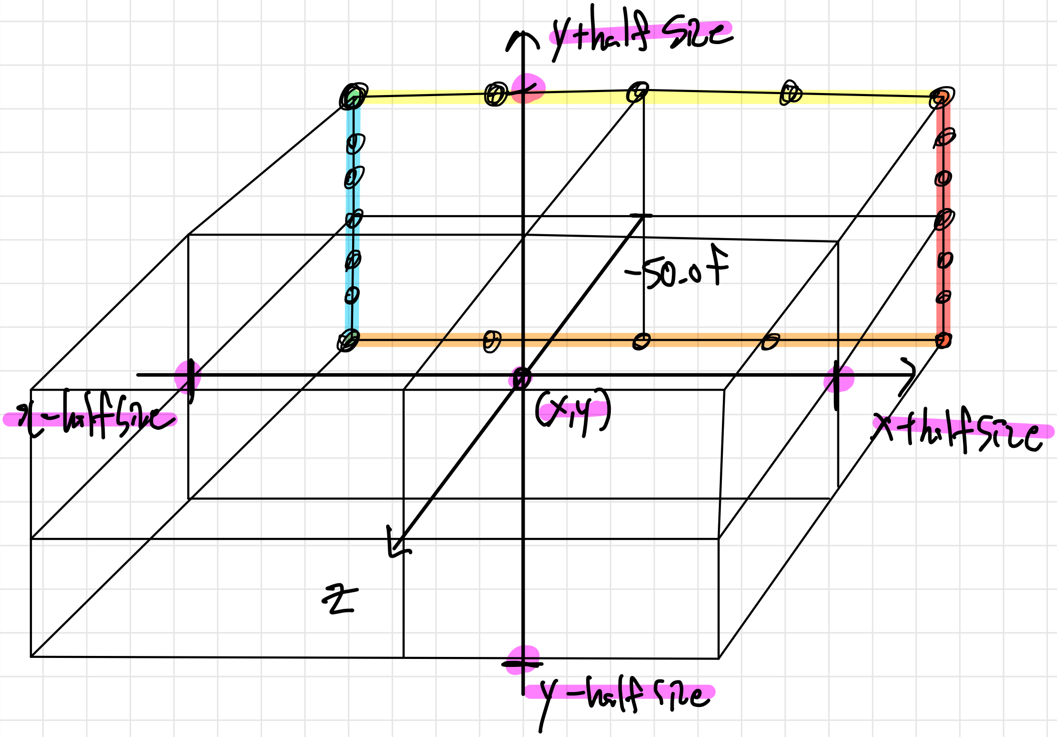
$\sin\theta = \frac{y}{50.0f}$ ∴ y = 50.0f * sinθ

```
for (float dx = -half size; dx <= half size; dx += spacing)
```

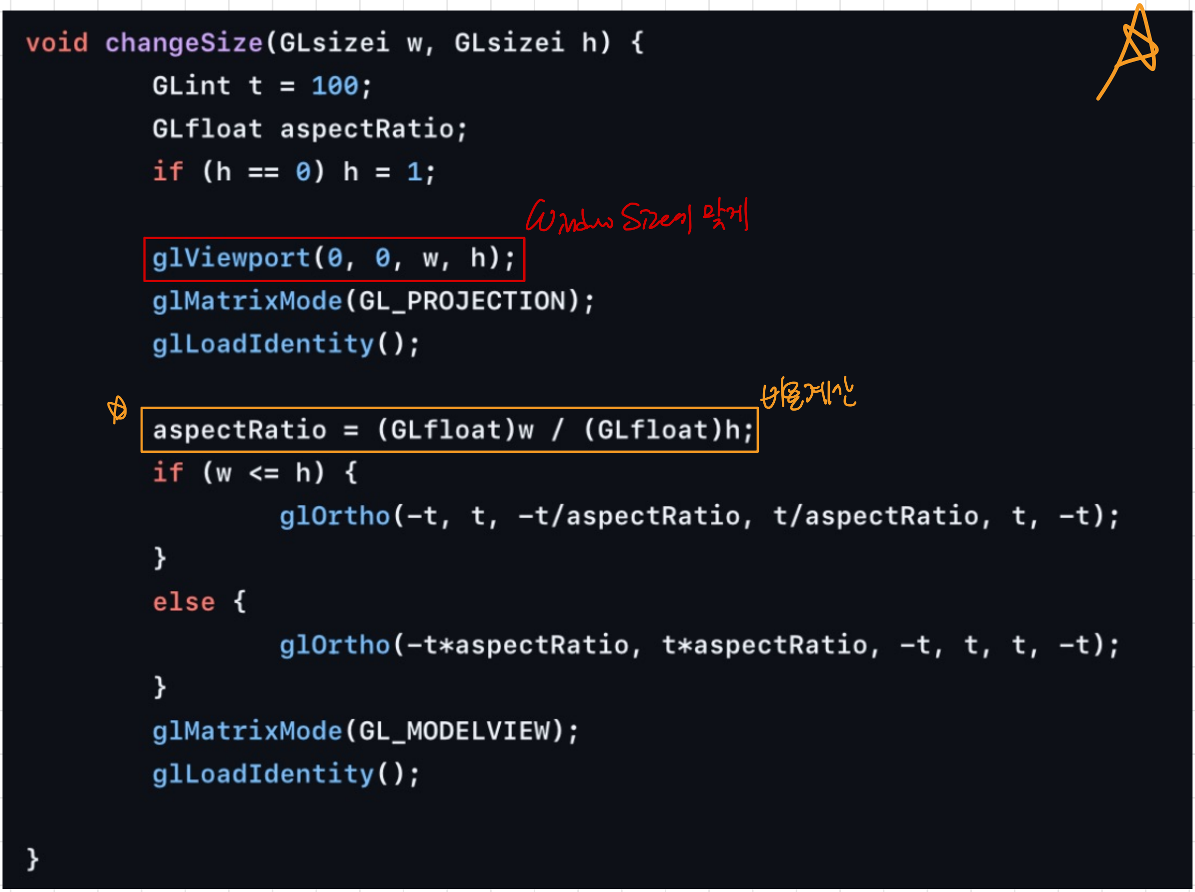
※ 점으로 정사각형을 그리기 위해 for을 구동.

∴ 점간격

```
glVertex3f(x+dx, y+half size, z)
glVertex3f(x+dx, y-halfsize, z)
```



```
glVertex3f (x-halfsize, y+dy, z);
glVertex3f (x+halfsize, y+dy, z);
```



```
void changeSize(GLsizei w, GLsizei h) {
        GLint t = 100;
        GLfloat aspectRatio;
        if (h == 0) h = 1;
```

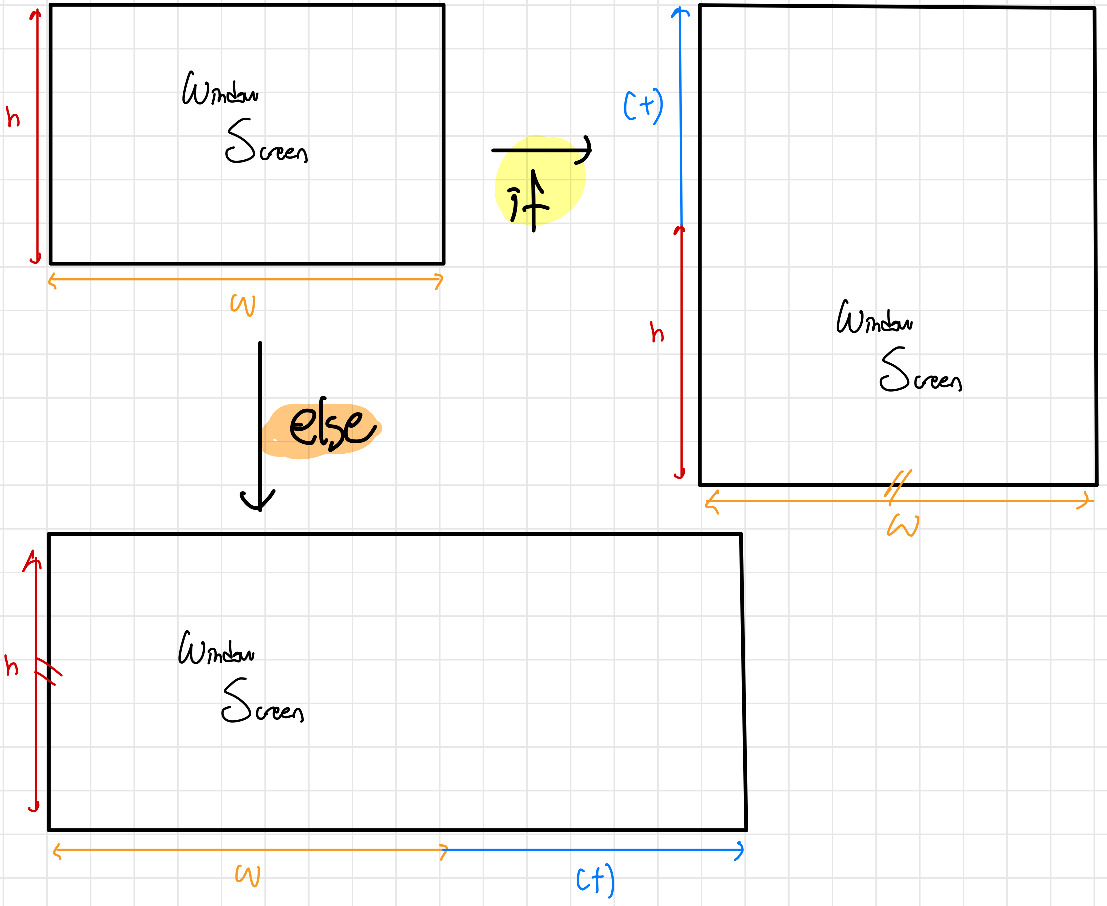
Window Size에 맞게

```
        glViewport(0, 0, w, h);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
```

비율계산

```
        aspectRatio = (GLfloat)w / (GLfloat)h;
        if (w <= h) {
                glOrtho(-t, t, -t/aspectRatio, t/aspectRatio, t, -t);
        }
        else {
                glOrtho(-t*aspectRatio, t*aspectRatio, -t, t, t, -t);
        }
        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

}
```



가) True { 세로가 넓어졌으니, 투영을 세로 쪽으로 조금 더 확장

$-w : w$ $-h : h$ $z : -z$

비율은 $-t : t$, $(+) -t \times \frac{h}{w} : (+) t \times \frac{h}{w}$, $t : -t$ 로 맞춤

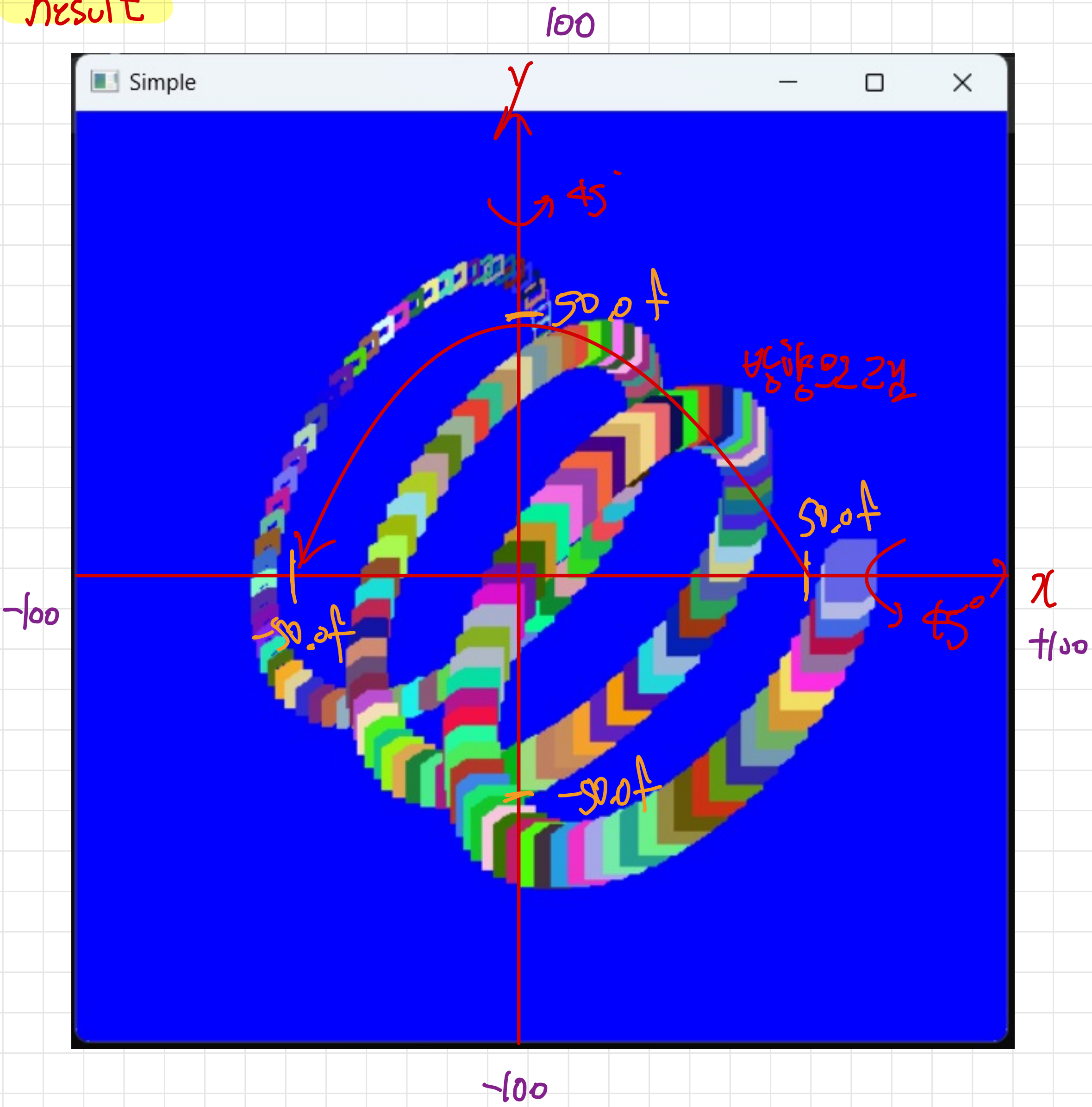
} else { 가로가 넓어졌으니, 투영을 가로 쪽으로 조금 더 확장

비율은 $(+) -t \times \frac{w}{h} : (+) t \times \frac{w}{h}$, $-t : t$, $t : -t$ 로 맞춤

$-w : w$ $-h : h$ $z : -z$

}

Result



glOrtho() 는 glMatrix(GL_PROJECTION) — 여기 존재 —

glMatrix(GL_MODELVIEW)

---

glVertex3f() 는 gl Begin(GL_POINTS) — 여기존재 —

gl End();