

기말고사

담당교수: 박경신

12/20 23:59까지 online.dankook.ac.kr 이러닝으로 학번_이름_final.zip 으로 묶어서 이러닝에 제출한다.
 주의: 단순 번역이나 있는 그대로 복사/붙이기는 금지함. 반드시 본인이 참고한 문서의 인용을 넣어줌.

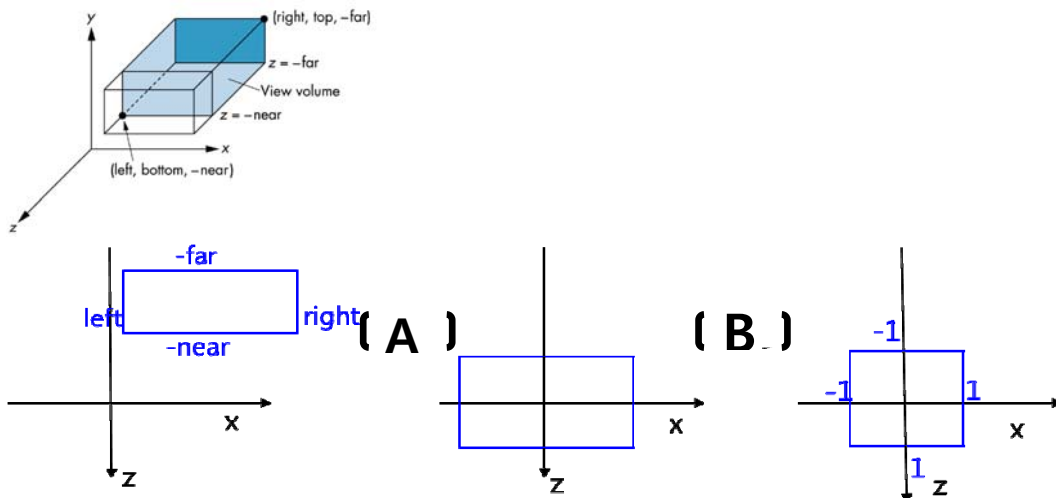
각 문제당 1장 이상씩 서술한다. 장수 제한 없음.

1. 다음 문제에 간단히 답하시오. (10점)

- 1) 버텍스 셰이더(vertex shader) 코드에서 in, out, uniform 변수란 무엇인가? 프래그먼트 셰이더 (fragment shader) 코드에서 in, out, uniform 변수란 무엇인가?
- 2) 직교 투영 (Orthographic Projection), 축측 투영 (Axonometric Projection), 등축투영 (Isometric Projection), 경사 투영 (Oblique Projection), 원근 투영 (Perspective Projection)의 특징을 자세히 서술하라
- 3) Flat Shading, Gouraud Shading, Phong Shading을 자세히 비교 서술하라.
- 4) Texture Mapping의 축소, 확대, mip맵필터를 자세히 비교 서술하라..

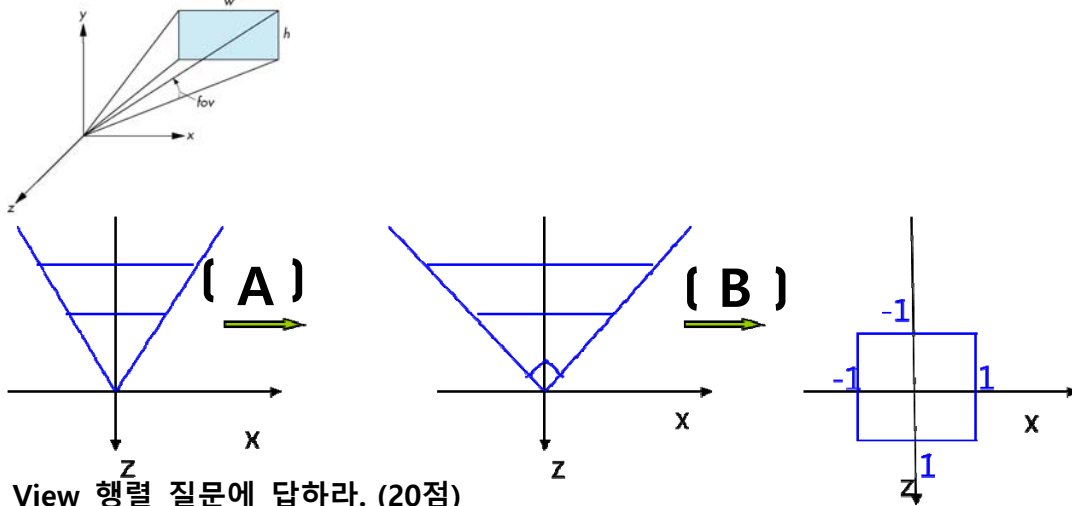
2. 다음 Projection 행렬 질문에 답하라. (25점)

- 1) 원근 정규화에 대해 설명하라. (5점)
- 2) glm::ortho(0, 10, 0, 10, 0, 10) 함수가 직육면체 관측공간에서 정규화된 관측 공간으로 변환시키는 직교 투영 행렬 (orthographic projection matrix)을 생성해 내는 과정을 보여주고 있다. 각 단계별로 필요한 (A)와 (B) 행렬 그리고 최종 행렬 ortho(4x4)을 계산하라. (10점)



- 3) glm::perspective(M_PI/4, 1, 1, 100) 함수가 절두체에서 정규화된 관측 공간으로 변환시키는 원근 투영 행렬 (perspective projection matrix)을 생성하는 과정을 보여주고 있다. 각 단계별로 필요한 (A)와 (B) 행렬 그리고 최종 행렬 perspective(4x4)을 계산하라. (10점)

$$\text{glm::perspective}(M_PI/4, 1, 1, 100) = \text{glm::frustum}(-1/\tan(22.5), 1/\tan(22.5), -1/\tan(22.5), 1/\tan(22.5), 1, 100)$$



3. 다음 View 행렬 질문에 답하라. (20점)

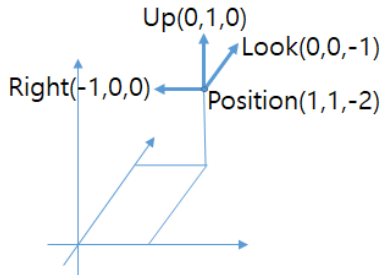
- 1) 다음 코드에서 밑줄 친 변환함수인 glm::translate, glm::rotate 함수를 제거하고, 화면에 동일하게 View가 나타나도록 glm::lookAt 사용하여 View 행렬 코드를 작성하라. (5점)

```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    spMain.useProgram();
    View = glm::mat4(1.0f); // glm::lookAt(_____);
    spMain.setUniform("gView", View);
    glm::mat4 T = glm::translate(glm::mat4(1.0f), glm::vec3(0, 0, -10));
    glm::mat4 R = glm::rotate(glm::mat4(1.0f), -M_PI/2.0, glm::vec3(0, 1, 0));
    World = T * R;
    spMain.setUniform("gModel", World);
    drawTeapot();
    glutSwapBuffers();
}
```

- 2) 위와는 반대로, 다음 코드에서 밑줄 친 glm::lookAt 함수를 제거하고, 화면에 동일하게 View가 나타나도록 변환함수인 glm::translate, glm::rotate 사용하여 World 행렬 코드를 작성하라. (5점)

```
void display(void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    spMain.useProgram();
    View = glm::lookAt(glm::vec3(0, 0, 15), glm::eye(0, 0, 0), glm::vec3(0, 1, 0));
    spMain.setUniform("gView", View);
    glm::mat4 T = glm::mat4(1.0f);
    glm::mat4 R = glm::mat4(1.0f);
    World = T * R;
    spMain.setUniform("gModel", World);
    drawTeapot();
    glutSwapBuffers();
}
```

- 3) 아래의 카메라 위치와 방향에서 90도 YAW 회전한 후에 다시 90도 Pitch 회전한 View 행렬을 계산하라. 최종 회전 후의 Right, Up, Look 벡터의 방향을 그림으로 도식화하고 View 행렬을 유도하라. (10점)



4. 다음은 조명 (lighting) 공식에 관한 문제이다. 아래의 질문에 답하시오. (20점)

$$I = K_a I_a + \sum_{i=0}^{m-1} f_{att}(d) \{ K_d I_d (N \cdot L) + K_s I_s (R \cdot V)^n \} + E$$

- 1) 조명공식에서 램버트 법칙 (Lambertian Law)을 자세히 설명하라. 이 부분에 대한 셰이더 코드를 적어라. (5점)
- 2) 블린 (Blinn) 직접 조명 모델을 설명하라. 블린 모델의 경우 위의 공식에서 어떤 부분을 바꿔서 사용하는지, 공식을 적고 자세히 설명하라. (5점)
- 3) (정점이 4개인) Quad와 (정점이 10000개인) Grid를 각각 Gouraud Shading과 Phong Shading를 사용했을 때 실행화면의 차이점을 설명하라. 각각 실행결과 화면(wireframe mode와 solid drawing mode 모두)과 코드의 차이점을 보여라. (10점)

5. 다음 OpenGL 텍스처 매핑(Texture Mapping)과 블렌딩(Blending) 질문에 답하라. (15점)

- 1) Quad의 각 꼭지점 p1(-2,-2,0), p2(2,-2,0), p3(2,2,0), p4(-2,2,0)에 텍스처 좌표(texture coordinates)를 t1(-1,-1), t2(0,-1), t3(0,0), t4(-1,0)로 지정했을 때, texture wrapping 방식 (1)GL_REPEAT, (2) GL_MIRRORED_REPEAT, (3)GL_CLAMP_TO_EDGE 따른 각각 출력 결과 실행화면을 보여라. 그리고 t1(-2,-2), t2(0,-2), t3(0,0), t4(-2,0)로 지정했을 때, texture wrapping 방식 (4)GL_REPEAT, (5) GL_MIRRORED_REPEAT, (6)GL_CLAMP_TO_EDGE 따른 각각 출력 결과 실행화면을 보여라. (opengl.jpg 사용할 것) (10점)
- 2) Blending을 사용한 Multi-texturing의 원리를 설명하라. 여기에서 사용한 glBlendFunc(GL_ONE, GL_ONE); 또는 glBlendFunc(GL_ZERO, GL_SRC_COLOR); 블렌딩 함수는 각각 무엇인가? (5점)

6. 본인의 이름(모빌 예제처럼 계층적 구조를 가지고, 움직이는)을 각각 geometryPositionColor, geometryPositionNormal, geometryPositionNormalTexture을 사용하여 프로그램을 만들어서 실행결과 화면을 보여라. source code와 실행파일을 포함한 프로젝트 파일 전체를 같이 제출하라. (30점)

- 끝 -