HW4　　20171624 김지선

문제. 다음은 간단한 모델링 변환에 관한 문제이다.

(a) 철수는 그림1과 같이 7마리의 소가 (-3, 0, -3)으로부터 (3,0,3)지점까지 z=x함수를 따라 이일정 속도로 이동하며 x축을 기준으로 360도 회전하는 모델링을 구현하려고 하였다. 그러나, 철수의 모델링 화면은 그림2와 같이 소가 회전하지 않았다. 아래의 코드는 소를 그려주는 프로그램의 일부코드로써 철수가 구현한 것이다. 그림1과 같이 고치기 위해서는 어떻게 수정해야 하는지 기술하시오.

그림1

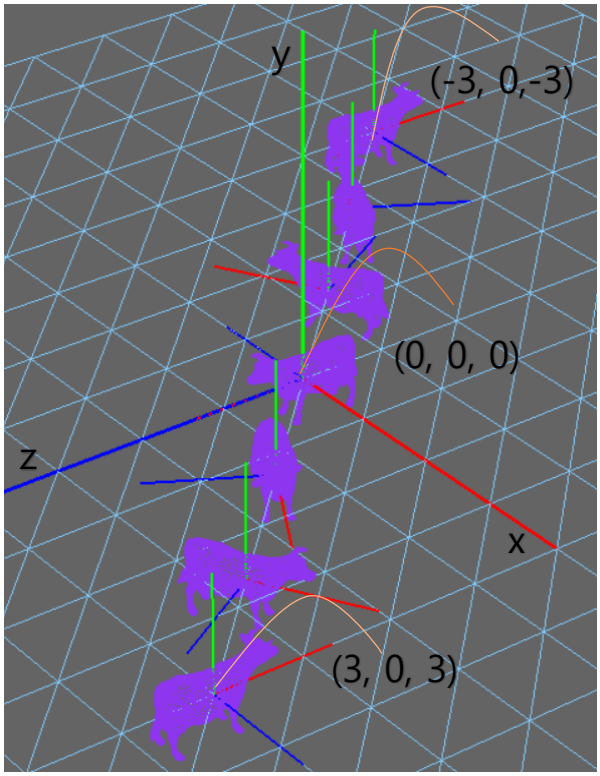
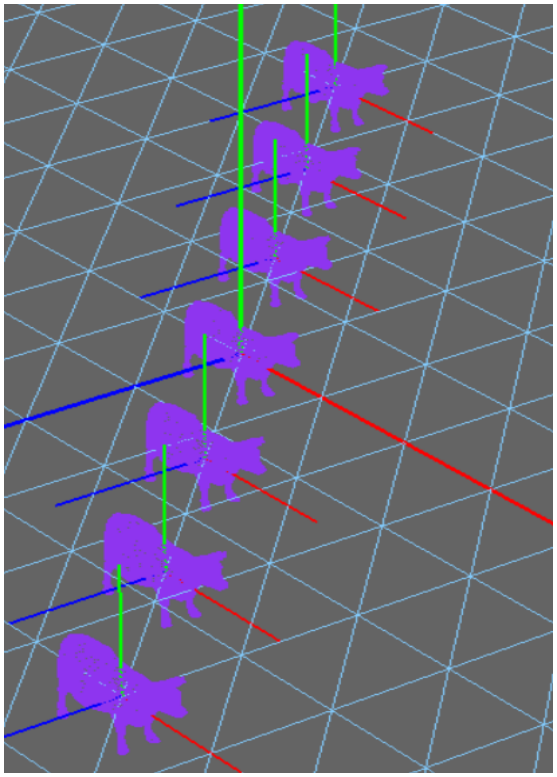


그림2



```
for (int i = 0; i <= 360; i += 60) {
		float angle = (float)i;
		ModelViewMatirx = glm::rotate(ModelViewMatirx, (angle + 90) * TO_RADIAN, glm::vec3(0, 1, 0));
		ModelViewMatirx = glm::translate(ViewMatrix, glm::vec3(   3*angle / 180 -3, 0, 3*angle/180 -3));

		ModelViewProjectionMatrix = ProjectionMatrix * ModelViewMatirx;
		glUniformMatrix4fv(loc_ModelViewProjectionMatrix, 1, GL_FALSE,
  &ModelViewProjectionMatrix[0][0]);   glLineWidth(2.0f);
		draw_axes();
		glLineWidth(1.0f);
		draw_object(OBJECT_COW, 142 / 255.0f, 53 / 255.0f, 239 / 255.0f);
}
```

답:

```
ModelViewMatirx = glm::translate(ViewMatrix, glm::vec3(3 * angle / 180 - 3, 0, 3 * angle / 180 - 3));
ModelViewMatirx = glm::rotate(ModelViewMatirx, (angle + 90) * TO_RADIAN, glm::vec3(0, 1, 0));
```

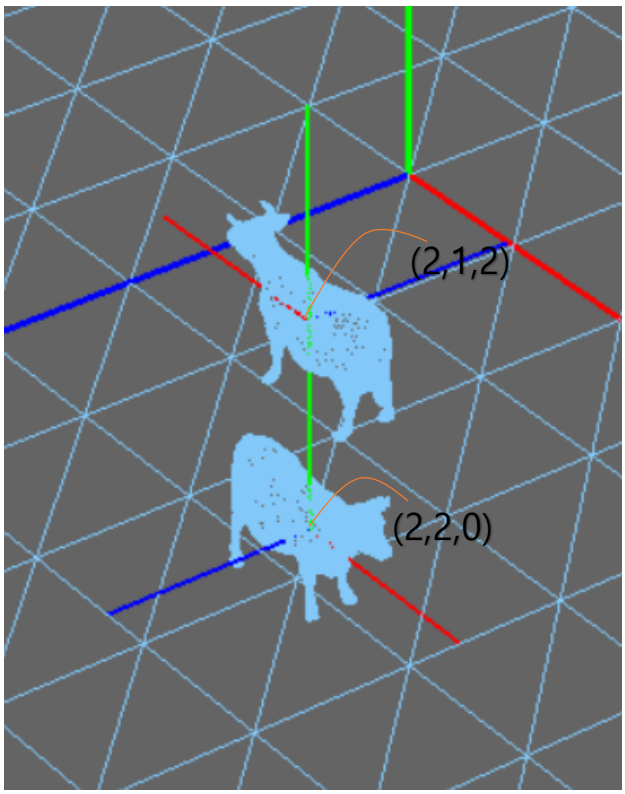
의 순서로 호출하여 수정한다.

해설:

좌표계 변환은 물체변환과 달리 함수호출 순서대로 실행되기에 translate을 먼저 한 뒤에 rotate를 해야 물체에 적용이 된다. 따라서, 철수와 같이 rotate를 한 후 translate를 실행하면 결과에 적용되지 않는다.

(b) 그림3과 같이 (2,2,0)위에 놓인 소와 y축으로 1만큼 떨어진 (2,1,2)지점에 반대방향으로 머리를 향하고 있는 소가 있다. 이와 같이 모델링을 구현하기 위한 아래 코드의 빈칸을 채우시오.

그림3



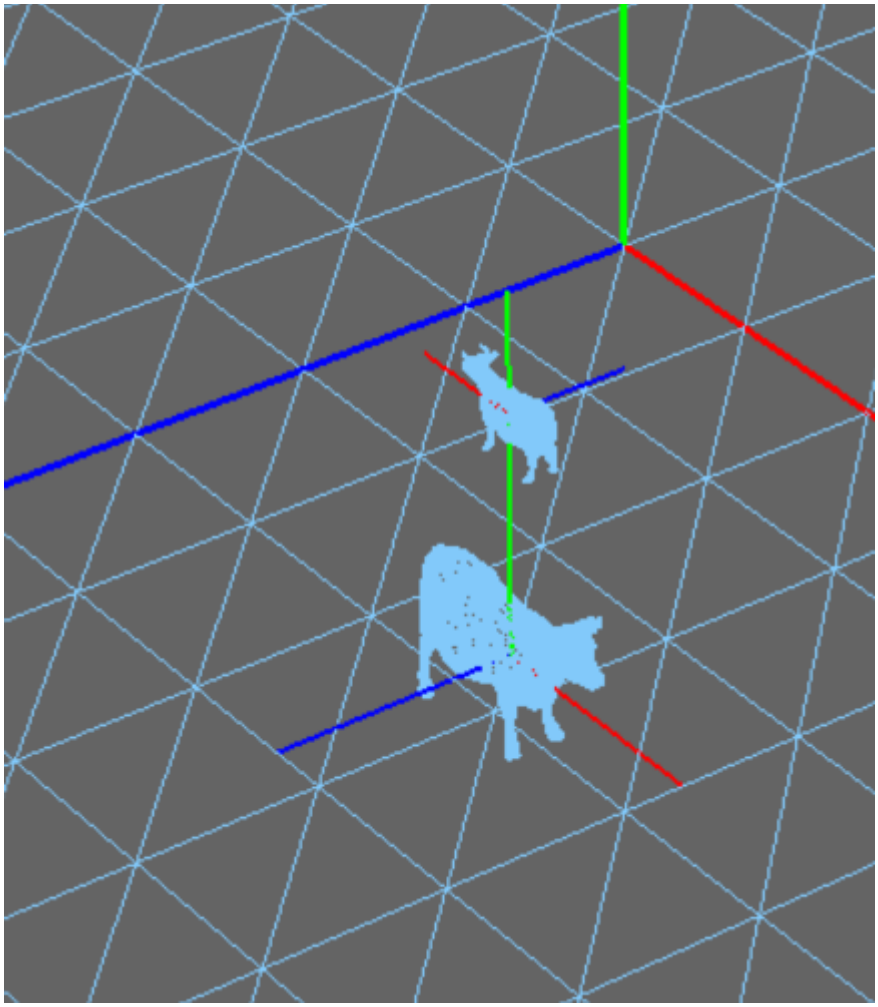
```
ModelViewMatirx = glm::translate(ViewMatrix, glm::vec3(2.0f, 0.0f, 2.0f));
ModelViewMatirx = glm::rotate(ModelViewMatirx, (A)* TO_RADIAN, glm::vec3((B), (C), (D));
ModelViewProjectionMatrix = ProjectionMatrix * ModelViewMatirx;
glUniformMatrix4fv(loc_ModelViewProjectionMatrix, 1, GL_FALSE,
&ModelViewProjectionMatrix[0][0]); glLineWidth(2.0f);
draw_axes();
glLineWidth(1.0f);
draw_object(OBJECT_COW, 130 / 255.0f, 201 / 255.0f, 250 / 255.0f);

ModelViewMatirx = glm::translate(ViewMatrix, glm::vec3(2.0f, (E), 2.0f));
ModelViewMatirx = glm::rotate(ModelViewMatirx, (F) * TO_RADIAN, glm::vec3((B), (C), (D));
ModelViewProjectionMatrix = ProjectionMatrix * ModelViewMatirx;
glUniformMatrix4fv(loc_ModelViewProjectionMatrix, 1, GL_FALSE,
&ModelViewProjectionMatrix[0][0]); glLineWidth(2.0f);
draw_axes();
glLineWidth(1.0f);
draw_object(OBJECT_COW, 130 / 255.0f, 201 / 255.0f, 250 / 255.0f);
```

답: (A) 0, (B) 0, (C) 1, (D) 0, (E)1.0f (F) 180

해설: (2,2,0)지점에 있는 소는 양의 x축을 바라보며 위치하기에 (A)의 값이 0이다. 또한, 소의 등이 y축의 양의 방향을 바라보고 있기에 (B)(C)(D)는 각각 0,1,0의 값을 갖는다. 또한, 그 위의 소는 y축으로 1만큼 떨어져 있기에 (E)의 값은 1이다. 그리고 이 소는 x축의 음의 방향을 바라보기에 (F)의 값은 180으로 180도 만큼 회전된다.

(c) 위 그림3에서 (2,2,1)지점에 위치한 소의크기를 그림 4와 같이 반으로 줄이려 한다. 코드의 어떤 부분을 수정하여야 하는지 위치와 수정내용을 서술하시오.

그림4



답:

```
ModelViewMatirx = glm::translate(ViewMatrix, glm::vec3(2.0f, 1.0f, 2.0f));
ModelViewMatirx = glm::scale(ModelViewMatirx, glm::vec3(0.5f, 0.5f, 0.5f));
ModelViewMatirx = glm::rotate(ModelViewMatirx, 180 * TO_RADIAN, glm::vec3(0, 1, 0));
```

또는

```
ModelViewMatirx = glm::translate(ViewMatrix, glm::vec3(2.0f, 1.0f, 2.0f));
ModelViewMatirx = glm::rotate(ModelViewMatirx, 180 * TO_RADIAN, glm::vec3(0, 1, 0));
ModelViewMatirx = glm::scale(ModelViewMatirx, glm::vec3(0.5f, 0.5f, 0.5f));
```

해설: scale 함수를 사용하여 소의 크기를 x축, y축, z축을 기준으로 반으로 줄인다. 이때, scale조정은 translate 또는 rotate다음에서 수행될 수 있다.