

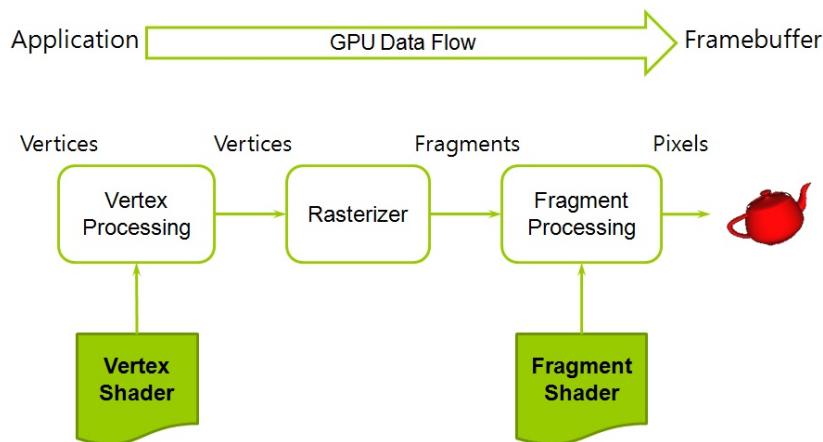
중간고사 (Extra 10%)

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 뒷쪽이나 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.

2. 다음 문제에 답하시오. (25점)

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



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

- 3) 래스터 그래픽스 시스템의 특징을 서술하라. (5점)
- 4) GeometryPrimitives 에서 glDrawArrays (GL_TRIANGLES, 0, 6) 방식과
glDrawElements(GL_TRIANGLES, 6, GL_UNSIGNED_INT, 0) 그림 그리기 방식의 차이점을
설명하라. (5점)
- 5) OpenGL 기하 객체의 GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN 그리기
모드의 차이점을 정점리스트(v1, v2, ~, v8)를 예로 들어 설명하라. (5점)

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] = _____
        vertex[1] = _____
        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(_____, _____, _____);
    else
        glDrawArrays(_____, _____, _____);
}

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] = _____;
        vertex[2] = p[2] + radius * sinf(theta * i);
        vbo.addData(&vertex, sizeof(glm::vec3));
        vbo.addData(&color, sizeof(glm::vec3));
        numVertices++;

        vertex[1] = _____;
        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(_____, _____, _____);
    else
        glDrawArrays(_____, _____, _____);
}
```

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)

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

벡터 a 와 벡터 $v (x, y, z)$ 의 외적 (cross product) $= a \times v$ 를 구하라. (5점)

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

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

$$Symm = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix} \quad Skew = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

$$R = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

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;`

$$M_1 = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

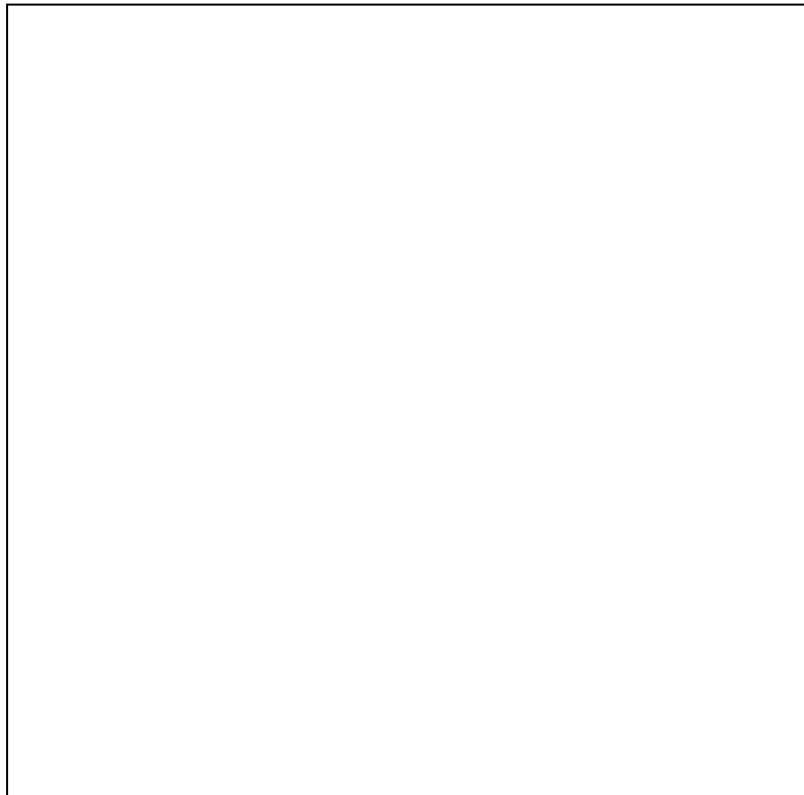
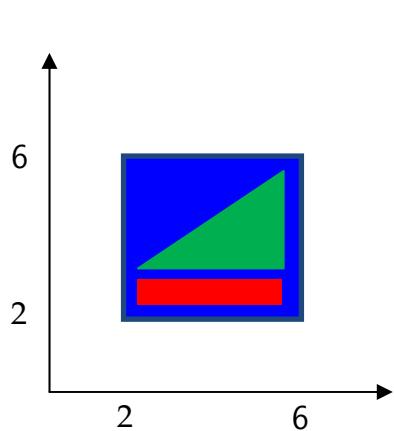
$$= \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

- 2) 다음 GLM 변환 함수가 만들어 내는 3차원 아핀 변환 행렬 (4x4 matrix) M2을 구하라.
`glm::mat4 M2 = S * R * T;`

$$M_2 = \begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

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

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



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

6. HW1의 **GeometryPositionColor** 클래스를 이용하여, 위(4번 문제)의 기본 도형을 그리는 코드를 작성하라. (10점)

```
Triangle(glm::vec3 p=glm::vec3(0, 0, 0), glm::vec3 v1=glm::vec3(-1, -1, 0), glm::vec3 v2=glm::vec3(1, -1, 0),  
glm::vec3 v3=glm::vec3(0, 1, 0));
```

```
Quad(glm::vec3 p=glm::vec3(0, 0, 0), glm::vec3 n=glm::vec3(0, 0, 1), glm::vec3 u=glm::vec3(0, 1, 0), float  
width=1.0f, float height=1.0f);
```

```
Geo::Geo(glm::vec3 p_) : GeometryPositionColor()  
{
```

```
p = p_-;  
quad1 = Quad();  
quad2 = Quad();  
tri = Triangle();  
init();  
}  
void Geo::init() // "기본 도형" 비슷하게 작성  
{  
}  
}  
void Geo::draw(bool wireframe)  
{  
    quad1.draw();  
    quad2.draw();  
    tri.draw();  
}
```

- 끝 -