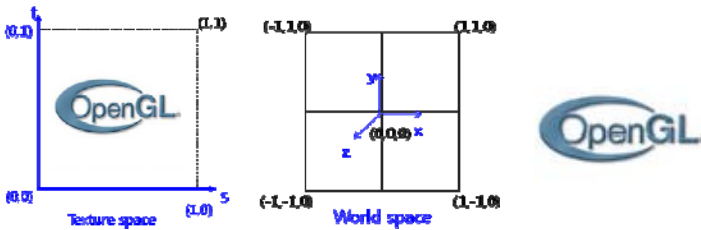


기말고사

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. 다음은 OpenGL 환경에서 텍스처 매핑과 블렌딩에 관한 문제이다. 아래의 질문에 답하시오. (45점)



1) 다음은 Quad 메쉬를 그리는 OpenGL 프로그램의 일부이다. 위 결과 그림과 같이 나타나게 하기 위하여 빈 칸의 텍스처 좌표를 채우시오. (10점)

```
void drawTextureQuad()
{
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, textureID);
    glBegin(GL_QUADS);
    glNormal3f(0, 0, 1);
        glTexCoord2f(0.0, 0.0);
        glVertex3f(-1, -1, 0);
        glTexCoord2f(0.5, 0.0);
        glVertex3f( 0, -1, 0);
        glTexCoord2f(0.5, 0.5);
        glVertex3f( 0,  0, 0);
        glTexCoord2f(0.0, 0.5);
        glVertex3f(-1,  0, 0);

        glTexCoord2f(_____, _____);
        glVertex3f(-1,  0,  0);
        glTexCoord2f(_____, _____);
        glVertex3f( 0,  0,  0);
        glTexCoord2f(_____, _____);
        glVertex3f( 0,  1,  0);
        glTexCoord2f(_____, _____);
        glVertex3f(-1,  1,  0);

        glTexCoord2f(_____, _____);
        glVertex3f( 0, -1,  0);
        glTexCoord2f(_____, _____);
        glVertex3f( 1, -1,  0);
```

```

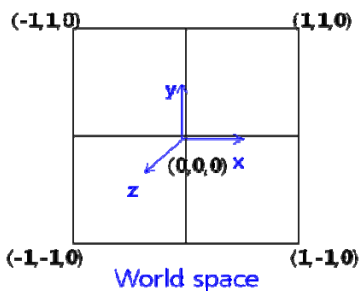
glTexCoord2f(_____, _____);
glVertex3f( 1,  0,  0);
glTexCoord2f(_____, _____);
glVertex3f( 0,  0,  0);

glTexCoord2f(_____, _____);
glVertex3f( 0,  0,  0);
glTexCoord2f(_____, _____);
glVertex3f( 1,  0,  0);
glTexCoord2f(_____, _____);
glVertex3f( 1,  1,  0);
glTexCoord2f(_____, _____);
glVertex3f( 0,  1,  0);
glEnd();
glDisable(GL_TEXTURE_2D);
}
    
```

- 2) 1)번 문제의 Quad 메쉬 OpenGL 프로그램에 텍스처 랩 (wrap) 모드는 GL_CLAMP을 사용하고 모두 동일한 렌더링 인자를 사용하는 상황에서, 아래의 OpenGL 코드에서 보이는 텍스처 행렬 스택만 조작하여 삽입하였을 때 나타나는 결과 화면을, 아래 그림 위에 표시하고 설명하라. (10점) (힌트: 정점은 그대로인 상태에서 텍스처 좌표계만 변환)

```

glMatrixMode(GL_TEXTURE);
glTranslatef(0.5, 0.5, 0);
glScalef(2, 2, 1);
glTranslatef(-0.5, -0.5, 0);
drawTextureQuad();
    
```



- 3) 1) 2)번 문제의 Quad 메쉬를 그리는 OpenGL 프로그램에서, 만약 조명 (Lighting)이 작동되지 않는다면, 무엇이 문제인지 해결 방법을 자세히 설명하라. (10점) (힌트: 텍스처 매핑을 위한 환경 변수 설정)

4) 다음은 OpenGL Extension에서 제공하는 Point Sprites에 대해 자세히 설명하라. (5점)

5) 다음은 블렌딩 방식에 따른 SourceFactor와 DestinationFactor값을 빈 칸에 적어라. (5점)

블렌딩 공식: $C = \text{SourceFactor} * C_s + \text{DestinationFactor} * C_d$

알파 블렌딩 (alpha blending): $C = (S_a, S_a, S_a, S_a) * C_s + (1-S_a, 1-S_a, 1-S_a, 1-S_a) * C_d$

SourceFactor는 Source의 alpha값인 (S_a, S_a, S_a, S_a)

DestFactor는 Source의 inverse alpha값인 (1-S_a, 1-S_a, 1-S_a, 1-S_a)

블렌딩 공식에서 C_s와 C_d는 무엇인가?

C_s = _____

C_d = _____

덧셈 블렌딩 (addition blending) 의 경우

SourceFactor는 _____

DestinationFactor는 _____

C = _____

곱셈 블렌딩 (multiply blending)의 경우,

SourceFactor는 _____

DestinationFactor는 _____

C = _____

블렌딩을 설정한 채 불투명한 것을 그리고자 할 (no blending) 경우

SourceFactor는 _____

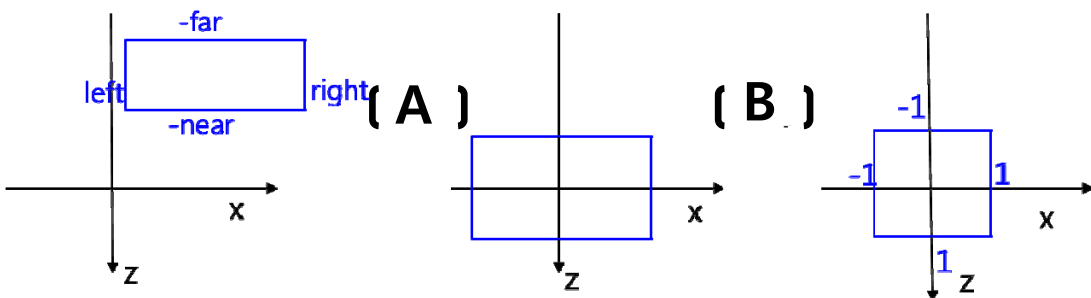
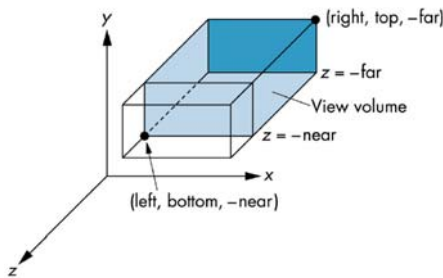
DestinationFactor는 _____

C = _____

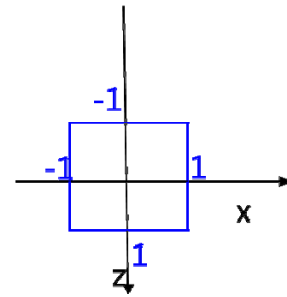
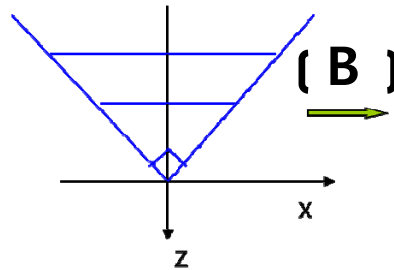
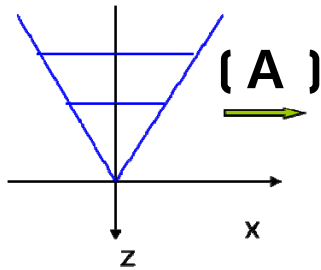
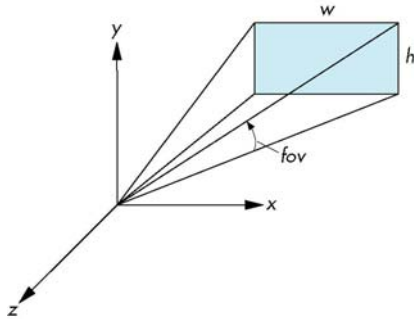
6) 블렌딩을 이용한 라이트 매핑(light mapping)은 어떻게 만드는지 자세히 설명하라. (5점)

2. 관측과 관련된 다음 문제에 답하시오. (25점)

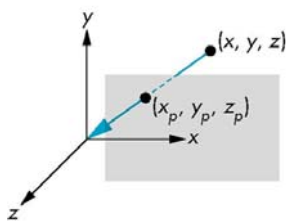
1) 다음은 `glOrtho(left, right, bottom, top, near, far)` 함수가 직육면체 관측공간에서 정규화된 관측 공간으로 변환시키는 직교 투영 행렬 (orthographic projection matrix)을 생성해 내는 과정을 보여주고 있다. 각 단계별로 필요한 (A)와 (B) 변환행렬을 자세히 설명하시오. (5점)



- 2) 다음은 gluPerspective(fov, aspect, near, far) 함수가 절두체에서 정규화된 관측 공간으로 변환시키는 원근 투영 행렬 (perspective projection matrix)을 유도하는 과정을 보여주고 있다. 각 단계별로 필요한 (A)와 (B) 변환행렬을 자세히 설명하시오. (5점)



- 3) 다음 M_{pers} 는 3차원 공간의 한 점 $p(x, y, z)$ 를 투영면의 한 점 $q(x, y)$ 로 투영하는 원근 투영 행렬(Perspective Projection Matrix)이다. M_{pers} 4x4 행렬을 유도하라. (5점)



$$p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \xrightarrow{M_{pers}} q = \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix}$$

$$M_{pers} = \left(\begin{array}{c} \\ \\ \\ \end{array} \right)$$

- 4) 다음 OpenGL 코드에서 밑줄 친 (A) gluLookAt 함수를 제거하고, 화면에 동일하게 그림이 나타나도록 OpenGL 기본 변환함수인 glTranslatef, glRotatef, glScalef 등을 사용하여 코드를 작성하라. (5점)

```
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0); // (A)
    glutWireTeapot(2);
    glutSwapBuffers();
}
```

- 5) Depth Fighting이 무엇인지 간단히 서술하시오. (5점)

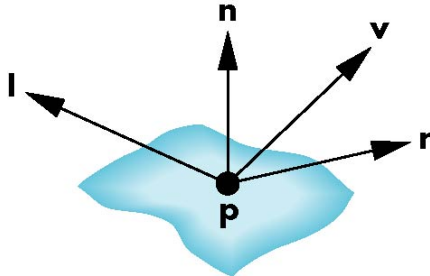
3. 다음은 조명 (lighting)과 셰이딩 (shading)에 관한 문제이다. 아래의 질문에 답하시오. (30점)

- 1) 다음은 Phong 조명 모델 (Phong Illumination Model) 공식을 보여주고 있다. 램버트 코사인 법칙 (Lambert's Cosine Law)이 무엇인지 설명하고 이 공식에서 어느 변수인지 서술하라. (5점)

$$I = K_a I_a + \sum_{i=0}^{m-1} \frac{1}{a + bd + cd^2} \{ K_d I_d (N \cdot L) + K_s I_s (R \cdot V)^\alpha \} + E$$

- 2) 이 공식에서 $R \cdot V$ 대신 다른 변수를 사용하여 표현하라. (힌트: OpenGL에서 기본으로 사용하는 조명 모델) (5점)

- 3) 카메라의 위치 $V (2.0, 2.0, 1.0)$ 이고, 광원의 위치가 $(-2.0, 3.0, -1.0)$ 이고, 물체 표면의 점 좌표 $P (0.0, 0.0, 0.0)$ 이고, 표면의 법선 벡터 $N (0.0, 1.0, 0.0)$ 일 때, 정반사 계산에 쓰이는 반사 벡터 R 의 값을 구하는 계산과정을 보여라. (5점)



- 4) OpenGL에서 기본값으로 설정된 셰이딩(Shading)이 무엇인지 간단히 설명하라. (5점)
- 5) 다음 OpenGL 코드에서 LIGHT0와 GL_LIGHT_MODEL_AMBIENT 광원은 무엇인지(색과 위치) 설명하고, 광원과 재질의 ambient, diffuse, specular는 어떻게 사용되는 지 간단히 설명하라. (10점)

```
void init(void)
{
    GLfloat ambient[] = {0.0, 0.0, 0.0, 1.0};
    GLfloat diffuse[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat specular[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat position[] = {0.0, 3.0, 3.0, 0.0};
    GLfloat lmodel_ambient[] = {0.2, 0.2, 0.2, 1.0};
    glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, specular);
    glLightfv(GL_LIGHT0, GL_POSITION, position);
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel_ambient);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
}

void renderTeapot(GLfloat x, GLfloat y,
    GLfloat ambr, GLfloat ambg, GLfloat ambb,
    GLfloat difr, GLfloat difg, GLfloat difb,
    GLfloat specr, GLfloat specg, GLfloat specb, GLfloat shine)
{
```

```
GLfloat mat[4];
glPushMatrix();
glTranslatef(x, y, 0.0);
mat[0] = ambr; mat[1] = ambg; mat[2] = ambb; mat[3] = 1.0;
glMaterialfv(GL_FRONT, GL_AMBIENT, mat);
mat[0] = difr; mat[1] = difg; mat[2] = difb;
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat);
mat[0] = specr; mat[1] = specg; mat[2] = specb;
glMaterialfv(GL_FRONT, GL_SPECULAR, mat);
glMaterialf(GL_FRONT, GL_SHININESS, shine * 128.0);
glCallList(teapotList);
glPopMatrix();
}
```

-끝-