

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 **암호**를 기입하면 성적공고시 학번 대신 암호를 사용할 것임.

1. 다음은 oglclass에서 제공하는 matrix4x4 클래스를 이용하여 변환을 수행한 예를 보여주고 있다. 이 코드에서 밑줄 친 `glMultMatrixf()` 부분을 `glTranslatef()`, `glRotatef()`, `glScalef()` 함수로 대체하여 작성하라 (10점).

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    drawObject();           // 아래 왼쪽 기본 도형

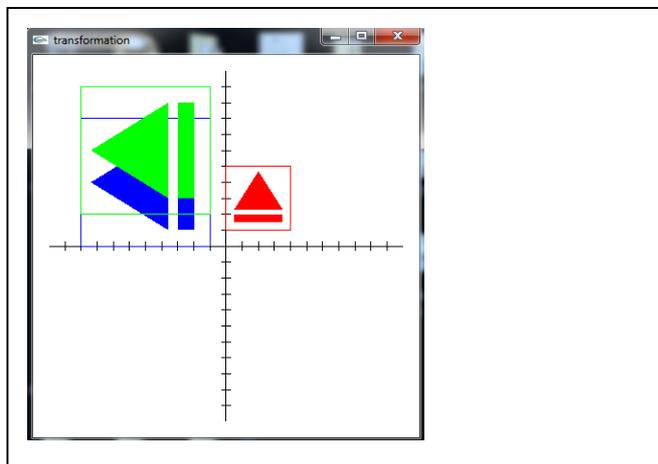
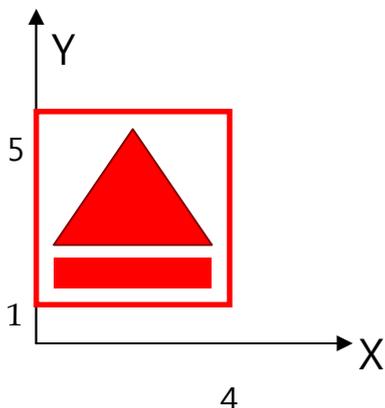
    matrix4x4 M, T, R, S;
    T.translate(vector3(1, 0, 0));
    R.rotate(90, 'z');
    S.scale(vector3(2, 2, 1));
    M = T * R * S;
    glPushMatrix();
    glMultMatrixf(M);
    drawObject();
    glPopMatrix();

    glPushMatrix();
    glMultMatrixf(M);
    glMultMatrixf(T);
    drawObject();
    glPopMatrix();
}
```

```
glTranslatef(1, 0, 0);
glRotatef(90, 0, 0, 1);
glScalef(2, 2, 1);
```

```
glTranslatef(1, 0, 0);
glRotatef(45, 0, 0, 1);
glScalef(2, 2, 1);
glTranslatef(1, 0, 0);
```

그리고 다음 왼쪽 기본 도형에, 위의 최종 실행 결과를 아래 네모 칸 안에 정확한 척도를 그려서 나타내라 (10점).



2. 다음은 관측 (Viewing) 변환과 투영 (Projection) 변환에 관한 질문이다 (20점).

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(90, 1, 1, 100);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef(0, 0, -10);
    glRotatef(-90, 0, 1, 0);

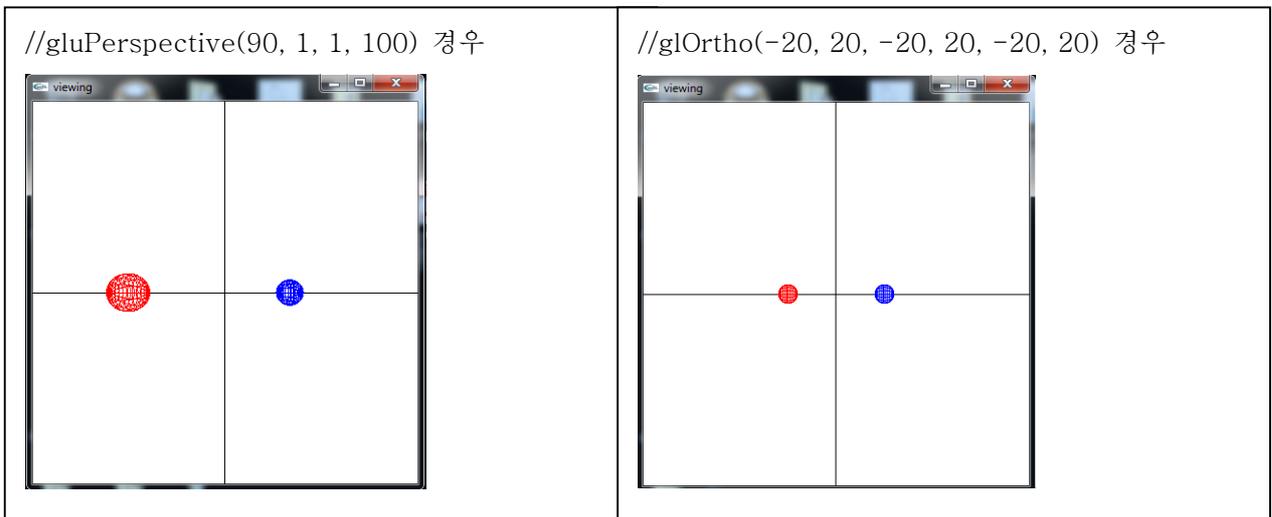
    glPushMatrix();
    glTranslatef(0, 0, 5);
    glColor3f(1, 0, 0); // 빨간공
    glutWireSphere(1, 16, 16);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(-5, 0, -5);
    glColor3f(0, 0, 1); // 파란공
    glutWireSphere(1, 16, 16);
    glPopMatrix();
}
```

- 1) 위 코드에서 밑줄 친 이동 변환과 회전 변환 함수를 gluLookAt() 함수로 대체하려는데, 같은 그림이 나타나도록 하기 위해서 이 함수의 9개 인자를 정확히 기술하라. 어떤 방식으로 문제를 풀었는지 상세히 기술할 것 (10점).

물체전체를 y축으로 -90도 회전하여 z축으로 -10 이동한 결과는, gluLookAt를 사용하여 카메라 위치 (10, 0, 0)에서 x축 방향으로 원점(0, 0, 0)을 바라보는 관측과 같다.

```
gluLookAt(10, 0, 0, 0, 0, 0, 0, 1, 0);
eye position (10, 0, 0)
at position (0, 0, 0)
up (0, 1, 0)
```

- 2) 위 코드에서 이탤릭체 gluPerspective()를 glOrtho(-20, 20, -20, 20, -20, 20) 함수로 대체하면, 최종 실행 결과 화면이 어떻게 바뀌는지 그림으로 (빨간 공과 파란 공의 위치와 크기) 자세히 설명하라 (5점).



- 3) 그리고 직교 투영 (Orthographic Projection), 축측 투영 (Axonometric Projection), 등축투영 (Isometric Projection), 경사 투영 (Oblique Projection), 원근 투영 (Perspective Projection)의 특징을 자세히 서술하라 (5점).

직교 투영은 투영선(projector)이 투영면(projection plane)에 수직이다.

축측 투영은 투영선(projector)은 투영면(projection plane)에 수직이지만, 투영면은 객체에 대해 어떠한 방향에서도 존재할 수 있다.

등축 투영은 축측 투영의 하나인데 그 중에서 투영면이 사각형 객체의 모서리에서 만나는 세 개의 주면에 대해서 대칭으로 놓여지는 것이다.

경사 투영은 투영선(projector)은 투영면(projection plane)과 임의의 각을 가질 수 있으나 투영면에 평행한 면내의 각은 보존된다.

원근(투시) 투영은 객체가 관측자로부터 멀리 떨어질수록 크기가 축소된다. 이러한 크기변화는 자연스러운 모습의 관측을 제공한다.

3. 다음은 조명 (lighting)에 관한 문제이다. 아래의 질문에 답하시오. (20점)

- 1) OpenGL에서 기본으로 사용하는 블린 조명 모델 (Blinn Illumination Model) 공식을 보여주고 있다. 이 공식에서 각 부분이 무엇을 의미하는지 자세히 설명하라 (5점).

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

$K_a I_a$: 환경반사 (ambient reflection) 부분

Σ : 여러 개의 광원이 있는 경우 (최대 8개까지)

$f_{att}(d)$: 빛의 감쇄 공식

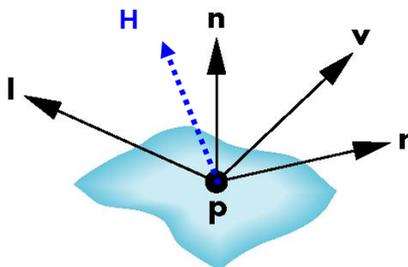
$K_d I_d (N \cdot L)$: 확산 반사 (diffuse reflection) 부분으로, 면의 밝기가 입사각 (광원벡터와 법선 벡터의 사이각)의 코사인에 정비례한다는 램버트 코사인 법칙에 따르는 빛이다.

$K_s I_s (N \cdot H)^{n'}$: 블린모델은 중간각 H를 사용하여 정반사 (specular reflection) 경면광을 나타내준다.

E: 자체 발광 방출 조명 부분

- 2) 블린 모델에서 사용하는 H란 무엇인지 아래 그림에 표시하고 자세히 설명하라 (5점).

정반사 (specular reflection)에서 풍 모델의 경우 $K_s I_s (R \cdot V)^n$ 을 사용하는데 반사벡터 R을 모든 정점에 대해 계산해야 한다. 반면, 블린 모델의 경우 정점에서 광원으로 향하는 광원벡터 L과 시점벡터 V의 사이인 중간각 $H = L + V / |L + V|$ 를 사용한다.



3) OpenGL에서 기본으로 사용하는 구로우 셰이딩(Gouraud Shading)을 자세히 설명하라. (5점)

구로우 셰이딩 (Gouraud shading)은 정점의 색으로부터 내부면의 색을 선형보간함. 정점의 법선 벡터를 요함. OpenGL에서는 smooth shading이라고 함. 면에 전체적으로 부드러운 음영을 제공 함. 그러나 경면광을 감안하지 않음 (실제적인 정점의 법선벡터와 근사적으로 계산된 법선벡터 가 완전히 일치하지 않기 때문).

4) 만약 OpenGL 프로그램에서 조명 (lighting)을 활성화하고, 광원(light source)와 물체의 재질(material)을 지정했는데 물체에 음영이 제대로 동작하지 않는다면 무엇이 문제인가? 조명에 반드시 필요한 3번째 요소가 무엇인지 자세히 설명하라 (5점).

OpenGL에서 lighting은 빛이 표면에서 얼마나 반사되는지 계산하기 위해서 해당 정점에서 표면이 어느 방향으로 향하고 있는지 알아야 한다. 따라서 면/정점의 법선 벡터 (normal vector)가 필요하다. OpenGL에서는 glNormal3f(...) 함수로 지정한다.

4. 다음은 OpenGL 텍스처 매핑(texture mapping)과 블렌딩 (blending) 프로그램의 일부이다. 아래의 질문에 답하시오. (40점)

```
void initTexture(char * filename, GLuint& textureID, GLuint param)
{
    unsigned char *imageBuffer;
    int imageWidth = 0, imageHeight = 0, numComponents = 0;
    imageBuffer = simage_read_image(filename, &imageWidth, &imageHeight, &numComponents);
    if (imageBuffer != NULL) {
        glGenTextures(1, &textureID);
        glBindTexture(GL_TEXTURE_2D, textureID);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, param);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, param);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
        glTexImage2D(GL_TEXTURE_2D, 0, numComponents == 3 ? GL_RGB : GL_RGBA,
                    imageWidth, imageHeight, 0, numComponents == 3 ? GL_RGB : GL_RGBA,
                    GL_UNSIGNED_BYTE, imageBuffer);
        glBindTexture(GL_TEXTURE_2D, 0);
    }
}
```

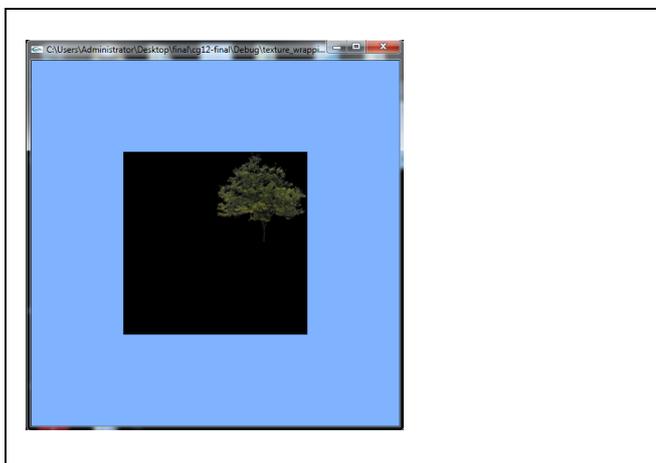
```
void drawTextureQuad(int textureID)
{
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, textureID);
    glBegin(GL_QUADS);
        glNormal3f(0, 0, 1);
        glTexCoord2f(-1, -1);
        glVertex3f(-1, -1, 0);
        glTexCoord2f(1, -1);
        glVertex3f( 1, -1, 0);
        glTexCoord2f(1, 1);
        glVertex3f( 1, 1, 0);
        glTexCoord2f(-1, 1);
        glVertex3f(-1, 1, 0);
    glEnd();
}
```

```

glEnd();
glDisable(GL_TEXTURE_2D);
}
void display()
{
    // 중간생략...
    drawTextureQuad(texID);
}
int main(int argc, char *argv[])
{
    // 중간생략...
    initTexture("tree02S.png", texID, GL_CLAMP); // 주변이 투명한 나무 그림
}
    
```



1) 위 코드의 실행 결과를 그림으로 나타내라 (10점).



2) 만약 flipbook animation 효과를 제공하고 싶다면, 위 코드 중에서 어느 부분 값이 바뀌어야 하는 지 설명하라 (5점).

여러 개의 텍스처에 대해서 initTexture를 여러 번 호출하여 각각 다른 textureID로 바인딩하고, 시간에 따라서 textureID를 바꿔주면 플립북 애니메이션이 동작된다.

3) 만약 텍스처 매핑이 거칠게 나타나서 좀 더 부드럽게 나타나게 하고 싶다면, 위 코드 중에서 glTexParameteri() 변수를 어떻게 설정해야 할지 자세히 설명하라 (5점).

확대필터(MAG_FILTER)와 축소필터(MIN_FILTER)에 최근점 필터링 (GL_NEAREST) 대신 포인트 샘플링에 의해 결정된 텍셀의 이웃을 포함한 텍셀 그룹의 가중 평균을 사용한 선형 필터링 (GL_LINEAR)를 사용해준다.

```

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    
```

4) 만약 mip맵 (Mipmap) 을 활성화하고 싶다면, 위 코드 중에서 어느 부분이 어떻게 바뀌어야 하는 지 자세히 설명하라 (5점).

일련의 축소된 크기의 mip맵 텍스처를 생성하여, 작은 객체에 텍스처 맵핑을 할 시 보간 문제를 줄여주는 필터로, `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);`

```
glTexImage2D(GL_TEXTURE_2D, 0, numComponents == 3 ? GL_RGB : GL_RGBA,
            imageWidth, imageHeight, 0, numComponents == 3 ? GL_RGB : GL_RGBA,
            GL_UNSIGNED_BYTE, imageBuffer);
```

아래와 같이 바꿔서 사용하면 된다.

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
gluBuild2DMipmaps(GL_TEXTURE_2D, numComponents == 3 ? GL_RGB : GL_RGBA,
                imageWidth, imageHeight, numComponents == 3 ? GL_RGB : GL_RGBA,
                GL_UNSIGNED_BYTE, imagePtr);
```

5) 만약 텍스처 매핑에 조명 (Lighting) 효과를 제공하고 싶지 않다면, 위 코드 중에서 `glTexEnvf()` 환경변수를 어떻게 설정해야 할지 자세히 설명하라 (5점).

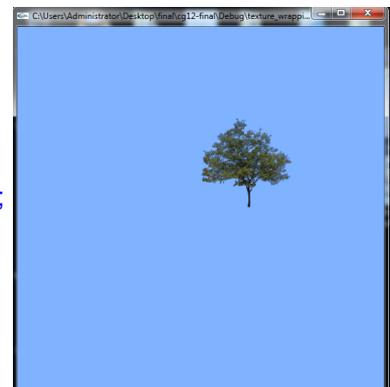
OpenGL에서 default texture environment는 `GL_MODULATE`, 즉 조명 (lighting)과 텍스처 매핑 (texture mapping) 색이 혼합이 되는 것이 기본값이다.

그런데, 만약 코드에서 `glTexEnv (GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL); /* or GL_REPLACE */`로 되어 있다면 텍스처 매핑의 색만을 이용하게 되므로 조명(Lighting)이 적용되지 않는다.

6) 위 코드는 투명한 나무 그림으로 나타나지 않는다. 문제가 무엇인지 지적하고 `display()` 함수의 내부 코드를 수정하라 (10점).

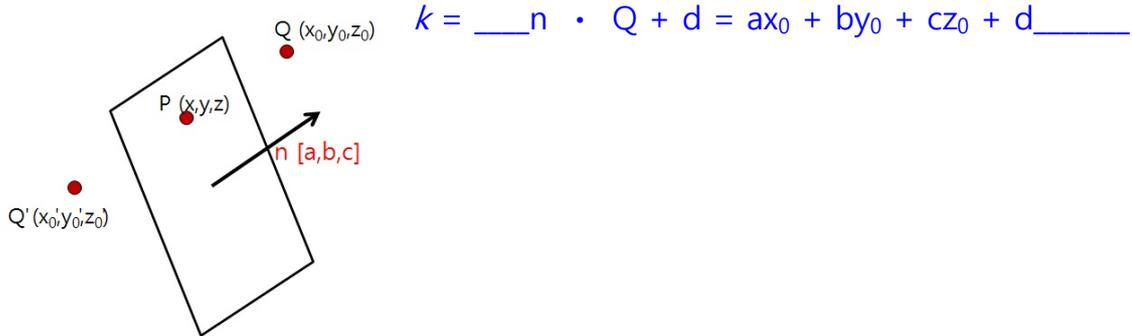
OpenGL은 기본적으로 블렌딩이 활성화 되어 있지 않으므로 텍스처는 알파블렌딩이 나타나지 않는다. 투명한 나무그림을 보여주려면 블렌딩활성화와 알파블렌딩 함수를 설정해야한다.

```
void display()
{
    // 중간생략...
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
    drawTextureQuad(texID);
    glDisable(GL_BLEND);
}
```



5. 다음 문제에 답하시오. (extra 15점)

평면 ($ax + by + cz + d = 0$)의 법선 벡터 $n(a, b, c)$ 가 단위벡터일 때, 평면 밖의 한 점 $Q(x_0, y_0, z_0)$ 에서 평면에 가장 가까운 거리 k 를 구하라.



그리고 점 $Q(x_0, y_0, z_0)$ 가 평면에 반사되는 점 $Q'(x'_0, y'_0, z'_0)$ 공식을 구하라. (힌트 $q' = q - 2kn$)

$$\begin{aligned} x'_0 &= x_0 - 2(ax_0 + by_0 + cz_0 + d)a \\ &= (1 - 2a^2)x_0 - 2aby_0 - 2acz_0 - 2ad \end{aligned}$$

$$\begin{aligned} y'_0 &= y_0 - 2(ax_0 + by_0 + cz_0 + d)b \\ &= 2abx_0 + (1 - 2b^2)y_0 - 2bcz_0 - 2bd \end{aligned}$$

$$\begin{aligned} z'_0 &= z_0 - 2(ax_0 + by_0 + cz_0 + d)c \\ &= -2acx_0 - 2bcy_0 + (1 - 2c^2)z_0 - 2cd \end{aligned}$$

점 $Q(x_0, y_0, z_0)$ 가 xz 평면에 반사되는 점 $Q'(x'_0, y'_0, z'_0)$ 값을 계산하라.

$$x'_0 = \underline{\quad} x_0 \underline{\hspace{2cm}}$$

$$y'_0 = \underline{\quad} - y_0 \underline{\hspace{2cm}}$$

$$z'_0 = \underline{\quad} z_0 \underline{\hspace{2cm}}$$