단국대학교 멀티미디어공학 그래픽스 프로그래밍 중간고사 (2011년 봄학기) 2011년 4월 26일

학과 _____ 학번 _____ 이름 _____

중간고사

담당교수: 단국대학교 멀티미디어공학전공 박경신

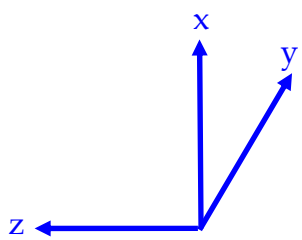
- **답은 반드시 답안지에 기술할 것. 공간이 부족할 경우 반드시 답안지 몇 쪽의 뒤에 있다고 명기한 후 기술할 것. 그 외의 경우의 답안지 뒤쪽이나 연습지에 기술한 내용은 답안으로 인정 안 함. 답에는 반드시 네모를 쳐서 확실히 표시할 것.**
- **답안지에 학과, 학번, 이름 외에 본인의 암호를 기입하면 성적공고시 학번 대신 암호를 사용할 것임.**

1. 맞으면 true, 틀리면 false를 적으시오. (20점)

1) 벡터는 공간 내에 고정된 위치를 갖지 않는다. ___T___
2) 아핀 공간 (Affine Space)에는 거리는 없으나 한 직선 위에 있다는 개념은 있으며 거리의 비가 보존된다. ___T___
3) 기하 파이프라인에서 클리핑(Clipping)은 정점 단위로 이루어 진다. ___F___
4) 안티에일러싱(Anti-aliasing)이 필요한 장치는 벡터 그래픽 시스템이다. ___F___
5) OpenGL 디스플레이 리스트 (Display List)는 대용량 그래픽의 렌더링 속도 향상을 위해 사용된다. ___T___
6) glEnable(GL_DEPTH_TEST)는 렌더링 속도의 향상을 위해 눈에 보이지 않는 면을 제거하는 후면 추리기(Backface culling) 기법이다. ___F___
7) OpenGL 기본 카메라는 물체 공간의 원점 (Origin)에 위치하며 +z 방향을 향하고 있다. ___F___
8) glutPostRedisplay() 함수는 윈도우 크기가 변화된 이벤트 발생시 호출된다. ___F___
9) glutStrokeCharacter() 함수는 문자를 3차원 선으로 그리는 방식이므로, OpenGL의 변환에 의해 영향을 받는다. ___T___
10) 컴퓨터가 영상을 생성하는 방법은, 합성 카메라 모델을 사용하여 카메라 영상면에 생성된 영상이 전후좌우가 뒤집혀 나타난다. ___F___

2. 다음 문제에 답하시오. (45점)

1) +x축이 화면 위 방향, +y축이 화면의 안쪽으로 들어가는 방향일 때, 오른손 좌표계 (right-handed coordinate system)에서 +z축이 가리키는 방향이 어느 곳인지 x, y, z축을 그리시오.



학과 _____ 학번 _____ 이름 _____

2) 벡터 그래픽 시스템과 래스터 그래픽 시스템을 간단히 설명하라. (5점)

벡터 그래픽 시스템은 레이더나 오실로스코프, 플로터 등에서 사용한 그래픽 시스템으로써, 화소(pixel)의 개념이 없다. 즉 프레임 버퍼를 사용하지 않는다. 무한해상도를 갖으며, aliasing이 없다. 전자 편향판을 조정해서 화면에 직접 선을 그리는 방식이다.

래스터 그래픽 시스템은 현재 대부분의 컴퓨터 시스템에서 사용하는 프레임 버퍼를 사용하여, 프레임 버퍼에서 화소 (pixel)의 배열 (array)인 래스터 (raster) 이미지로 생성하여 모니터에 출력하는 방식의 시스템이다. 화소를 사용하는 래스터 이미지 방식이라서 연속적인 영상을 유한 개의 화소를 사용하는 영역으로 표현하므로 오차가 발생하는데 이를 aliasing이라고 부른다.

3) RGB 색 모델 (color model) 과 CMY 색 모델에 대해 간단히 설명하시오. 그리고 각각의 색 모델에서 (1, 1, 1), (1, 1, 0), (1, 0, 0) 이 의미하는 색이 무엇인지 적으시오.

RGB 색 모델은 빛의 삼원색으로 컬러 디스플레이 시스템에 적합한 모델이다.
RGB 색 모델은 Red, Green, Blue 값을 더해서 표현하는 방식이다.
RGB (1, 1, 1) 흰색, (1, 1, 0) 노란색, (1, 0, 0) 빨간색

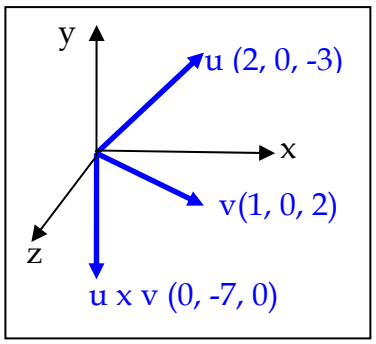
CMY 색 모델은 색의 삼원색으로 인쇄 시스템에 적합한 모델이다.
CMY 색 모델은 RGB의 보색인 Cyan, Magenta, Yellow을 사용하며 CMY를 빼서 표현하는 방식이다.
CMY (1, 1, 1) 검정색, (1, 1, 0) 파란색, (1, 0, 0) 청록색

4) 다음은 벡터의 내적(dot product) 특성을 정리한 것이다. 아래의 빈 칸을 채우시오.

만약 벡터 u • v < 0 이라면 두 벡터 간의 사이각 θ는 ___90 < θ <= 180____에 있다.
만약 벡터 u • v = 0 이라면 두 벡터 간의 사이각 θ는 ___0 = 90_____ 에 있다.
만약 벡터 u • v > 0 이라면 두 벡터 간의 사이각 θ는 ___0 <= θ < 90_____에 있다.

5) OpenGL 환경에서 두 벡터 u (2, 0, -3), v(1, 0, 2)를 그림으로 표시하라. 그리고 두 벡터 간의 외적(cross product) u x v 을 계산하고 그림으로 표시하라.

u x v = (0*2 + 0*3, -2*2 - 3 *1, 2*0 – 1*0) = (0, -7, 0)



6) 컨벡스 헐(Convex Hull)이란 점들의 집합 {P_1, P_2, ..., P_n}을 포함하는 가장 작은 볼록 객체이다. 컨벡스 헐을 만족하는 아핀합(Affine Sums)을 설명하시오.

N points: P_1, P_2, ..., P_n
$P = \alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_n P_n$
$\alpha_1 + \alpha_2 + \ldots + \alpha_n = 1$
$\alpha_1, \alpha_2, \ldots, \alpha_n >= 0$

7) glutIdleFunc(void (func*)(void))와 glutTimerFunc(unsigned int millis, void (func*)(int value), int value) 함수는 무엇인가, 그리고 각각의 함수에 답신함수(callback)의 예를 적어라.

glutIdleFunc 함수란 idle 이벤트 (즉, 다른 이벤트가 없을 경우)에 이 함수에 등록된 답신함수가 불려진다. 주로, 애니메이션을 위한 업데이트 함수가 이곳에서 불려진다.

```
glutIdelFunc(update);
void update()
{
        g_List.Update();  // 각 도형의 위치, 크기 등 값이 변경됨
        glutPostRedisplay();
}
```

glutTimerFunc 함수란 지정한 시간이 지나면 이벤트가 발생하여 이 함수에 등록된 답신함수가 불려지고 타이머는 종료된다. 업데이트 함수에 사용하려면 답신함수 내부에서 다시 glutTimerFunc 함수를 호출하여 일정 간격으로 불려지게 만들어야한다.

```
glutTimerFunc(100, timer, 1);
void timer(int value)
{
        g_List.Update();  // 각 도형의 위치, 크기 등 값이 변경됨
        glutPostRedisplay();
        glutTimerFunc(100, timer, 1); // 자신을 호출하여 100millisec단위로 불리게 함
}
```

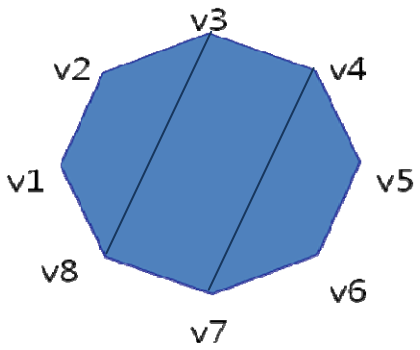
8) 더블 버퍼링(double buffering)에서 스왑 버퍼 (swap buffer)가 무엇인지 간단히 서술하시오. 그리고 더블 버퍼링과 싱글 버퍼링(single buffering)의 차이점을 예를 들어 설명하라.

더블 버퍼링이란 색 버퍼를 전면과 후면버퍼로 나누어서 사용하는 것으로,
비디오 제어기가 항상 완성된 이미지를 그리도록 하기 위하여, 프로세서는 이미지의 내용을 계산하여 후면 버퍼에 축적하고, 그동안 비디오 제어기는 전면 버퍼의 내용을 읽어 화면에 이미지 도시하는데, 양쪽에서 모두 다 그리고 나면 전면버퍼와 후면버퍼를 서로 바꾸는 (swap buffer)를 수행한다.

회전하는 사각형 같은 애니메이션을 그려야 할 경우, 더블 버퍼링을 쓰면 싱글 버퍼링보다 부드러운 애니메이션을 생성한다. 만약, 애니메이션에 싱글 버퍼링을 썼을 경우 회전하는 사각형의 이미지가 번쩍번쩍거리는 현상이 발생한다.

9) OpenGL에서 GL_QUAD_STRIP를 사용해서 도형을 그릴 때, 다음 그림과 같은 결과가 되도록 코드를 작성하시오.

```
glBegin(GL_QUAD_STRIP);
    glVertex3fv(   v1   );
    glVertex3fv(   v2   );
    glVertex3fv(   v8   );
    glVertex3fv(   v3   );
    glVertex3fv(   v7   );
    glVertex3fv(   v4   );
    glVertex3fv(   v6   );
    glVertex3fv(   v5   );
glEnd();
```



학과 _____ 학번 _____ 이름 _____

3. 다음 행렬 문제에 답하시오. (35점)

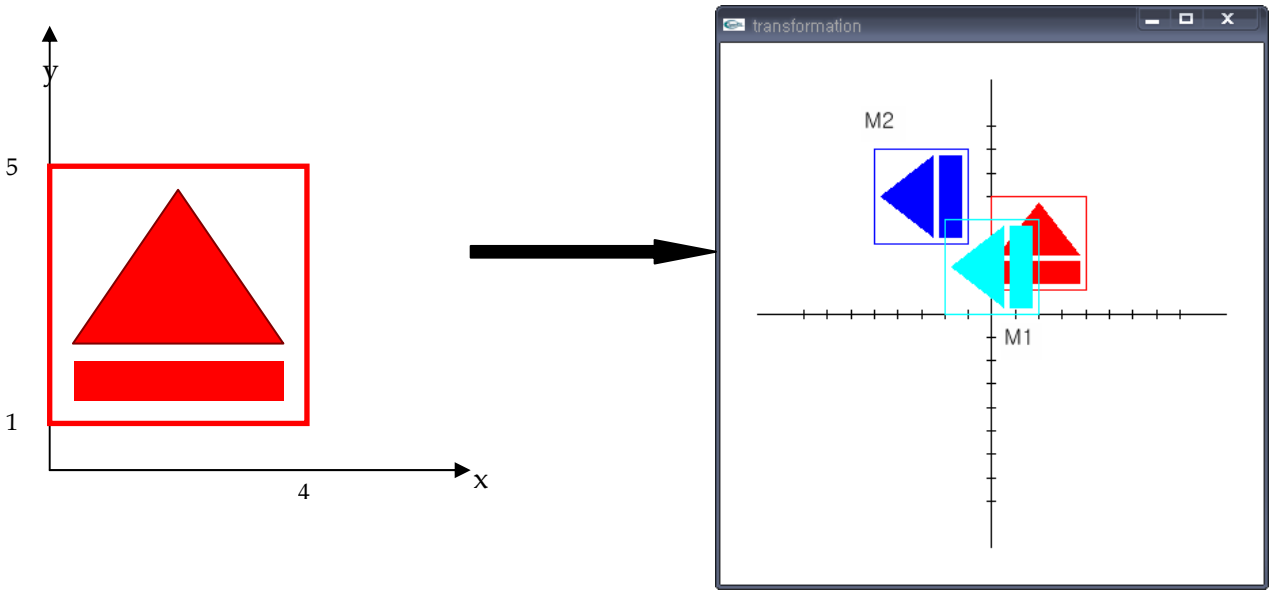
1) 다음 OpenGL 변환 함수가 만들어 내는 3차원 아핀 변환 행렬 M_1과 M_2을 간단히 설명하라. (5점)

```
glPushMatrix();
glTranslatef(3, 0, 0);
glRotatef(90, 0, 0, 1);
drawObject();
glGetFloatv(GL_MODELVIEW_MATRIX, M1);
glPopMatrix();

glPushMatrix();
glRotatef(90, 0, 0, 1);
glTranslatef(3, 0, 0);
drawObject();
glGetFloatv(GL_MODELVIEW_MATRIX, M2);
glPopMatrix();
```

M1은 z-축으로 90도 회전을 먼저 한 후, x-축으로 3만큼 이동한 결과

M2는 x-축으로 3만큼 이동한 후, z-축으로 90도 회전한 결과

2) 다음 왼쪽 기본 도형에, 위의 3차원 아핀 변환 행렬 M_1과 M_2를 적용하여 나타난 도형의 모습을 오른쪽에 그려서 나타내라 (정확한 척도를 그려서 답하라). (10점)

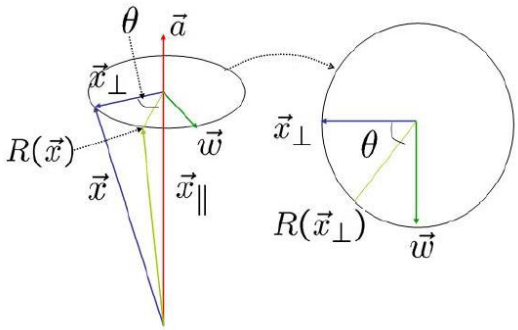


3) 다음 위의 기본 도형을 중심점 (2, 3)으로 x-축과 y-축으로 0.5배로 크기 변환하는 행렬 M을 OpenGL 코드로 작성하라. (5점)

```
glPushMatrix();
glTranslatef(2, 3, 0);
glScalef(0.5, 0.5, 1);
glTranslatef(-2, -3, 0);
drawObject();
glGetFloatv(GL_MODELVIEW_MATRIX, M);
glPopMatrix();
```

학과 _____ 학번 _____ 이름 _____

4) 다음 임의의 축 a에 회전하는 한 점 x가 a에 투영하여 평행한 성분인 $x_{\parallel}$을 표현하는 행렬을 구하라. (5점)



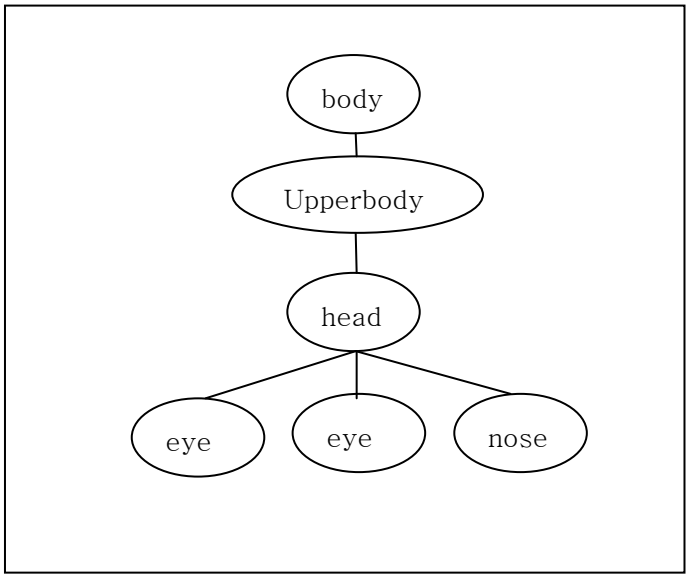
$$x_{\parallel} = (a \bullet x)a$$

$$SymmetricMatrix = \begin{pmatrix} a_x^2 & a_x a_y & a_x a_z \\ a_x a_y & a_y^2 & a_y a_z \\ a_x a_z & a_y a_z & a_z^2 \end{pmatrix}$$

5) 다음 간단한 OpenGL 코드를 보고, 3차원 객체를 계층적 변환 (Hierarchical Transformation) 트리 구조로 그림으로 표현하라. (10점)

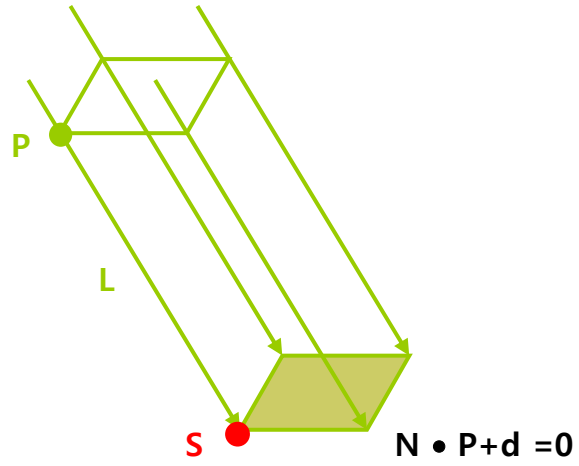
```
void initDL()
{
        GLfloat white[] = {1, 1, 1, 1};
        GLfloat black[] = {0, 0, 0, 1};
        GLfloat orange[] = {1.0f, 0.5f , 0.5f};

        glNewList(1, GL_COMPILE);

        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, white);

        glPushMatrix();
        glTranslatef(0.0f, 1.0f, 0.0f);
        gluSphere(quadric, 1.5f, 20, 20);      // body

        glPushMatrix();
        glTranslatef(0.0f, 2.0f, 0.0f);
        gluSphere(quadric, 1.0f, 20, 20);      // upper body

        glPushMatrix();
        glTranslatef(0.0f, 1.3f, 0.0f);
        gluSphere(quadric, 0.6f, 20, 20);      // head

        glPushMatrix();
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, black);
        glTranslatef(0.2f, 0.1f, 0.6f);
        gluSphere(quadric, 0.1f, 10, 10);      // eye
        glPopMatrix();
```

```
        glPushMatrix();
        glTranslatef(-0.2f, 0.1f, 0.6f);
        gluSphere(quadric, 0.1f, 10, 10);      // eye
        glPopMatrix();

        glPushMatrix();
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, orange);
        glTranslatef(0.0f, -0.1f, 0.6f);
        glRotatef(0.0f,1.0f, 0.0f, 0.0f);
        gluCylinder(quadric, 0.1f, 0.0, 0.35, 10, 2); // nose
        glPopMatrix();

        glPopMatrix();
        glPopMatrix();
        glPopMatrix();

        glEndList();
}
```



학과 _____ 학번 _____ 이름 _____

4. 다음 그림에서와 같이 3차원 공간의 점 P (x, y, z)가 벡터 L 방향으로 가는 광선 (P(t) = P + t L)은 평면에 한 점 S (Sx, Sy, Sz)를 만난다. 평면의 공식 (N • P + d = 0)을 이용하여 t 를 계산하라. 그리고 S 를 계산하는 공식을 유도하라. (extra 10점)



S = P + tL

N • S + d = 0

$$N \cdot (p + tL) + d = 0$$
$$tN \cdot L = -d - p \cdot N$$
$$t = \frac{-(d + p \cdot N)}{N \cdot L}$$
$$\therefore s = p + \left[\frac{-(d + p \cdot N)}{N \cdot L}\right] L$$