

심화전공실습 (CGL)

HW06_Transformation



Self-scoring table

	P1	P2	P3	E1	E2	Total
Score	1	1	1	1	1	5

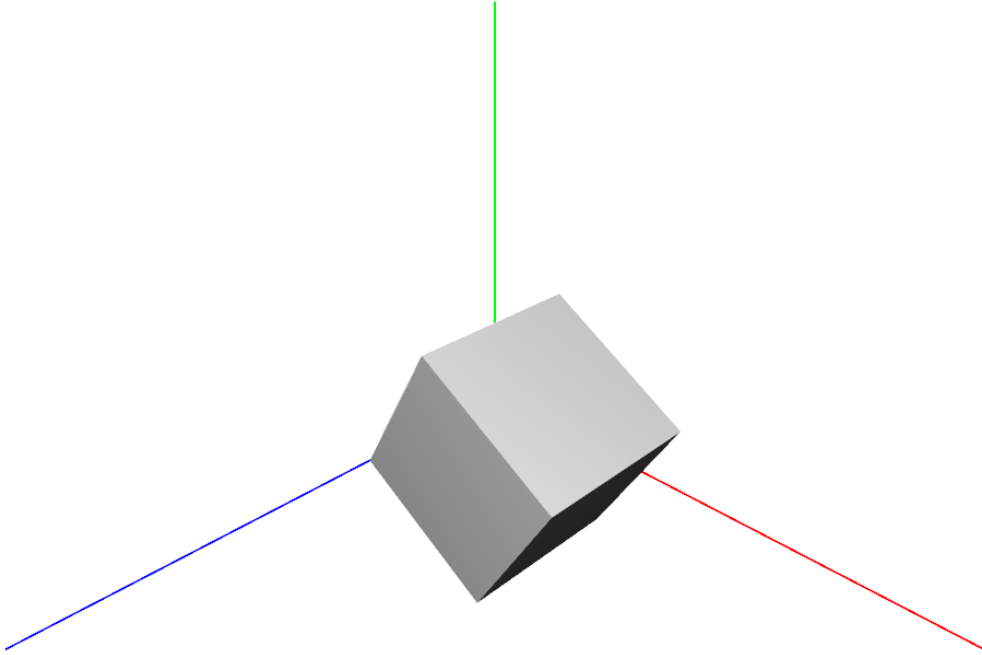
2018707068 김경환

KwangWoon University

Practice01 Snapshot, Explanation:

C:\Users\KimKyeongHwan\source\repos\practice\EXE\practice.exe

— □ ×

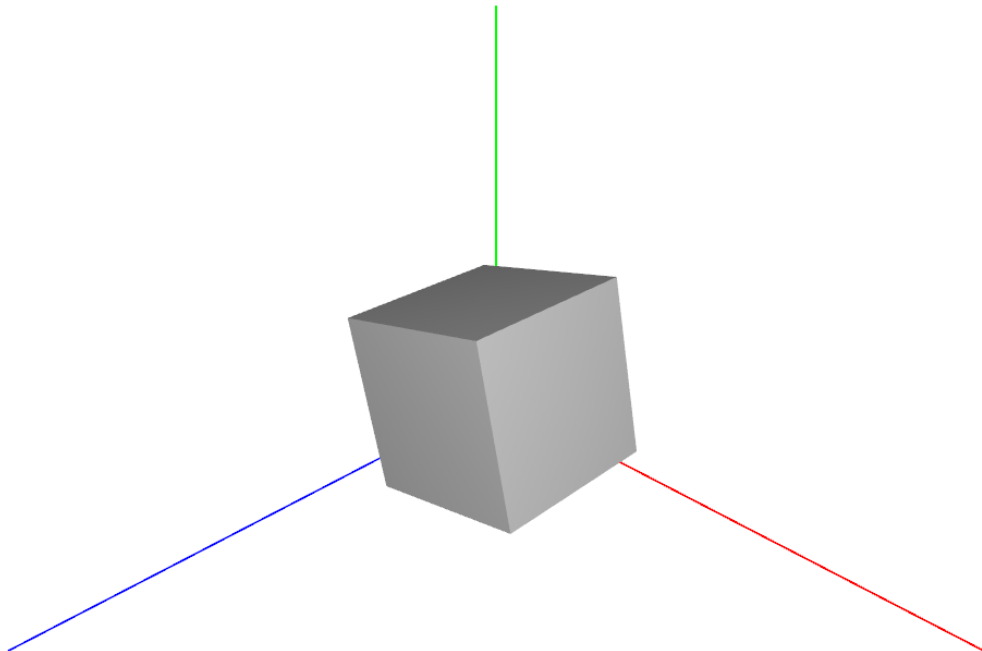


Keyboard Input으로 1을 입력했을 때 나오는 rotation example이다.

이는 회전축인 $\text{axis}(0, 0, 1)$ 의 z축(파란색 선)을 중심으로 회전하는 것을 볼 수 있다.

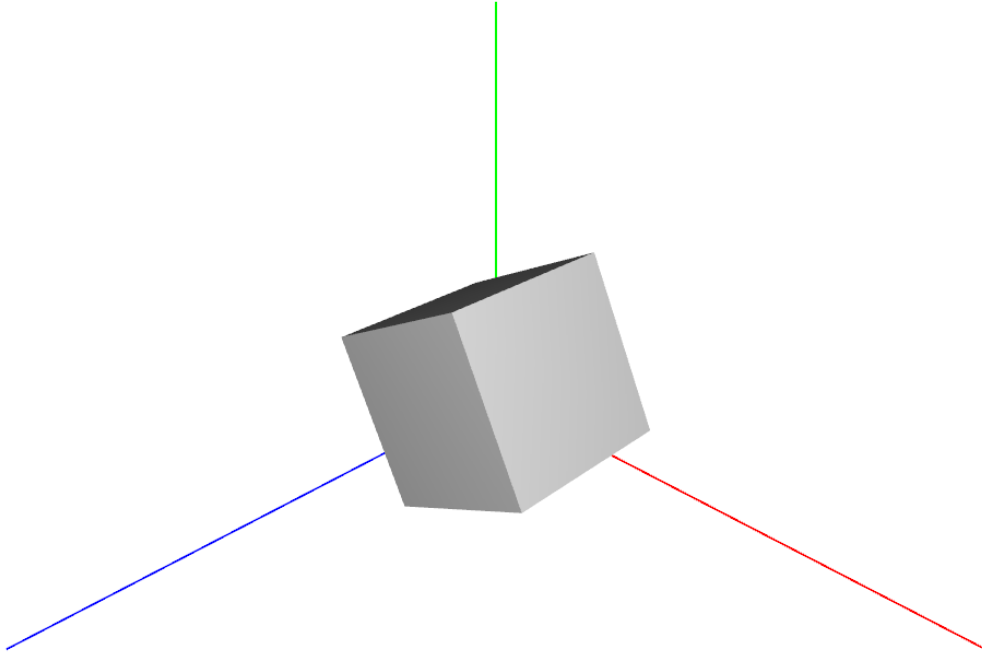
C:\Users\KimKyeongHwan\source\repos\practice\EXE\practice.exe

— □ ×



Keyboard Input으로 2를 입력했을 때 나오는 rotation wrt a pivot example이다.

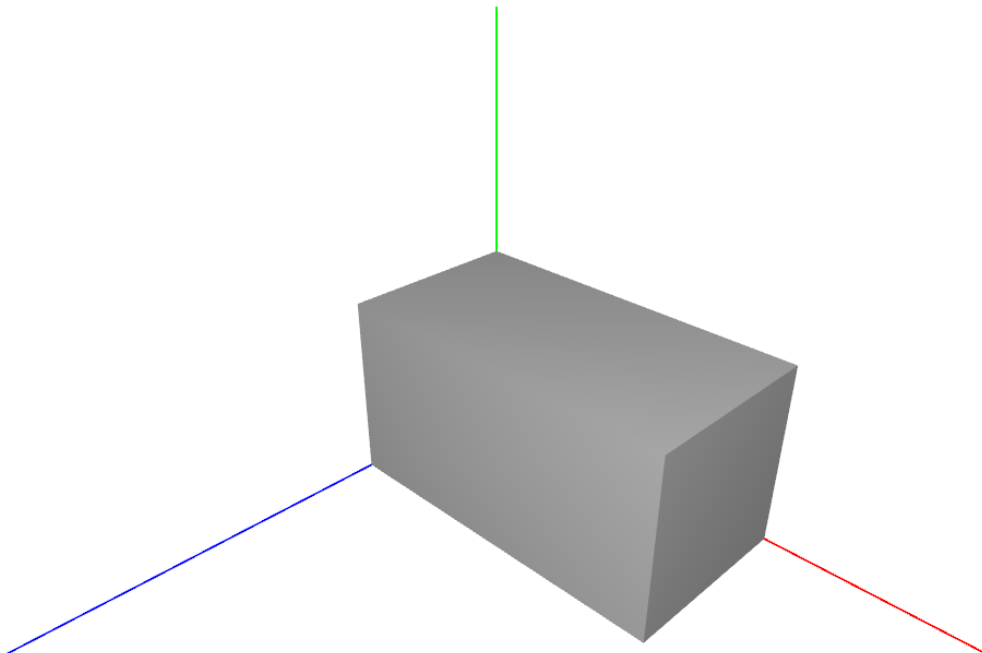
특정 $\text{pivot}(0.5, 0.5, 0)$, $\text{axis}(0, 0, 1)$ 에 대해 $\text{Translate}(-\text{pivot}) \rightarrow \text{rotate}(\text{axis}) \rightarrow \text{Translate}(\text{pivot})$ 로 구현되어 해당 pivot을 지나는 z축과 평행한 선분을 중심축으로 회전하고 있는 것을 볼 수 있다.



Keyboard Input으로 3을 입력했을 때 나오는 rotation wrt a pivot example using GLM이다.

GLM도 이전 OpenGL에서의 과정과 유사하게 $\text{Translate}(M, -\text{pivot}) \rightarrow \text{rotate}(M, \text{axis}) \rightarrow \text{Translate}(M, \text{pivot})$ 로 구현되지만, GLM의 경우 명백하게 $M(\text{matrix})$ 를 사용하여 해당 과정을 multiply해야한다. 그리고 그러한 matrix를 OpenGL로 넘겨 그림을 그리게 된다.

Practice02 Snapshot, Explanation:

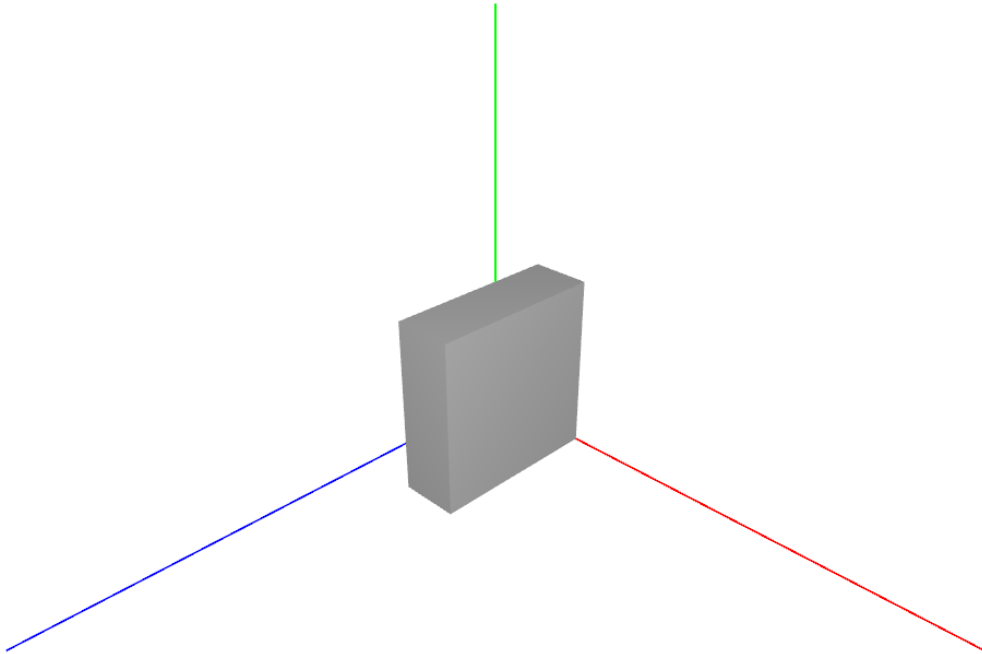


Keyboard Input으로 4을 입력했을 때 나오는 scaling example이다.

Scale(s, 1, 1)을 통해서 해당 물체의 y축, z축 성분의 값은 그대로이지만, x축 성분의 값이 달라지는 것을 볼 수 있다. 여기서 주의할 점은 해당 cube의 중심이 원점이 아니고, vertex x좌표가 전부 양수이기 때문에 x축의 양의 방향으로만 증가하고 감소하는 것을 볼 수 있다.

C:\Users\KimKyeongHwan\source\repos\practice\EXE\practice.exe

- □ ×

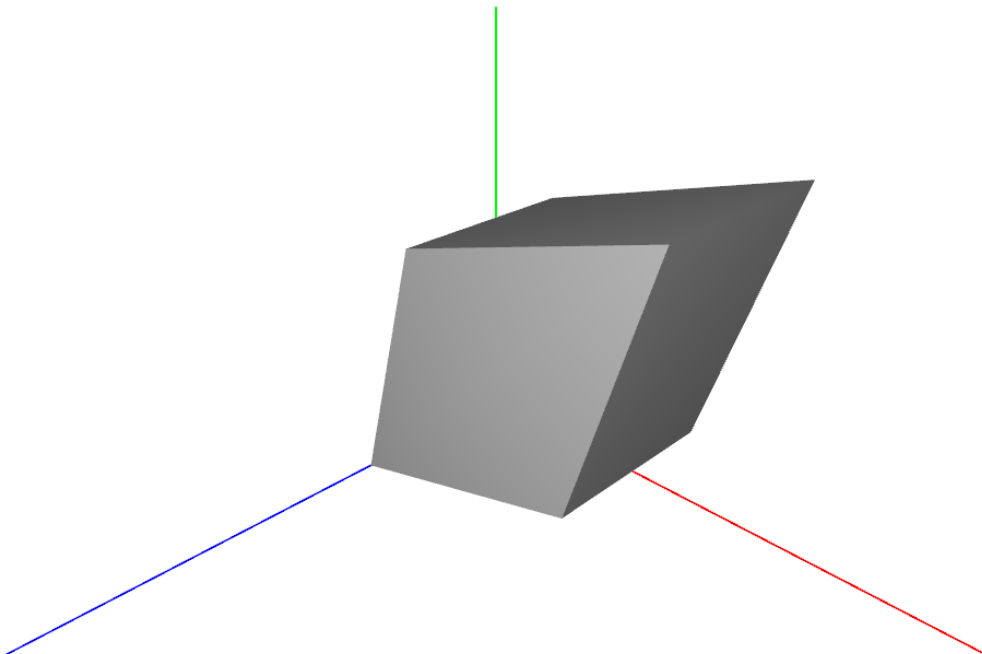


Keyboard Input으로 5을 입력했을 때 나오는 scaling wrt a pivot example이다.

특정 pivot(0.5, 0.5, 0.5)에 대해 Translate(-pivot) -> scale(s, 1, 1) -> Translate(pivot)로 구현되어 해당 pivot을 중심으로 x축 성분의 값이 달라지는 것을 볼 수 있다. 여기서 주의할 점은 특정 pivot에 대해 translate하여 cube의 중심을 원점으로 옮기고 scaling 하였으므로 x축의 양방향으로 증가하고 감소하는 것을 볼 수 있다.

C:\Users\KimKyeongHwan\source\repos\practice\EXE\practice.exe

- □ ×

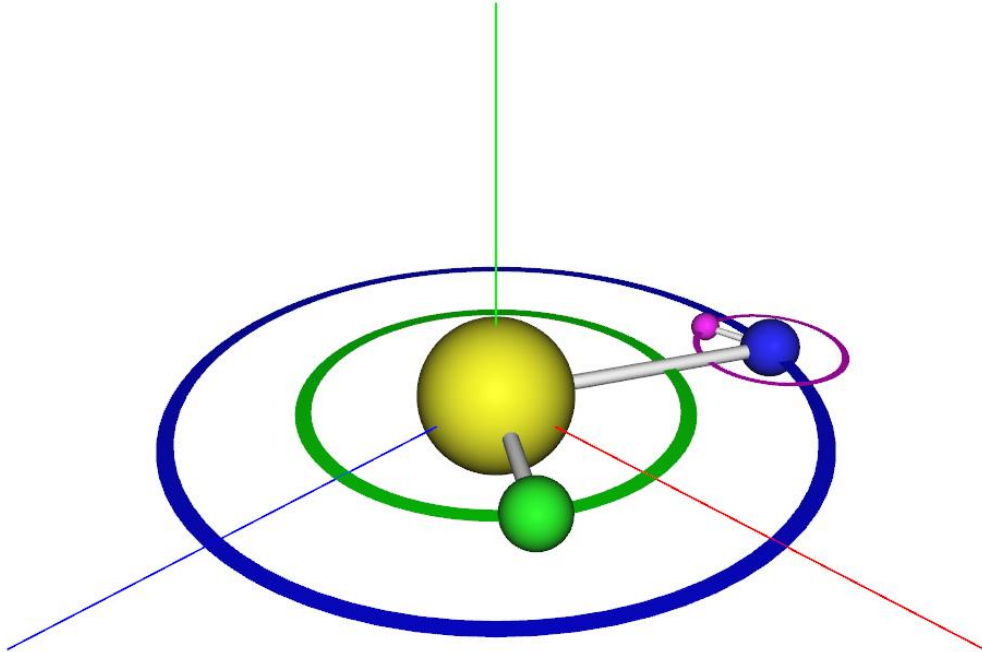


Keyboard Input으로 6을 입력했을 때 나오는 scaling wrt a direction example이다.

x축에 대해 scale을 할 것이기 때문에 특정 direction(1, 1, 0)과 x축(1, 0, 0)이 이루는 theta(각)를 구한다. 이는 x축과 direction의 cross product와 dot product를 통해 $\sin(\theta)$ 값과 $\cos(\theta)$ 값을 구하고, 해당 두 값으로 $\text{atan2}()$ 을 사용하여 theta를 구할 수 있다. 이렇게 구한 theta값으로 $\text{rotate}(-\theta, \text{axis}) \rightarrow \text{scale}(s, 1, 1) \rightarrow \text{rotate}(\theta, \text{axis})$ 을 거쳐 특정 direction으로부터의 scaling을 수행할 수 있다.

Practice03 Snapshot, Explanation:

C:\Users\KimKyeongHwan\source\repos\practice\EXE\practice.exe

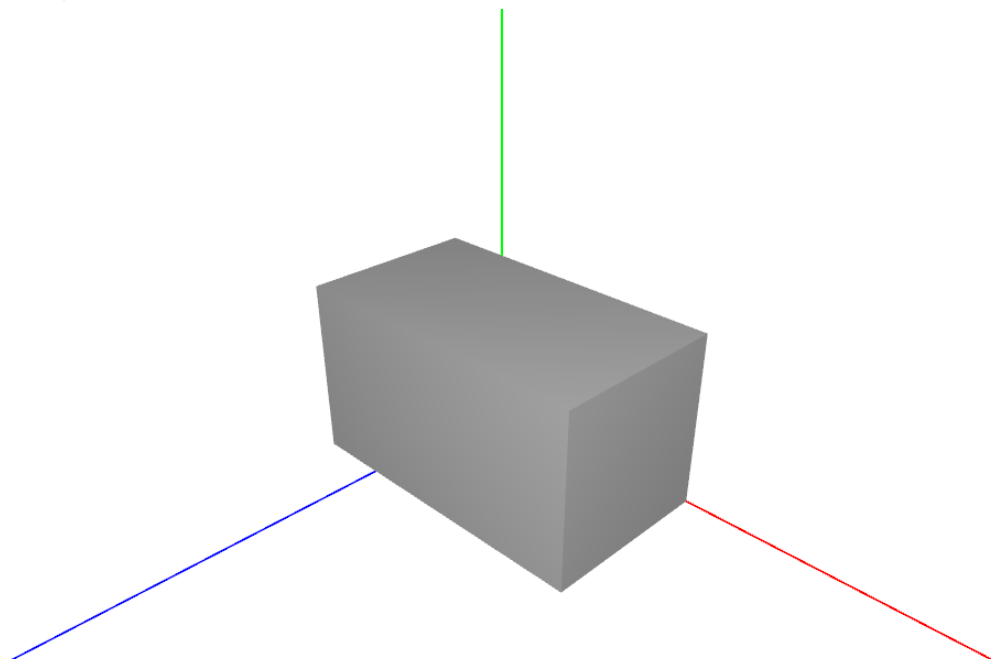


Keyboard Input으로 7을 입력했을 때 나오는 solar system example이다.

2개의 planet은 sun을 중심으로 pivot rotate를 수행하고, satellite의 경우에는 planet 2를 중심으로 pivot rotate를 수행한다. 그리하여 이러한 계층적인 transformation을 수행하기 위해 matrix stack을 사용한다.

Exercise01 SnapShot, Explanation:

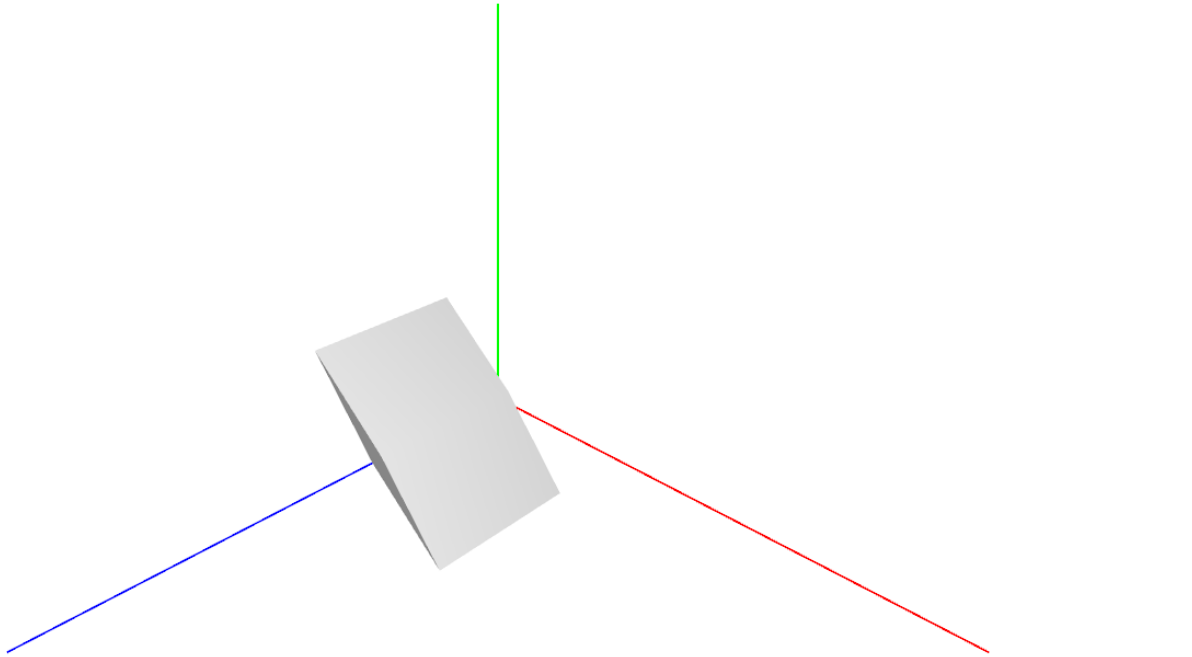
C:\Users\KimKyeongHwan\source\repos\exercise\EXE\exercise.exe



OpenGL에서의 scaling wrt a pivot과 같이 과정은 특정 pivot(0.5, 0.5, 0.5)에 대해 Translate(-pivot) -> scale(s, 1, 1) -> Translate(pivot)으로 구현되어 있으나 GLM에서는 명백하게 M(matrix)를 사용하여 해당 과정을 multiply해야한다. 그리하여 Translate(M, -pivot) -> scale(M, (s, 1, 1)) -> Translate(M, pivot)을 거치고 해당 matrix를 OpenGL로 넘겨 그림을 그리게 된다. 또한, rendering된 모습은 OpenGL에서의 scaling wrt a pivot과 같다.

Exercise02 SnapShot, Explanation:

C:\Users\KimKyeongHwan\source\repos\exercise\EXE\exercise.exe



여기서도 OpenGL에서의 scaling wrt a direction과 같이 특정 direction(1, 1, 0)과 scale 시킬 축 사이의 theta를 구하고, 이를 통해 rotate(-theta, axis) -> scale(s, 1, 1) -> rotate(theta, axis)을 거쳐 특정 direction으로부터의 scaling을 수행할 수 있다. 하지만 위에서 말한 것 과 같이 GLM에서는 명백하게 M(matrix)를 사용하여 해당 과정을 multiply해야한다. 그리하여 rotate(-theta, -axis) -> scale(s, 1, 1) -> Translate(M, axis)을 거치고 해당 matrix를 OpenGL로 넘겨 그림을 그리게 된다. 또한, rendering된 모습은 OpenGL에서의 scaling wrt a direction과 같다.