기초 컴퓨터 그래픽스 HW4

20171666 이예은

1. 다음은 3차원 모델링 변환에 관한 문제이다.

(a) 그림 2에는 그림1에서와 같이 좌표계의 (-6,0,0) 지점에서 (6,0,0) 지점까지의 곡선 경로를 따라 이동하는 소 모델을 적절한 모델링 변환을 통해 세상에 배치하는 프로그램의 일부가 주어져 있다. 여기서 소 물체는 2*sin 함수의 모양에 따라 45도씩 회전하며 움직이고 있다. 또한 원점을 지난 후 크기가 1.5배 커지는 이동을 하고 있다. 이 코드가 올바르게 작동하기 위하여 그림2의 (A)~(I)까지 들어가야 할 값을 정확히 기술하여라.

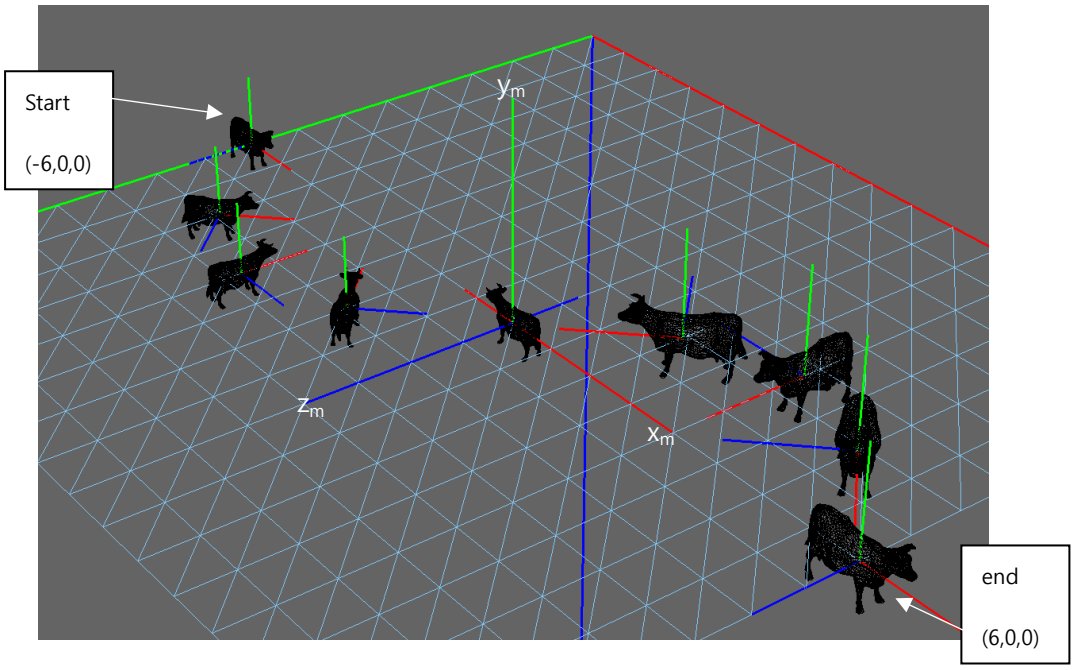


그림 1

```
for (int i = 0; i <= 360; i+=45) {
    float angle = (float)i;

    ModelViewProjectionMatrix = glm::translate(ViewProjectionMatrix, glm::vec3((angle - 180.0f) / 30, (A), (B)));
    ModelViewProjectionMatrix = glm::rotate(ModelViewProjectionMatrix, (C) * TO_RADIAN, glm::vec3((D), (E), (F)));

    if (angle > 180)
        ModelViewProjectionMatrix = glm::scale(ModelViewProjectionMatrix, glm::vec3((G), (H), (I)));

    glUniformMatrix4fv(loc_ModelViewProjectionMatrix, 1, GL_FALSE, &ModelViewProjectionMatrix[0][0]);
    glLineWidth(3.0f);
    draw_axes();
    glLineWidth(1.0f);
    draw_object(OBJECT_COW, 0 / 255.0f, 0 / 255.0f, 0 / 255.0f);
}
```

그림 2

(b) 그림3은 (a) 코드의 벡터 값을 적절히 변환하여 소를 모델링한 그림이다. 이 또한 2*sin 함수의 움직임을 보이나 그림에서 알 수 있듯 조금은 특별한 운동을 한다. 이와

같은 결과를 얻기 위하여 (A)~(I)까지 들어가야 할 적절한 값을 정확히 기술하여라. (필요시, abs(x): x의 절댓값을 구해주는 abs 함수를 사용하여라.)

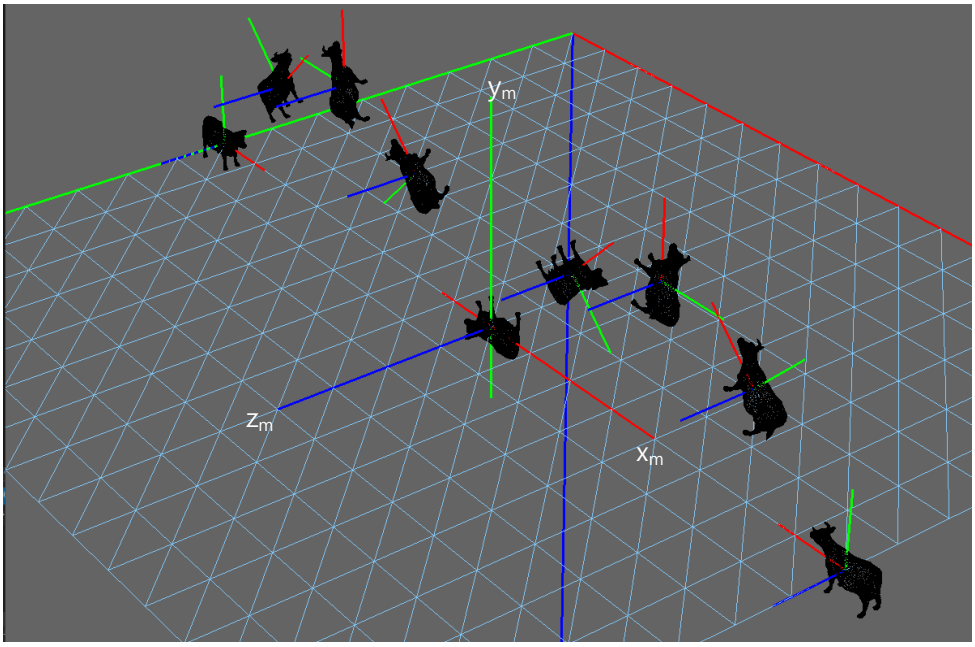


그림 3

(c) (b)의 코드에서 rotate 함수 실행 이후

```
ModelViewProjectionMatrix = glm::rotate(ModelViewProjectionMatrix,
                                        -90 * TO_RADIAN,
                                        glm::vec3(0.0f, 1.0f, 0.0f));
```

다음과 같은 코드를 추가한다면 소의 모습이 어떤 식으로 변화할 지

```
glm::rotate(glm::mat4(1.0f), -90 * TO_RADIAN, glm::vec3(0.0f, 1.0f, 0.0f))
```

다음 함수가 리턴해주는 4행 4열 행렬을 적절히 사용하여 기술하시오.

2. 정답 및 해설

(a) 정답:

A: 0.0f B: 2*sin(angle * TO_RADIAN) C: angle

D: 0.0f E: 1.0f F: 0.0f

G: 1.5f H: 1.5f I: 1.5f

해설: 소의 이동하는 모습만 본다면, z=2 * sin(x)의 형태로 이동하고 있다는 것을 알 수 있다. 또한 x 축의 이동은 복잡한 사칙연산으로 이루어져 있지만, sin의 값, z는 그에 반해 angle로만 결정된다는 것을 90도, 180도, 270도의 그림으로 알 수 있다. y축으로는 이동을 보이지 않는다. 그러므로 A는 0.0f, B는 2*sin(angle * TO_RADIAN)이 된다. 소가 회전하는 모습을 보면, y축을 기준으로 angle 만큼씩 회전하고 있다는 것을 볼 수 있다. 그러므로 C는 angle (*TO_RADIAN이 이미 있으므로), D는 0.0f, E는 1.0f, F는 0.0f가 된다. 소가 원점을 지난 후, 즉 180도 이후부터 크기가 1.5배 커진다고 했으므로 if문 안의 scale은 1.5배 커지는 내용이 들어가야 한다. 그러므로 G는 1.5f, H는 1.5f, I는 1.5f가 된다.

(b) 정답:

A: 2*abs(sin(angle * TO_RADIAN)) B: 0.0f C: angle

D: 0.0f E: 0.0f F: 1.0f

G: -1.0f H: 1.0f I: 1.0f

해설: 소의 이동하는 모습만 보면 y=2 * sin(x)의 형태로 이동하고 있다. 하지만 이 모델의 경우 음수 값이 양수 값으로 변환되어 이동하는 특이한 운동을 하므로 음수 값의 이동이 절댓값으로 변환되었다는 것을 알 수 있다. 이 이동 또한 sin의 값은 angle로만 결정된다는 것을 90, 180도 등의 그림에서 알 수 있으므로 A의 값은 2*abs(sin(angle * TO_RADIAN))으로 나타낼 수 있고, z축의 이동은 일어나지 않으므로 B는 0.0f이다. 소의 회전운동을 보면 또한 z축 기준으로 angle 만큼 회전하고 있다는 것을 볼 수 있다. 그러므로 C는 angle, D는 0.0f, E는 0.0f, F는 1.0f이다. 이 모델링에서는 크기 변환이 일어나고 있지는 않지만 소의 모습을 잘 보면 180도 이후부터 yz 평

면에 관해 반전이 일어나고 있는 것을 360도 모델 등을 통해 알 수 있다. 그러므로 if 문 안의 scale은 x의 부호가 바뀌는 내용이 들어가야 한다. 그러므로 G는 -1.0f, H는 1.0f, I는 1.0f가 된다.

(c) 정답:

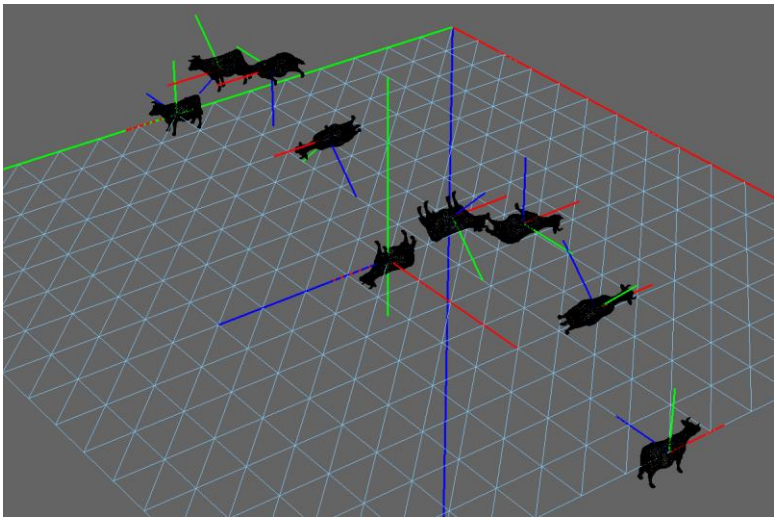
glm::rotate(glm::mat4(1.0f), -90 * TO_RADIAN, glm::vec3(0.0f, 1.0f, 0.0f)) 의 리턴 값은 $\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 이므로 해당 줄을 추가하게 되면 소가 angle만큼 z축에 관해 회전 후 y축에 관해 -90도 회전하게 된다. 그러므로 전체적인 모습은 그림 2에서 y축에 관해 -90도 회전한 모습으로 바뀌게 된다. (행렬 값이 있고 아래 그림을 표현한 설명이라면 정답으로 처리한다.)

해설:

$$R(\alpha, n_x, n_y, n_z) = \begin{bmatrix} \bar{n}_x^2(1-c)+c & \bar{n}_x\bar{n}_y(1-c)-\bar{n}_z s & \bar{n}_z\bar{n}_x(1-c)+\bar{n}_y s & 0 \\ \bar{n}_x\bar{n}_y(1-c)+\bar{n}_z s & \bar{n}_y^2(1-c)+c & \bar{n}_y\bar{n}_z(1-c)-\bar{n}_x s & 0 \\ \bar{n}_z\bar{n}_x(1-c)-\bar{n}_y s & \bar{n}_y\bar{n}_z(1-c)+\bar{n}_x s & \bar{n}_z^2(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{where } c = cos(\alpha),\ s = sin(\alpha), n = (n_x\, n_y\, n_z)^t, \text{ and } \bar{n} = (\bar{n}_x\, \bar{n}_y\, \bar{n}_z)^t = \frac{n}{|n|}$$

rotation 함수는 다음과 같은 행렬 식을 반환해주므로 α는 -90, n=(0.0, 1.0, 0.0)을 대입해 계산해보면 $\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 이러한 행렬식이 나온다. 이를 통해 y축 기준으로 -90도 돌린 그림을 추론해 낼 수 있다. 그러므로,



다음과 같이 y축 기준으로 -90도만큼 추가로 회전한 그림이 결과로 나타나게 된다.