Graphics Programming

OpenGL 소개

HyoungSeok Kim

OpenGL 개발환경

- Header File
  - gl.h
  - glu.h
  - glut.h(glaux.h)
- Library File
  - gl.lib
  - glu32.lib
  - glut32.lib
- DLL File
  - glut32.dll

OpenGL Pipeline

1. 기하 프리미티브로부터 도형을 만든다.
2. 객체들을 3차원 공간에 배치시킨 다음,
3. 이를 바라볼 시점을 선택한다.
4. 모든 도형들에 대한 색상을 계산한다.
5. 거리 정보와 색상 정보를 스크린 상의 픽셀로 변환한다.

용어 정리

- 렌더링(Rendering)
  - 컴퓨터가 모델로부터 이미지를 생성하는 과정
- 기하 프리미티브(primitives)
  - 기본적인 도형 : 점, 선, 다각형
- 픽셀(pixel : picture element)
  - 스크린에 표현할 수 있는 가장 작은 단위의 사각형
- 비트 평면(bitplane)
  - 각 픽셀에 대한 비트의 정보가 저장되는 메모리
- 프레임 버퍼(Frame buffer)
  - 비트 평면들의 모임

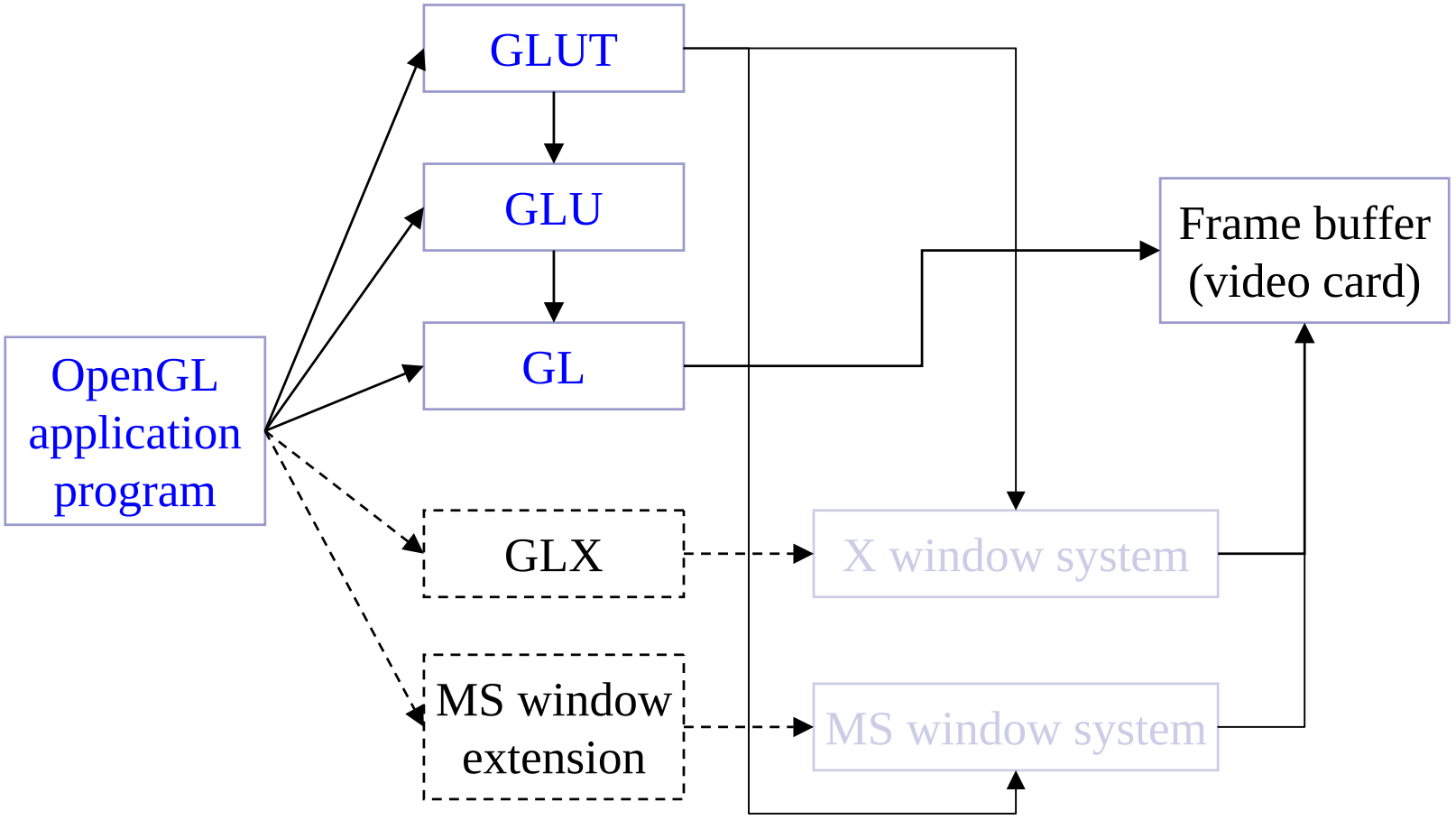
OpenGL Pipeline Architecture

- transformation
  - 좌표 변환을 위한 4×4 matrix multiplication
- clipping
  - projection plane 상에서 불필요한 부분 제거
- projection
  - 3D object → 2D image 로의 mapping
- rasterization
  - image 를 frame buffer에 저장하는 과정

OpenGL 구성

- 특성
  - C-based library (not object-oriented)
- 3개의 library 로 구성
  - GL : graphics library
    - H/W에서 지원하여야 하는 기본 primitives
  - GLU : graphics utility library
    - S/W로 지원해도 되는 확장 primitives
  - GLUT : GL utility toolkit
    - window system들을 위한 지원 함수들

OpenGL 구성


GLUT
GLU
GL
OpenGL application program
GLX
MS window extension
X window system
MS window system
Frame buffer (video card)

OpenGL 프로그램 구성도

이벤트 callback 함수

초기화 함수

Render() 함수

main() 함수

Main()

- 윈도우 관리 함수 + 이벤트 함수
  - void main (int argc, char** argv) {
    - glutInit(); // GLUT 초기화
    - glutInitDisplayMode (); // 그리기 설정
    - glutInitWindowPosition (); // 모니터에서 그려지는 윈도우의 위치 .
    - glutInitWindowSize (); // 윈도우의 크기.
    - glutCreateWindow (); // 원하는 이름의 윈도우 생성.
    - init(); // 가상 세계 및 환경 초기화.
    - glutDisplayFunc (display);
      - 윈도우의 내용을 다시 출력할 필요가 있을 때마다 등록된 display라는 함수를 호출
    - glutReshapeFunc (reshape);
      - 윈도우의 크기 변화가 있을 때마다 등록된 reshape 라는 함수를 호출
    - glutMouseFunc (mouse);
      - 마우스 이벤트가 발생할 때마다 등록된 mouse 라는 함수를 호출
    - **glutMainLoop();**
      - 지금까지 생성한 윈도우들과 그려진 그림들이 비로소 화면에 나타남
      - 이 함수가 없다면 그래픽스 프로그램이라 할 수 없음
  - }

- 입력 이벤트 처리
  - glutReshapeFunc (void (*func)(int w, int h));
    - 윈도우의 크기 변화가 있을 때마다 등록된 reshape 라는 함수를 호출
  - glutMouseFunc (void (*func)(int button, int state, int x, int y));
    - 마우스 이벤트가 발생할 때마다 등록된 함수 호출
    - button : 왼쪽 버튼 인지 오른쪽 버튼인지
    - state : 그 버튼이 눌려져 있는지 아니면 떨어져 있는지
    - (x, y) : 커서가 있는 위치
  - glutKeyboardFunc (void (*func)(unsigned char key, int x, int y));
    - 키를 누를 때 등록된 함수 호출
    - key : 어떤 키를 눌렀는지
    - (x, y) : 커서가 있는 위치
  - glutMouseFunc (void (*func)(int state, int x, int y));
    - 마우스 버튼을 누른 채 움직일 때 함수 호출
    - (x, y) : 커서의 위치
  - glutIdleFunc (void (*func)(void));
    - 처리해야 할 이벤트가 없을 때 지정된 함수 호출

Example 1

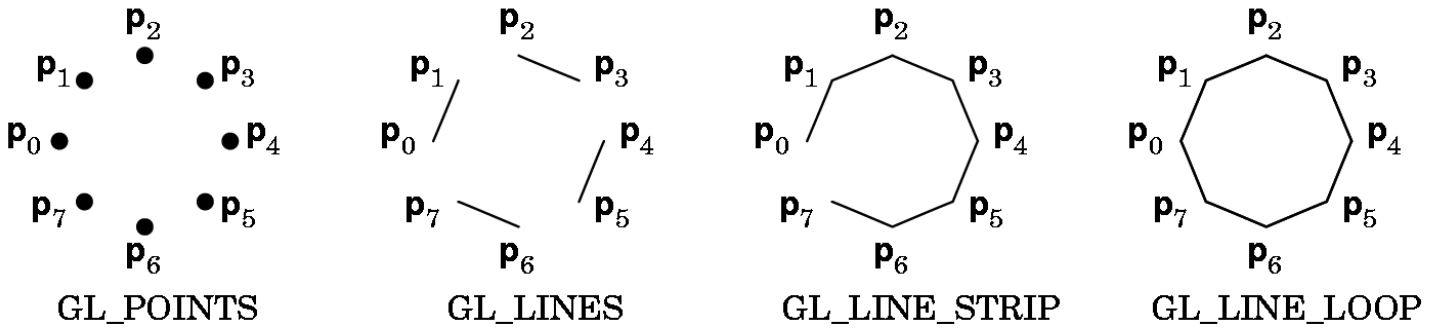

```
#include <gl/glut.h>   // (or others, depending on the system in use)

void init (void)
{
        glClearColor (1.0, 1.0, 1.0, 0.0) ;    // Set display-window color to white.
        glMatrixMode (GL_PROJECTION);    // Set projection parameters.
        gluOrtho2D (0.0, 200.0, 0.0, 150.0);
}
void lineSegment (void) {
        glClear (GL_COLOR_BUFFER_BIT);    // Clear display window.
        glColor3f(1.0, 0.0, 0.0);    // Set line segment color to red.
        glBegin (GL_LINES);
                glVertex2i (180, 15);    // Specify line-segment geometry.
                glVertex2i (10, 145);
        glEnd();
        glFlush ();    // Process all OpenGL routines as quickly as possible.
}
```

Example 1

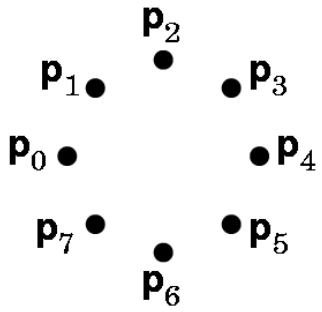

```
void main (int argc, char** argv)
{
   glutInit (&argc, argv);    // Initialize GLUT.
   glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);    // Set display mode.
   glutInitWindowPosition (50, 100);    // Set top-left display-window position.
   glutInitWindowSize (400, 300);    // Set display-window width and height.
   glutCreateWindow ("An Example OpenGL Program");    // Create display window.

   init();    // Execute initialization procedure.
   glutDisplayFunc (lineSegment);    // Send graphics to display window.
   glutMainLoop();    // Display everything and wait.
}
```

Line Primitives

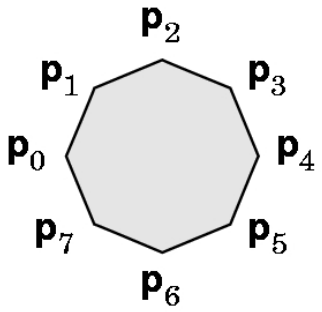
- 기본 구조
  - glBegin(TYPE);
    glVertex*(…);
    …
    glVertex*(…);
    glEnd( );
- type 별 출력 예제



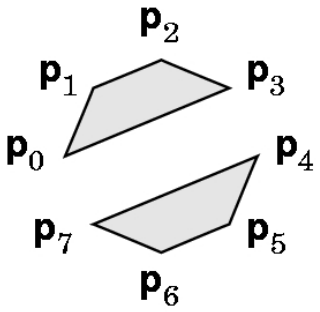
Polygon Primitives



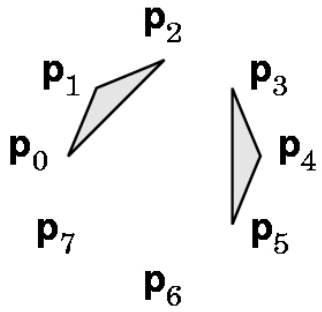
GL_POINTS



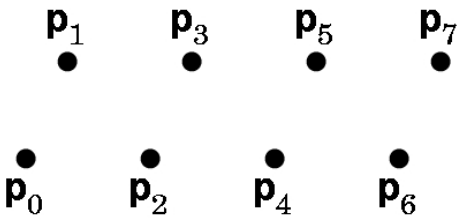
GL_POLYGON



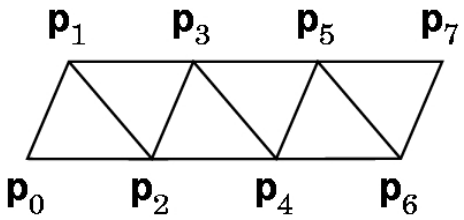
GL_QUADS



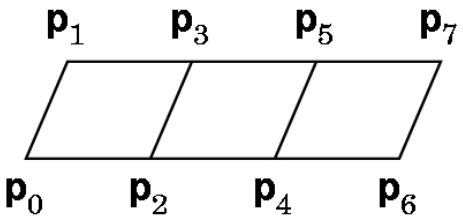
GL_TRIANGLES



GL_POINTS



GL_TRIANGLE_STRIP



GL_QUAD_STRIP

Example 2


```
#include <gl/glut.h>   // (or others, depending on the system in use)
static GLfloat spin = 0.0;

void init (void)
{
      glClearColor (0.0, 0.0, 0.0, 0.0) ;    // 검정색으로 칠하기
      glShadeModel(GL_FLAT);
}

void display(void)
{
      glClear(GL_COLOR_BUFFER_BIT);
      glPushMatrix();
      glRotatef(spin, 0.0, 0.0, 1.0);
      glColor3f(1.0, 1.0, 1.0);
      glRectf(-25.0, -25.0, 25.0, 25.0);
      glPopMatrix();
      glutSwapBuffers();
}
```

Example 2


```
void spinDisplay(void) {
        spin += 2.0;
        if (spin > 360.0)
                spin -= 360.0;
        glutPostRedisplay();
}

void spinReverse(void) {
        spin -= 2.0;
        if (spin < 0.0)
                spin += 360.0;
        glutPostRedisplay();
}

void reshape(int w, int h){
        glViewport(0, 0, (GLsizei) w, (GLsizei) h);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);
        glMatrixMode (GL_MODELVIEW);    // Set projection parameters.
        glLoadIdentity();
}
```

Example 2


```
void mouse(int button, int state, int x, int y) {
        switch(button) {
                case GLUT_LEFT_BUTTON:
                        if (state == GLUT_DOWN)  glutIdleFunc(spinDisplay);
                        break;
                case GLUT_RIGHT_BUTTON:
                        if (state == GLUT_DOWN)  glutIdleFunc(spinReverse);
                        break;
                default:      break;
        }
}
```

Example 2


```
void main (int argc, char** argv)
{
    glutInit (&argc, argv);    // Initialize GLUT.
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);    // Set display mode.
//      glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);    // Set display mode.
        glutInitWindowPosition (100, 100);    // Set top-left display-window position.
        glutInitWindowSize (250, 250);    // Set display-window width and height.
        glutCreateWindow (argv[0]);    // Create display window.
        init();    // Execute initialization procedure.
        glutDisplayFunc (display);    // Send graphics to display window.
        glutReshapeFunc (reshape);
        glutMouseFunc (mouse);
        glutMainLoop();    // Display everything and wait.
}
```

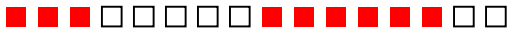
기하 프리미티브

- Void glBegin(Glenum mode)
  - 기하 프리미티브를 표현하는 정점 데이터 리스트의 시작 위치를 표시
  - Mode
    - GL_POINTS
    - GL_LINES
    - GL_LINE_STRIP
    - GL_LINE_LOOP
    - GL_TRIANGLES
    - GL_TRIANGLE_STRIP
    - GL_TRIANGLE_FAN
    - GL_QUADS
    - GL_QUAD_STRIP
    - GL_POLYGON
- Void glEnd();
  - 정점 데이터 리스트가 끝났다는 것을 표시

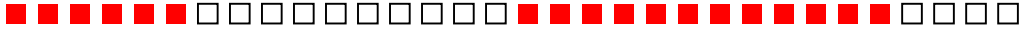
glBegin() – glEnd()

- glBegin()과 glEnd() 사이에 넣을 수 있는 명령어
  - glVertex*()
  - glColor*()
  - glIndex*()
  - glSecondaryColor*() : PostTexturing 에 대한 2차 색상 지정
  - glNormal*()
  - glMaterial()
  - glFogCoord*()
  - glTexCoord*()
  - glMultiTexCoord*ARB() : 멀티텍스링을 위한 텍스쳐 좌표 설정
  - glEdgeFlag*() : 모서리 그리기 제어
  - glArrayElement*() : 정점 배열 데이터를 가져오기
  - glEvalCoord*() : 좌표 생성
  - glCallList() : 디스플레이 리스트 실행
- 기본 상태 관리
  - Void glEnable(Glenum cap) : 지정한 기능 “cap” 을 활성화
  - Void glDisable(Glenum cap) : 지정한 기능 을 비활성화

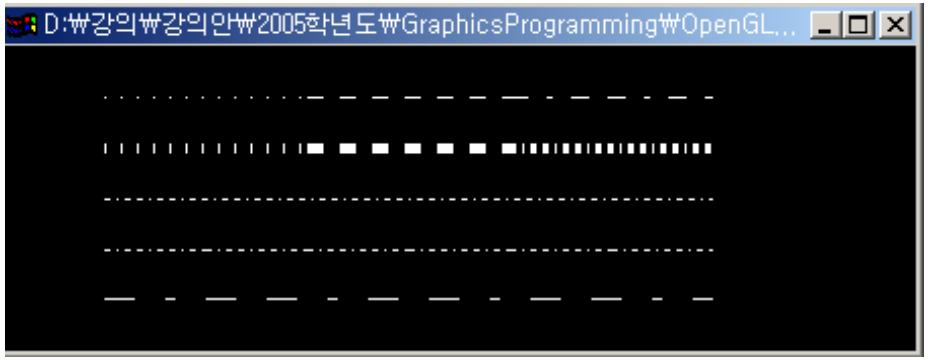
- Vertex
  - 점의 크기
    - Void glPointSize(GLfloat size)
- Line
  - 선분의 폭
    - Void glLineWidth(GLfloat width)
  - 선분의 스타일
    - Void glLineStipple(GLint factor, GLushort pattern) : 점선의 스타일 지정하는 함수
      - Factor : 그려지는 구간의 반복되는 정도
      - Pattern : 그릴 선의 패턴 : 16비트 이진수로 표현
    - 예제) glLineStipple(1, 0x3F07)
      - 0x3F07 = 3(0011)F(1111)0(0000)7(0111) = 0011111100000111



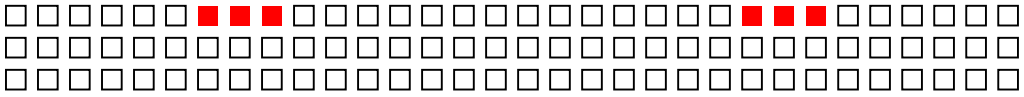
    - 예제) glLineStipple(2, 0x3F07)



- 예제
  - glEnable(GL_LINE_STIPPLE);
  - glLineStipple(1, 0x0101);
  - glBegin(GL_LINES);
    - glVertex2f(100, 100);
    - glVertex2f(300, 100);
  - glEnd();



- Polygon
  - 외형과 사용면 지정
    - Void glPolygonMode(Glenum face, Glenum mode)
      - Face : GL_FRONT_AND_BACK, GL_FRONT, GL_BACK
      - Mode : GL_POINT, GL_LINE, GL_FILL
  - 앞면 지정
    - Void glFrontFace(Glenum mode)
      - Mode : GL_CCW, GL_CW
  - 스티플링
    - Void glPolygonStipple(const Glubyte *mask)
      - Mask : 32 x 32 비트 윈도우 정렬 스티플 패턴
      - 0x00 0x00 0x00 0x00 (1)
      - 0x00 0x00 0x00 0x00 (2)
      - 0x03 0x80 0x01 0xC0 (3)
      - …



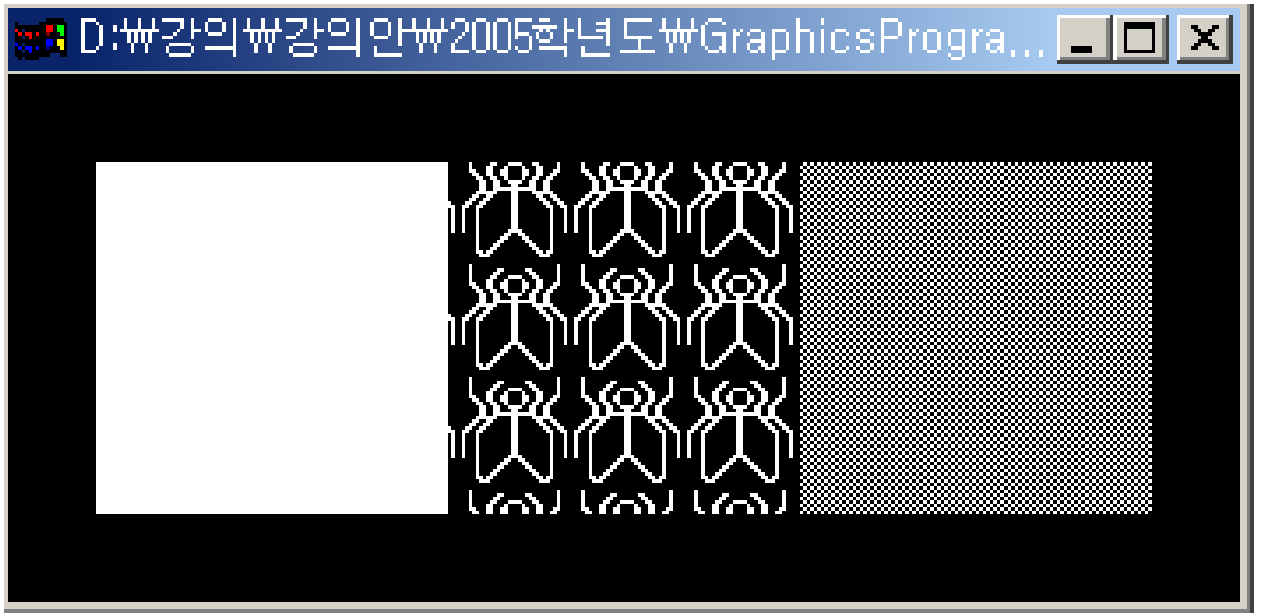
      - 0x00 0x00 0x00 0x00 (32)

```
void display(void)
{
        GLubyte fly[] = {
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x80, 0x01, 0xc0, 0x06, 0xc0, 0x03, 0x60,
        0x04, 0x60, 0x06, 0x20, 0x04, 0x30, 0x0c, 0x20, 0x04, 0x18, 0x18, 0x20, 0x04, 0x0c, 0x30, 0x20,
        0x04, 0x06, 0x60, 0x20, 0x44, 0x03, 0xc0, 0x22, 0x44, 0x01, 0x80, 0x22, 0x44, 0x01, 0x80, 0x22,
        0x44, 0x01, 0x80, 0x22, 0x44, 0x01, 0x80, 0x22, 0x44, 0x01, 0x80, 0x22, 0x44, 0x01, 0x80, 0x22,
        0x66, 0x01, 0x80, 0x66, 0x33, 0x01, 0x80, 0xcc, 0x19, 0x81, 0x81, 0x98, 0x0c, 0xc1, 0x83, 0x30,
        0x07, 0xe1, 0x87, 0xe0, 0x03, 0x3f, 0xfc, 0xc0, 0x03, 0x31, 0x8c, 0xc0, 0x03, 0x33, 0xcc, 0xc0,
        0x06, 0x64, 0x26, 0x60, 0x0c, 0xcc, 0x33, 0x30, 0x18, 0xcc, 0x33, 0x18, 0x10, 0xc4, 0x23, 0x08,
        0x10, 0x63, 0xc6, 0x08, 0x10, 0x30, 0x0c, 0x08, 0x10, 0x18, 0x18, 0x08, 0x10, 0x00, 0x00, 0x08
        };
```

```
GLubyte halftone[] = {
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55,
0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55, 0xAA, 0xAA, 0xAA, 0xAA, 0x55, 0x55, 0x55, 0x55};
```

```
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glRectf(25.0, 25.0, 125.0, 125.0);

    glEnable(GL_POLYGON_STIPPLE);
    glPolygonStipple(fly);

    glRectf(125.0, 25.0, 225.0, 125.0);
    glEnable(GL_POLYGON_STIPPLE);
    glPolygonStipple(halftone);

    glRectf(225.0, 25.0, 325.0, 125.0);
    glDisable(GL_POLYGON_STIPPLE);

    glFlush();

}
```

Attributes of Primitives



- Polygon
  - 경계 선분 여부 표시
    - Void glEdgeFlag(Glboolean flag)
      - flag : GL_TRUE, GL_FALSE
      - GL_TRUE : 그 정점부터 그려지는 선분은 경계 선분이므로 그려지게 함
      - GL_FALSE : 그 정점부터 그려지는 선분은 내부 선분으로 인식 그려지지 않게 함
    - 예제
    - glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    - glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
    - glBegin (GL_POLYGON);
      - glEdgeFlag(GL_TRUE);
      - glVertex2i (100, 200);
      - glEdgeFlag(GL_FALSE);
      - glVertex2i (100, 100);
      - glEdgeFlag(GL_TRUE);
      - glVertex2i (200, 100);
      - glEdgeFlag(GL_TRUE);
      - glVertex2i (200, 200);
    - glEnd();

- Polygon
  - 경계 선분 여부 표시
    - Void glEdgeFlag(Glboolean flag)
      - flag : GL_TRUE, GL_FALSE
      - GL_TRUE : 그 정점부터 그려지는 선분은 경계 선분이므로 그려지게 함
      - GL_FALSE : 그 정점부터 그려지는 선분은 내부 선분으로 인식 그려지지 않게 함
    - 예제
    - glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    - glPolygonMode(GL_FRONT_AND_BACK, GL_POINT);
    - glBegin (GL_POLYGON);
      - glEdgeFlag(GL_TRUE);
      - glVertex2i (100, 200);
      - glEdgeFlag(GL_FALSE);
      - glVertex2i (100, 100);
      - glEdgeFlag(GL_TRUE);
      - glVertex2i (200, 100);
      - glEdgeFlag(GL_TRUE);
      - glVertex2i (200, 200);
    - glEnd();

```
void display (void)
{
        glClear (GL_COLOR_BUFFER_BIT);
        glColor3f(1.0, 0.0, 0.0);
        glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
        glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
        glPointSize(10.0f);
        glBegin (GL_POLYGON);
                glEdgeFlag(GL_TRUE);
                glVertex2i (100, 200);
                glEdgeFlag(GL_FALSE);
                glVertex2i (100, 100);
                glEdgeFlag(GL_TRUE);
                glVertex2i (200, 100);
                glEdgeFlag(GL_TRUE);
                glVertex2i (200, 200);
        glEnd();
        glFlush ();
}
```

Normal Vectors

- Normal Vector
  - 표면의 수직을 향하는 벡터
  - 각 정점마다 하나의 법선 벡터를 지정
  - 좌표 지정 : glNormal3f(1.0, 2.0, 3.0);
  - 벡터 지정 : glNormal3fv(vector); float vector[] = {1.0, 2.0, 3.0};
  - 사용법
    - glBegin(primitive type);
      - glNormal3f(nx, ny, nz):
      - glVertex3f(x, y, z);
    - glEnd();
  - How to compute the normal vector of vertices ?

Example : circle


```
#define  DIVIDE  36
float   circle[DIVIDE][2];
void   MakeCircle(void) {
        float          delta = (float)(2*PI/DIVIDE);
        for ( int i = 0; i < DIVIDE; i++ ) {
                circle[i][0] = radius * cos(delta*i);
                circle[i][1] = radius * sin(delta*i);
        }
}
void   DrawCircle(void){
        glBegin(GL_POLYGON);
        for ( int i = 0; i < DIVIDE; i++ ) {
                glColor3f(1.0f, 0.0f, 0.0f);
                glVertex2f (circle[i][0], circle[i][1]);
        }
        glEnd();
}
void main (int argc, char** argv) {
    MakeCircle();
}
```

Example : REGULAR POLYGON(1)


```
int  divide=36;
float  circle[MAX_DIVIDE][2];
void  MakeCircle(void) {
        float        delta = (float)(2*PI/divide);
        for ( int i = 0; i < divide; i++ ) {
                circle[i][0] = radius * cos(delta*i);
                circle[i][1] = radius * sin(delta*i);
        }
}

void  DrawCircle(void){
        glBegin(GL_POLYGON);
        for ( int i = 0; i < divide; i++ ) {
                glColor3f(1.0f, 0.0f, 0.0f);
                glVertex2f (circle[i][0], circle[i][1]);
        }
        glEnd();
}
```

Example : REGULAR POLYGON(2)


```
void  keyfunction(unsigned char key, int x, int y) {
        switch(key) {
            case 'x' : divide += 1;
                       glutPostRedisplay();
                       break;
            case 'z' : divide -= 1;
                       glutPostRedisplay();
                       break;
            default:   break;
}
void main (int argc, char** argv) {
    MakeCircle();
    glutKeyboardFunc (keyfunction);
}
```

Example : Radius of Circle(1)


```
#include PI 3.1415926
float  delta;

void  DrawCircle(void){
    delta = 2.0 * PI / divide;
     glBegin(GL_POLYGON);
     for ( int i = 0; i < divide; i++ )
                glVertex2f (radius*cos(delta*i), radius*sin(delta*i));
     glEnd();
}
```

Example : Radius of Circle(2)


```
void  keyfunction(unsigned char key, int x, int y) {
        switch(key) {
            case ‘i' :  radius += 1;
                        glutPostRedisplay();
                        break;
            case ‘d' :  radius -= 1;
                        glutPostRedisplay();
                        break;
            default:
                        break;
}
void main (int argc, char** argv) {
    glutKeyboardFunc (keyfunction);
}
```

Example : Motion(1)


```
int    now_x = 250, now_y = 250;
void display(void) {
        now_x += xstep;
        now_y += ystep;

        glPushMatrix();
            glTranslatef(now_x, now_y, 0.0);
            glPushMatrix();
                glRotatef(spin, 0.0, 0.0, 1.0);
                DrawCircle();
            glPopMatrix();
        glPopMatrix();
        glutSwapBuffers();
}
```

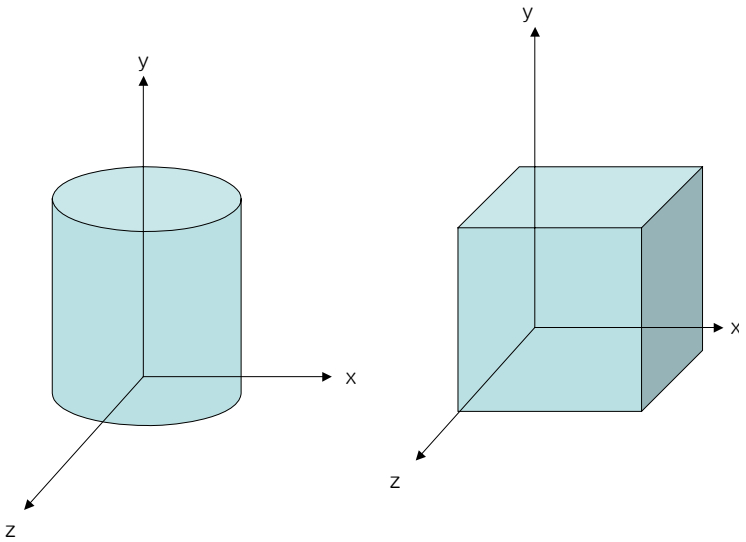
Example : Motion(2)


```
void mouse(int button, int state, int x, int y) {
        now_x = x;
        now_y = HEIGHT - y ;
}
void  keyfunction(unsigned char key, int x, int y)
{
        switch(key) {
        case 'm' :   xstep = 1.0f ;  ystep = 1.0f;
                     glutPostRedisplay();
                     break;
        case 's' :   xstep = 0.0f;   ystep = 0.0f;
                     glutPostRedisplay();
                     break;
        default:     break;
        }
}
```

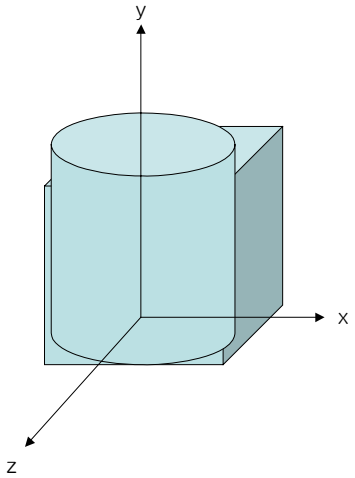
Example : Collision Detection(1)


```
int    now_x = 250, now_y = 250;
void display(void) {
    if (( now_x - radius <= 0.0 ) || ( now_x + radius >= WIDTH ))
                    xstep *= (-1.0f);
    if (( now_y - radius <= 0.0 ) || ( now_y + radius >= HEIGHT ))
                    ystep *= (-1.0f);

    now_x += xstep;
    now_y += ystep;
    glPushMatrix();
        glTranslatef(now_x, now_y, 0.0);
        glPushMatrix();
            glRotatef(spin, 0.0, 0.0, 1.0);
            DrawCircle();
        glPopMatrix();
     glPopMatrix();
     glutSwapBuffers();
}
```

월드 변환 -1

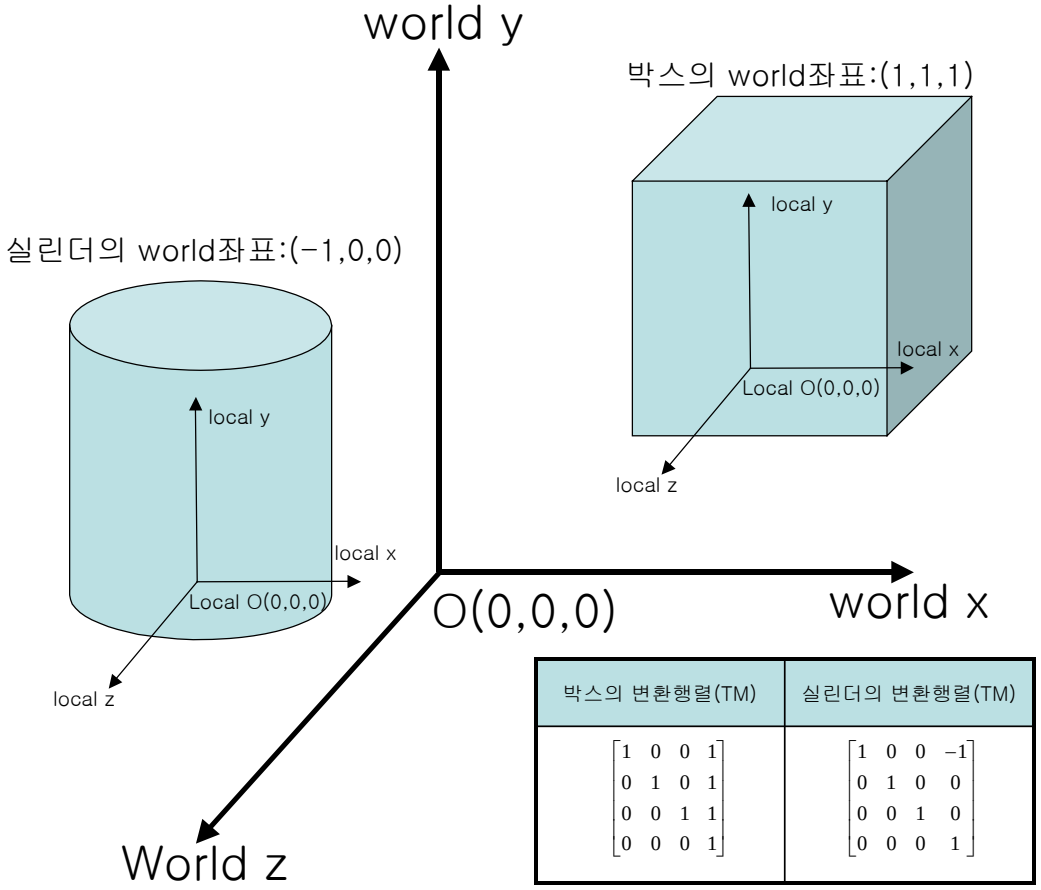


실린더와 박스



원점을 공유한 두 개의 물체

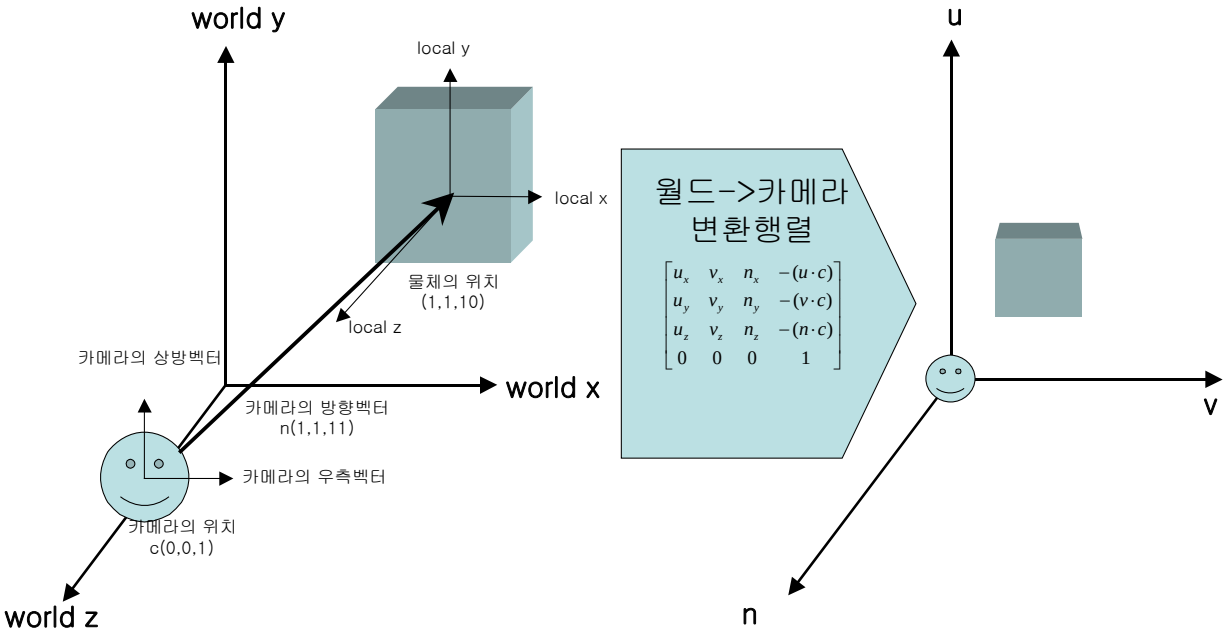
월드 변환 -2



박스의 변환행렬(TM)	실린더의 변환행렬(TM)
$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

TM을 적용해서 그린 경우

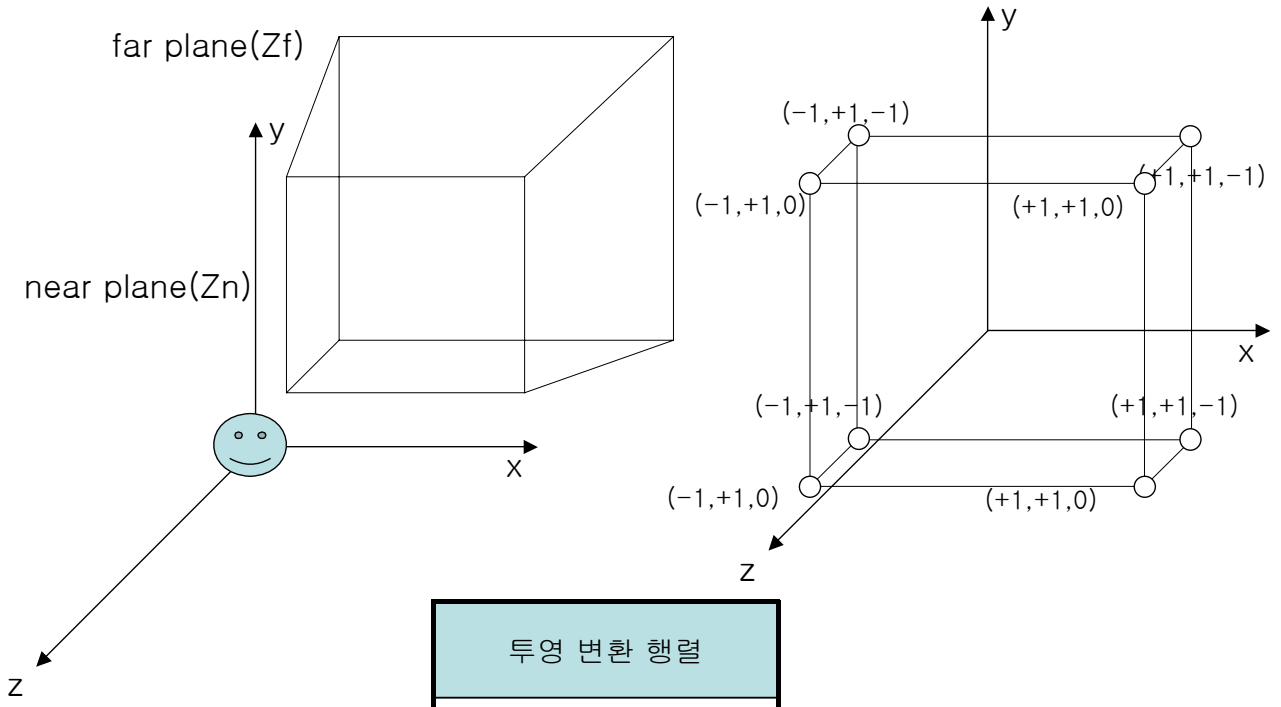
카메라 변환



월드좌표계 카메라좌표계

월드 -> 카메라 좌표계 변환

투영 변환



카메라 좌표계

투영 변환 행렬

$$\begin{bmatrix} \frac{2\times Z_n}{S_w} & 0 & 0 & 0 \\ 0 & \frac{2\times Z_n}{S_h} & 0 & 0 \\ 0 & 0 & \frac{Z_f}{Z_f - Z_f} & \frac{-Z_f \times Z_n}{Z_f \times Z_n} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

투영 좌표계

카메라 -> 투영 좌표계 변환

D3D의 전체 렌더링 파이프라인