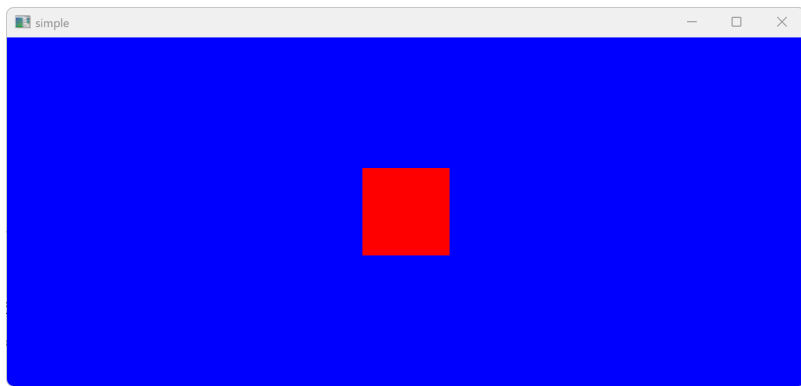
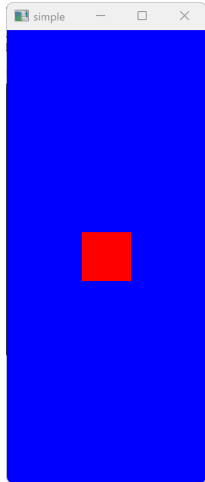


| 2025-04-18_CG_06_예제_창 크기 변화에 따른 모델 비율 유지

| 📁 예제 설명: 창 크기 변화에도 정사각형 비율 유지

창의 크기가 변할 때 사각형이 찌그러지지 않고 비율을 유지하면서 화면에 표시되도록, `ChangeSize` 함수에서 투영 행렬(Projection Matrix)을 설정하기.

| 📄 목표 출력



| 📄 조건 (주어진 코드)

- 창 크기가 변해도 도형 비율이 유지돼야 한다.
- `glOrtho()` 를 종횡비(aspectRatio)에 따라 다르게 설정해야 한다.
- 논리적 좌표계를 기준으로 확대/축소해 비율을 맞춘다.
- `glViewport()` 는 항상 (0, 0, w, h)로 설정한다. (창 전체 크기)

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

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

```

    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(1.0f, 0.0f, 0.0f);
    glRectf(-25.0f, 25.0f, 25.0f, -25.0f);

    glFlush();
}

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

    GLint wSize = 100;
    GLfloat aspectRatio;

    if (h == 0) h = 1;

    glViewport(0, 0, w, h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // ✖ 작성

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void SetupRC(void) {
    glClearColor(0.0f, 0.0f, 1.0f, 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();
}

```

I 📁 해결 코드

I 📄 핵심 코드

```
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);
}
```

I 📄 전체 코드

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

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

    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(1.0f, 0.0f, 0.0f);
    glRectf(-25.0f, 25.0f, 25.0f, -25.0f);

    glFlush();
}

// 🚩 창 크기 변경 콜백 함수
void ChangeSize(GLsizei w, GLsizei h) {
    std::cout << "ChangeSize" << std::endl;

    GLint wSize = 100; // 논리 좌표계 기본 반지름 (기준 단위)
    GLfloat aspectRatio; // 가로/세로 비율(종횡비) 계산용

    if (h == 0) h = 1; // 0으로 나누기 방지

    glViewport(0, 0, w, h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // 🚩
    aspectRatio = (GLfloat)w / (GLfloat)h;
    if (aspectRatio >= 1.0f) { // 가로가 더 길 경우 → x축 확대
```

```

        glOrtho(-wSize*aspectRatio, wSize*aspectRatio, -wSize, wSize, -1, 1);
    }
    else { // 세로가 더 길 경우 → y축 확대
        glOrtho(-wSize, wSize, -wSize/aspectRatio, wSize/aspectRatio, -1, 1);
    }

    // 모델뷰 행렬로 전환 및 초기화 (좌표계 안정화)
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void SetupRC(void) {
    glClearColor(0.0f, 0.0f, 1.0f, 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); // 디스플레이 콜백 등록
    glutReshapeFunc(ChangeSize); // 창 크기 변경 시 콜백 등록

    glutMainLoop();
}

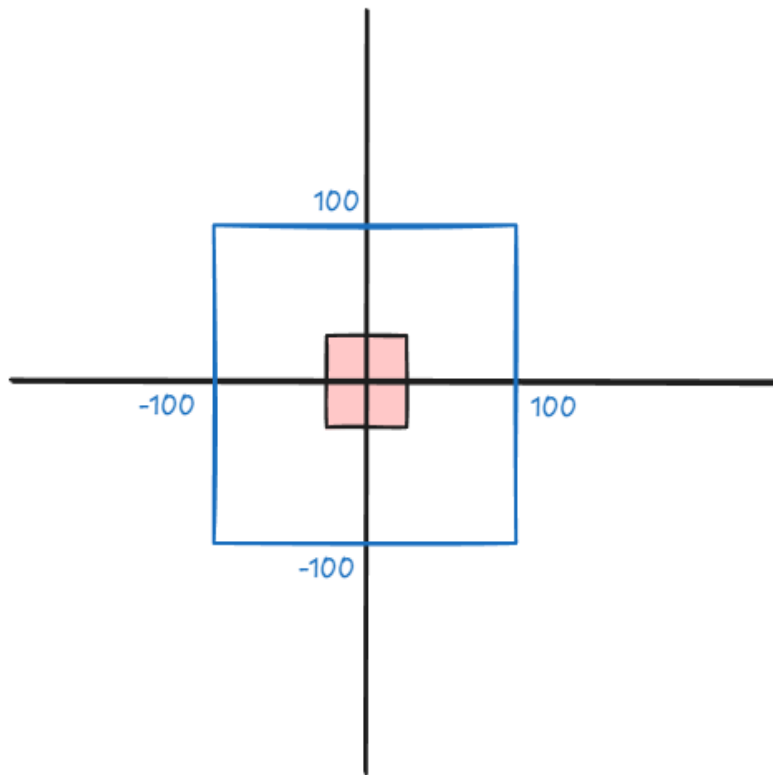
```

📁 설명

📄 `glViewport(0, 0, w, h);`

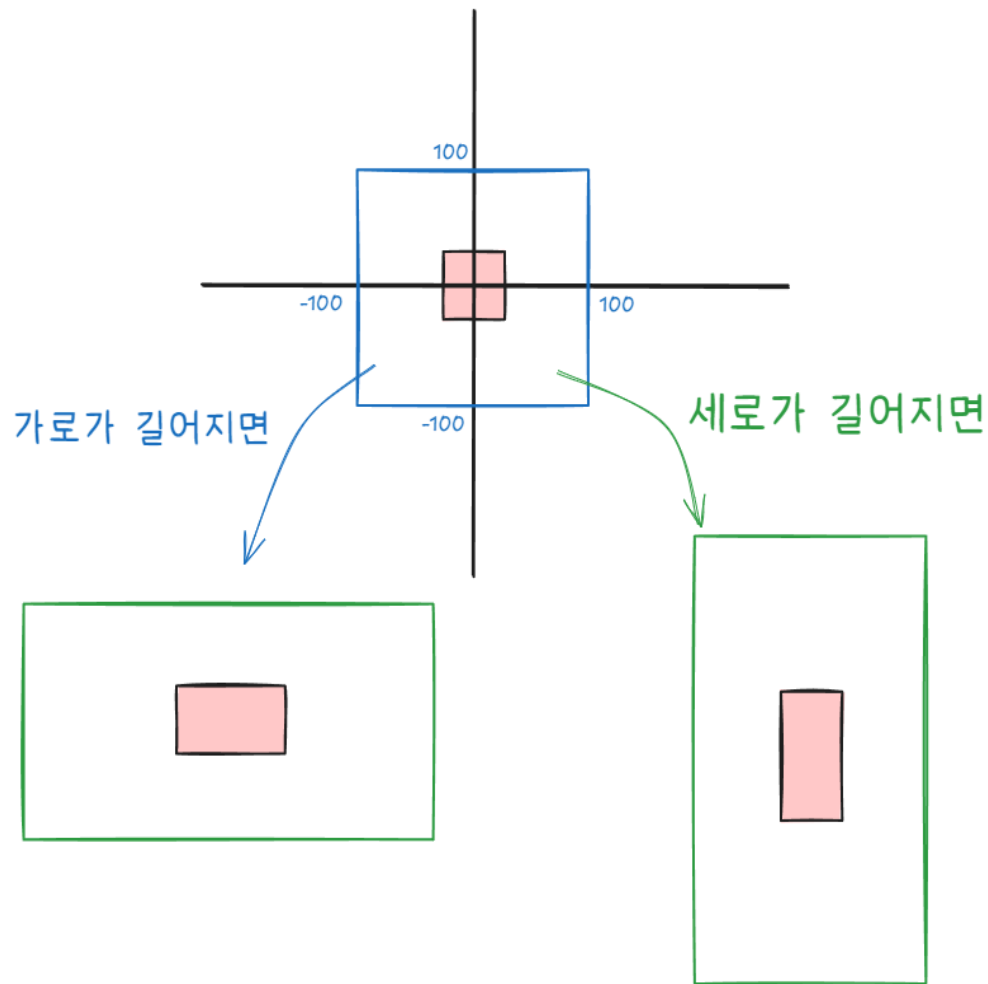
- viewport는 창 크기에 따라 업데이트되며,
 - 만약 `glOrtho`를 고정시킨다면, 창의 종횡비가 바뀔 때 논리 좌표계와 실제 픽셀 비율이 달라져, 정사각형 같은 도형이 찌그러져 보일 수 있다.
 - 쉽게 설명하면, **논리 좌표계**에서 정사각형을 그렸지만, 그림을 실제로 보여주는 종이(=**창의 픽셀 비율**)가 다르면 그 정사각형이 찌그러져 보일 수 있다는 것.
- 따라서, `glOrtho()`를 **종횡비(aspect ratio)에 따라 조정**하지 않으면 도형이 찌그러지게 된다.

📄 `glOrtho(-wSize, wSize, -wSize, wSize, -1, 1)` 고정 이라면?



- 논리 좌표계 상에서는 정사각형이 그려져있다.

! 🤖 가로 또는 세로가 길어지면

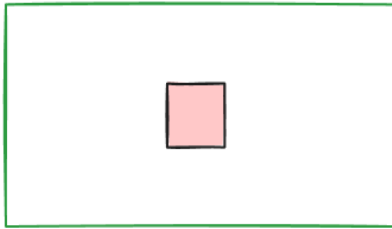
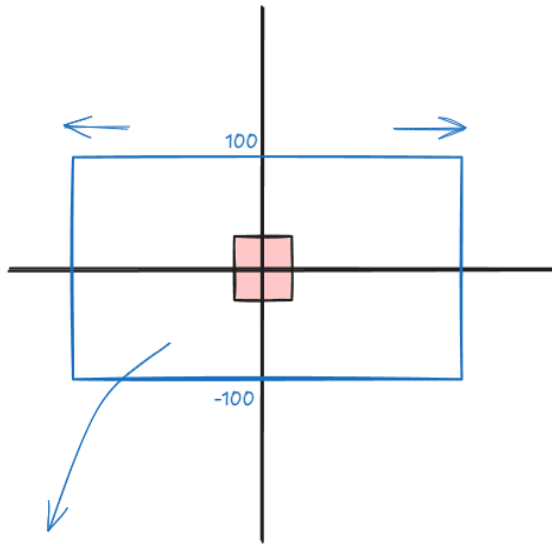


- 논리 좌표계가 정사각형으로 고정되어 있기 때문에, 비율이 다른 viewport에서는 도형이 왜곡되어 보이게 된다.

! 📄 가로가 더 긴 경우 (aspectRatio >= 1.0)

```
glOrtho(-wSize * aspectRatio, wSize * aspectRatio, -wSize, wSize, -1, 1);
```

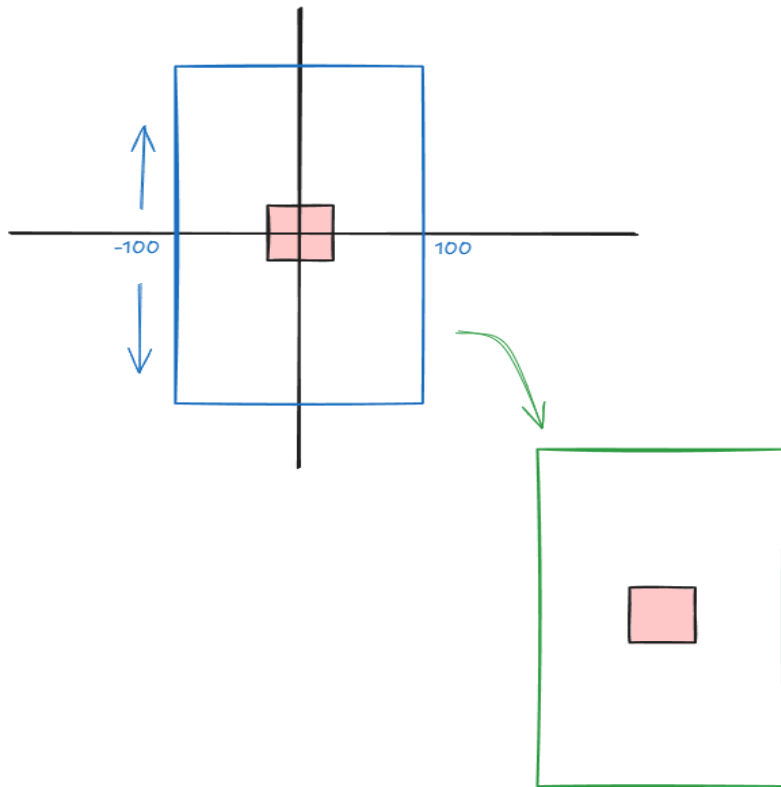
- Y축의 범위는 그대로: $[-100, 100]$
- X축 범위를 늘린다: $[-100 * a, 100 * a]$ ($a = \text{aspect ratio}$)
- 가로 공간이 늘어난 만큼 논리 좌표계도 X축을 넓게 잡아서 **비율을 보정**한다.



| ✎ 세로가 더 긴 경우 ($\text{aspectRatio} < 1.0$)

```
glOrtho(-wSize, wSize, -wSize / aspectRatio, wSize / aspectRatio, -1, 1);
```

- X축 범위는 그대로: $[-100, 100]$
- Y축 범위를 늘린다: $[-100 / a, 100 / a]$ ($a = \text{aspect ratio}$)
- 세로 공간이 늘어난 만큼 논리 좌표계도 Y축을 넓게 잡아서 **비율을 보정**한다.



I 📁 추가 설명

I 📄 glutReshapeFunc()란?

```
glutReshapeFunc(ChangeSize);
```

- 창 크기가 바뀔 때마다 호출될 콜백 함수를 지정하는 함수.
- 주로 `glViewport()` 와 `glOrtho()` 를 다시 설정해서 도형이 찌그러지지 않도록 한다.

구분	<code>glutDisplayFunc()</code>	<code>glutReshapeFunc()</code>
역할	화면을 그릴 때 호출	창 크기 변경 시 호출
함수 내부	도형 렌더링 (<code>glBegin</code> , <code>glEnd</code> , <code>glDraw</code> , <code>glFlush</code>)	뷰포트, 투영 설정 (<code>glViewport</code> , <code>glOrtho</code>)

I 📄 ChangeSize() 끝의 두 줄의 필요성

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
```

- 이 코드는 행렬모드를 다시 모델뷰 행렬로 전환하고, 초기화한다.

- `glMatrixMode(GL_MODELVIEW)` 로 돌려놓지 않으면, 나중에 도형을 그릴 때 모든 좌표 변환이 Projection 행렬에 쌓일 수 있다.
 - 따라서 투영 행렬 변경 이후, 모델뷰 행렬도 초기화해서 **안정적인 상태로 되돌리는 것이 좋다.** (관례)

! `ChangeSize(GLsizei w, GLsizei h)`의 매개변수 `w`, `h`

- GLUT이 창의 크기가 바뀔 때 **새로운 창의 크기**를 전달해준다.
- GLUT 내부에서 창의 크기가 바뀔 때 → 자동으로 `ChangeSize()` 를 호출하며, **현재 창의 너비와 높이**를 `w`, `h` 에 넣어준다.