

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) 두 개의 3차원 벡터 u 와 v 에 대해서 $(u \times v) \cdot u = 0$ 를 성립한다. ___T___
- 2) 두 개의 3차원 벡터 u 와 v 에 대해서 내적 (dot product)와 외적 (cross product) 는 각각 교환법칙이 성립한다. 즉, $u \cdot v = v \cdot u$ & $u \times v = v \times u$ ___F___
- 3) 아핀 공간 (Affine Space)에는 거리는 없으나 한 직선 위에 있다는 개념은 있으며 거리의 비가 보존된다. ___T___
- 4) 두 개의 4x4 행렬 A 와 B 의 곱에 대해서 교환법칙은 성립한다. ___F___
- 5) 반사행렬은 각도와 크기를 보존한다. ___T___

2. 다음 문제에 답하시오. (30점)

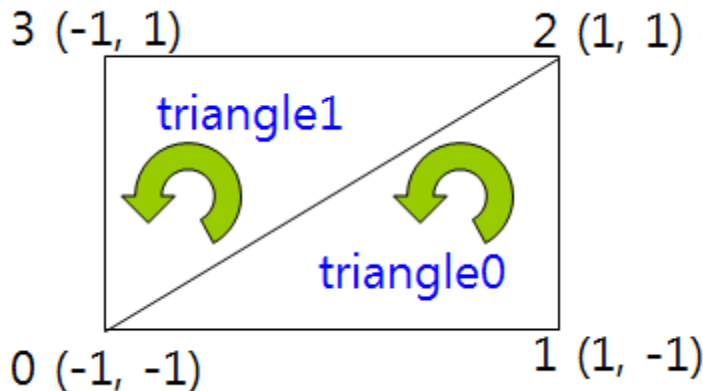
- 1) 컴퓨터 그래픽에서 3차원 정점(vertex)들로부터 3차원 객체(3D object)를 생성하는데 삼각면(triangle face)을 사용한다. 그 이유를 간단히 설명하라. (5점)

OpenGL에서 물체를 표현하는데 있어서 가장 기본이 되는 요소 (Primitive)를 그릴 때 선분들이 서로 교차하면 안되고, 볼록 다각형(convex)을 사용해야 하고, 그리고 꼭지점들이 한 평면 상에 존재해야 함. 따라서 삼각형을 사용할 때 이러한 요구 사항을 모두 만족하기 때문에 사용함

- 2) 동차 좌표 (Homogeneous Coordinates)는 무엇인가, 그리고 컴퓨터 그래픽에서 동차 좌표를 사용하는 이유는?

동차 좌표란 2차원의 점 (x,y) 의 경우의 동차좌표로는 (x, y, w) 로 표현한다. 따라서 동차 좌표에서 $(1,2,3)$ 과 $(2,4,6)$ 은 같은 동차좌표 표현이다. 3차원 점 (x, y, z) 은 동차좌표로 $(x, y, z, 1)$ 로 표현하고, 벡터 (x, y, z) 는 동차좌표로는 $(x, y, z, 0)$ 으로 표현한다.
컴퓨터 그래픽에서 동차 좌표를 사용하는 이유는, 그래픽 렌더링 파이프라인에서 각종 좌표계(예: 모델 좌표계, 세계 좌표계, 카메라 좌표계, 투영변환, 등) 변환을 수행해야 하는데, 동차좌표를 사용한다면 3차원 점과 벡터를 4차원 열행렬로 나타낼 수 있고 주어진 프레임에서 점과 벡터의 표현을 변환하는 단일함수로 변환을 정의할 수 있다. 이 동(translation), 회전(rotation), 크기변환 (scaling)등 그래픽스에서 필요로 하는 대부분의 변환은 아핀 변환이다.

- 3) 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. 이 두 가지 방식의 차이점을 설명하라. (5점)



`glDrawArrays`(`GL_TRIANGLES`, 0, 6);

4개의 정점 정보를 3개씩 2개의 삼각형에 대한 정점리스트(즉 6개)를 사용하여 그림을 그리는 방식

`glDrawElements`(`GL_TRIANGLES`, 6, `GL_UNSIGNED_INT`, 0);

4개의 정점 정보를 가지고 있는 정점리스트(즉 4개)와 3개씩 2개의 삼각형에 대한 인덱스 리스트(즉 6개)를 사용하여 그림을 그리는 방식

- 4) 위의 사각형 그리기(Filled Square) 예제에서 Wireframe으로 바뀌서 그리려면, 코드 수정을 어떻게 해야 하는가? (5점)

`GL_TRIANGLES` 를 `GL_LINE_LOOP`로 바꾼다.

`glDrawArrays`(`GL_LINE_LOOP`, 0, 6);

`glDrawElements`(`GL_LINE_LOOP`, 6, `GL_UNSIGNED_INT`, 0);

- 5) Modern OpenGL은 초기 OpenGL의 Fixed Graphics Pipeline에서 벗어나 Programmable Graphics Pipeline을 사용하고 있다. Programmable Graphics Pipeline이란 무엇인가? (5점)

NVIDIA나 ATI 등 그래픽스 카드 제조회사에서 제공한 GPU (Graphics Processing Unit)을 사용하는 구조로, 렌더링 파이프라인의 일부분을 프로그래머가 vertex shader와 fragment shader를 통하여 원하는 방식으로 프로그래밍을 할 수 있으며 과거 고정 그래픽스 렌더링 파이프라인을 사용할 시 불가능 하였던 다양한 실시간 렌더링 효과를 적용가능함.

- 6) Modern OpenGL 프로그래밍에 사용하는 Vertex Array Object와 Vertex Buffer Object란 무엇인가? 예를 들어 색깔 있는 삼각형을 두 개 그린다고 할 때 VAO와 VBO를 얼마나 생성해야 하는 지 등 (5점)

3차원 객체 하나마다 Vertex Array Object는 생성하여 사용해야 함. 만약 Colored Triangle을 두 개 그린다고 할 때 VAO는 2개 생성함.

3차원 객체를 구성하는 정점의 정보(예, 정점 위치나 색상 등)를 VBO로 생성하여 사용해야 함. 각각의 Colored Triangle마다 정점의 위치값에 대한 VBO와 색상정보에 대한 VBO를 생성한 후 `glVertexAttribPointer`를 사용하여 여러 개 지정할 수 있음.

3. 다음 OpenGL 코드는 Cylinder 클래스 함수를 보여준다. Cylinder radius=1.0, height=1.0, slices=4로 했을 때 나타나는 결과를 그림으로 그려라 (정점의 위치 값과 정점의 순서를 정확히 표시할 것). (10점)

```
Cylinder::Cylinder(float radius_, float height_, int slices_) : Geometry()
{
    radius = radius_;
    height = height_;
    slices = slices_;
    init();
}
void Cylinder::init()
{
    glm::vec3 vertex;

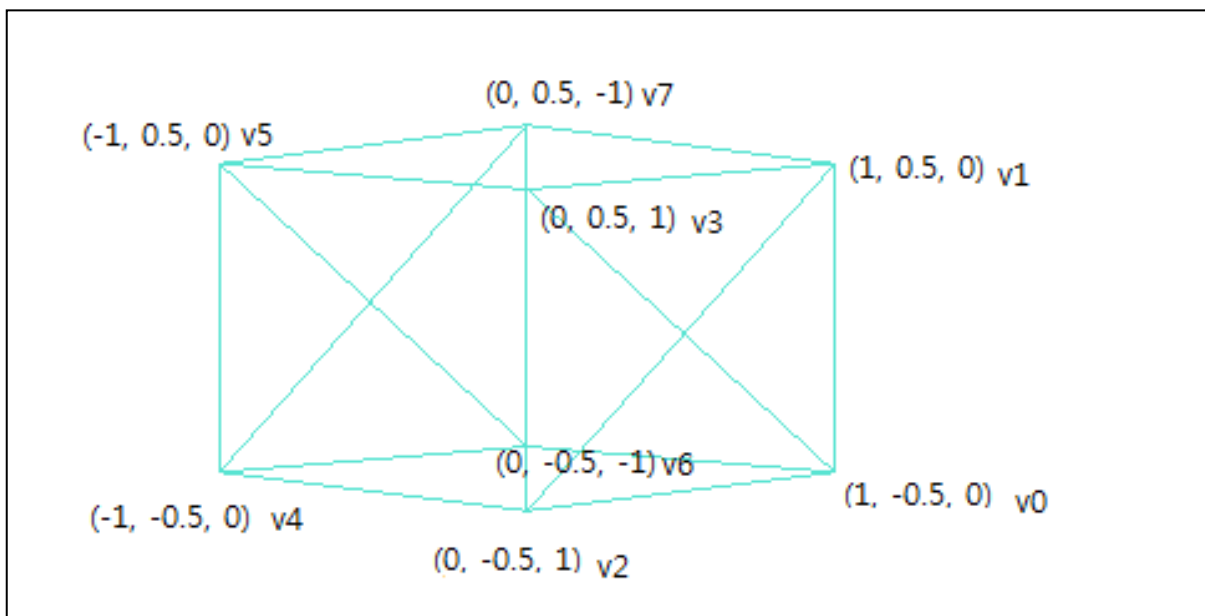
    float theta = (float) (2*M_PI/slices);
    for (int i=0; i<=slices; i++)
    {
        vertex[0] = radius * cosf(theta * i);
        vertex[1] = -height/2.0f;
        vertex[2] = radius * sinf(theta * i);
        vbo.addData(&vertex, sizeof(glm::vec3));
        numVertices++;

        vertex[1] = height/2.0f;
        vbo.addData(&vertex, 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);

    isLoaded = true;
}
void Cylinder::draw()
{
    if (!isLoaded) return;
    glBindVertexArray(vao);
    glDrawArrays(GL_TRIANGLE_STRIP, 0, numVertices); // draw cylinder
}
```



4. 다음 문제에 답하십시오. (20점)

- 1) X-축으로 180도 회전하고, Y-축으로 0도 회전 한 후, Z-축으로 180도 회전하는 행렬 $M_1(4 \times 4 \text{ 아핀변환행렬})$ 을 구하라.

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

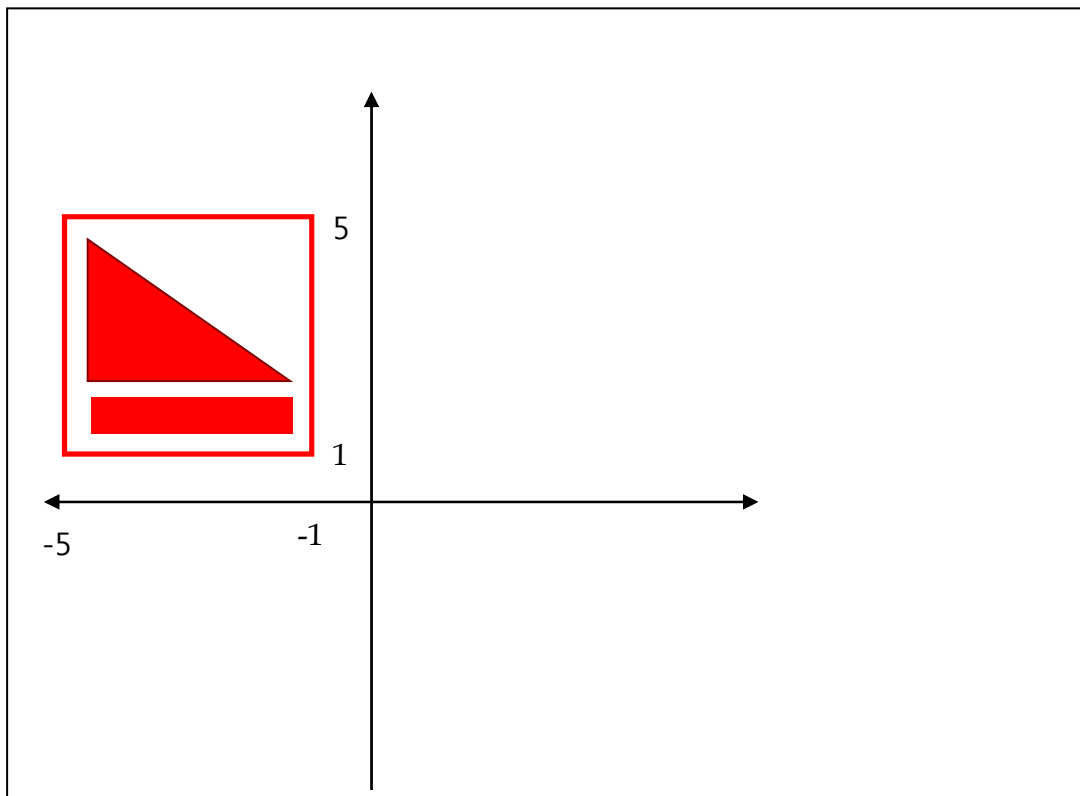
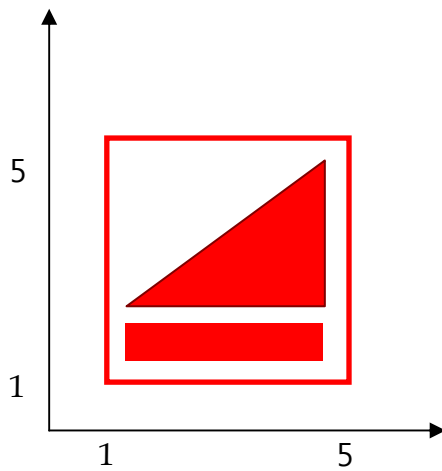
$$= \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- 2) X-축으로 0도 회전하고, Y-축으로 180도 회전 한 후, Z-축으로 0도 회전하는 행렬 $M_2(4 \times 4 \text{ 아핀변환행렬})$ 을 구하라.

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

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

- 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)



- 4) 점 $P(x, y, z)$ 를 $P'(x, -y + 1, -z)$ 로 변환시키는 행렬 $M_3(4 \times 4 \text{ 아핀변환행렬})$ 을 유도하라.

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

5. 다음은 3차원 그래픽에서 임의의 축 (arbitrary axis) $a = (2, 3, 4)$ 에 대한 θ 도 회전 행렬을 유도하는 문제이다. (20점)

임의의 축 a 에 대해 θ 만큼 회전 (rotate)하는 행렬 (matrix)는 아래의 공식과 같이 간단히 정의될 수 있다. 계산 시, a 는 단위벡터로 바꾸지 않고 그냥 사용, \sin 과 \cos 도 그대로 사용함.

$$R = I \cos\theta + \text{Symmetric} (1 - \cos\theta) + \text{Skew} \sin\theta$$

이 때, Symmetric matrix A 는 A 와 $v (x, y, z)$ 를 곱한 것 $Av = a(a \cdot v)$ 이다. A 의 결과를 유도하라.

$$x' = 4x + 6y + 8z, y' = 6x + 9y + 12z, z' = 8x + 12y + 16z$$

$$A = \begin{pmatrix} 4 & 6 & 8 & 0 \\ 6 & 9 & 12 & 0 \\ 8 & 12 & 16 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Skew matrix B 는 B 와 $v (x, y, z)$ 를 곱한 것 $Bv = a \times v$ 이다. B 의 결과를 유도하라.

$$x' = 3z - 4y, y' = 4x - 2z, z' = 2y - 3x$$

$$B = \begin{pmatrix} 0 & -4 & 3 & 0 \\ 4 & 0 & -2 & 0 \\ -3 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

최종 회전형렬 R 의 결과를 유도하라.

$$R = \begin{pmatrix} 4 - 3\cos\theta & 6 - 6\cos\theta - 4\sin\theta & 8 - 8\cos\theta + 3\sin\theta & 0 \\ 6 - 6\cos\theta + 4\sin\theta & 9 - 8\cos\theta & 12 - 12\cos\theta - 2\sin\theta & 0 \\ 8 - 8\cos\theta - 3\sin\theta & 12 - 12\cos\theta + 2\sin\theta & 16 - 15\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

6. GLUT 프로그래밍은 이벤트에 답신하는 Callbacks (답신함수) 구조를 가지고 있다. GLUT Callbacks의 예를 5개 이상 들고 사용법을 소개하라. (10점)

glutDisplayFunc(display) – void display()에서는 그리기를 수행하는 함수를 지정

glutIdleFunc(update) – void update()에서는 이벤트가 없을 때 업데이트하는 함수를 지정

glutKeyboardFunc(keyboard) – void keyboard(unsigned char key, int x, int y)에서는 키보드의 ASCII 키가 눌러졌을 때 수행하는 함수를 지정

glutSpecialFunc(specialkey) – void specialkey(int key, int x, int y)에서는 키보드의 특수키가 눌러졌을 때 수행하는 함수를 지정

glutMouseFunc(mouse) – void mouse(int button, int state, int x, int y)에서는 마우스 버튼이 눌러졌을 때 수행하는 함수를 지정

glutMotionFunc(motion) – void motion(int x, int y)에서는 마우스 버튼이 눌러진 상태에서 마우스가 움직일 때 수행하는 함수를 지정