단국대학교 멀티미디어공학 그래픽스 프로그래밍 중간고사 (2014년 봄학기) 2014년 4월 22일 화요일

학과 _____ 학번 _____ 이름 _____

중간고사

담당교수: 단국대학교 멀티미디어공학전공 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. 맞으면 true, 틀리면 false를 적으시오. (10점)

1) glutSpecialFunc 답신함수(Callback Function)에서 ACII 키보드 입력을 지원한다.

____F____

2) 그래픽 파이프라인에서 클리핑(Clipping)은 정점 단위로 이루어 진다. ____F____

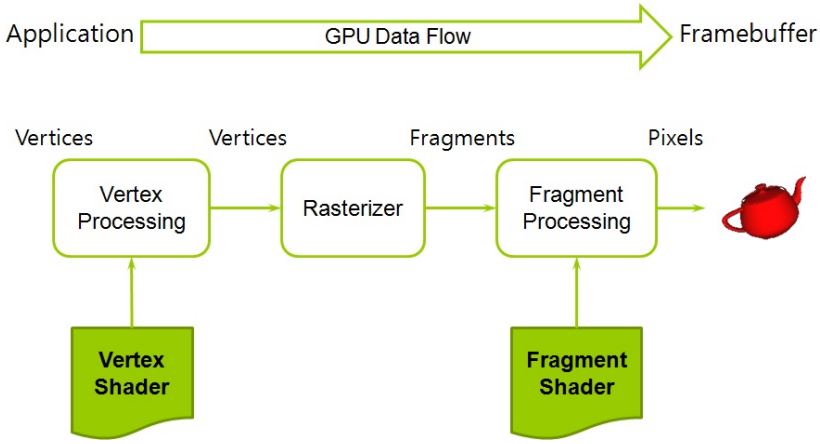
3) 아핀 공간에서는 벡터와 점의 표현이 가능하다. ____T____

4) 두 개의 3차원 벡터 u와 v에 대해서 (u x v) · u = 0를 성립한다. ____T____

5) 두 개의 3차원 벡터 u와 v에 대해서 내적 (dot product)와 외적 (cross product) 는 각각 교환법칙이 성립한다. 즉, u · v = v · u & u x v = v x u ____F____

2. 다음 문제에 답하시오. (25점)

1) Modern OpenGL (OpenGL 3.x)의 Programmable Graphics Pipeline에서 사용하는 Vertex Shader와 Fragment Shader를 설명하라. (10점)



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.

학과 _____ 학번 _____ 이름 _____

2) 3D Gasket 프로그램에서 사용한 깊이 버퍼가 무엇인지 간단히 서술하라. (5점)

일명 깊이버퍼라 불리며, 색 버퍼와 동일한 크기를 가지며, 픽셀 단위로 기하개체(geometry)의 깊이 값 (즉, Z-값)이 가장 작은 것이 앞에 그리도록 깊이 정보를 축적하는 버퍼이다.
OpenGL/GLUT에서 Z-버퍼링을 사용하기 위하여 깊이버퍼를 사용한다는 초기화가 필요하며 깊이정보 테스트를 활성화해야 하며, 디스플레이 함수 내에서 매 프레임마다 색 버퍼를 지울 때 깊이버퍼도 같이 지워야 한다.
glutInitDisplayMode(GLUT_DEPTH| ....);
glClear(GL_DEPTH_BUFFER_BIT | ....);
glEnable(GL_DEPTH_TEST);

3) 래스터 그래픽스 시스템의 특징을 서술하라. (5점)

래스터 그래픽 시스템은 현재 대부분의 컴퓨터 시스템에서 사용하는 프레임 버퍼를 사용하여, 프레임 버퍼에서 화소 (pixel)의 배열 (array)인 래스터 (raster) 이미지로 생성하여 모니터에 출력하는 방식의 시스템이다. 화소를 사용하는 래스터 이미지 방식이라서 연속적인 영상을 유한 개의 화소를 사용하는 영역으로 표현하므로 오차가 발생하는데 이를 aliasing이라고 부른다.

4) GeometryPritimives 에서 glDrawArrays (GL_TRIANGLES, 0, 6) 방식과 glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, 0) 그림 그리기 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

5) OpenGL 기하 객체의 GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN 그리기 모드의 차이점을 정점리스트(v1, v2, ~, v8)를 예로 들어 설명하라. (5점)

GL_TRIANGLES는 정점 리스트에서 v1, v2, v3가 triangle1, v4, v5, v6가 triangle2를 그린다.

GL_TRIANGLE_STRIP은 정점 리스트에서 v1, v2, v3가 triangle1, v2, v3, v4가 triangle2, v3, v4, v5가 triangle3, v4, v5, v6가 triangle4, v5, v6, v7가 triangle5, v6, v7, v8가 triangle6을 그린다.

GL_TRIANGLE_FAN은 정점 리스트에서 v1, v2, v3가 triangle1, v1, v3, v4가 triangle2, v1, v4, v5가 triangle3, v1, v5, v6가 triangle4, v1, v6, v7가 triangle5, v1, v7, v8가 triangle6을 그린다.

학과 _____ 학번 _____ 이름 _____

3. 다음 OpenGL 코드 안에 빈 칸을 채우시오. (15점)

```
void Circle::init()
{
	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;

	// create a VBO
	vbo.createVBO();
	vbo.bindVBO();
	vbo.uploadDataToGPU(GL_STATIC_DRAW);

	// create a VAO
	glGenVertexArrays(1, &vao);
	glBindVertexArray(vao);
	glEnableVertexAttribArray(0);
	glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 0, 0);
	 // 중간 생략…
	isLoaded = true;
}

void Circle::draw()
{
	if (!isLoaded) return;
	glBindVertexArray(vao);
	if (wireframe)
		glDrawArrays(GL_LINE_LOOP, 0, numVertices);
	else
		glDrawArrays(GL_TRIANGLE_FAN, 0, numVertices);
}

void Cylinder::init()
{
	numVertices = 0;
	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] - height/2.0f;
		vertex[2] = p[2] + radius * sinf(theta * i);
		vbo.addData(&vertex, sizeof(glm::vec3));
		vbo.addData(&color, sizeof(glm::vec3));
		numVertices++;

		vertex[1] = p[1] + height/2.0f;
		vbo.addData(&vertex, sizeof(glm::vec3));
		vbo.addData(&color, sizeof(glm::vec3));
		numVertices++;
	}

	// create a VBO
```

```
        vbo.createVBO();
        vbo.bindVBO();
        vbo.uploadDataToGPU(GL_STATIC_DRAW);

        // create a VAO
        glGenVertexArrays(1, &vao);
        glBindVertexArray(vao);
        glEnableVertexAttribArray(0);
        glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 0, 0);
         // 중간 생략…
        isLoaded = true;
}

void Cylinder::draw()
{
        if (!isLoaded) return;
        glBindVertexArray(vao);
        if (wireframe)
                glDrawArrays(GL_LINE_STRIP, 0, numVertices);
        else
                glDrawArrays(GL_TRIANGLE_STRIP, 0, numVertices);
}
```

4. 다음은 3차원 그래픽에서 임의의 축 (arbitrary axis) a = (1, 1, 0)에 대한 90도 회전 행렬을 유도하는 문제이다. (30점)

벡터 a (1, 1, 0)를 정규화 (normalize) 하라. (5점)

$$a(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}},0)$$

벡터 v (x, y, z)가 벡터 a에 평행한 성분 $x_{\parallel}$ = (a·v) a 을 구하라. (5점)

$$x' = \frac{1}{2}x + \frac{1}{2}y$$

$$y' = \frac{1}{2}x + \frac{1}{2}y$$

$$z' = 0$$

$$x_{\parallel}\left(\frac{1}{2}x+\frac{1}{2}y,\frac{1}{2}x+\frac{1}{2}y,0\right)$$

벡터 a와 벡터 v (x, y, z)의 외적 (cross product) = a x v를 구하라. (5점)

$$x' = \frac{1}{\sqrt{2}}z$$

$$y' = -\frac{1}{\sqrt{2}}z$$

$$z' = -\frac{1}{\sqrt{2}}x + \frac{1}{\sqrt{2}}y$$

$$a \times v = \left(\frac{1}{\sqrt{2}}z,-\frac{1}{\sqrt{2}}z,-\frac{1}{\sqrt{2}}x+\frac{1}{\sqrt{2}}y\right)$$

학과 _____ 학번 _____ 이름 _____

임의의 축 a (1, 1, 0)대해 90도만큼 회전 (rotate)하는 행렬 (matrix)는 아래의 공식을 이용하여 최종 R의 결과를 유도하라. (cos 90 = 0, sin 90 = 1) (15점)

R = *I* cosθ + *Symmetric* (1 − cosθ) + *Skew* sinθ

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

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

5. 다음 문제에 답하시오. (20점)

1) 다음 GLM 변환 함수가 만들어 내는 3차원 아핀 변환 행렬 (4x4 matrix) M1을 구하라.

```
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(0.5, 1, 1));
glm::mat4 M1 = T * R * S;
```

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0.5 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

학과 _____ 학번 _____ 이름 _____

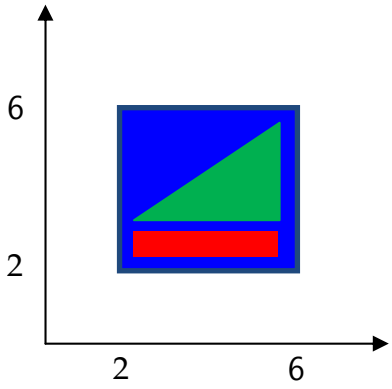
2) 다음 GLM 변환 함수가 만들어 내는 3차원 아핀 변환 행렬 (4x4 matrix) M2을 구하라.

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

$$M_2 = \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

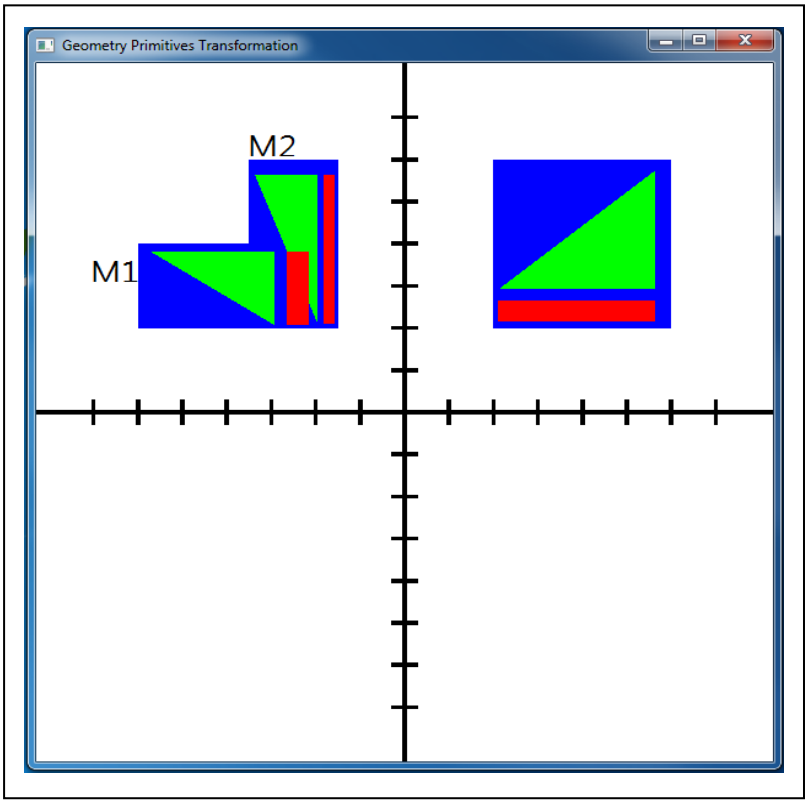
$$= \begin{pmatrix} 0 & -0.5 & 0 & -0.5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3) 다음 아래 기본 도형에, 위의 아핀 변환 행렬 **M1**과 **M2**를 적용하여 나타난 도형의 모습을 아래 네모 칸 안에 그려서 나타내라. (M1과 M2 결과를 각각 정확한 척도를 그려서 답하라) **좌측하단과 우측상단의 점 위치 값을 정확히 명시해 줄 것.**



M1 (2, 2, 0) => (-2, 2, 0)
(6, 6, 0) => (-6, 4, 0)

M2 (2, 2, 0) => (-1.5, 2, 0)
(6, 6, 0) => (-3.5, 6, 0)



4) 다음 위의 기본 도형을 GLM translate, rotate, scale 변환 함수를 사용하여, 기본 도형의 중심에서 x,y축으로 2배 크기를 키우는 코드를 작성하라.

```
glm::mat4 T1 = glm::translate(glm::mat4(1.0f), glm::vec3(-4, -4, 0));
glm::mat4 S = glm::scale(glm::mat4(1.0f), glm::vec3(2, 2, 1));
glm::mat4 T2 = glm::translate(glm::mat4(1.0f), glm::vec3(4, 4, 0));
glm::mat4 M = T2 * S * T1;
```

6. HW1의 GeometryPositionColor 클래스를 이용하여, 위의 기본 도형을 그리는 코드를 작성하라. (extra 10점)

```
Geo::Geo(glm::vec3 p_) : GeometryPositionColor()
{
	p = p_;
	quad1 = Quad();
	quad2 = Quad();
	tri = Triangle();
	init();
}
void Geo::init() // "기본 도형" 비슷하게 작성
{
	glm::vec3 p1 = p + glm::vec3(4.f, 4.f, 0.f);
	quad1.set(p1, glm::vec3(0.0f, 0.0f, 1.0f), glm::vec3(0.0f, 1.0f, 0.0f), 4.0f, 4.0f);
	glm::vec3 p2 = p + glm::vec3(3.85f, 2.4f, 0.5f);
	quad2.set(p2, glm::vec3(0.0f, 0.0f, 1.0f), glm::vec3(0.0f, 1.0f, 0.0f), 3.5f, 0.5f);
	glm::vec3 p3 = p + glm::vec3(2.1f, 2.9f, 0.5f);
	tri.set(p3, glm::vec3(0, 0, 0), glm::vec3(3.5, 0, 0), glm::vec3(3.5, 2.8, 0));
}
void Geo::draw(bool wireframe)
{
	quad1.draw();
	quad2.draw();
	tri.draw();
}
```

- 끝 -