단국대학교 응용컴퓨터공학 컴퓨터 그래픽스 중간고사 (2017년 가을학기) 2016년 10월 26일 목요일

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.**

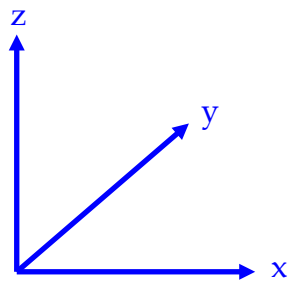
1. 다음 문제에 간단히 답하시오. (30점)

1) **아핀 공간 (Affine Space)과 벡터 공간 (Vector Space)을 간단히 서술하라.** (5점)

Affine Space – 벡터공간에 점을 추가한 공간. 임의의 벡터와 임의의 점의 표현이 가능.

Vector Space – 벡터 공간은 실수(scalar)와 벡터(vector)만 표현이 가능한 공간. 임의의 벡터 v = 기저벡터 (v1, v2, .. vn)

2) **+y 축이 화면의 안쪽으로 들어가는 방향, +z가 화면의 위쪽을 향하는 방향일 때 오른손 좌표계(Right Hand Coordinate System)에서 +x 축이 가리키는 방향이 어느 곳인지 x, y, z 축을 그리시오. 그리고 각 축에서 회전 방향 (오른손 좌표계에서는 회전은 반시계방향)을 그리시오. (5점)**



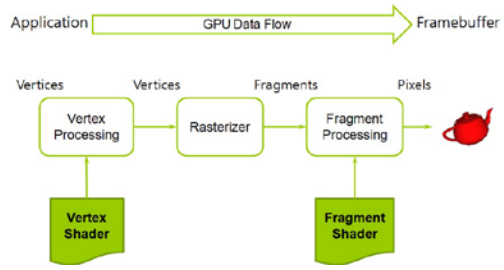
3) **Pinhole-Camera Model과 Synthetic-Camera Model에 대해 간단히 서술하라. 이 두 카메라 모델에서 물체(Object), 이미지 평면(Image Plane), 투영선(Projection), 카메라 중심(Center of Projection)을 그림으로 나타내어 비교하라. (5점)**

Object (객체)는 영상생성 과정이나 관측자와 관계없이 공간에 존재
Viewer (관측자)는 물체의 영상을 형성하는 존재. 즉, 컴퓨터 그래픽에서 카메라.
Light (광원)은 객체와 관측자가 있다 하더라도 광원이 없다면 객체는 어두워서 영상에 나타나지 않는다.

Synthetic-Camera Model (합성 카메라 모델)은 Image Plane(영상면)이 Center of Projection보다 앞에 있다. 인간시각시스템이나 핀홀카메라모델이 영상면이 카메라 후면에 있어서 영상면에 전후 좌우가 뒤바뀌어 이미지가 생기는 것과는 달리 영상이 그대로 투영되는 모델이다.

학과 _____ 학번 _____ 이름 _____

4) **Modern OpenGL (OpenGL 3.x)의 Programmable Graphics Pipeline에서 사용하는 Vertex Shader와 Fragment Shader를 설명하라. (5점)**



Programmable Graphics Pipeline은 과거 고정 그래픽스 렌더링 파이프라인을 사용할 시 불가능 하였던 다양한 실시간 렌더링 효과를 적용 가능함. OpenGL에서 객체를 최종 프레임버퍼의 이미지로 표출하는데 있어서,
Vertex shader는 정점의 정보를 받아서 예를 들어 Model-View-Projection Transformation, Morphing, Wave motion, Fractals, 또는 lighting과 정점과 관련된 속성을 바꿔서, 새로운 정점의 정보를 생성해 준다.
Fragment shader는 Texture mapping, Per-fragment lighting과 같은 잠정적 픽셀인 Fragment와 관련된 속성을 바꿔주고 최종 픽셀 이미지, 즉 프레임버퍼를 생성해 준다.

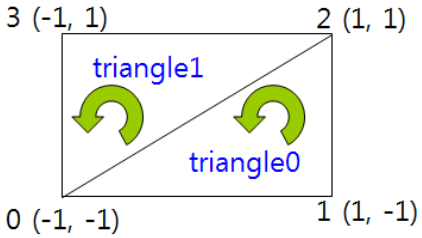
5) **Modern OpenGL 렌더링을 하기 위해 사용하는 Vertex Array Object (VAO), Vertex Buffer Object (VBO), Index Buffer Object (IBO)이 무엇인지 간단히 설명하라. (5점)**

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) **다음 OpenGL에서 삼각형 2개를 사용하여 사각형(Square)을 그릴 때, glDrawArrays (GL_TRIANGLES, 0, 6)나 glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, 0)를 사용할 수 있다. 아래의 그림을 사용하여 이 두 가지 방식의 차이점을 설명하라. (5점)**

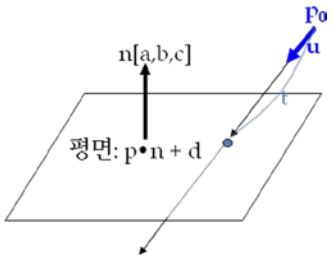


glDrawArrays(GL_TRIANGLES, 0, 6);는 Vertex 정보만을 사용해서 그리는 방식
즉, 4개의 정점 정보를 3개씩 2개의 삼각형에 대한 정점리스트(즉 6개)를 사용하여 그림을 그리는 방식

glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, 0);는 Vertex와 Index를 동시에 사용해서 그리는 방식
4개의 정점 정보를 가지고 있는 정점리스트(즉 4개)와 3개씩 2개의 삼각형에 대한 인덱스 리스트(즉 6개)를 사용하여 그림을 그리는 방식

학과 _____ 학번 _____ 이름 _____

2. 다음은 벡터(Vector), 평면(Plane), 광선(Ray)에 관한 질문이다. (25점)



$$t = \frac{-(p_0 \cdot n + d)}{u \cdot n}$$

$$p(t) = p_0 + \frac{-(p_0 \cdot n + d)}{u \cdot n} u$$

1) **벡터 (0, 3, -4)와 벡터 (4, -3, 0)의 내적(dot product)을 구하시오. (3점)**

0*4 + 3*-3 + -1*0 = -9

2) **벡터 (0, 3, -4)와 벡터 (4, -3, 0)의 외적(dot product)을 구하시오. (3점)**

(0*-3 – 4*3, -4*4 – 0*0, 0*-3 – 3*4) = (-12, -16, -12)

3) **벡터 v (4, 3, -1)가 n (0, 1, 0)에 평행한 $v_{\parallel}$를 계산하시오. (3점)**

$$v_{\parallel} = n\frac{v \cdot n}{\|n\|^2} = (0,1,0)\frac{(4,3,-1) \cdot (0,1,0)}{1^2}$$

$$= (0,1,0)\frac{(4*0 + 3*1 + -1*0)}{1}$$

$$= (0,3,0)$$

4) **벡터 v (4, 3, -1)가 n (0, 1, 0)에 수직인 $v_{\perp}$를 계산하시오. (3점)**

$$v_{\perp} = v - v_{\parallel} = (4,3,-1) - (0,3,0) = (4,0,-1)$$

5) **평면 y = 0 일 때 평면의 법선 벡터 n은 무엇인가? (3점)**

n (0, 1, 0)

6) **점 Q (2, 1, -1)은 평면 y=0의 바깥쪽, 안쪽, 아니면 평면에 있는 점인가? (3점)**

glm::dot(Q, n) + d = 2*0 + 1*1 + (-1)*0 + 0 = 1 => 평면의 바깥쪽

7) **점 Q (2, 1, -1)가 평면 y = 0에서 가장 가까운 평면의 점은 무엇인가? (3점)**

P = Q – (glm::dot(Q, n) + d)n = (2, 1, -1) – 1(0, 1, 0) = (2, 0, -1)

8) **광선 v는 시작 점이 P_0=(0, 1, -2)이고 방향은 u=(1, 1, 1)이다. 광선 v가 평면 y = 0을 만나는 점은 무엇인가? 계산 과정을 보여라. (4점)**

glm::dot(u, n) = 1*0 + 1*1 + 1*0 = 1
t = -(glm::dot(P_0, n) + d) / glm::dot(u, n) = -(0*0 + 1*1 + -2*0 + 0) / 1 = -1
t<0므로, 광선 v가 평면 x + 1 = 0과 만나는 점은 없다.

학과 _____ 학번 _____ 이름 _____

3. 다음은 행렬에 관한 질문이다. (20점)

1) 다음 GLM 변환 함수가 만들어 내는 3차원 아핀 변환 행렬 (4x4 matrix) **T, R, S**를 구하라. (cos 90 = 0, sin 90 = 1)

```
glm::mat4 T = glm::translate(glm::mat4(1.0f), glm::vec3(0, 1, 0));
glm::mat4 R = glm::rotate(glm::mat4(1.0f), 90.0f, glm::vec3(0, 0, 1));
glm::mat4 S = glm::scale(glm::mat4(1.0f), glm::vec3(-1, 1, 1));
```

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

2) 다음 3차원 아핀 변환 행렬 (4x4 matrix) **M1**을 구하라.

```
glm::mat4 M1 = T * R * S;
```

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

3) 다음 GLM 변환 함수가 만들어 내는 3차원 아핀 변환 행렬 (4x4 matrix) **M2**를 구하라.

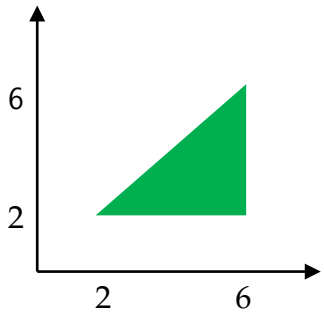
```
glm::mat4 M2 = S * R * T;
```

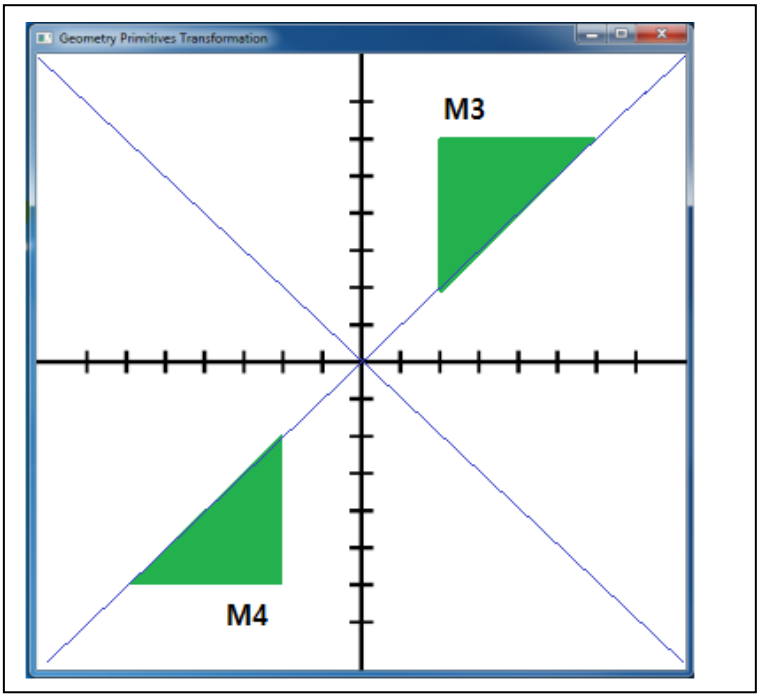
$$M_2 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

학과 _____ 학번 _____ 이름 _____

4) 다음 삼각형에, 다음 3차원 아핀 변환 행렬 (4x4 matrix) **M3**과 **M4**를 적용하여 변환된 모습을 아래 네모 칸 안에 그려서 나타내라. **삼각형의 점 위치 값을 정확히 표시한다.**





$$M_3 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

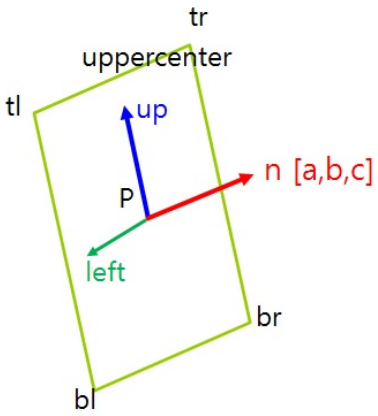
$$M_4 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

M3 * p1 (2, 2, 0) => (2, 2, 0)
M3 * p2 (6, 2, 0) => (2, 6, 0)
M3 * p3 (6, 6, 0) => (6, 6, 0)

M4 * p1 (2, 2, 0) => (-2, -2, 0)
M4 * p2 (6, 2, 0) => (-2, -6, 0)
M4 * p3 (6, 6, 0) => (-6, -6, 0)

4. 다음은 Quad 클래스 코드 일부이다. 다음 빈 칸을 채우시오. (10점)

```
void Quad::init() {
        glm::vec3 left = glm::cross(n, up);
        glm::vec3 uppercenter = ____(up * height/2.0f) + p;_____
        glm::vec3 tl = ____uppercenter + (left * width/2.0f);_____
        glm::vec3 tr = ____uppercenter - (left * width/2.0f);_____
        glm::vec3 bl = ____tl – (up * height);_____
        glm::vec3 br = ____tr - (up * height);_____
        // face1
        vbo.addData(&bl[0], sizeof(glm::vec3));
        vbo.addData(&color[0], sizeof(glm::vec3));
        vbo.addData(&br[0], sizeof(glm::vec3));
        vbo.addData(&color[0], sizeof(glm::vec3));
        vbo.addData(&tr[0], sizeof(glm::vec3));
        vbo.addData(&color[0], sizeof(glm::vec3));
// 중간 생략…
}
```

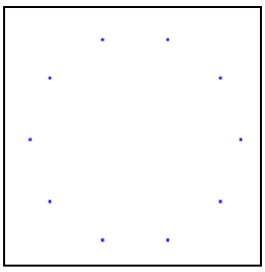


학과 _____ 학번 _____ 이름 _____

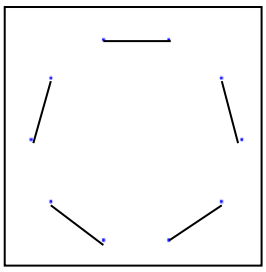
5. 다음 OpenGL 코드는 Circle 클래스의 일부이다. 만약 glDrawArrays(GL_POINTS,...) 대신 GL_LINES, GL_LINE_LOOP, GL_LINE_STRIP, GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN, GL_POLYGON 을 사용했을 때 실행 결과를 그려라. (15점)

```
void Circle::init() // radius = 0.5 & slices = 10
{
	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);
}
```

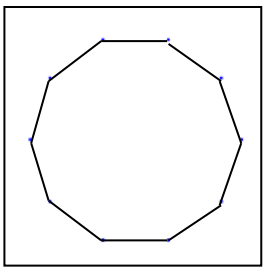
GL_POINTS



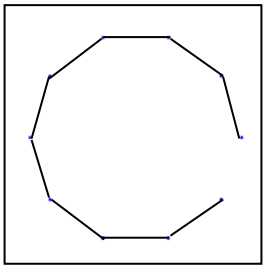
GL_LINES



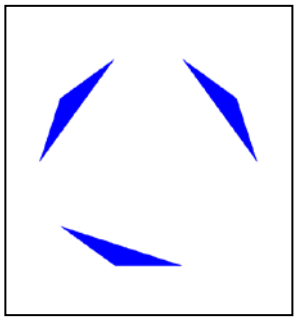
GL_LINE_LOOP



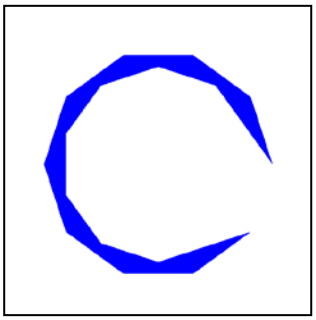
GL_LINE_STRIP



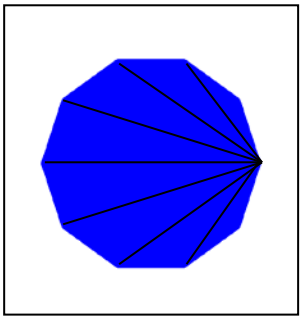
GL_TRIANGLES



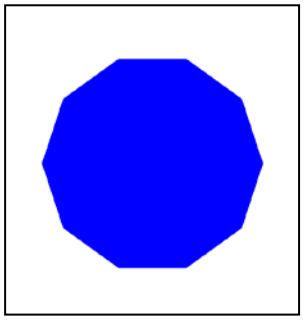
GL_TRIANGLE_STRIP



GL_TRIANGLE_FAN



GL_POLYGON



- 끝 -