

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. (10점)

- 1) OpenGL 디스플레이 리스트 (Display List)는 glFlush()를 포함하여 모든 OpenGL 함수를 쓸 수 있다. ___F___
- 2) OpenGL/GLUT 프로그램은 반드시 glutDisplayFunc 답신함수(Callback Function)를 정의해야 한다. ___T___
- 3) glutStrokeCharacter 함수는 문자를 선으로 그리는 방식이므로, OpenGL의 변환에 의해 영향을 받지 않는다. ___F___
- 4) 프레임 버퍼의 깊이는 시스템이 얼마나 많은 색을 한 픽셀에서 표현할 수 있는 지를 결정한다. ___T___
- 5) 부드러운 애니메이션을 생성하기 위해 깊이 버퍼가 필요하다. ___F___

2. OpenGL/GLUT 프로그램의 이벤트 답신함수를 적어라. (10점)

예: ASCII 키가 눌렸을 때 호출하는 이벤트 답신함수는 glutKeyboardFunc 에 등록한다.

- 1) 특수키가 눌렸을 때 호출하는 이벤트 답신함수는 __glutSpecialFunc__
- 2) 마우스 버튼이 눌렸을 때 호출하는 이벤트 답신함수는 __glutMouseFunc__
- 3) 마우스가 움직일 때 호출하는 이벤트 답신함수는 __glutMotionFunc__
- 4) 이벤트가 없을 때 호출하는 이벤트 답신함수는 __glutIdleFunc__
- 5) 윈도우 크기가 변했을 때 호출하는 이벤트 답신함수는 __glutReshapeFunc__

3. 다음 문제에 답하시오. (30점)

- 1) 컴퓨터 그래픽에서 사용하는 3차원 객체(3D object)는 무엇인지 설명하라. (5점)

3차원 공간 내의 정점들의 집합이다.

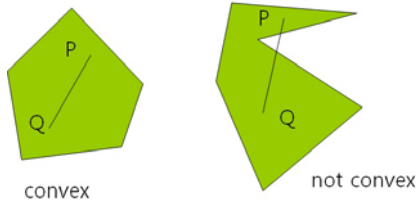
표면 (surface)에 의해 기술되고 속이 비어있다 (hollow)

표면은 볼록 다각형 (convex polygon)으로 구성되어 있다.

임의의 다각형 (arbitrary polygons)은 삼각 다각형(triangular polygons)으로 분할 (tessellate)하여 그려야 한다.

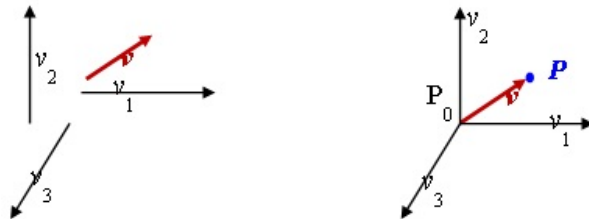
- 2) 컴퓨터 그래픽에서 화면에 도시되는 객체는 볼록성(Convexity)을 가져야 한다. 볼록성이 무엇인지 간단히 설명하라. (5점)

컴퓨터 그래픽에서 볼록성(Convexity)란 객체 내의 임의의 두 점을 연결하는 선분 위에 놓인 임의의 점이 모두 객체 내에 있어야 한다는 것으로, 컴퓨터 그래픽에서 볼록성을 갖지 않는 객체는 삼각 다각형으로 분할(Tessellation)해서 사용한다.



- 3) 벡터 공간 (Vector Space)와 아핀 공간 (Affine Space)를 간단히 설명하고, 특징을 서술하라. 벡터 공간과 아핀 공간을 그림으로 그려서 차이를 보여라. (5점)

벡터 공간은 실수 (즉 scalar)와 벡터(vector)를 가지고 있다. 벡터는 공간 내에 고정된 위치를 가지고 있지 않으며 크기와 방향성을 갖는다. 아핀 공간은 벡터 공간에 점(point)이 추가된다. 아핀 공간에는 거리는 없으나 한 직선 위에 있다는 개념은 있으며 거리의 비가 보존된다.



- 4) 동차 좌표 (Homogeneous Coordinates)는 무엇인가, 그리고 컴퓨터 그래픽에서 동차 좌표를 사용하는 이유는? (5점)

동차 좌표란 2차원의 점 (x,y) 의 경우의 동차좌표로는 (x, y, w) 로 표현한다. 따라서 동차 좌표에서 $(1,2,3)$ 과 $(2,4,6)$ 은 같은 동차좌표 표현이다. 3차원 점 (x, y, z) 은 동차좌표로 $(x, y, z, 1)$ 로 표현하고, 벡터 (x, y, z) 는 동차좌표로는 $(x, y, z, 0)$ 으로 표현한다.

컴퓨터 그래픽에서 동차 좌표를 사용하는 이유는, 그래픽 렌더링 파이프라인에서 각종 좌표계(예: 모델 좌표계, 세계 좌표계, 카메라 좌표계, 투영변환, 등) 변환을 수행해야 하는데, 동차좌표를 사용한다면 3차원 점과 벡터를 4차원 열행렬로 나타낼 수 있고 주어진 프레임에서 점과 벡터의 표현을 변환하는 단일함수로 변환을 정의할 수 있다. 이동(translation), 회전(rotation), 크기변환 (scaling)등 그래픽스에서 필요로 하는 대부분의 변환은 아핀 변환이다.

- 5) 컴퓨터 그래픽 영상의 생성에서 필요한 3가지 요소인 객체, 관측자, 광원을 설명하라. 그리고 합성 카메라 모델 (Synthetic-Camera Model)을 설명하라. (5점)

Object (객체)는 영상생성 과정이나 관측자와 관계없이 공간에 존재
Viewer (관측자)는 물체의 영상을 형성하는 존재. 즉, 컴퓨터 그래픽에서 카메라.
Light (광원)은 객체와 관측자가 있다 하더라도 광원이 없다면 객체는 어두워서 영상에 나타나지 않는다.

Synthetic-Camera Model (합성 카메라 모델)은 Image Plane(영상면)이 Center of Projection보다 앞에 있다. 인간시각시스템이나 핀홀카메라모델이 영상면이 카메라 후면에 있어서 영상면에 전후 좌우가 뒤바뀌어 이미지가 생기는 것과는 달리 영상이 그대로 투영되는 모델이다.

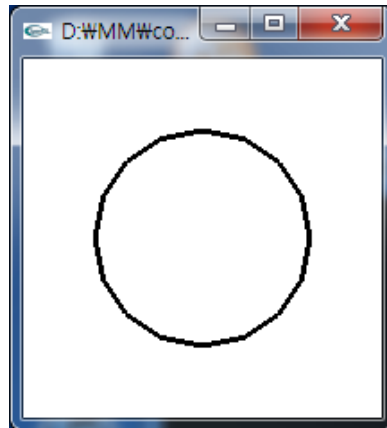
- 6) 컴퓨터 그래픽 기하 파이프라인 (geometric pipeline)에서 응용프로그램의 정점들 (vertices)을 정점처리(Vertex Processor) 한 후 클리핑(Clipping)과 기본 요소로 조립 (Primitive Assembly)하여 래스터화(Rasterization)하고 픽셀처리(Pixel Processor) 한 후 프레임버퍼로 나오는 과정을 수행한다. 이 때 기본요소 (Primitive)란 무엇을 의미하는가? 기본요소 조립은 왜 필요한지 설명하라. (5점)

기본요소(Primitive)란 그래픽을 그리는 최소 단위인 선분 (Line segments), 다각형 (Polygons), 곡선 곡면 (Curves and surfaces) 등을 말한다.

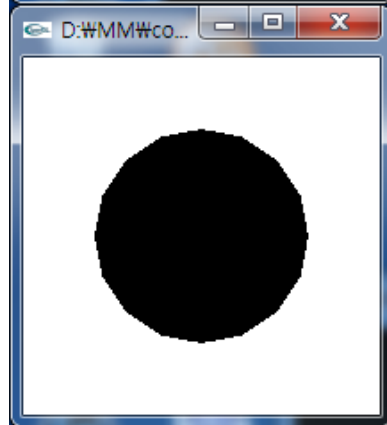
기본요소조립 과정이 필요한 이유는 카메라가 찍을 수 있는 시야각 영역 내부만 보이게 처리하는 과정, 즉 클리핑(Clipping),이 정점(Vertex)의 단위로 이루어지기보다는 기본요소(Primitive) 단위로 이루어지고 있기 때문에 기본 요소 단위로 조립을 한 후에 클리핑을 수행하여야 한다.

4. 다음 OpenGL 코드는 원의 공식을 이용한 도형 그리기에서 GL_LINE_LOOP, GL_TRIANGLE_FAN, GL_TRIANGLES 차이를 보여준다. 결과를 오른쪽 네모 칸 안에 그리시오. (20점)

```
float x, y, theta;
theta = 2 * M_PI / 16;
glBegin(GL_LINE_LOOP);
for (int i = 0; i < 16; i++)
{
    x = cosf(theta * i);
    y = sinf(theta * i);
    glVertex3fv(x, y, 0);
};
glEnd();
```



```
float x, y, theta;
theta = 2 * M_PI / 16;
glBegin(GL_TRIANGLE_FAN);
for (int i = 0; i < 16; i++)
{
    x = cosf(theta * i);
    y = sinf(theta * i);
    glVertex3fv(x, y, 0);
};
glEnd();
```



```
float x, y, theta;
theta = 2 * M_PI / 16;
glBegin(GL_TRIANGLES);
for (int i = 0; i < 16; i++)
{
    x = cosf(theta * i);
    y = sinf(theta * i);
    glVertex3fv(x, y, 0);
};
glEnd();
```



```
float x, y, theta;
theta = 2 * M_PI / 16;
glBegin(GL_QUADS);
    for (int i = 0; i < 16; i++)
    {
        x = cosf(theta * i);
        y = sinf(theta * i);
        glVertex3fv(x, y, 0);
    };
glEnd();
```



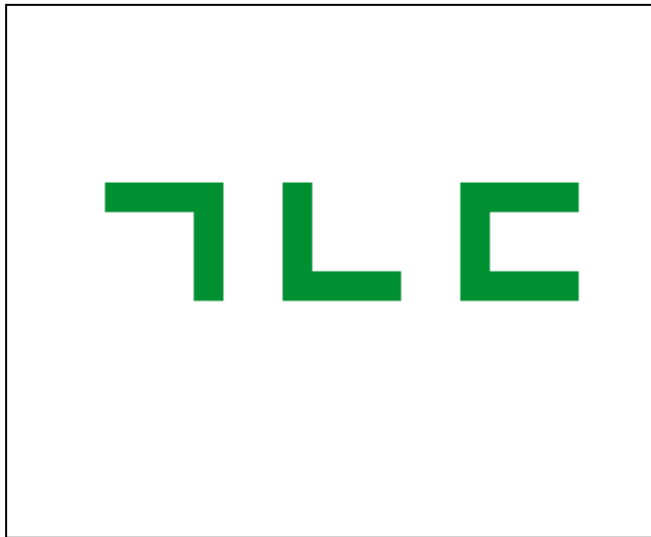
5. 다음 OpenGL 코드는 HW1에서 (pos[0], pos[1])을 중심으로 한 도형 그리기이다. 최종 결과를 아래 네모 칸 안에 그리시오. (10점)

```
glBegin(GL_QUADS);
glVertex2f(pos[0] - 2.f, pos[1] + 0.5f);
glVertex2f(pos[0] - 2.f, pos[1] + 1.f);
glVertex2f(pos[0] - 4.f, pos[1] + 1.f);
glVertex2f(pos[0] - 4.f, pos[1] + 0.5f);
glVertex2f(pos[0] - 2.f, pos[1] - 1.f);
glVertex2f(pos[0] - 2.f, pos[1] + 1.f);
glVertex2f(pos[0] - 2.5f, pos[1] + 1.f);
glVertex2f(pos[0] - 2.5f, pos[1] - 1.f);
glEnd();

glBegin(GL_QUADS);
glVertex2f(pos[0] - 1.f, pos[1] - 1.f);
glVertex2f(pos[0] - 0.5f, pos[1] - 1.f);
glVertex2f(pos[0] - 0.5f, pos[1] + 1.f);
glVertex2f(pos[0] - 1.f, pos[1] + 1.f);
glVertex2f(pos[0] + 1.f, pos[1] - 1.f);
glVertex2f(pos[0] + 1.f, pos[1] - 0.5f);
glVertex2f(pos[0] - 1.f, pos[1] - 0.5f);
glVertex2f(pos[0] - 1.f, pos[1] - 1.f);
glEnd();

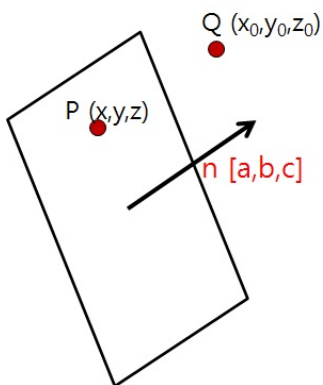
glBegin(GL_QUADS);
glVertex2f(pos[0] + 4.f, pos[1] + 1.f);
glVertex2f(pos[0] + 2.f, pos[1] + 1.f);
glVertex2f(pos[0] + 2.f, pos[1] + 0.5f);
glVertex2f(pos[0] + 4.f, pos[1] + 0.5f);
glVertex2f(pos[0] + 4.f, pos[1] - 0.5f);
```

```
glVertex2f(pos[0] + 2.f, pos[1] - 0.5f);
glVertex2f(pos[0] + 2.f, pos[1] - 1.f);
glVertex2f(pos[0] + 4.f, pos[1] - 1.f);
glVertex2f(pos[0] + 2.f, pos[1] + 1.f);
glVertex2f(pos[0] + 2.f, pos[1] - 1.f);
glVertex2f(pos[0] + 2.5f, pos[1] - 1.f);
glVertex2f(pos[0] + 2.5f, pos[1] + 1.f);
glEnd();
```



6. 다음 문제에 답하십시오. (20점)

- 1) 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. (10점)



$$k = \underline{\quad} n \cdot Q + d = ax_0 + by_0 + cz_0 + d \underline{\quad}$$

그리고 점 Q에서 가장 가까운 평면 상의 점 P(x, y, z)를 구하라.

$$P = Q - kn$$

$$x = \underline{\quad} x_0 - k a \underline{\quad}$$

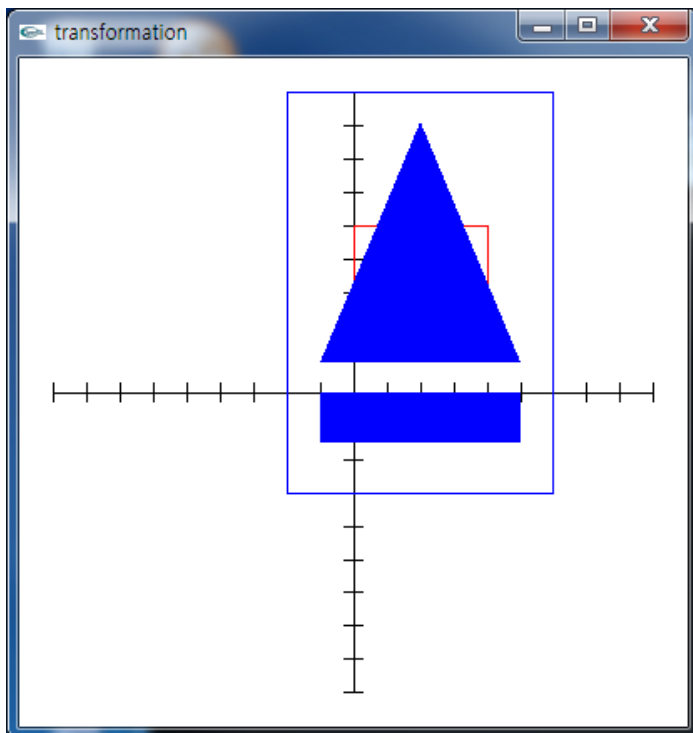
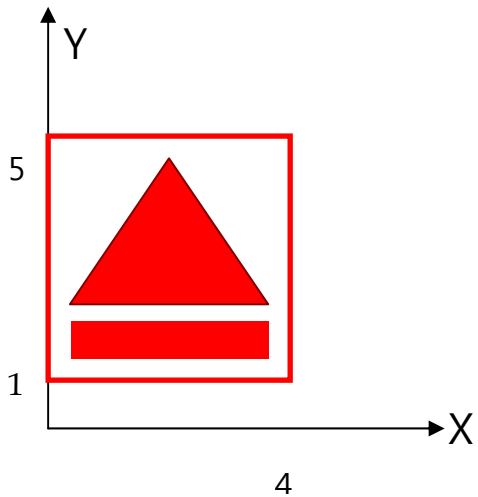
$$y = \underline{\quad} y_0 - k b \underline{\quad}$$

$$z = \underline{\quad} z_0 - k c \underline{\quad}$$

- 2) 중심점 (2, 3, 0)으로 X-축으로 2배, Y-축으로 3배 크기 변환하는 행렬 M(4x4아핀행렬)을 작성하라. (10점)

$$\begin{pmatrix} 2 & 0 & 0 & -2 \\ 0 & 3 & 0 & -6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

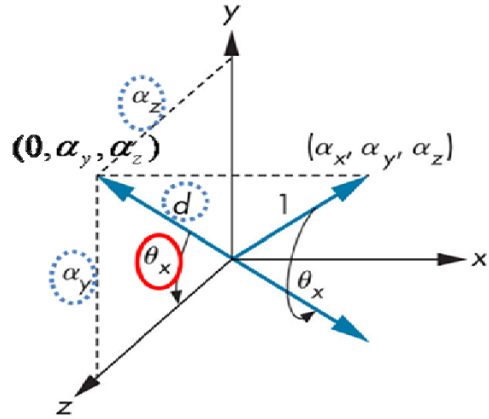
그리고 다음 왼쪽 기본 도형에, 위의 아핀 변환 행렬 M을 적용하여 나타난 도형의 모습을 아래 네모 칸 안에 정확한 척도를 그려서 나타내라.



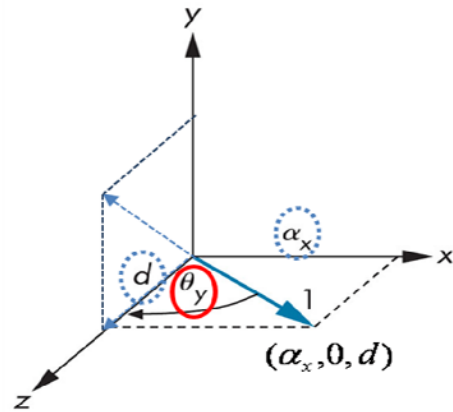
7. 다음은 3차원 그래픽에서 임의의 축 (arbitrary axis) $v = (2, 3, 4)$ 에 대한 45도 회전 행렬을 유도하는 문제이다. 아래의 행렬에서 $R_x(\theta_x), R_y(\theta_y), R_z(\theta_z)$ 행렬을 답하시오. $\cos(45) = \sin(45) = 0.707$ (extra 10점)

$$M = T(P_0)R_x(-\theta_x)R_y(-\theta_y)R_z(\theta)R_v(\theta_v)R_r(\theta_r)T(-P_0)$$

$$R_x(\theta_x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{4}{5} & -\frac{3}{5} & 0 \\ 0 & \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$R_y(\theta_y) = \begin{pmatrix} 5 & 0 & -2 & 0 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$R_z(\theta_z) = R_z(45^\circ) = \begin{pmatrix} 0.707 & -0.707 & 0 & 0 \\ 0.707 & 0.707 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$