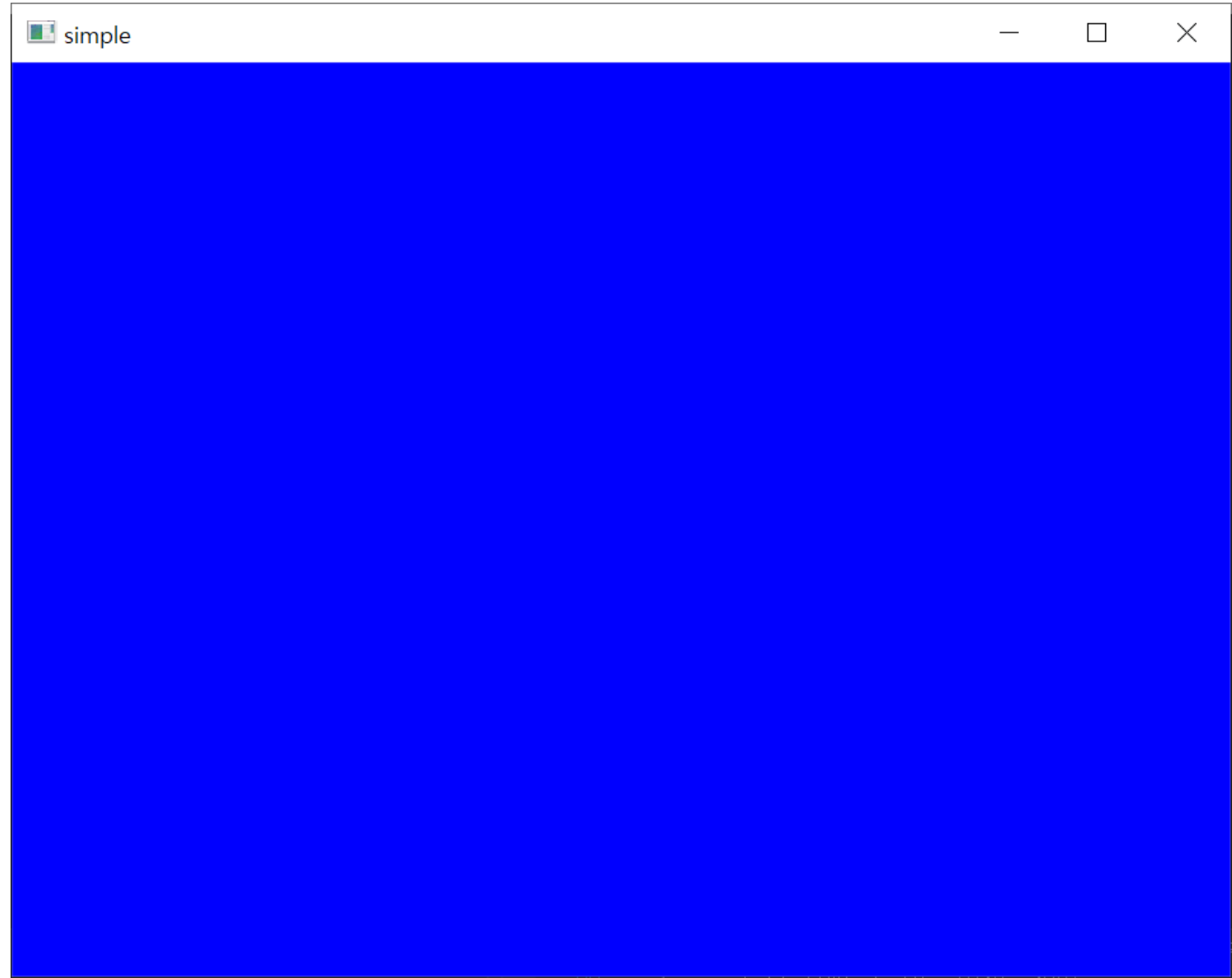




simple(2-1)

- 파란색으로 채워진 창 출력하기

- code





simple(2-1)

- `glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);`
 - 싱글 버퍼 창, RGBA 색상 모드 사용
 - 싱글 버퍼 창
 - ✓ 모든 드로잉 명령이 출력 창에서 진행
 - 더블 버퍼 창
 - ✓ 드로잉 명령이 오프 스크린 버퍼에서 실행된 후, 출력 창으로 옮겨지는 방식으로 진행
 - ✓ 애니메이션 효과를 낼 때 주로 사용
- `SetupRC();` ➔ 정해진 이름 아님..
 - 렌더링이 이루어지기 전에 필요한 OpenGL 초기화
- `glutMainLoop();`
 - GLUT 프레임웍 실행
 - 프로그램이 종료될 때까지 운영체제에 관련된 메시지, 키 입력 등을 처리



simple(2-1)

- `glClearColor(0.0f, 0.0f, 1.0f, 1.0f);`
 - 화면을 어떤 색으로 지울지 결정
 - 실제로 화면을 지우기 위해서는 **`glClear` 호출**
- `glClear(GL_COLOR_BUFFER_BIT);`
 - 가장 최근에 그려진 이미지를 지움 : Buffer...
- `glFlush();`
 - 드로잉 명령을 전달, 명령을 강제로 실행하여 즉시 렌더링
 - 주로 `GLUT_SINGLE` (단일 버퍼)인 경우 사용
- `glutDisplayFunc(func)`
 - `func`함수를 호출하는 콜백함수
 - 창의 이동/크기변경 될 때마다 호출 됨



simple

Try these...

- 1) 빨간색 배경
- 2) 녹색 배경
- 3) 노란색 배경



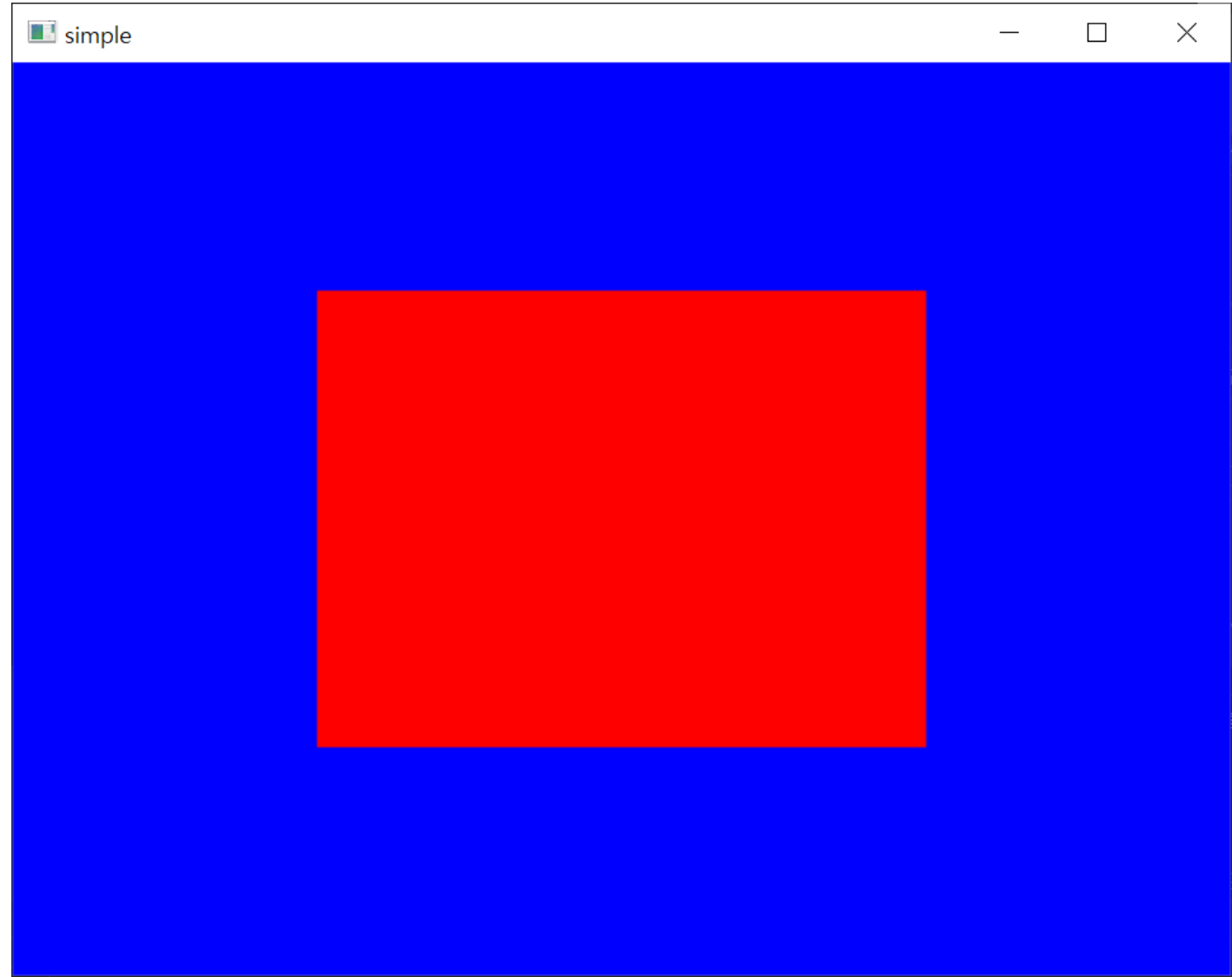
simple(2-1), error

- 출력창 초기 실행 시, 검정색만 출력(초기 셋팅값=파란색)
or, 흰색 화면만 출력
 - 가정 : gpu 혹은 os 종류 및 버전에 따라 더블 버퍼를 기본값으로 사용하는 경우
 - 해결
 - ✓ `glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);` ➔ `glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);`
 - ✓ `glFlush();` ➔ `glutSwapBuffers();`



도형 그리기(2-2-1)

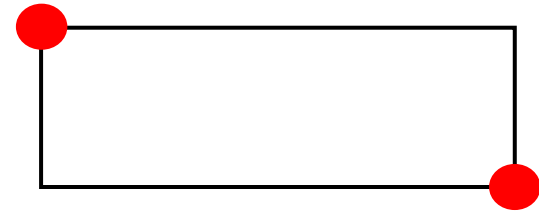
- 중간에 사각형 그리기
- 창 크기 변경 없는 경우
- **glViewport() 없음**
- code





도형 그리기(2-2-1)

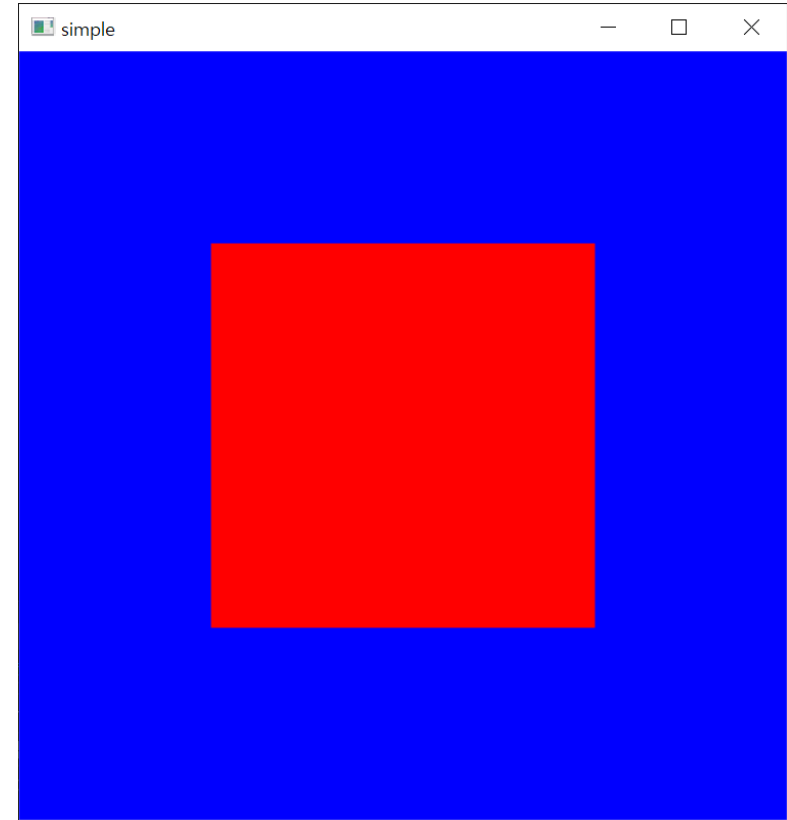
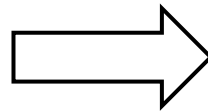
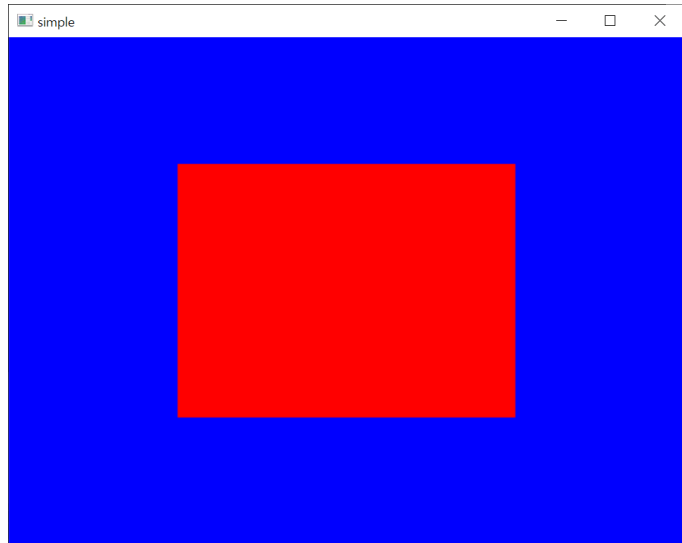
- `glColor3f(1.0f, 0.0f, 0.0f);`
 - 드로잉 색상을 적색으로 설정
- `glRectf(-0.5f, 0.5f, 0.5f, -0.5f);`
 - 사각형 그려주는 함수
 - (왼 위 x, 왼 위 y, 오른 아래 x, 오른 아래 y)
- 일반적으로 직교 투영으로 $[-1,1]$ 범위 내에서 작동
 - `glOrtho(-1, 1, -1, 1, 1, -1)`





도형 그리기(2-2-1)

- 그런데, 난 정사각형을 그렸는데... ——;;
 - `glRectf(-0.5f, 0.5f, 0.5f, -0.5f);`



How?

Try



도형 그리기(2-2-1)

■ solution

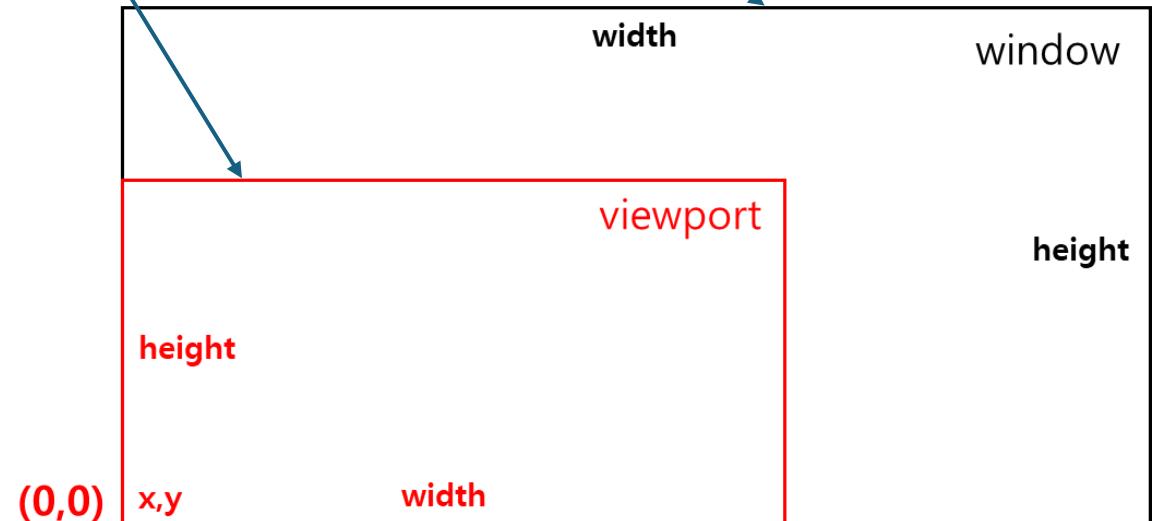
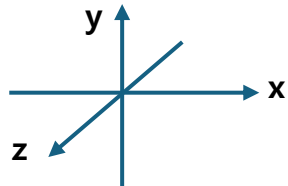
- `glutInitWindowSize(500, 500)`으로 변경
- `glViewport()`의 default값은 `glutInitWindowSize()`와 같음



View(2-2-2)

- `glutInitWindowSize(500, 500)`
- **`glViewport(0, 0, 500, 500)`**
 - `glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`
 - ✓ (x, y) : 원 아래 점
 - 일반적으로 `glutInitWindowSize`와 동일하게 많이 설정
- projection할 공간의 영역은 viewport 크기에 맞춰서 출력

- 기본 3D좌표계에서 drawing하고 projection 공간을 설정하면, 그 공간이 viewport를 거쳐서 window에 출력됨

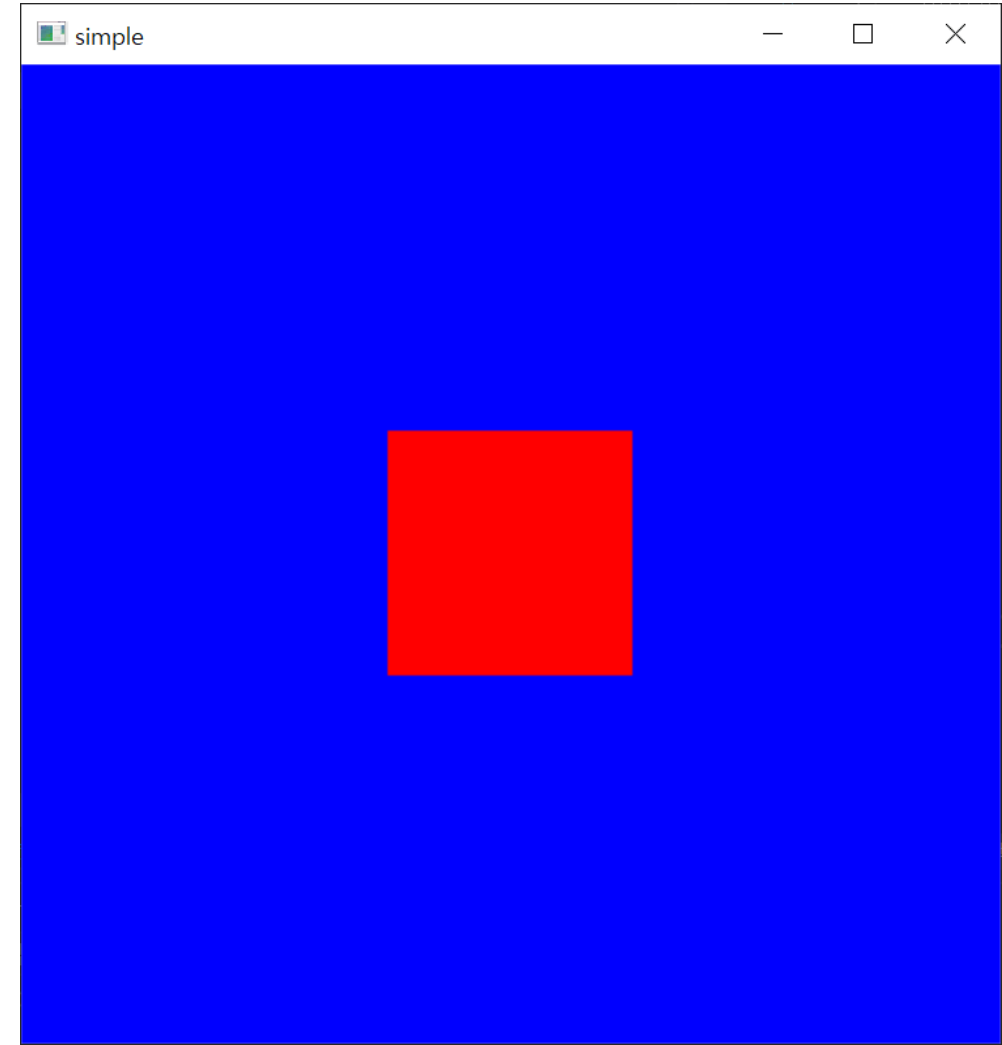




View(2-2-2)

- `glutInitWindowSize(500, 500), glViewport(0, 0, 500, 500)`의 값은 fix 시키고 다음과 같은 크기의 정사각형 그리기
 - 한 변의 길이가 창 한 변의 길이의 $1/4$

- code





View(2-2-2)

- `glMatrixMode(GL_PROJECTION);` → 카메라/뷰포트 설정
 - 3D 좌표를 2D 화면에 매핑하는 과정
 - `glOrtho()`, `gluPerspective()` 등을 사용하여 카메라의 시야(FOV) 설정
 - `glMatrixMode(GL_MODELVIEW);` → 객체 변환(이동, 회전, 크기 변경)
 - `glTranslatef()`, `glRotatef()`, `glScalef()` 등을 사용하여 모델(도형)의 위치 및 크기 변환
- ➔ 일반적으로 투영 행렬 (`GL_PROJECTION`)을 먼저 설정한 후, 모델뷰 행렬 (`GL_MODELVIEW`)을 설정



View(2-2-2)

■ glutInitWindowSize

- 실제 화면 창 크기 설정
- 모니터의 픽셀 단위 기준

■ glViewport

- 내가 실제 그리는 도화지의 크기 설정

■ glOrtho(2D인 경우 gluOrtho2D - glut지원X) / gluPerspective

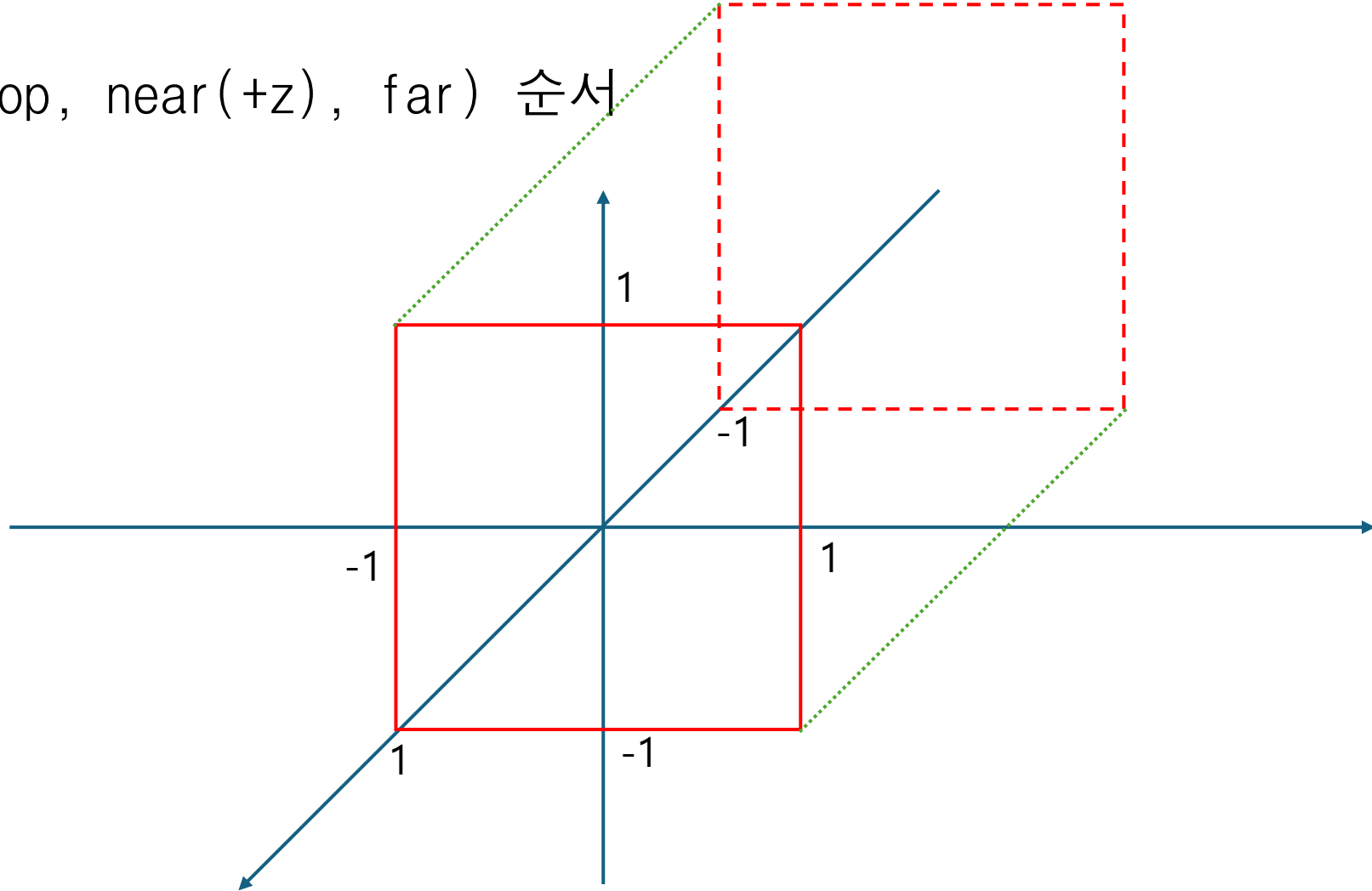
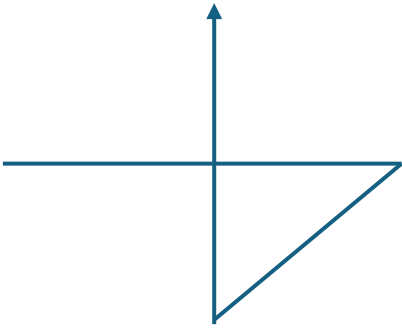
- projection할 공간 영역 설정(카메라 뷰)
- 2D인 경우 : glutInitWindowSize= glViewport= glOrtho2D 이면 모니터 픽셀 단위가 됨



View(2-2-2)

■ `glOrtho(-1, 1, -1, 1, 1, -1)`

- 카메라 뷰
- (left, right, bottom, top, near(+z), far) 순서
- viewport 크기에 맞춰짐





View(2-2-2)

■ 해보자

- viewport는 window size와 동일하게..
- glRectf(0, 240, 320, 0)으로..

■ glOrtho()를 수정

```
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);  
glutInitWindowSize(640, 480);  
glutInitWindowPosition(500, 500);  
glutCreateWindow("simple");
```

