

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) 벡터 자신의 내적은 벡터의 길이의 제곱과 같다. ___T___
- 2) 하나의 윈도우 안에 더블버퍼와 깊이버퍼를 동시에 사용할 수 없다. ___F___
- 3) 이동(translation)과 회전변환(rotation)은 아핀강체변환(affine rigid-body transformation)이다. ___T___
- 4) 인간 눈에 망막의 원추체 (Cones)는 명암 (Luminance)을 구분하고 간상체 (Rods)는 색깔 인지 (Color Perception)를 한다. ___F___
- 5) 정방행렬 (Square Matrix)는 반드시 역행렬 (Inverse Matrix)를 갖는다. ___F___

2. 빈칸을 채우시오. (10점)

- 1) OPENGL 3차원 좌표계는 x+는 오른쪽, y+는 위쪽, z+는 화면 밖으로 나오는 방향이다.
- 2) 프레임 버퍼에서 이미지를 생성하는데 필요한 색상 값을 갖는 버퍼를 색깔버퍼 (Color Buffer)라 한다.
- 3) 점과 벡터의 덧셈은 점(을)를 만든다.
- 4) 점 p가 평면 (n, d)에서 만약 $n \cdot p + d < 0$ 이라면 p는 평면 안쪽에 있다.
- 5) 점들의 집합 $\{P_1, P_2, \dots, P_n\}$ 을 포함하는 가장 작은 볼록한 객체를 컨벡스 헐이라 부른다.

3. 다음 문제에 답하시오. (50점)

- 1) HSV(Hue, Saturation, Value) 색깔 모델이 무엇인지 간단히 설명하라. 선명한 빨간색을 HSV 색깔모델로 표현하라.

HSV(Hue, Saturation, Value)는 색상, 채도, 명도를 의미하며, Hue는 색깔, Saturation(채도)는 높을 수록 원색에 가깝고 낮으면 파스텔톤에 가깝다. Value(명도)가 낮으면 검정색에 가까워지고 높으면 흰색에 가까워진다.

선명한 빨간색 Hue값은 0~360도에서 0, Saturation값은 0~1에서 1, Value값은 0~1에서 1

- 2) 깊이 버퍼 (Depth buffer)가 무엇인지 간단히 설명하시오. 그리고, 다음 간단한 OpenGL 프로그램에서 깊이 버퍼를 위해 사용하는 함수에 모두 동그라미 표시하시오.

```

void divide_tetra(GLfloat* a, GLfloat* b, GLfloat* c, GLfloat* d, int m)
{
    // 생략..
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
    divide_tetra(v[0], v[1], v[2], v[3], 5);
    glEnd();
    glFlush();
}

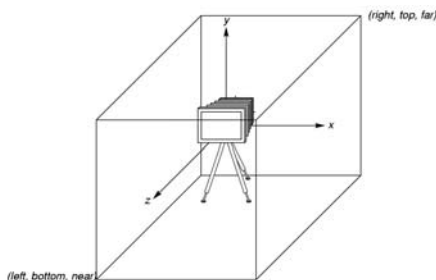
void init()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho3D(-2.0, 2.0, -2.0, 2.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glColor3f(0.0, 0.0, 0.0);
}

void main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB|GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutCreateWindow("3D Gasket");
    glutDisplayFunc(display);
    glEnable(GL_DEPTH_TEST);
    glutMainLoop();
}
    
```

깊이버퍼는 일명 Z-buffer라 불리며, 픽셀단위로 기하개체 (geometry)의 깊이값 (z)이 가장 작은 평면을 그리도록 깊이정보를 축적하는 버퍼이다.

OPENGL에서 깊이버퍼 알고리즘을 사용하기 위하여, 깊이버퍼를 사용한다는 초기화를 하고, 깊이정보 테스트를 활성화해야 하며, 디스플레이 함수 내에서 매 프레임마다 색버퍼를 지울 때 깊이버퍼도 같이 지워야 한다.

- 3) OpenGL에서 기본 카메라(Default Camera)의 위치와 관측공간을 간단히 설명하고 그림으로 표시하라.



OpenGL에서는 카메라가 물체의 공간(drawing coordinates)의 원점(origin)에 위치하며 z- 방향으로 향하고 있다. 관측공간을 지정하지 않는다면, 디폴트로 2x2x2 입방체의 viewing volume을 사용한다.

- 4) 입력장치가 응용프로그램에게 제공하는 3가지 입력모드 방식이 무엇인지 간단히 설명하고 비교하라.

요구모드 (Request Mode): 프로그램에서 입력을 요구(Request)하고 입력장치에서 트리거(trigger)되면 측정치(measure)가 프로그램으로 반환된다.

샘플 모드 (Sample Mode): 프로그램에서 함수의 호출을 만나는 즉시 즉각적인 측정치(measure)가 반환된다.

이벤트 모드 (Event Mode): 프로그램에서는 이벤트를 기다리고, 입력장치에서 트리거(trigger)될 때마다 이벤트(event)가 생성되고, 장치의 측정치(measure)가 장치 식별자와 함께 이벤트 큐에 들어가고, 프로그램에서는 해당 이벤트를 처리한다.

- 5) 다음은 간단한 OpenGL/GLUT 프로그램의 일부이다. 빈 칸에 함수의 기능을 간단히 설명하라.

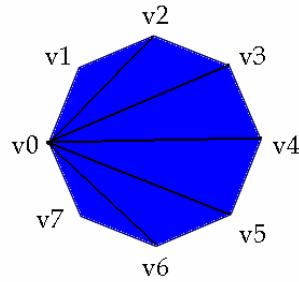
```
void reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, w, 0, h); // (1) _직교투영 함수 관측공간은 (0,0)에서 윈도우 크기/높이_____
    glMatrixMode(GL_MODELVIEW);
}

void keyboard(unsigned char key, int x, int y)
{
    if (key == 'q' || key == 'Q') {
        exit(0);
    }
    else if (key == 'r') {
        r = 1.0; g = 0.0; b = 0.0;
    }
    glutPostRedisplay(); // (2)_윈도우가 새로 그려져야할 필요가 있는 경우를 표시_____
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv); // 초기화
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB); // (3)_디스플레이의 표시모드를 설정.
    // RGB 칼러와 더블버퍼를 사용함_____
    glutInitWindowSize(400, 400); // 윈도우의 크기는 400x400
    glutInitWindowPosition(0,0); // 윈도우의 초기위치를 (0,0)
    glutCreateWindow(argv[0]); // 윈도우를 생성
    glutDisplayFunc(display); // 윈도우가 새로그리기를 요할 때 디스플레이 답신함수를 호출
    glutMouseFunc(mouse); // 마우스 버튼이 눌러졌을 때 호출
    glutMotionFunc(motion); // (4)_마우스 버튼이 눌러진 상태에서 움직일 때 호출_____
    glutKeyboardFunc(keyboard); // ACII character 키가 눌러졌을 때 호출
    glutSpecialFunc(special); // (5)_F1~F12나 방향키 등 특수키가 눌러졌을 때 호출_____
    glutReshapeFunc(reshape); // 윈도우가 처음 열렸을 때 크기가 바뀔 때 재구성 답신함수를 호출
    glutMainLoop();
}
```

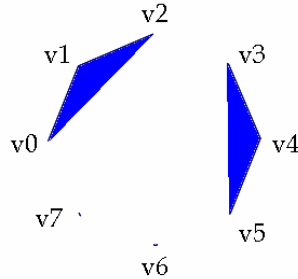
6) 다음은 간단한 OpenGL 프로그램의 일부이다. 그 출력 결과를 그림으로 그려라.

```
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glBegin(GL_TRIANGLE_FAN);
for (int i = 0; i < 8; i++)
    glVertex2fv(v[i]);
glEnd();
```



7) 다음은 간단한 OpenGL 프로그램의 일부이다. 그 출력 결과를 그림으로 그려라.

```
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glBegin(GL_TRIANGLES);
for (int i = 0; i < 8; i++)
    glVertex2fv(v[i]);
glEnd();
```



8) 원근감이 있는 3차원 물체를 그리고자 할 때 OPENGL 파이프 라인 (Graphics Pipeline)에서 다음 중 어디에서 원근감이 생성되는가? 답 투영변환

- 모델링 변환 (modeling transformation)
- 관측 변환 (viewing transformation)
- 투영 변환 (projection transformation)
- 뷰포트 변환 (viewport transformation)

9) 괄호 안을 채우시오.

$a \cdot b$	θ
$a \cdot b = 0$	$\theta = 90$
$a \cdot b < 0$	$90 < \theta \leq 180$
$a \cdot b > 0$	$0 \leq \theta < 90$

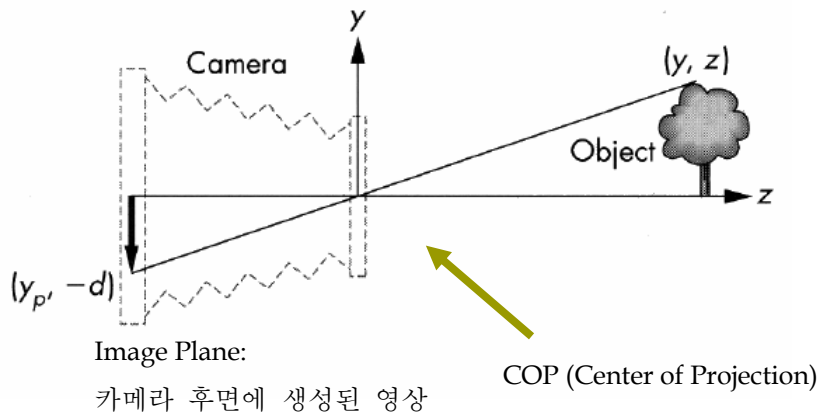
10) 회전축 $\mathbf{a}(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0)$ 에 90° 각도로 회전 (Rotate)하는 사원수 (Quaternion)을 유도하

라. ($\cos 90^\circ=0, \sin 90^\circ=1, \cos 45^\circ= \sin 45^\circ=0.707=\frac{\sqrt{2}}{2}$ 을 이용할 것)

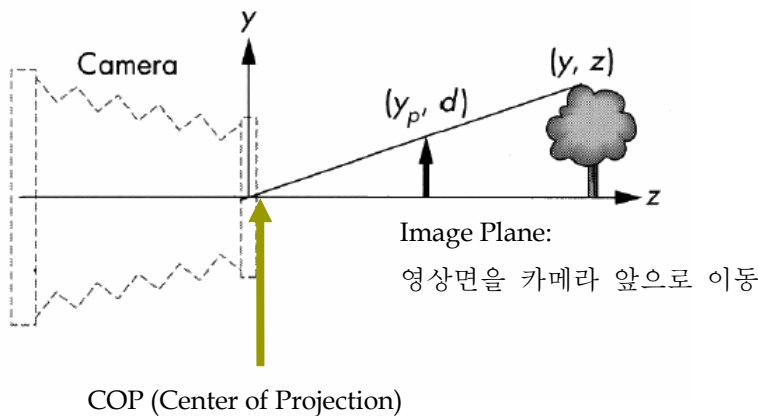
$$\begin{aligned}
 q &= \left[a_x \sin \frac{\theta}{2}, a_y \sin \frac{\theta}{2}, a_z \sin \frac{\theta}{2}, \cos \frac{\theta}{2} \right] \\
 &= \left[\frac{\sqrt{2}}{2} \sin 45, \frac{\sqrt{2}}{2} \sin 45, 0 \sin 45, \cos 45 \right] \\
 &= \left[\frac{1}{2}, \frac{1}{2}, 0, \frac{\sqrt{2}}{2} \right] = [0.5, 0.5, 0, 0.707]
 \end{aligned}$$

4. 바늘구멍 카메라 모델(Pinhole-Camera Model)과 합성 카메라 모델 (Synthetic-Camera Model)에 대해 물체(Object), 이미지 평면 (Image Plane), 투영선 (Projection), 카메라 중심 (Center of Projection)을 그림으로 나타내어 비교하라. (10점)

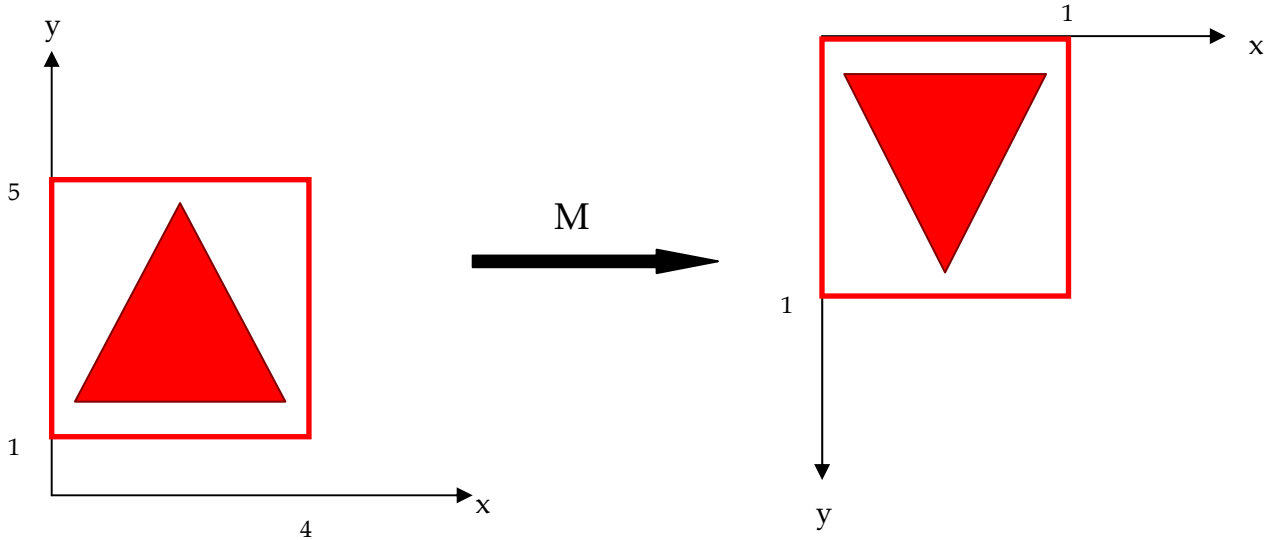
바늘구멍 카메라 모델 (Pinhole-Camera Model)



합성 카메라 모델 (Synthetic-Camera Model)



5. 아래 그림에서와 같이 왼쪽 삼각형을 오른쪽 삼각형으로 매핑(Mapping) 해주는 2차원 변환에 대한 3x3 행렬을 기본 아핀 변환 행렬인 $T(tx, ty)$, $S(sx, sy)$, $R(\theta)$ 등의 곱으로 표현한 후, 그 결과 행렬 M 을 구하라. (10점)



$$M = S\left(\frac{1}{4}, \frac{1}{4}\right)T(0, -1) = T(0, 1)S\left(\frac{1}{4}, \frac{1}{4}\right)T(0, -5) = T\left(\frac{1}{2}, \frac{1}{2}\right)S\left(\frac{1}{4}, \frac{1}{4}\right)T(-2, -3)$$

$$= \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6. 다음은 간단한 OPENGL 프로그램 일부를 보여주고 있다. 그 출력 결과 (즉, 1번부터 5번 삼각형을 표시하여)를 그려라. (10점)

```
void drawTriangle()
{
    glBegin(GL_TRIANGLES);
        glVertex2f(1, 1);
        glVertex2f(4, 1);
        glVertex2f(4, 6);
    glEnd();
}

void display(void)
{
    GLfloat M[16];
    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(0.0, 0.0, 0.0);
    glBegin(GL_LINES); // lines
```

```

    glVertex2f(-7, 0);
    glVertex2f(7, 0);
    glVertex2f(0, -7);
    glVertex2f(0, 7);

glEnd();

glPushMatrix(); // 1. gray triangle
glColor3f(0.5, 0.5, 0.5);
drawTriangle();
glPopMatrix();

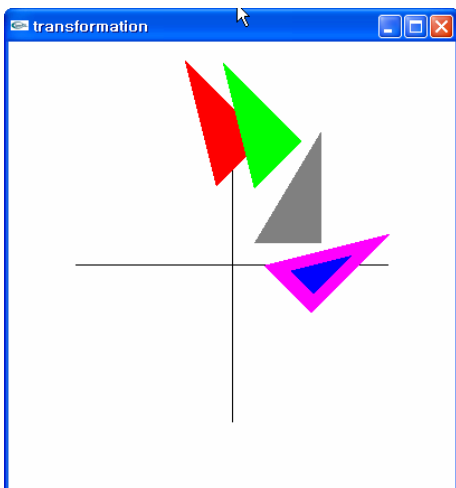
glPushMatrix(); // 2. red triangle
glRotatef(45, 0, 0, 1);
glTranslatef(1, 2, 0.0);
glColor3f(1.0, 0.0, 0.0);
drawTriangle();
glPopMatrix();

glPushMatrix(); // 3. green triangle
glTranslatef(1, 2, 0.0);
glRotatef(45, 0, 0, 1);
glColor3f(0.0, 1.0, 0.0);
drawTriangle();
glPopMatrix();

glPushMatrix();
glRotatef(-45, 0, 0, 1);
glColor3f(1.0, 0.0, 1.0);
drawTriangle(); // 4. magenta triangle
glTranslatef(3, 2.34, 0.0);
glScalef(0.5, 0.5, 1.0);
glTranslatef(-3, -2.34, 0.0);
glColor3f(0.0, 0.0, 1.0);
drawTriangle(); // 5. blue triangle
glPopMatrix();

glutSwapBuffers();
}

```



7. 벡터 v (3,4,2)가 단위벡터 n ($\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0$)에 평행한 v_{\parallel} 와 수직인 v_{\perp} 를 계산하시오. (보너스 문제 extra 10점)

$$\begin{aligned} v_{\parallel} &= n \frac{v \cdot n}{\|n\|^2} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right) \frac{(3,4,2) \cdot \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right)}{1^2} \\ &= \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right) \left(\frac{3\sqrt{2}}{2} + \frac{4\sqrt{2}}{2} + 0 \right) \\ &= \left(\frac{7}{2}, \frac{7}{2}, 0 \right) \end{aligned}$$

$$v_{\perp} = v - v_{\parallel} = (3,4,2) - \left(\frac{7}{2}, \frac{7}{2}, 0 \right) = \left(-\frac{1}{2}, \frac{1}{2}, 2 \right)$$