

학과 \_\_\_\_\_

학번 \_\_\_\_\_

이름 \_\_\_\_\_

## 중간고사

담당교수: 박경신

- 답은 반드시 답안지에 기술할 것. 공간이 부족할 경우 반드시 답안지 몇 쪽의 뒤에 있다고 명기한 후 기술할 것. 그 외의 경우의 답안지 뒤쪽이나 연습지에 기술한 내용은 답안으로 인정 안 함. 답에는 반드시 네모를 쳐서 확실히 표시할 것.
- 답안지에 학과, 학번, 이름 외에 본인의 암호를 기입하면 성적공고시 학번 대신 암호를 사용할 것임.

### 1. 빈칸을 채우시오. (10점)

- 1) OPENGL 3차원 좌표계는 x+는 오른쪽, y+는 위쪽, z+는 화면 밖으로 나오는 방향이다.
- 2) 프레임 버퍼에서 이미지를 생성하는데 필요한 색상 값을 갖는 버퍼를 색깔버퍼 (Color Buffer)라 한다.
- 3) 3차원 그래픽스 렌더링에서 픽셀 단위로 기하 객체의 z 값을 저장하는 버퍼를 깊이버퍼 (Depth Buffer)라 한다.
- 4) 아핀 공간에서 점과 벡터의 덧셈은 점(을)를 만든다.
- 5) 아핀 공간에서 점과 점의 뺄셈은 벡터(을)를 만든다.
- 6) 아핀 공간에서 동차좌표 (x, y, z, 0)은 벡터이다.
- 7) 아핀 공간에서 동차좌표 (x, y, z, 1)은 점이다.
- 8) 점 p가 평면 (n, d)에서 만약  $n \cdot p + d < 0$  이라면 p는 평면 안쪽에 있다.
- 9) 점 p가 평면 (n, d)에서 만약  $n \cdot p + d > 0$  이라면 p는 평면 바깥쪽에 있다.
- 10) 점들의 집합  $\{P_1, P_2, \dots, P_n\}$ 을 포함하는 가장 작은 볼록한 객체를 컨벡스 헐이라 부른다.

### 2. 다음 문제에 간단히 답하시오. (30점)

- 1) 아핀 공간 (Affine Space)과 아핀 합 (Affine Sum)을 간단히 서술하라. (5점)

아핀 공간 (Affine Space) - 아핀공간은 벡터공간에 점을 추가한 공간. 임의의 벡터 (vector)와 임의의 점(point)의 표현이 가능.

아핀 합 (Affine Sum) - 아핀 공간에서 점의 덧셈은 각 점들의 앞의 계수의 합이 1일때는 허용된다. 아핀 공간에서 n개 점의 덧셈에서 계수의 합이 1이 되는 경우를 affine sum이라고 한다.  $P = \alpha_1 P_1 + \alpha_2 P_2 + \dots + \alpha_n P_n, \alpha_1 + \alpha_2 + \dots + \alpha_n = 1$

- 2) 동차 좌표(Homogeneous Coordinates)란 무엇인가? 그리고 동차 좌표 (1, 0, -3, 2)로 표현된 3차원 공간의 점의 좌표는 무엇인가? (5점)

동차 좌표는 현재 차원의 좌표(N)에 한 차원을 추가한 좌표(N+1)로 표현을 하는 것

동차좌표 (1, 0, -3, 2) => (1/2, 0, -3/2) 3차원 공간의 점의 좌표

학과 \_\_\_\_\_ 학번 \_\_\_\_\_ 이름 \_\_\_\_\_

- 3) **싱글 버퍼(Single Buffer)와 더블 버퍼(Double Buffer)를 사용할 때의 차이점을 설명하라.**

싱글버퍼링은 전면버퍼만 사용하는 것이고, 더블버퍼링은 전면버퍼와 후면버퍼를 동시에 사용하는 것이다. 애니메이션 그래픽 렌더링을 할 때, 싱글버퍼를 사용하면 프레임 버퍼를 화면에 그리는 동안 다음 장면의 프레임버퍼를 갱신할 수 없기 때문에 지우고 다시 그려주기를 반복하면서 화면에 깜빡임이 발생한다.

더블버퍼를 사용하면 전면 버퍼가 화면에 그려지는 동안 후면 버퍼에서는 다음 장면의 프레임버퍼를 갱신하고, 둘 다 수행을 완료한 후 전면버퍼와 후면버퍼를 스왑버퍼링 (swap)을 하여 부드러운 애니메이션을 생성할 수 있다.

- 4) **다음은 Lab5~Lab8에서 사용한 Vertex Shader와 Fragment Shader이다. 한 줄 단위로 코드에 주석을 달아라. (5점)**

```
// simple4.vs
uniform mat4 gModel, gView, gProjection; // 프로그램 model, view, projection
in vec3 vPosition; // 입력 정점 위치
in vec3 inColor; // 입력 정점 색
out vec4 Color; // fragment shader로 보내는 색
void main()
{
    mat4 MVP = gProjection * gView * gModel; // 내부변수.mvp
    gl_Position = MVP * vec4(vPosition, 1); // 최종 정점위치는.mvp * vPosition
    Color = vec4(inColor, 1); // inColor를 받아 Color로 보냄
}

// simple4.fs
in vec4 Color; // vertex shader에서 입력받은 색
out vec4 fColor; // 최종 픽셀 색
void main()
{
    fColor = Color; // Color를 받아 최종픽셀 색으로 결정
}
```

- 5) **Modern OpenGL 렌더링을 하기 위해 사용하는 Vertex Array Object (VAO), Vertex Buffer Object (VBO), Index Buffer Object (IBO)이 무엇인지 간단히 설명하라. (5점)**

VAO (Vertex Array Object) – 모든 정점자료(즉, position, color, ...)를 하나로 묶어줌. 보통 하나의 Mesh (즉, 기하객체)마다 하나의 VAO를 사용함. 즉, 하나의 VAO안에는 하나 또는 여러 개의 VBO가 존재함.

VBO (Vertex Buffer Object) – 정점자료(즉, position, color, normal, 등등)의 데이터를 저장하는 메모리 버퍼. 대용량 자료를 GPU에 보내줄 수 있음.

IBO (Index Buffer Object) – 인덱스자료(즉, 정점의 index) 데이터를 저장하는 버퍼

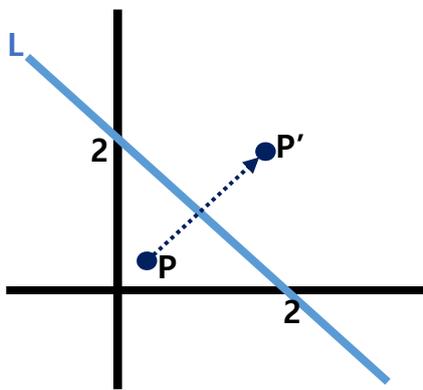
- 6) OpenGL/Glut 프로그램에서 glutIdleFunc(), glutPostRedisplay(), 함수가 무엇인지 간단히 서술하라. (5점)

glutIdleFunc()는 idle callback 함수를 등록하는 함수이다. 이벤트가 없을 때 호출되는 함수이며, 주로 애니메이션 움직임 계산 등 update를 수행한다.

glutDisplayFunc()는 display callback 함수를 등록하는 함수이다. 디스플레이 이벤트 시 호출되는 함수이며, 그리기를 수행한다. 모든 GLUT 프로그래밍에서 반드시 사용해야 하는 함수이다.

glutPostRedisplay()는 윈도우가 새로 그려져야 할 필요가 있는 경우에 호출해서 사용한다. Display callback 함수를 직접 호출하는 것보다 glutPostRedisplay() 함수를 사용하여 display callback이 다시 그리기를 지시하는 방식이다. 예를 들어 키보드나 마우스 이벤트를 사용하여 도형의 색상을 바꾸거나 움직임을 주거나 한 후 glutPostRedisplay() 함수를 사용하여 바뀌어진 도형의 색상이나 움직임으로 다시 그리기를 수행하게 한다.

3. 다음은 행렬에 관한 질문이다. (20점)

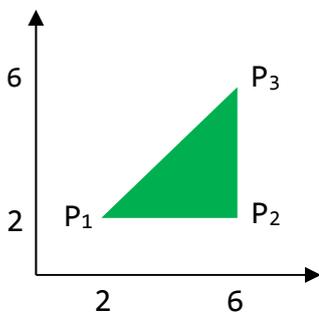


$$\begin{aligned} x' &= -y + 2 \\ y' &= -x + 2 \\ z' &= z \end{aligned}$$

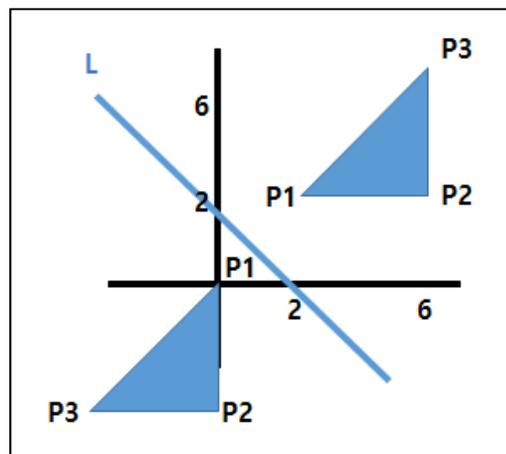
- 1) 다음은 3차원 공간의 점  $P(x, y, z)$ 를 직선 L에 대하여 반사된 점  $P'(x', y', z')$ 의 3차원 아핀 변환 행렬 M (4x4 matrix)를 구하라.

$$M = \begin{pmatrix} 0 & -1 & 2 & 0 \\ -1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- 2) 다음 삼각형에, 위의 M 행렬을 적용하여 변환된 모습을 아래 네모 칸 안에 그려서 나타내라. 삼각형의 점 위치 값을 정확히 표시한다.



$$\begin{aligned} M * p_1 (2, 2, 0) &\Rightarrow (0, 0, 0) \\ M * p_2 (6, 2, 0) &\Rightarrow (0, -4, 0) \\ M * p_3 (6, 6, 0) &\Rightarrow (-4, -4, 0) \end{aligned}$$



학과 \_\_\_\_\_ 학번 \_\_\_\_\_ 이름 \_\_\_\_\_

3) 다음 GLM 변환함수가 만들어 내는 3차원 아핀 변환 행렬 (4x4 matrix) T, R, S를 구하라.

```
glm::mat4 T = glm::translate(glm::mat4(1.0f), glm::vec3(2, 2, 0));
glm::mat4 R = glm::rotate(glm::mat4(1.0f), 45.0f, glm::vec3(0, 0, 1));
glm::mat4 S = glm::scale(glm::mat4(1.0f), glm::vec3(1, -1, 1));
```

$$T = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} R = \begin{pmatrix} 1 & -1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4) 다음은 위의 T, R, S 행렬을 아래와 같은 행렬의 곱으로 나타난 3차원 아핀 변환 행렬 M<sub>2</sub> (4x4 matrix)를 구하라.

```
glm::mat4 M2 = T * R-1 * S * R; // R-1은 R의 역행렬
```

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 0 & 2 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 2 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 0 & -1 & 0 & 2 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

학과 \_\_\_\_\_ 학번 \_\_\_\_\_ 이름 \_\_\_\_\_

4. 다음 OpenGL 코드는 Circle 클래스의 일부이다. 점의 개수가 10개 일때, `glDrawArrays(GL_POINTS,...)` 대신 `GL_LINES`, `GL_LINE_LOOP`, `GL_LINE_STRIP`, `GL_TRIANGLES`, `GL_TRIANGLE_STRIP`, `GL_TRIANGLE_FAN`, `GL_POLYGON` 을 사용했을 때 실행 결과를 그려라. (10점)

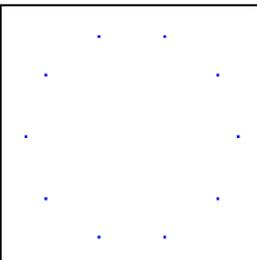
```
void Circle::init() // radius = 0.5 & slices = 8
{
    glm::vec3 vertex;

    float theta = (float) (2*M_PI/slices);
    for (int i=0; i<=slices; i++)
    {
        vertex[0] = p[0] + radius * cosf(theta * i);
        vertex[1] = p[1] + radius * sinf(theta * i);
        vertex[2] = p[2];
        vbo.addData(&vertex, sizeof(glm::vec3));
        vbo.addData(&color, sizeof(glm::vec3));
    }
    numVertices = slices;

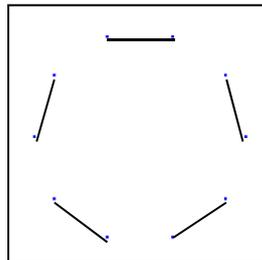
    // 중간 생략...
    isLoaded = true;
}

void Circle::draw()
{
    if (!isLoaded) return;
    glBindVertexArray(vao);
    glDrawArrays(GL_POINTS, 0, numVertices);
}
```

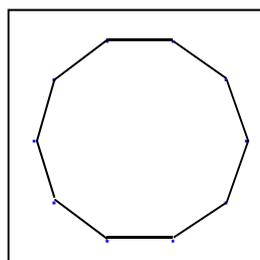
GL\_POINTS



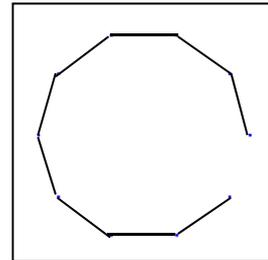
GL\_LINES



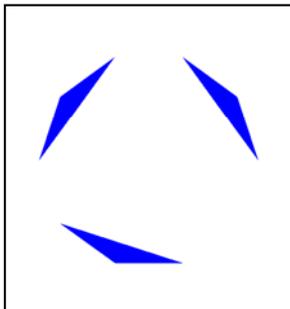
GL\_LINE\_LOOP



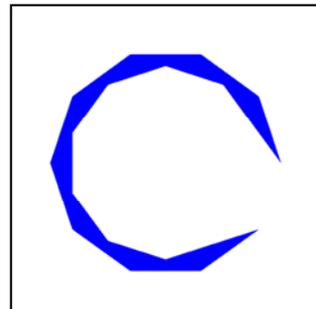
GL\_LINE\_STRIP



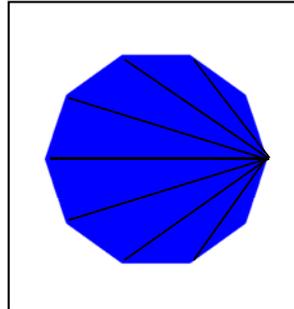
GL\_TRIANGLES



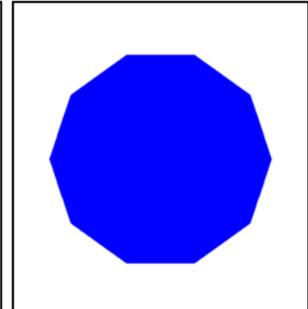
GL\_TRIANGLE\_STRIP



GL\_TRIANGLE\_FAN



GL\_POLYGON



학과 \_\_\_\_\_ 학번 \_\_\_\_\_ 이름 \_\_\_\_\_

5. 다음은 3차원 공간에서 정점  $X(6, 2, 1)$ 이 임의의 축 (arbitrary axis)  $(2, 2, 0)$ 에 대한 180도 각도 회전 Rotation Vector를 계산하는 과정이다. ( $\cos 180 = -1, \sin 180 = 0$ ) (35점)

$$\bar{a} = \left( \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right)$$

1)  $\bar{a}$ 와 정점  $X(6, 2, 1)$ 간의 외적(cross product)  $w$ 를 계산하라. (5점)

$$w = \bar{a} \times X = \left( \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right) \times (6, 2, 1) = \left( \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, -2\sqrt{2} \right)$$

2) 정점  $X(6, 2, 1)$ 이  $\bar{a}$ 에 평행한 벡터  $x_{||}$ 와 수직인 벡터  $x_{\perp}$ 를 계산하라. (5점)

$$x_{||} = (\bar{a} \cdot x)\bar{a} = 4\sqrt{2} \left( \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0 \right) = (4, 4, 0)$$

$$x_{\perp} = x - (\bar{a} \cdot x)\bar{a} = (6, 2, 1) - (4, 4, 0) = (2, -2, 1)$$

3) 정점  $X(6, 2, 1)$ 이 임의의 축  $a$ 에 180도 회전한 벡터  $R(X)$ 를 계산하라. (5점)

$$\begin{aligned} R(x) &= R(x_{||}) + R(x_{\perp}) \\ &= x_{||} + \cos \theta x_{\perp} + \sin \theta w \\ &= (4, 4, 0) + (-2, 2, -1) \\ &= (2, 6, -1) \end{aligned}$$

4) 임의의 축  $a$ 에 대해 180도만큼 회전 (rotate)하는 행렬을 구하는 공식이다. Symmetric 행렬, Skew 행렬, 최종 R 행렬을 구하라. (15점)  
 $R = I \cos \theta + \text{Symmetric} (1 - \cos \theta) + \text{Skew} \sin \theta$

$$\text{Symm} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

학과 \_\_\_\_\_ 학번 \_\_\_\_\_ 이름 \_\_\_\_\_

$$Skew = \begin{pmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & -\frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5) 정점  $X(6, 2, 1)$ 이 임의의 축  $a$  에 180도 회전한 벡터  $R(X)=R \cdot X$ 를 계산하라. (5점)

$$R(X) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ -1 \\ 1 \end{pmatrix}$$

- 끝 -