

[CSE4170: 기초 컴퓨터 그래픽스]

중 간 고 사

(담당교수: 임 인 성)

- 답은 연습지가 아니라 답안지에 기술할 것. 답안지 공간이 부족할 경우, 답안지 뒷면에 기술하고, 해당 답안지 칸에 그 사실을 명기할 것. 문제지와 연습지는 수거하지 않음.

1. 2차원 아핀 변환인 이동변환 $T(t_x, t_y)$, 크기변환 $S(s_x, s_y)$, 그리고 회전변환 $R(\theta)$ 에 대한 3행 3열 행렬들을 고려하자.

- $R(\theta) \cdot S(-1, 1) = S(-1, 1) \cdot R(\alpha)$ 식을 만족시켜주는 α 값은 무엇인가?
- $T(t_x, t_y) \cdot S(-1, 1) = S(-1, 1) \cdot T(\beta, \gamma)$ 식을 만족시켜주는 β 와 γ 값은 무엇인가?
- $R(\theta) \cdot T(t_x, t_y) = T(\delta, \epsilon) \cdot R(\theta)$ 식을 만족시켜주는 δ 와 ϵ 값은 무엇인가?

2. 그림 1에는 2차원 세상 좌표계의 원점 주위에 모델링 좌표계에서 정의된 모자 물체가 도시되어 있다.

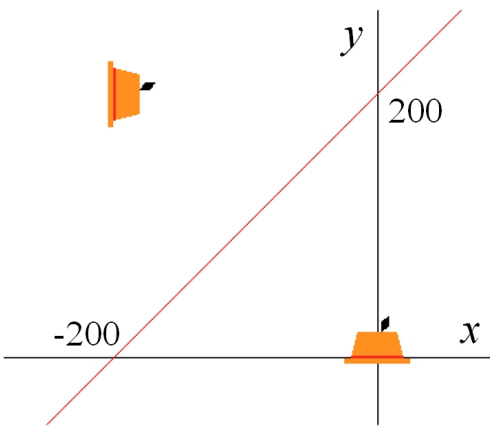


Figure 1: 2차원 공간에서의 반사

이제 이 원점 주변의 물체를 $y = x + 200$ 인 직선에 대하여 반사시키려 한다.

- 이에 필요한 모델링 변환을 OpenGL 환경에서 glm 함수를 사용하여 다음과 같이 구현하려 한다.

```
-----
ModelMatrix =
    glm::translate(glm::mat4(alpha),
        glm::vec3(-200.0f, 0.0f, 0.0f));
ModelMatrix = glm::rotate(ModelMatrix,
    -45.0f*TO_RADIAN,
    glm::vec3(0.0f, 0.0f, 1.0f));
ModelMatrix = glm::scale(ModelMatrix,
    glm::vec3(1.0f, -1.0f, 1.0f));
ModelMatrix = glm::rotate(ModelMatrix,
    45.0f*TO_RADIAN,
    glm::vec3(0.0f, 0.0f, 1.0f));
ModelMatrix =
    glm::translate(ModelMatrix,
        glm::vec3(200.0f, 0.0f, 0.0f));
-----
```

먼저 문맥상 α 에 들어갈 내용을 문법에 맞게 정확히 기술하라.

- 이 코드에는 명백한 오류가 있는데 잘못된 부분 지적한 후 이를 올바르게 고쳐라. (주의: 문장의 순서는 바꾸지 말것)
- (위 문제 오류 수정 후) 이 모델링 변환은 다음과 같이 구현할 수도 있다.

```
-----
ModelMatrix =
    glm::translate(glm::mat4(alpha),
        glm::vec3(-200.0f, 0.0f, 0.0f));
ModelMatrix = glm::scale(ModelMatrix,
    glm::vec3(1.0f, -1.0f, 1.0f));
ModelMatrix = glm::rotate(ModelMatrix,
    beta*TO_RADIAN,
    glm::vec3(0.0f, 0.0f, 1.0f));
ModelMatrix =
    glm::translate(ModelMatrix,
        glm::vec3(200.0f, 0.0f, 0.0f));
-----
```

이때 β 에 들어갈 값을 정확히 기술하라. (α 는 앞 문제의 구현에서와 동일함)

- (위 문제 오류 수정 후) 또한 이 모델링

변환은 다음과 같이 구현할 수도 있다.

```
ModelMatrix =
    glm::translate(glm::mat4( $\alpha$ ),
        glm::vec3(-200.0f, 0.0f, 0.0f));
ModelMatrix = glm::scale(ModelMatrix,
    glm::vec3(1.0f, -1.0f, 1.0f));
ModelMatrix =
    glm::translate(ModelMatrix,
        glm::vec3( $\gamma$ ,  $\delta$ , 0.0f));
ModelMatrix = glm::rotate(ModelMatrix,
     $\beta$ *TO_RADIAN,
    glm::vec3(0.0f, 0.0f, 1.0f));
```

이때 γ 와 δ 에 들어갈 값을 정확히 기술하라.
(α 와 β 는 앞 문제의 구현에서와 동일함)

3. 그림 2는 2차원 세상 좌표계의 원점 주위에 주어진 모자 물체를 좌표가 (250, 200)인 점을 중심으로 반지름이 150인 원을 따라 회전하면서 그려준 모습을 도시하고 있다. 이에 필요한 모델링 변환 행렬 Mm 을 아래의 코드에서 구현하려 한다. 이때 (Here) 부분에 들어갈 내용을 다음과 같은 형태의 문장들을 사용하여 구현하라. (동일한 종류의 문장을 중복해서 사용해도 무방하며, 각 문장 사용 시 필요한 인자값을 적절히 변경하여 기술할 것. 편의상 이 코드에서 $glm::$ 은 생략함)

- $Mm = \text{translate}(Mm, \text{vec3}(tx, ty, tz));$
- $Mm = \text{rotate}(Mm, \text{angle}*TO_RADIAN, \text{vec3}(0.0f, 0.0f, 1.0f));$
- $Mm = \text{scale}(Mm, \text{vec4}(sx, sy, sz));$

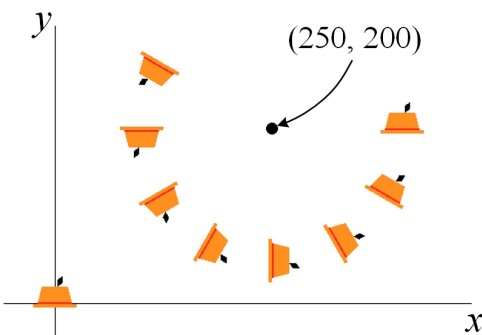


Figure 2: 2차원 기하 변환 합성 1

```
for (int i = 0; i < 8; i++) {
    Mm = mat4(1.0f);
```

(Here) ;

```
Mmvp = Mvp * Mm;
glUniformMatrix4fv(loc_Mmvp, 1,
    GL_FALSE, &Mmvp[0][0]);
draw_hat();
}
```

4. 그림 3은 2차원 세상 좌표계의 원점 주위에 주어진 모자 물체를 아래의 코드를 사용하여 그려준 모습을 도시하고 있다. 이때 (Here) 부분에서는 $mat4$ 타입의 어떤 행렬 $Mtmp$ 의 내용을 설정해 주고 있는데, 이에 필요한 코드를 위 문제의 세 문장들을 참조하여 문법에 맞게 정확히 기술하라. (동일한 종류의 문장을 중복해서 사용해도 무방하며, 각 문장 사용 시 필요한 인자값을 적절히 변경하여 기술할 것. 편의상 이 코드에서 $glm::$ 은 생략함)

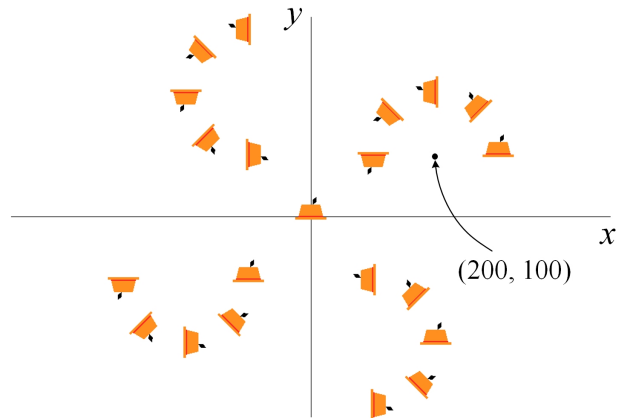


Figure 3: 2차원 기하 변환 합성 2

```
for (int i = 0; i < 4; i++) {
    Mtmp = mat4(1.0f);
    (Here) ;
    for (int j = 0; j < 5; j++) {
        Mm = rotate(Mtmp, j*45.0f*TO_RADIAN,
            vec3(0.0f, 0.0f, 1.0f));
        Mm = translate(Mm,
            vec3(100.0f, 0.0f, 0.0f));

        Mmvp = Mvp * Mm;
        glUniformMatrix4fv(loc_Mmvp, 1,
            GL_FALSE, &Mmvp[0][0]);
        draw_hat();
    }
}
```

5. 그림 4는 3차원 세상 좌표계의 점 $(x_w \ y_w \ z_w)^t$ 를 눈 좌표계의 점 $(x_e \ y_e \ z_e)^t$ 로 바꾸어주는 OpenGL 렌더링 파이프라인에서의 뷰잉 변환에 관한 그림이다. 여기서 $PRP = (e_x \ e_y \ e_z)^t$, $u = (u_x \ u_y \ u_z)^t$, $v = (v_x \ v_y \ v_z)^t$, 그리고 $n = (n_x \ n_y \ n_z)^t$ 이고, 내적 연산 $\circ$ 에 대해 $u \circ u = v \circ v = n \circ n = 1$ 이고 $u \circ v = v \circ n = n \circ u = 0$ 이며, 이때 각 기하 데이터의 의미는 수업시간에 배운 것을 가정한다.

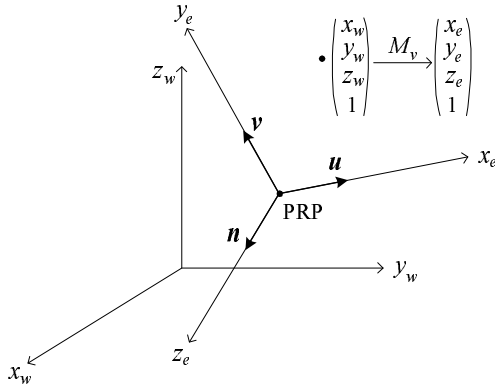


Figure 4: OpenGL 시스템에서의 뷰잉 변환

- (a) 이 두 좌표계 공간의 상호 관계는 아래와 같이 회전변환 행렬 R 과 이동변환 행렬 T 의 곱으로 표현할 수 있는데, 이때 이동변환 인자 a_x, a_y , 그리고 a_z 의 값을 기술하라. (힌트: 이 변환과 뷰잉변환과의 관계를 고려할 것)

$$\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} = T(a_x, a_y, a_z) \cdot R \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix}$$

- (b) 이 두 좌표계 공간의 상호 관계는 아래와 같이도 표현할 수 있는데, 이때 3행 3열 행렬의 원소값들을 기술하라. (즉 b_{11} 부터 b_{33} 까지)

$$\begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix}$$

- (c) 위 문제에서 c_y 값을 PRP , u , v , 그리고 n 들의 좌표값을 사용하여 표현하라. (벡터 내적 연산 $\circ$ 를 사용하여 기술해도 무방함)

6. 그림 5는 세상 좌표계 공간에서 한 변의 길이가 1이고 서로 수직인 세 선분에 의해 정의되는 두 프레임 Frame A와 Frame B를 도시하고 있는데, 지금 Frame A를 각 점이 아래와 같이 매핑이 되도록 Frame B로 변환하려 한다.

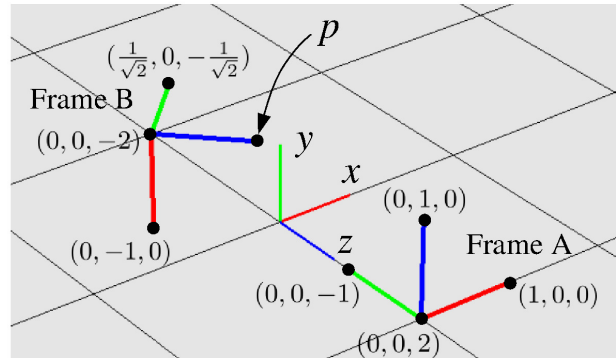


Figure 5: 두 프레임간의 변환

- $(0, 0, 2) \rightarrow (0, 0, -2)$
- $(1, 0, 0) \rightarrow (0, -1, 0)$
- $(0, 0, -1) \rightarrow (\frac{1}{\sqrt{2}}, 0, -\frac{1}{\sqrt{2}})$
- $(0, 1, 0) \rightarrow p$

(주의: 여기서 두 Frame의 원점의 좌표 $(0, 0, 2)$ 와 $(0, 0, -2)$ 만 점의 좌표이고 p 를 포함 나머지 좌표는 벡터를 나타냄)

- (a) Frame B의 p 의 좌표 (x, y, z) 를 기술하라.
- (b) 이때 Frame A를 Frame B로 변환해주는 아핀 변환에 해당하는 4행 4열 행렬 M 을 $M = T_1 \cdot R \cdot T_2$ 와 같이 두 개의 이동변환 행렬 T_1, T_2 와 회전변환 행렬 R 의 곱으로 표현하려할 때, 각 4행 4열 행렬의 16개 원소들을 정확히 기술하라.
- (c) 만약 Frame A의 세 방향 $(1, 0, 0)$, $(0, 0, -1)$, 그리고 $(0, 1, 0)$ 을 각각 세상 좌표계의 x_w , y_w , 그리고 z_w 축으로 일치시켜주는 기하변환을 “이동변환 후 회전변환”을 통하여 구현하려 한다. 이때 이 회전변환은 과연 어떤 방향 (n_x, n_y, n_z) 둘레로 어떤 각도 α ($\alpha > 0$)만큼 회전하는 변환에 해당할까? (참고: 필요하다면 그림 6에 주어진 회전 변환 행렬을 참조하면서 답하라)

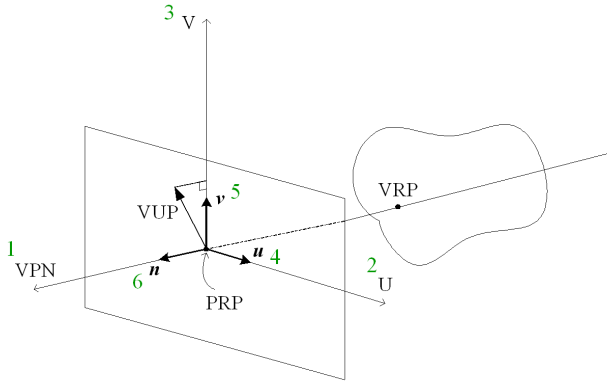
7. 다음의 `glm` 함수는 수업 시간에 배운 OpenGL 렌더링 과정에서의 뷰잉 변환 행렬 M_v 를 계산하여 리턴해주는 함수이다.

```
mat4 lookAt(vec3 eye, vec3 center vec3 up);
```

그림 7을 참조하면서 아래의 질문에 답하라. (편의상 이 문제의 코드에서 `glm::`은 생략함)

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

Figure 6: 회전 변환 행렬

Figure 7: 뷰잉 변환 행렬 M_V 의 계산

- (a) 이 그림에서 `lookAt()` 함수의 세 입력 인자 `eye`, `center`, 그리고 `up` 각각에 대응되는 기호를 기술하라.
- (b) 이 함수의 구현에서 지역변수로 `vec3` 타입의 변수 `f`, `g`, 그리고 `h`가 정의되어 있다. 이제 다음의 코드를 통하여 OpenGL 눈 좌표계(EC)의 세 축 x_e , y_e , 그리고 z_e 방향의 “길이가 1이고 서로 수직”인 세 방향을 계산하고 있다. 문맥상 (Here)에 들어갈 내용을 문법에 맞게 정확히 기술하라.

```
f = normalize(eye - center);
g = normalize(cross(up, f));
h = (Here) ;
```

- (c) 위 코드 수행 후, 세 축 x_e , y_e , 그리고 z_e 의 방향에 대응되는 변수를 순서대로 나열하라.
- (d) 이제 이 함수 내에서 정의한 `mat4` 타입의 지역변수 `M`에 대해 M_V 의 회전에 해당하는 3행 3열 부행렬의 내용을 설정하고 있다.

```
M = mat4(1.0f); // Line (A)
M[0][0] = (a); M[0][1] = (b);
M[0][2] = (c); M[1][0] = (d);
M[1][1] = h.y; M[1][2] = (e);
M[2][0] = (f); M[2][1] = (g);
M[2][2] = (h);
```

여기서 Line (A)의 문장의 수행 목적은 무엇인가?

- (e) 위 코드에서 (a)에서 (h)까지 각 문장에 들어갈 내용을 `M[1][1] = h.y`; 문장을 참조하며 문법에 맞게 정확히 기술하라.
- (f) 이제 마지막으로 아래의 코드에서 `M`의 나머지 내용을 계산하여 결과 행렬을 리턴해주려 한다. 현재 이 코드는 잘못 작성되었는데 `lookAt()` 함수가 올바르게 작동할 수 있도록 오류를 수정하라.

```
M[3][0] = -dot(f, eye);
M[3][1] = -dot(g, eye);
M[3][2] = dot(h, eye);
return M;
```

8. 그림 8에 도시한 카메라의 위치와 방향에 대한 4행 4열 뷰잉변환 행렬 M_V 를 기술하라.

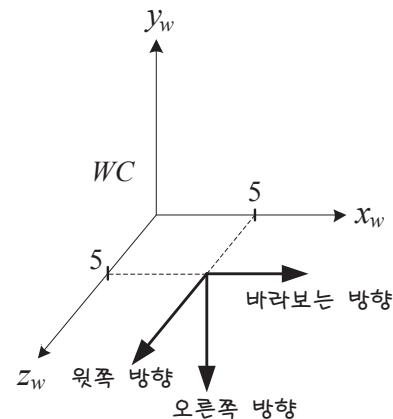


Figure 8: 카메라의 위치와 방향 설정

9. 그림 9는 COP (Center of Projection)가 $(0, 0, -1)$ 이고 PP (Projection Plane)이 $z = 5$ 인 3차원 공간에서 2차원 공간으로의 원근 투영을 도시하고 있다.

- (a) 이때 한 점 $p = (x \ y \ z)^t$ 의 깊이(depth) 값은 얼마인가? ($z > -1$ 이라 가정함)

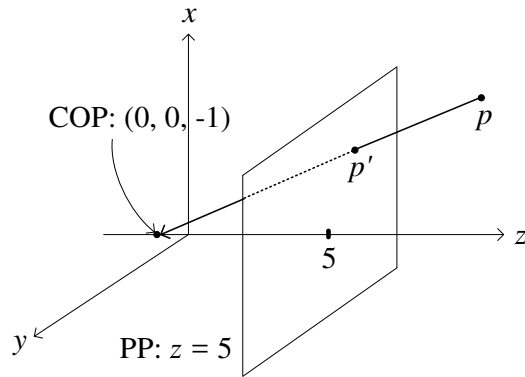


Figure 9: 원근 투영 변환

- (b) 임의의 점 $p = (x \ y \ z \ 1)^t$ 을 $p' = (x' \ y' \ z' \ 1)^t$ 로 투영해주는 4행 4열의 원근 투영 변환 행렬 M_P 를 다음과 같이 표현할 때, 나머지 열의 내용을 기술하라.

$$M = \begin{bmatrix} 1 & a & e & i \\ 0 & b & f & j \\ 0 & c & g & k \\ 0 & d & h & l \end{bmatrix}.$$

- (c) 주어진 점 $p = (4 \ 5 \ 8 \ 1)^t$ 에 대한 투영점 $p' = (x' \ y' \ z' \ 1)^t$ 의 좌표값을 아래와 같이 glm 함수를 사용하여 구하려 한다.

```
glm::vec4 p, p_prime;
glm::mat4 Mp;
```

```
p = glm::vec4(4.0f, 5.0f, 8.0f, 1.0f);
Mp = glm::mat4(1.0f, a, 0.0f, 0.0f,
               b, c, 0.0f, 0.0f,
               d, e, f, g,
               h, i, j, k);
```

```
p_prime = Mp * p;
p_prime = (Here);
```

이때 a에서 k까지 들어갈 상수값을 기술하라.

- (d) 위 코드에서 변수 p_prime에 올바른 투영점 좌표를 저장할 수 있도록 (Here) 부분에 들어갈 내용을 위 코드 문법에 맞게 정확히 기술하라.

10. 다음은 3차원 모델링 변환에 관한 문제이다.

- (a) 그림 10에서와 같이 세상 좌표계의 원점 주변에 도시한 소 모델을 적절한 모델링 변환을 통하여 세상에 배치해주는 프로그램의 일

부가 그림 11에 주어져 있다. 이 코드가 올바르게 작동하기 위하여 (A)에서부터 (H)까지 들어갈 값을 정확히 기술하라. (참고: 변수 ViewMatrix와 ProjectionMatrix에는 그 이름이 의미하는 행렬이 이미 계산되어 지정이 되어 있음)

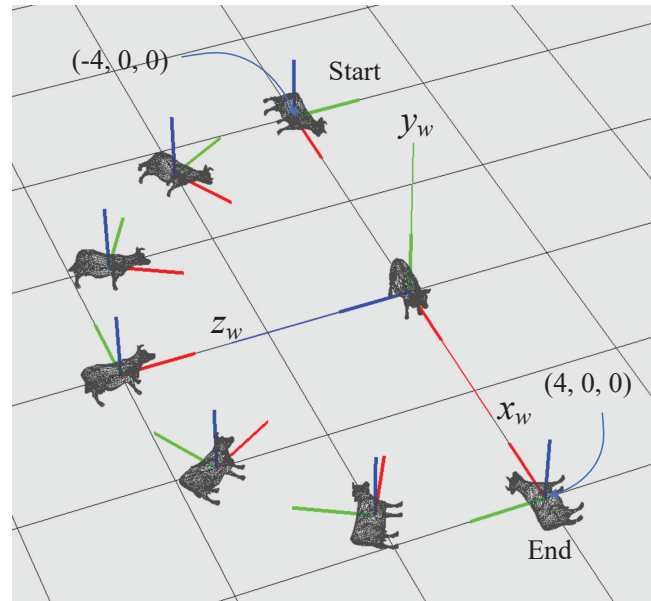


Figure 10: 3차원 모델링 변환

- (b) (위 문제에서 구한 값들에 대해) 다음 함수 호출이 리턴해주는 4행 4열 행렬을 정확히 기술하라. (힌트: 이 변환에 의해 기본 프레임이 어떻게 회전하는지 생각할 것)

```
glm::rotate(glm::mat4(1.0f),
            (E)*TO_RADIAN,
            glm::vec3((F), (G), (H)))
```

11. 다음은 3차원 공간에서의 카메라의 조작에 관한 문제이다. 그림 12의 코드를 보면서 답하라. 현재 가상의 카메라는 다음과 같이 정의가 되어 있는데,

```
enum { LMB_PRESSED = 0, RMB_PRESSED };
typedef struct _CAMERA {
    glm::vec3 pos;
    glm::vec3 uaxis, vaxis, naxis;
    float fovy, aspect, near, far;
    int move[2];
} CAMERA;
CAMERA camera;
```

주요 변수의 의미는 수업 시간에 설명하였다.


```

for (i = 0; i <= 180; i += 30) {
    float angle = (float) i;

    ModelViewMatrix = glm::rotate(ViewMatrix, (A)*TO_RADIAN, glm::vec3((B), (C), (D)));
    ModelViewMatrix = glm::translate(ModelViewMatrix, glm::vec3(-4.0f, 0.0f, 0.0f));
    ModelViewMatrix = glm::rotate(ModelViewMatrix, (E)*TO_RADIAN, glm::vec3((F), (G),
(H)));
    ModelViewProjectionMatrix = ProjectionMatrix * ModelViewMatrix;

    glUniformMatrix4fv(loc_MVPMatrix, 1, GL_FALSE, &ModelViewProjectionMatrix[0][0]);
    glLineWidth(5.0f);
    draw_axes();
    glLineWidth(1.0f);
    draw_cow(0.3f, 0.3f, 0.3f); // Cow color = (0.3, 0.3, 0.3)
}

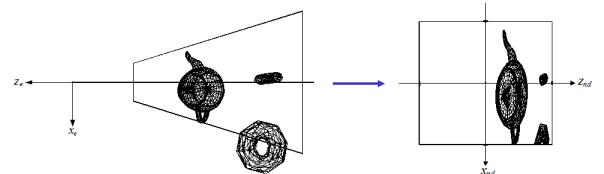
```

Figure 11: 3차원 모델링 변환 코드

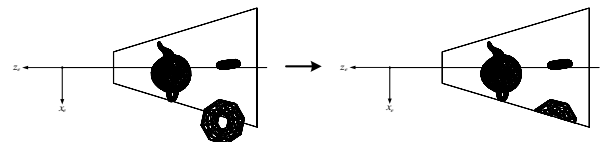
- (a) `set_ViewMatrix_from_camera_frame()` 함수는 전역 변수로 정의된 `camera`의 내용을 사용하여 `glm::mat4` 타입으로 선언한 뷰잉 변환 행렬 `ViewMatrix`를 설정하고 있다. 이때 (C), (D), 그리고 (G)에 들어갈 내용들을 이 함수의 문법에 맞게 정확히 기술하라.
- (b) 이때 (I)와 (J)에 들어갈 내용들을 이 함수의 문법에 맞게 정확히 기술하라.
- (c) `renew_cam_orientation()` 함수는 카메라 프레임을 3차원 공간의 벡터인 `axis`가 가리키는 방향으로 `angle`에 비례하는 각도만큼 회전시켜주는 역할을 한다. 이때 (Here1)에 들어갈 내용들을 이 함수 문법에 맞게 정확히 기술하라. (참고: 여러 문장일 수 있음)
- (d) 사용자가 오른쪽 마우스 버튼을 누른 후 오른쪽으로 움직일 때 시선의 방향이 오른쪽으로 회전하게 하기 위하여 (고개를 오른쪽으로 돌리듯이) (Here2)에 들어갈 내용들을 이 함수의 문법에 맞게 기술하라.
- (e) 사용자가 오른쪽 마우스 버튼을 누른 후 왼쪽으로 움직일 때 시선의 방향이 아래쪽으로 회전하게 하기 위하여 (고개를 숙이듯이) (Here3)에 들어갈 내용들을 이 함수의 문법에 맞게 기술하라.
- (f) 사용자가 왼쪽 마우스 버튼을 누른 후 왼쪽으로 움직일 때 카메라 프레임이 세상 좌표계 공간에서 어떠한 변화가 있는지 정확히 기술하라.

OpenGL의 좌표계의 이름 EC (Eye Coordinate), WdC (Window Coordinate), MC (Modeling Coordinate), CC (Clip Coordinate), 그리고 WC (World Coordinate)를 사용하여 답하라.

- (a) NDC 공간은 어느 박스와 어느 박스 사이의 지점에 해당할 지 그 기호를 기술하라.
- (b) 박스 (a)는 어느 좌표계에서 어느 좌표계로의 기하변환을 수행할까?
- (c) 어느 두 박스 사이에서 꼭지점의 좌표값이 항상 -1과 1 사이의 값만 가질 지 두 박스의 기호를 기술하라.
- (d) 다음 그림이 암시하는 렌더링 계산은 어느 박스 직전부터 어느 박스 직후까지에 해당하는지 두 박스의 기호를 기술하라.



- (e) 다음 그림이 암시하는 계산과 가장 직접적인 관련이 있는 박스의 기호를 기술하라.



12. 그림 13에 주어진 OpenG 렌더링 파이프라인의 한 부분을 보면서 답하라. 필요할 경우

```
void set_ViewMatrix_from_camera_frame(void) {
    ViewMatrix = glm::mat4( (A) , (B) , (C) , 0.0f, (D) , camera.vaxis.y,
        (E) , 0.0f, (F) , (G) , (H) , 0.0f, 0.0f, 0.0f, 0.0f, 1.0f);
    ViewMatrix = glm::translate( (I) , (J) );
}
```

(a) 뷰잉 변환 행렬의 설정

```
#define CAM_TSPEED 0.05f
void renew_cam_position(int del, glm::vec3 axis) {
    camera.pos += CAM_TSPEED * del * axis;
}

#define CAM_RSPEED 0.1f
void renew_cam_orientation(int angle, glm::vec3 axis) {
    glm::mat3 RotationMatrix;
    RotationMatrix = glm::mat3(glm::rotate(glm::mat4(1.0),
        CAM_RSPEED * TO_RADIAN * angle, axis));
    (Here1)
}
```

(b) 카메라 프레임의 재설정

```
int prevx, prevy;
void motion(int x, int y) {
    if (camera.move[LMB_PRESSED]) { // if the left mouse button is pressed down
        renew_cam_position(prevx-x, camera.uaxis);
        renew_cam_position(prevy-y, camera.naxis);
        set_ViewMatrix_from_camera_frame();
        ViewProjectionMatrix = ProjectionMatrix * ViewMatrix;
        glutPostRedisplay();
    }
    else if (camera.move[RMB_PRESSED]) { // if the right mouse button is pressed down
        renew_cam_orientation(prevx-x, (Here2) );
        renew_cam_orientation(prevy-y, (Here3) );
        set_ViewMatrix_from_camera_frame();
        ViewProjectionMatrix = ProjectionMatrix * ViewMatrix;
        glutPostRedisplay();
    }
    prevx = x; prevy = y;
}
```

(c) 모션 콜백 함수

Figure 12: 카메라의 조작

(f) 원근 투영 변환 시 원근감이 생성되는 수학 계산과 가장 밀접한 연관이 있는 박스의 기호를 기술하라.

관이 있는 박스의 기호를 기술하라.

(g) OpenGL 렌더링 과정을 사진 촬영 과정과 비교할 때, 필름에 현상한 이미지를 인화지의 특정 영역에 인화하는 작업과 가장 밀접한 연

(h) 아래의 행렬과 가장 관련이 있는 박스의 기

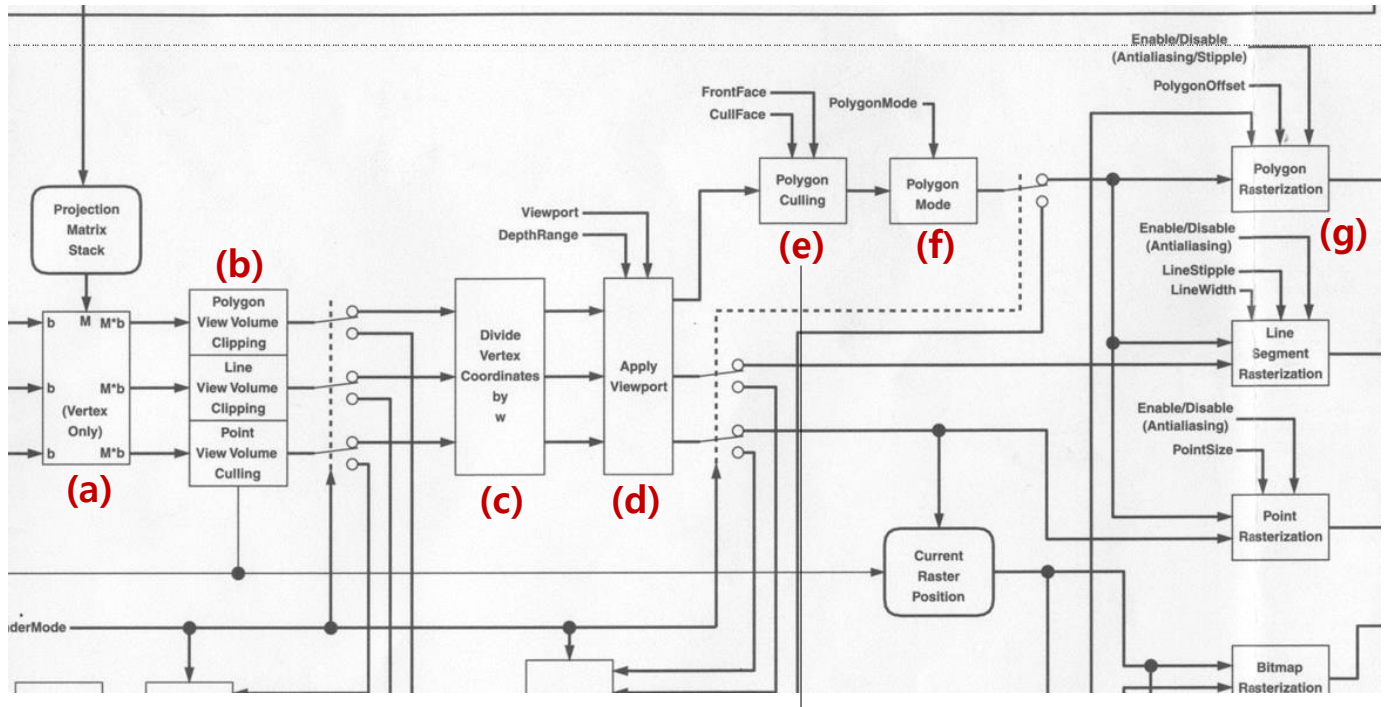


Figure 13: OpenGL 렌더링 파이프라인의 일부

호를 기술하라.

$$\begin{bmatrix} \frac{w}{2} & 0 & 0 & o_x \\ 0 & \frac{h}{2} & 0 & o_y \\ 0 & 0 & \frac{f-n}{2} & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- (i) 보편적인 원근투영 과정에서 네 번째 좌표값이 1이 아닐 수도 있는 3차원 투영 공간을 잠시 지나가는데 이 시점은 과연 어느 박스와 어느 박스 사이의 지점에 해당할 지 기호를 기술하라.

13. 다음 단답식 문제에 답하라.

- (a) 3차원 투영 공간의 점 $(6.0, -6.0, -9.0, -3.0)$ 에 해당하는 아핀 공간의 점의 좌표 (x, y, z) 를 기술하라.
- (b) 2차원 공간의 두 직선 $3x + 2y + 5 = 0$ 과 $6x + 4y + 5 = 0$ 의 교점에 대한 투영 공간에서의 동차 좌표를 상수만 사용하여 기술하라.
- (c) 투영 참조점이 투영 평면에서 유한 거리만큼 떨어진 투영변환의 이름은 무엇인가?
- (d) 모든 투영선들이 투영 평면에 수직으로 투영되는 투영변환의 이름은 무엇인가?
- (e) OpenGL 렌더링 파이프라인에서 '카메라의 위치와 방향을 설정'해주는 과정에 해당하는 기하 변환의 이름은 무엇인가?

- (f) 다음과 같은 4행 4열 아핀 변환 행렬 M 을 $M = R \cdot T$ 와 같이 이동 변환 행렬 T 와 회전 변환 행렬 R 의 곱으로 표현할 때, 4행 4열 행렬 T 과 R 의 내용을 기술하라.

$$M = \begin{bmatrix} 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 5 \\ 1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

- (g) 위 문제의 행렬 M 의 역행렬 M^{-1} 16개 원소의 값들을 정확히 기술하라. (힌트: 바로 위 문제의 답을 이용할 것)
- (h) 3차원 공간의 점을 위 문제의 아핀 변환 M 을 통하여 기하 변환 할 때, '방향과 크기'를 가지는 임의의 법선 벡터 $n = (n_x \ n_y \ n_z)^t$ 가 M 에 의해 어떻게 변환이 될 지 그 변환 후의 벡터 $n' = (n'_x \ n'_y \ n'_z)^t$ 의 좌표값을 기술하라.
- (i) 매개변수 t 를 사용하여 표현한 원점을 지나 는 직선 $L(t) = (t \cdot a_x \ t \cdot a_y \ t \cdot a_z)^t$ 를 고려하자(t 는 임의의 실수이며, 3차원 벡터 $(a_x \ a_y \ a_z)^t$ 의 길이는 1이라고 가정함). 원 점과 한 점 $p = (3.0 \ -1.0 \ 5.0)^t$ 에 의해 정의되는 선분이 주어졌을 때, p 를 $L(t)$ 에 수선을 내렸을 때의 $L(t)$ 의 점에 해당하는 t 값을 기술하라.

서강대학교