

Computer Graphics Assignment #02

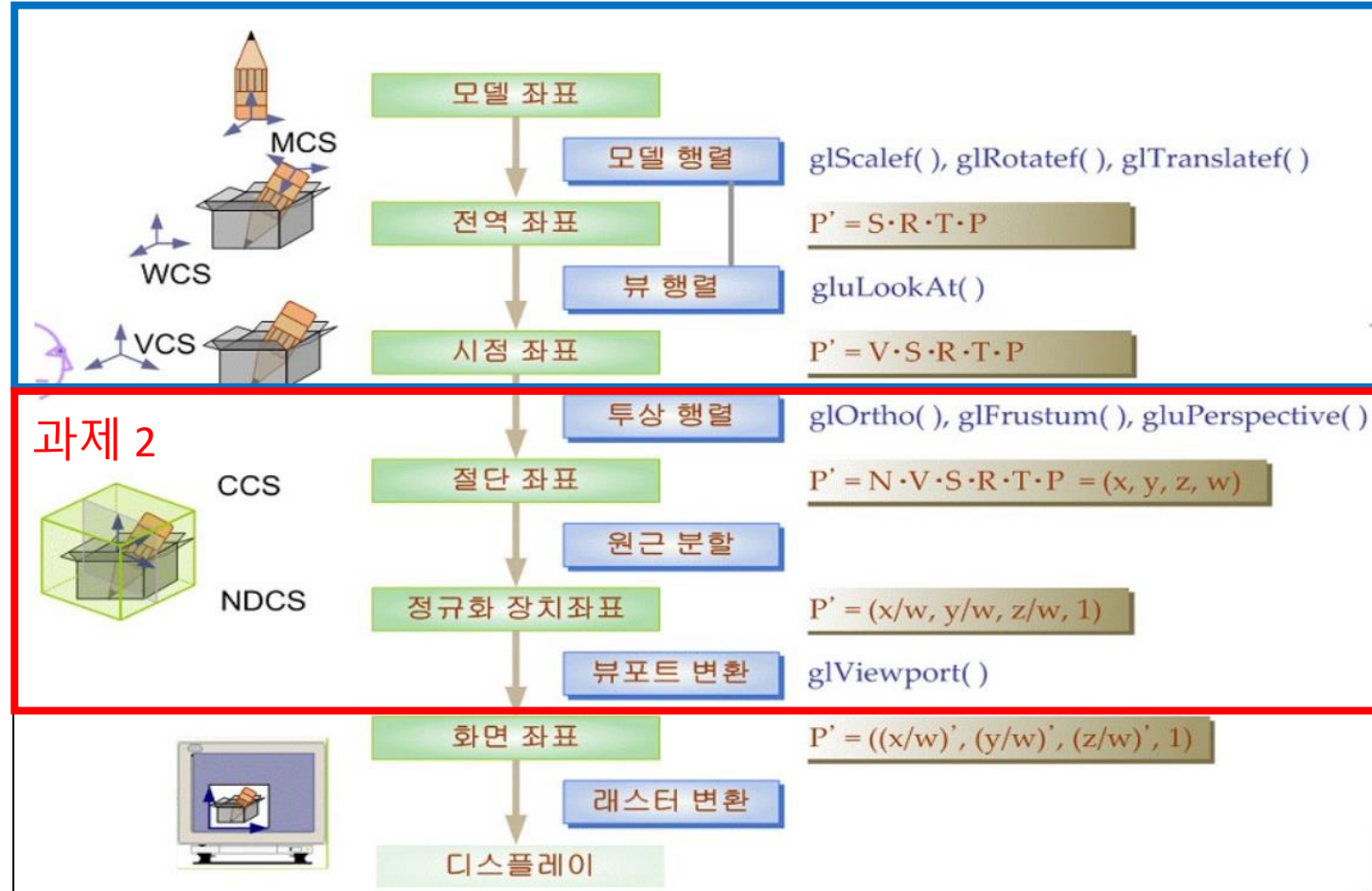
- Projection Transformation
- Light System
- Input/Output functions

과제 안내

• 과제 2의 목표

1. 투영/뷰포트 변환 실습
2. 광원 시스템의 이해
3. 입출력 기능 구현

과제 1



과제 제출 방식

1. 제출물

1. 실행코드 (.cpp, python, etc.)
2. 실행 결과를 캡처한 .pdf 파일

2. 제출하는 코드에는 모두 주석이 달려있어야 함

1. `Int main(void); return 0;` 등의 기초 라인들은 제외해도 무방

3. 추가 라이브러리 존재 시, 주석으로 설명

4. 제한기간 내, LMS 시스템에 제출

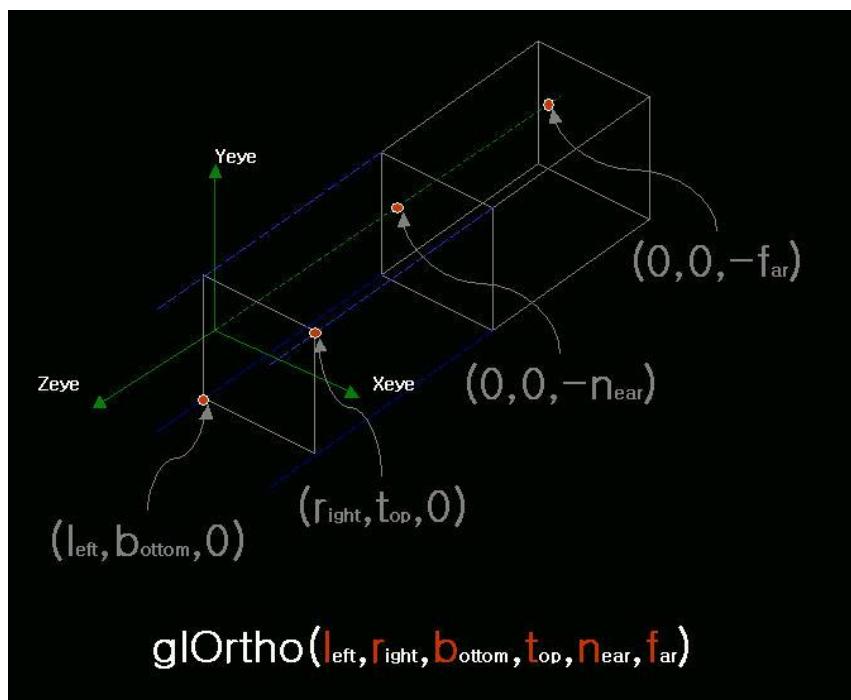
문제 1

- Projection Transformation

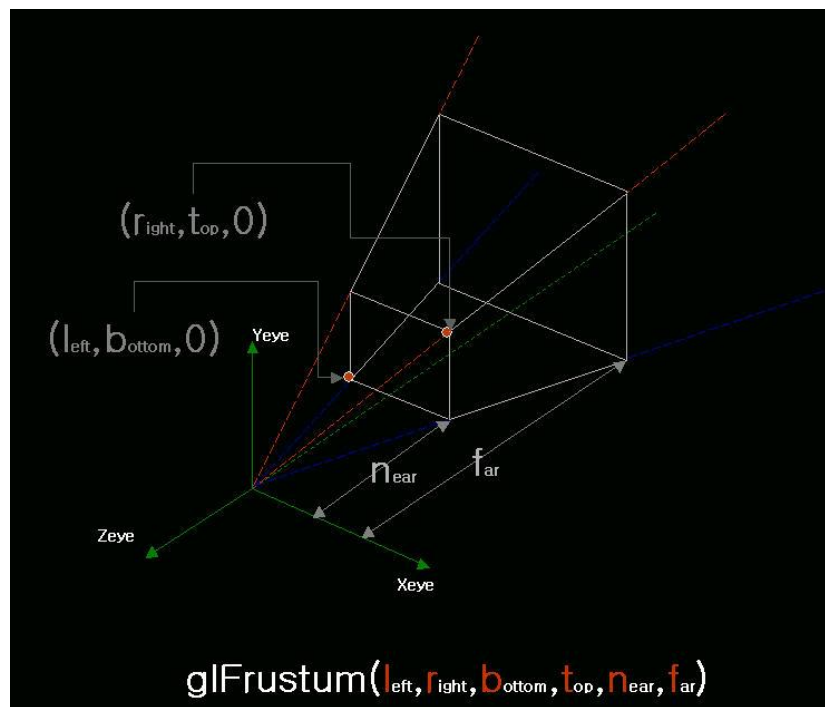
문제 1 - 투영 변환

- OpenGL에는 크게 두가지 투영 변환이 존재함.

- 직교투영

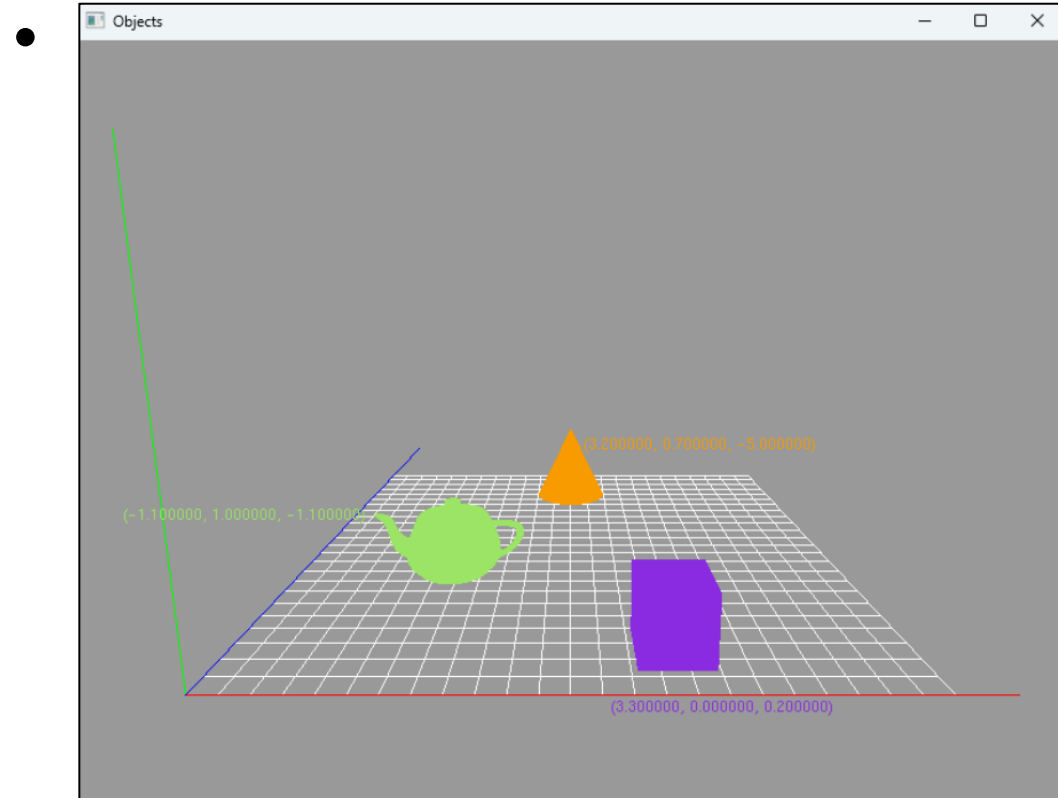


- 원근투영



문제 1 – 투영 변환

- 과제 1에서는 원근투영으로 진행되었음.

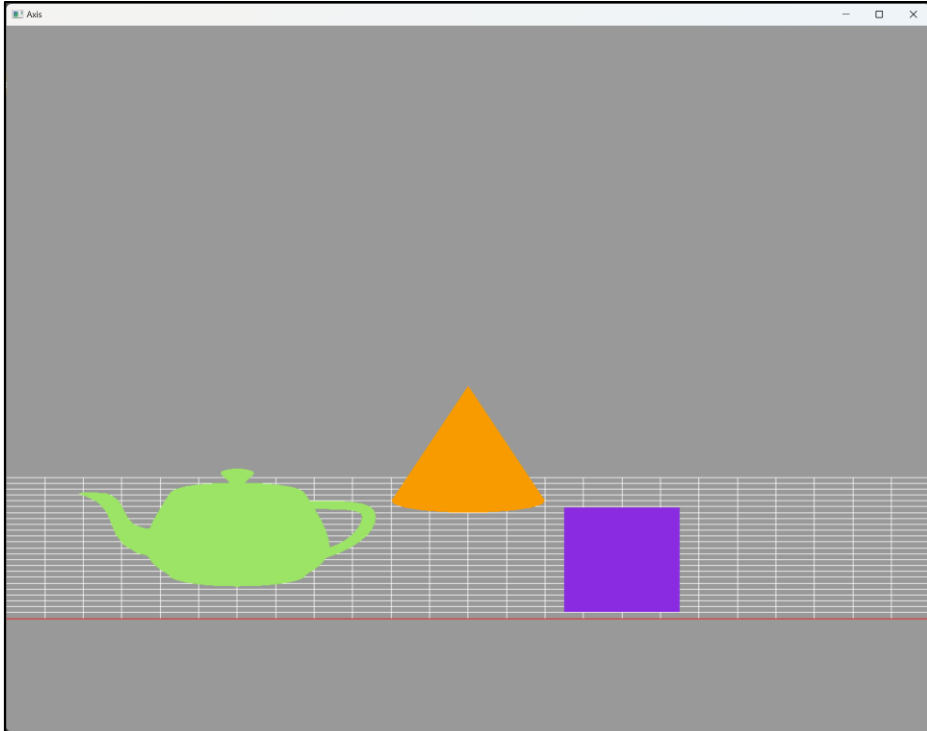


```
glLoadIdentity();  
gluPerspective(60.0, 4.0 / 3.0, 1.0, 15.0);
```

*gluPerspective
: glFrustum를 편하게 사용할 수 있는 glu 함수

문제 1 – 투영 변환

- 과제 1의 6번 코드의 gluPerspective() 함수를 gluOrtho() 함수를 적절히 사용하여 아래와 같은 결과를 만들 것.

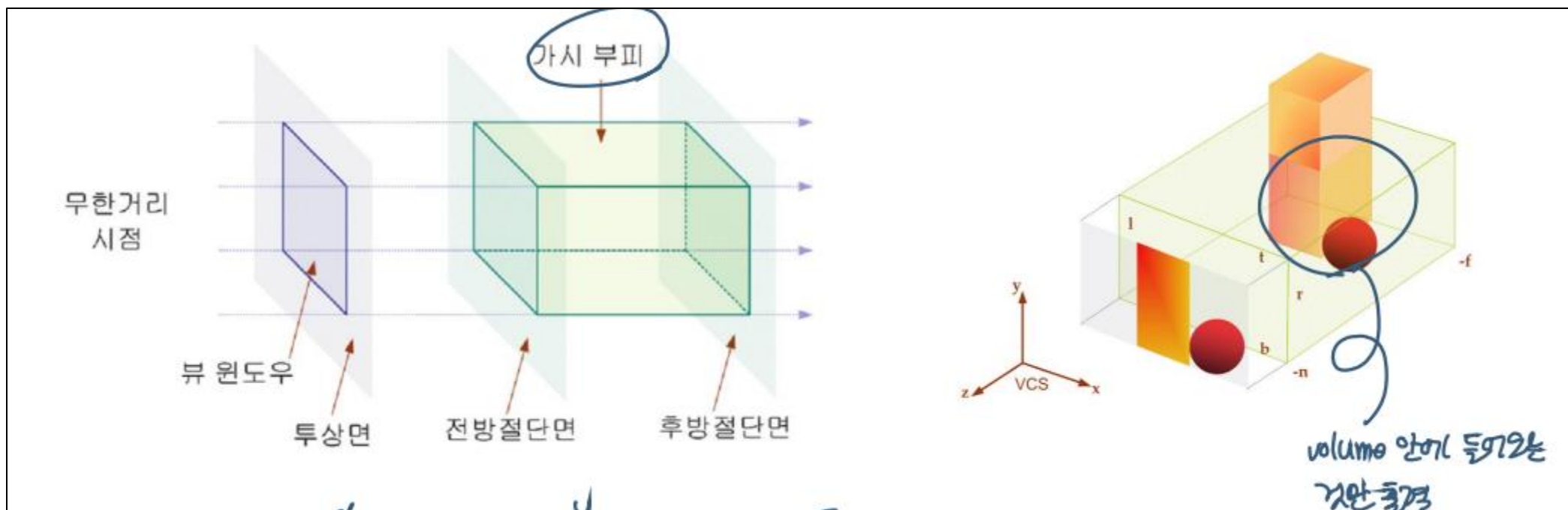


문제 2

- Projection Transformation : **Viewing Volume**

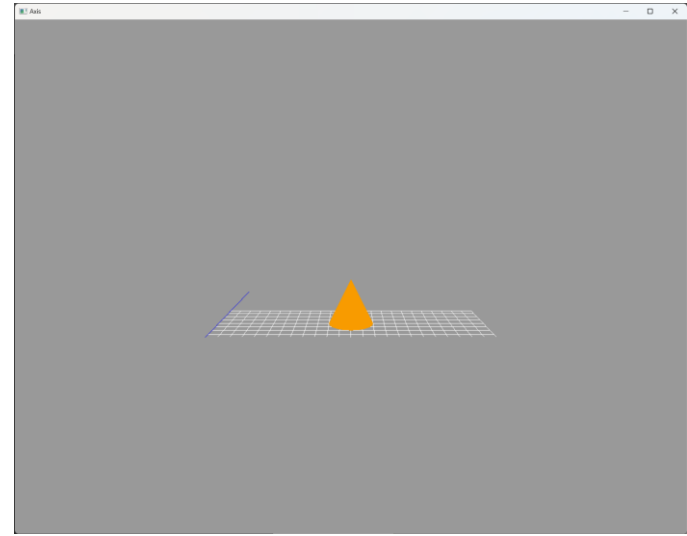
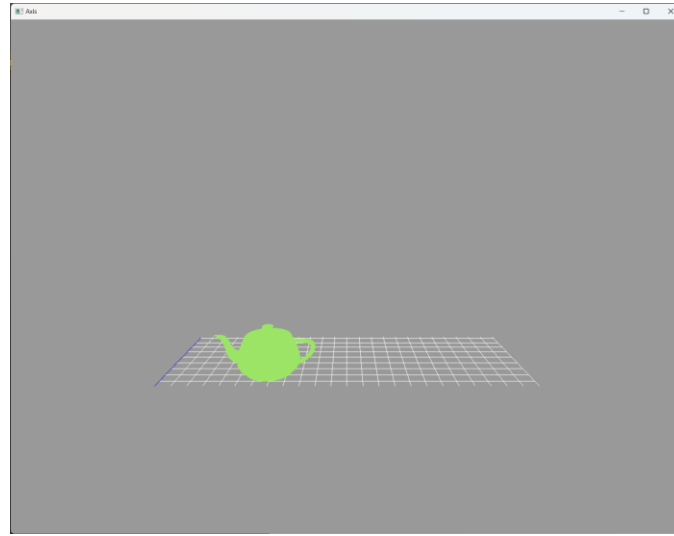
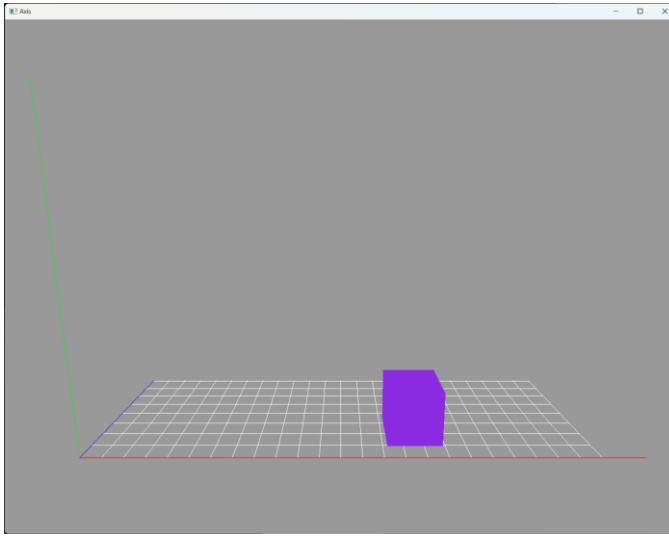
문제 2 – 뷰잉 볼륨

- 투영 변환 함수는 가시 부피(카메라가 보는 공간)를 정의할 수 있음



문제 2 – 뷰잉 볼륨

- 3가지의 물체를 적절한 위치에 옮긴 후, gluPerspective() 혹은 glOrtho() 함수로 예시를 참고하여 물체를 각각 따로 보일 수 있게 할 것



문제 3

- Viewport Transformation

문제 3 – 뷰포트 변환

- 뷰포트 변환(viewport transformation)은 이전 단계인 투영 변환의 결과를 이용하여 출력 장치 좌표계로 매핑하여 실제 장치에 출력된 2D 영상을 만들어내는 단계이다.
- 뷰포트 변환을 이용하면 투영변환의 결과를 생성된 윈도우 내에 자유롭게 위치시킬 수 있다.

문제 3 – 뷰포트 변환

- 참고) 강의 자료 1



뷰포트 변환

- ❖ 정규화 장치좌표계(NDCS: Normalized Device Coordinate System)
 - ❖ 절단 이후 원근분할에 의해 물체 정점을 3차원 좌표로 변환한 것
 - ❖ $(x', y', z', 1) = (x/w, y/w, z/w, 1)$
- ❖ 뷰포트 변환(Viewport Transformation)
 - ❖ 정규화 장치좌표계에서 화면 좌표계로 가는 작업
 - ❖ 화면 좌표계(SCS: Screen Coordinate System), 뷰포트 좌표계(Viewport Coordinate System), 윈도우 좌표계(Window Coordinate System)



(a) (b)



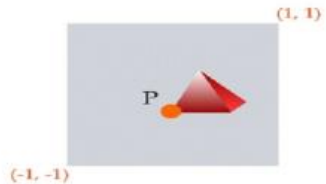
Kyonggi University, Computer Graphics Lab. 56

문제 3 – 뷰포트 변환

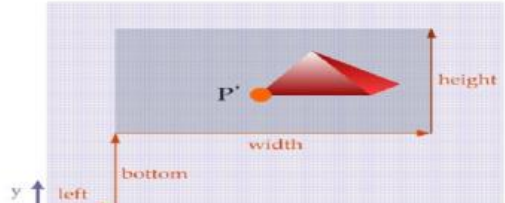
- 참고) 강의 자료 2

KGU
뷰포트 설정

❖ void glViewport(GLint left, GLint bottom, GLsizei width, GLsizei height):



(a)




(b)

$$x' = \frac{x - (-1.0)}{(1.0) - (-1.0)} \times width + left$$

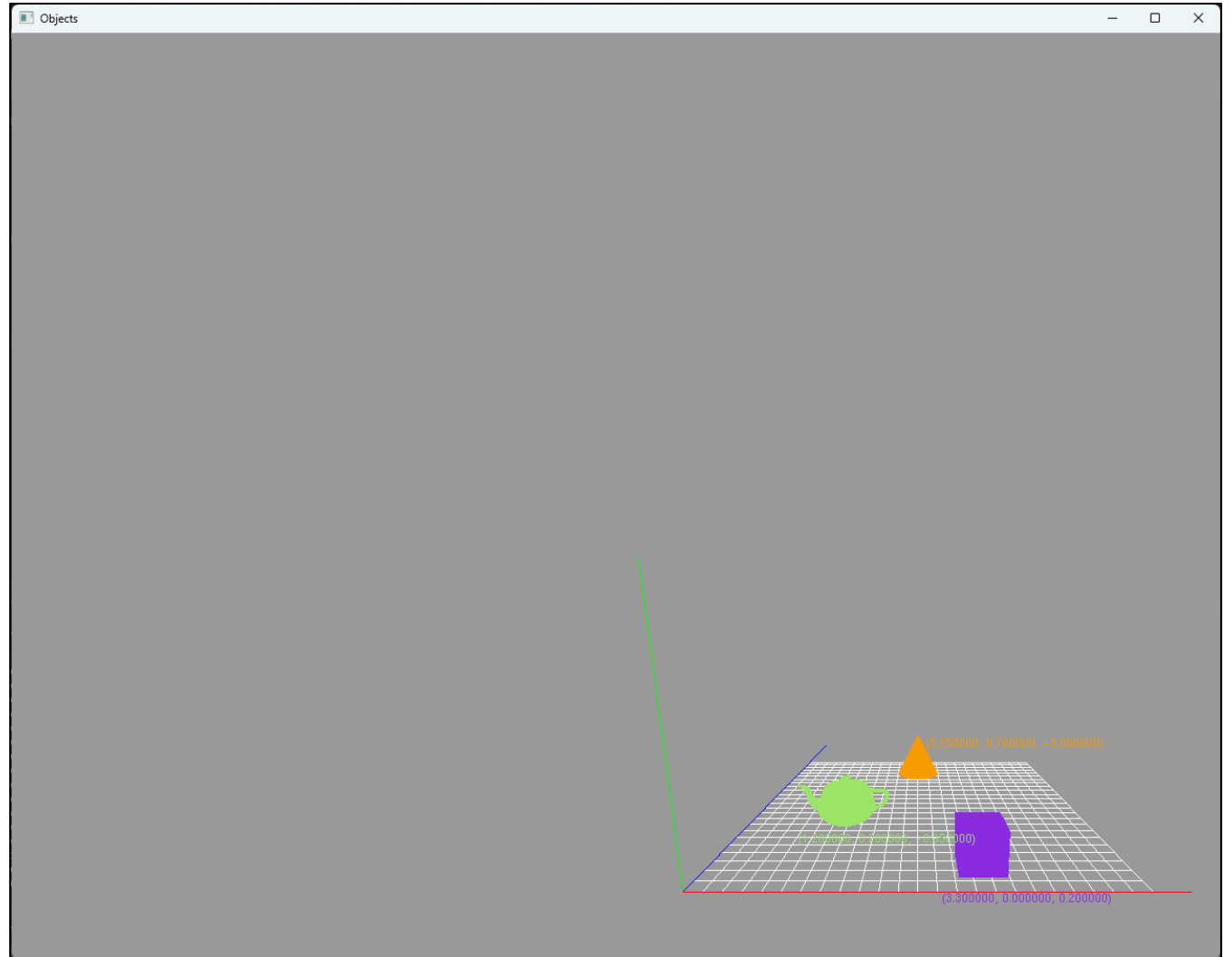
$$y' = \frac{y - (-1.0)}{(1.0) - (-1.0)} \times height + bottom$$

$$P' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{width}{2} & 0 & 0 & left + \frac{width}{2} \\ 0 & \frac{height}{2} & 0 & bottom + \frac{height}{2} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



문제 3 – 뷰포트 변환

- (이전 문제 코드 계속)
display() 함수 내에
glViewport()를 사용하여
임의의 위치에
컨텐츠 표시
(예시 뷰포트 : 우하단)

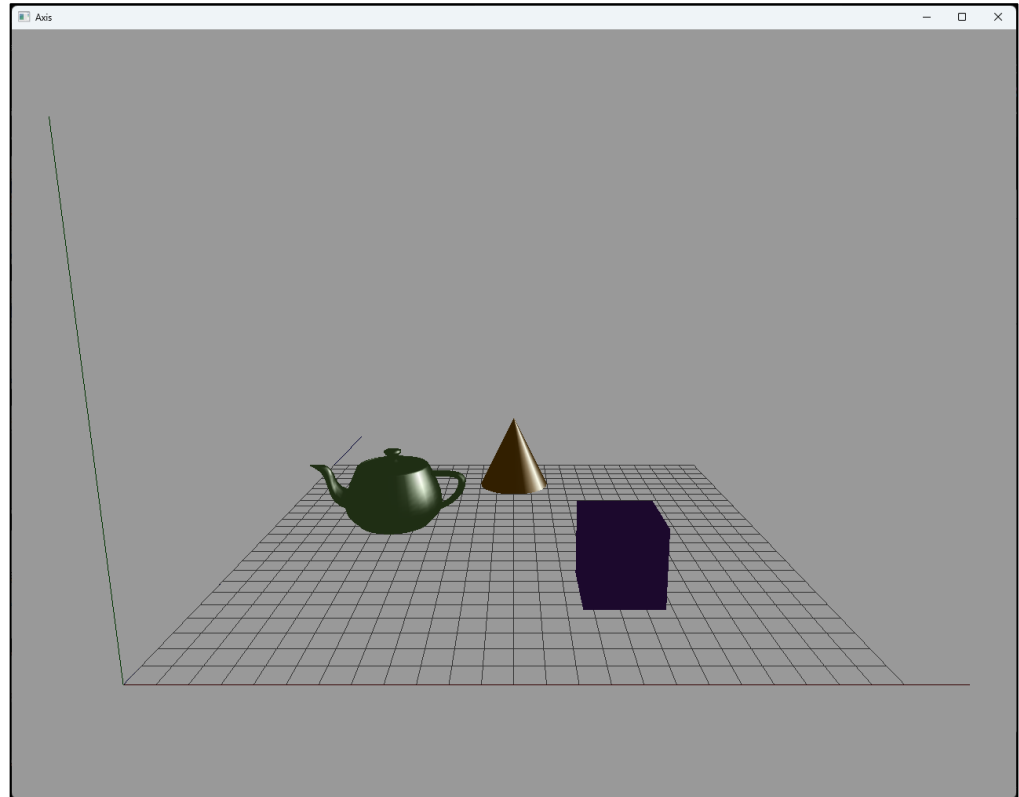
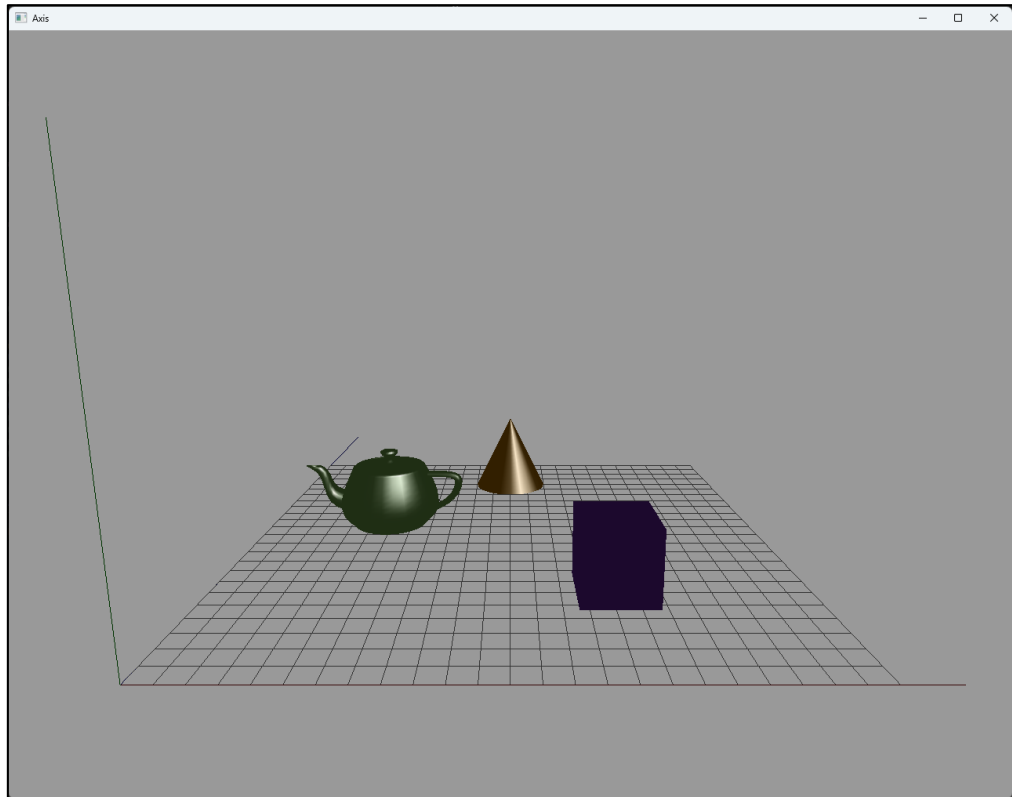


문제 4

- Light System

문제 4 – 조명 시스템

- OpenGL의 기본 조명 시스템에 대해 알아보자
- 첨부된 **LightSystem.txt** 파일을 실행해보고 주석처리 후 제출

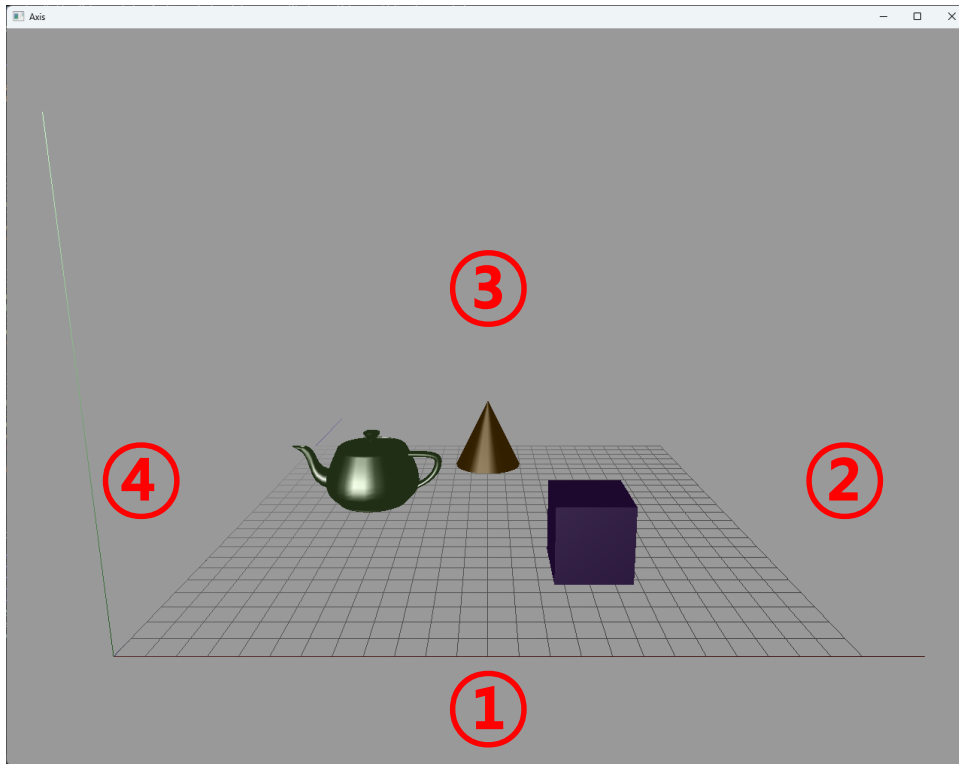


문제 5

- Input / Output function

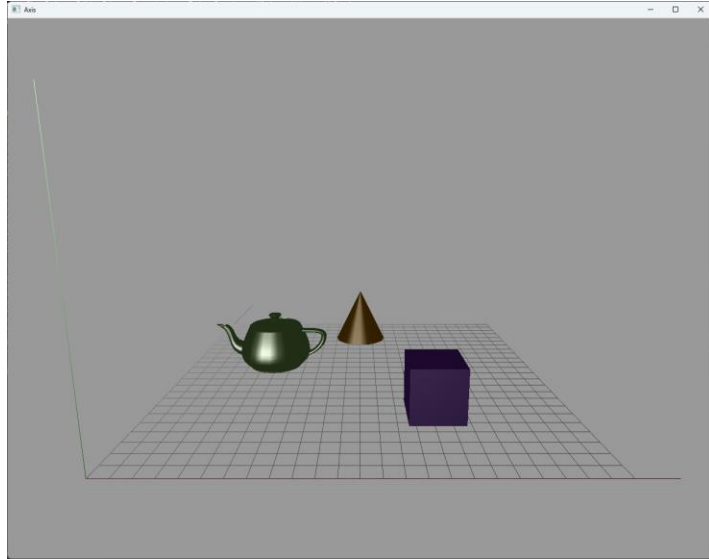
문제 5 – 입출력 처리

- 광원의 위치를 변경하는 **LightSystem.txt** 를 참고하여
다음과 같이 4방향에서 바라볼 수 있는 입출력 시스템 구현

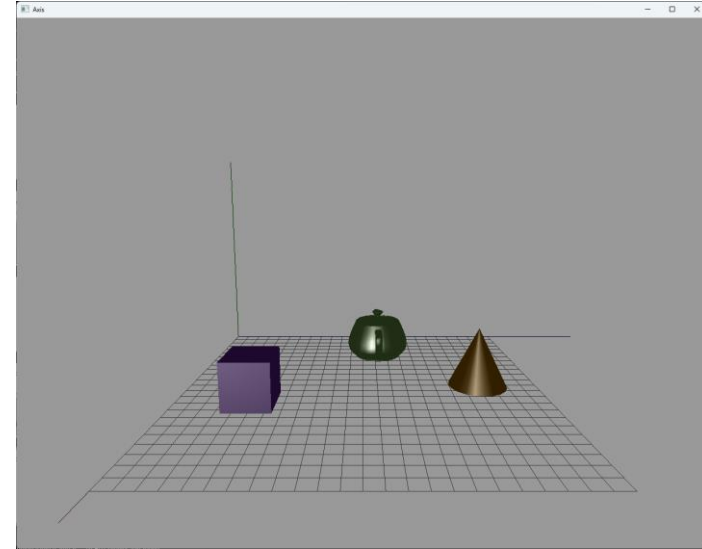


문제 5 – 입출력 처리 구현 결과

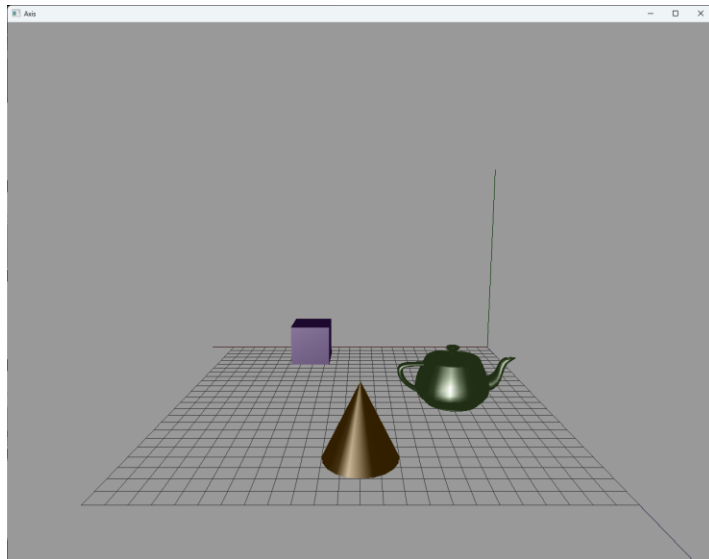
①



②



③



④

