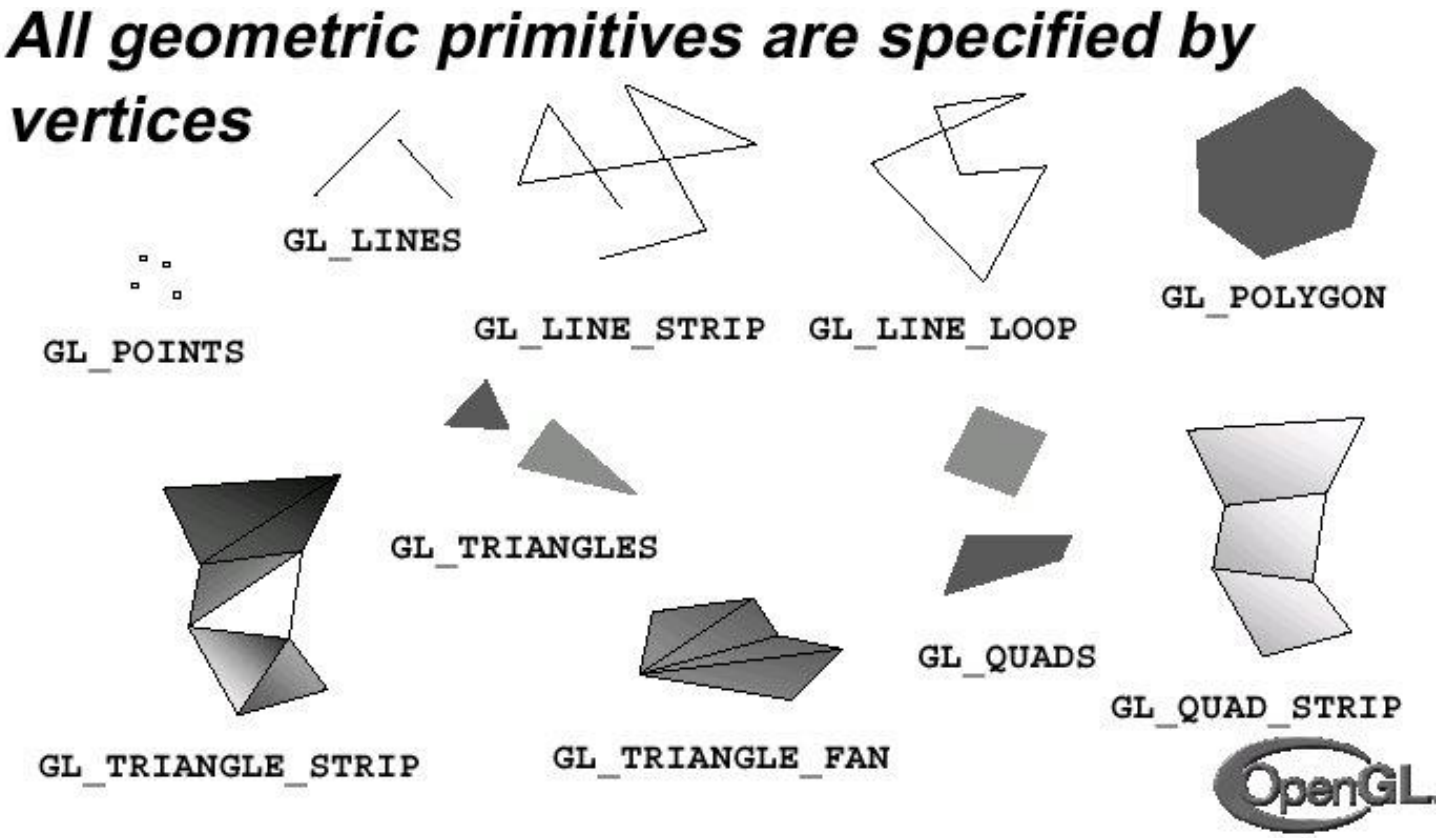
OpenGL 프로그래밍

- OpenGL Geometric Primitives
  - point, line, triangle, quad, polygon



도형 그리기

- glBegin(mode);

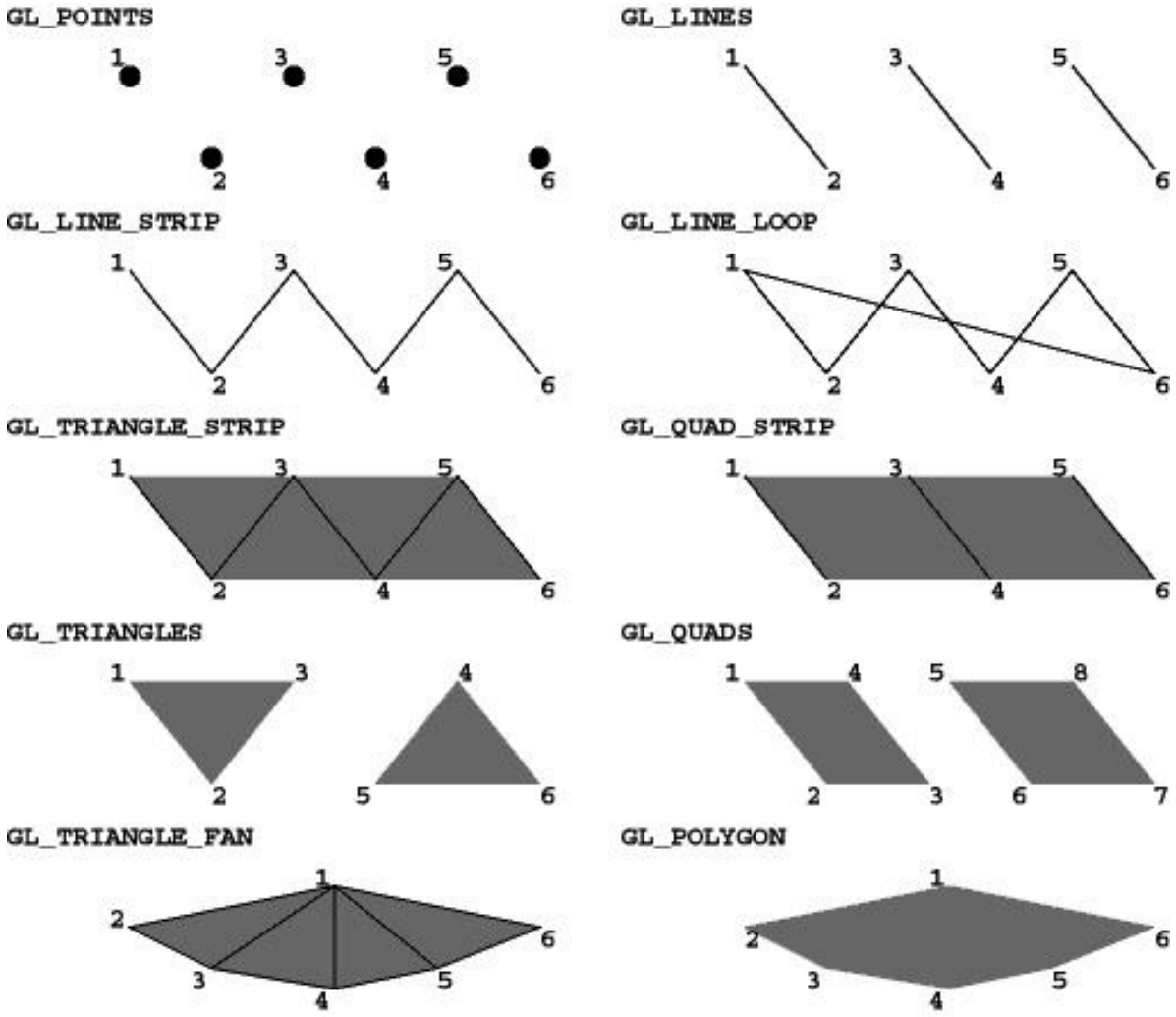
  glVertex{234}{sfid}[v](정점 위치 정보);

  gleEnd(mode);

  - glBegin과 glEnd 사이에 glVertex를 사용하여 필요한 정점들을 선언
  - mode를 통해 원하는 geometric primitive를 선택

Mode	설 명	Mode	설 명
GL_POINTS	각 정점의 위치에 점을 그림	GL_TRIANGLE_STRIP	정점의 인덱스 순서대로 삼각형을 그림
GL_LINES	정점 두개씩 묶어서 선을 그림	GL_TRIANGLE_FAN	하나의 정점을 기준으로 삼각형을 그려 부채 모양으로 그림
GL_LINE_STRIP	정점들을 각각의 순서에 따라 연결	GL_QUADS	네개의 정점을 연결해 사각형을 그림
GL_LINE_LOOP	STRIP과 비슷하지만 맨 끝과 맨 처음의 정점을 잇는 선을 그림	GL_QUAD_STRIP	사각형들을 정점의 인덱스 순서대로 그림
GL_TRIANGLES	세개의 정점들을 연결해 삼각형을 그림	GL_POLYGON	모든 정점들을 연결해 하나의 다각형을 그림

도형 그리기


GL_POINTS
1
3
5
2
4
6
GL_LINES
1
3
5
2
4
6
GL_LINE_STRIP
1
3
5
2
4
6
GL_LINE_LOOP
1
3
5
2
4
6
GL_TRIANGLE_STRIP
1
3
5
2
4
6
GL_QUAD_STRIP
1
3
5
2
4
6
GL_TRIANGLES
1
3
2
4
5
6
GL_QUADS
1
4
2
3
5
8
6
7
GL_TRIANGLE_FAN
1
2
6
3
5
4
GL_POLYGON
1
2
6
3
5
4

실습 Code: 03_1 도형 그리기

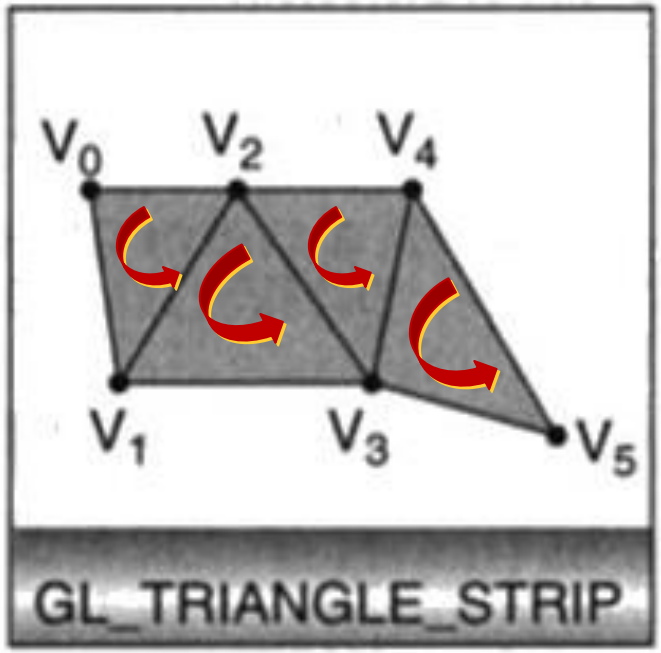
```
#include <GL/glut.h>
#include <stdlib.h>

float v1[3] = { 75.0, 400.0, 0.0};  // 첫번째 정점 위치
float v2[3] = {150.0, 100.0, 0.0};  // 두번째 정점 위치
float v3[3] = {225.0, 400.0, 0.0};  // 세번째 정점 위치
float v4[3] = {300.0, 100.0, 0.0};  // 네번째 정점 위치
float v5[3] = {375.0, 400.0, 0.0};  // 다섯번째 정점 위치
float v6[3] = {450.0, 100.0, 0.0};  // 여섯번째 정점 위치

void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 500.0, 0.0, 500.0);
}

// 빨강색으로 점을 그리는 코드
void draw_points()
{
    glColor3f(1.0, 0.0, 0.0);  // 빨강색

    glPointSize(4);
    glBegin(GL_POINTS);
        glVertex3fv(v1);
        glVertex3fv(v2);
        glVertex3fv(v3);
        glVertex3fv(v4);
        glVertex3fv(v5);
        glVertex3fv(v6);
    glEnd();
}

// 녹색으로 선(lines)을 그리는 코드
void draw_lines()
{
    glColor3f(0.0, 1.0, 0.0);  // 녹색
    glLineWidth(1);
    glBegin(GL_LINES);
        glVertex3fv(v1);
        glVertex3fv(v2);
        glVertex3fv(v3);
        glVertex3fv(v4);
        glVertex3fv(v5);
        glVertex3fv(v6);
    glEnd();
}
```

```
// 녹색으로 삼각형(triangle fan)을 그리는 코드
void draw_triangle_fan()
{
    glColor3f(0.0, 1.0, 0.0);  // 녹색

    // 순서에 주의
    glBegin(GL_TRIANGLE_FAN);
        glVertex3fv(v1);
        glVertex3fv(v2);
        glVertex3fv(v3);
        glVertex3fv(v4);
        glVertex3fv(v5);
        glVertex3fv(v6);
    glEnd();
}

// 녹색으로 사각형(quads)을 그리는 코드
void draw_quads()
{
    glColor3f(0.0, 1.0, 0.0);  // 녹색

    // 순서에 주의
    glBegin(GL_QUADS);
        glVertex3fv(v1);
        glVertex3fv(v2);
        glVertex3fv(v4);
        glVertex3fv(v3);
    glEnd();
}


// 녹색으로 사각형(quad_strip)을 그리는 코드
void draw_quad_strip()
{
    glColor3f(0.0, 1.0, 0.0);  // 녹색

    // 순서에 주의
    glBegin(GL_QUAD_STRIP);
        glVertex3fv(v1);
        glVertex3fv(v2);
        glVertex3fv(v3);
        glVertex3fv(v4);
        glVertex3fv(v5);
        glVertex3fv(v6);

    glEnd();
}
```

```
// 녹색으로 다각형(polygon)을 그리는 코드
void draw_polygon()
{
    glColor3f(0.0, 1.0, 0.0);  // 녹색

    // 순서에 주의
    glBegin(GL_POLYGON);
        glVertex3fv(v1);
        glVertex3fv(v2);
        glVertex3fv(v4);
        //glVertex3fv(v6);
        glVertex3fv(v5);
        glVertex3fv(v3);

    glEnd();
}

void display(void)
{
     glClear(GL_COLOR_BUFFER_BIT);
     draw_points();
     //draw_lines();
     //draw_line_strip();
     //draw_line_loop();
     //draw_triangles();
     //draw_triangle_strip();
     //draw_triangle_fan();
     //draw_quads();
     //draw_quad_strip();
     draw_polygon();
     glFlush();
}

int main(int argc, char** argv)
{
     glutInit(&argc, argv);
     glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
     glutInitWindowSize(500, 500);
     glutInitWindowPosition(300, 300);
     glutCreateWindow("My First GL Program");
     init();

     glutDisplayFunc(display);

     glutMainLoop();

     return 0;
}
```

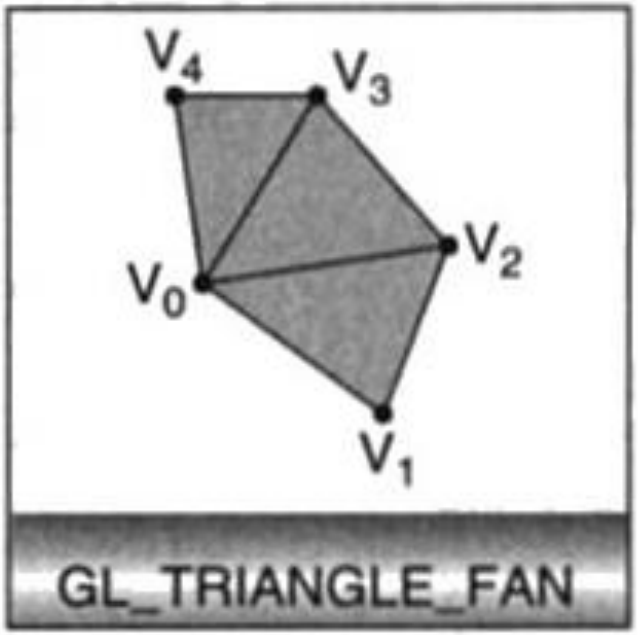
도형 그리기: GL_TRIANGLE_STRIP

- GL_TRIANGLE_STRIP
  - 연속된 정점들을 연결하여 삼각형들을 그려나감. V0, v1, v2, v3, v4, v5로 구성된 정점들에 대해서, v0, v1, v2로 삼각형을 그리고, v2, v1, v3로 삼각형을 그리고, v2, v3, v4, 마지막으로 v4, v3, v5의 순서로 삼각형을 그림
  - 같은 방향의 정점들로 삼각형이 구성됨



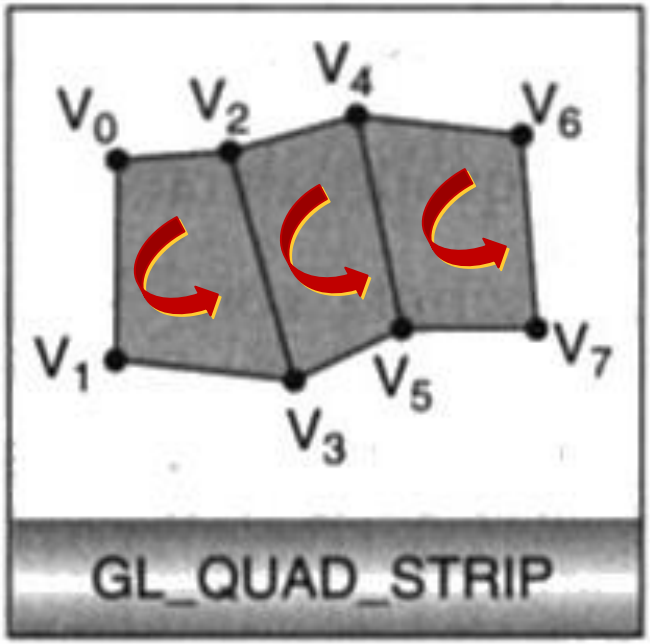
도형 그리기: GL_TRIANGLE_FAN

- GL_TRIANGLE_FAN
  - 연속된 정점들을 시작점을 기준으로 연결하여 삼각형들을 그려나감. V0, v1, v2, v3, v4로 구성된 정점들에 대해서, v0, v1, v2로 삼각형을 그리고, v0, v2, v3로 삼각형을 그리고, 마지막으로 v0, v3, v4의 순서로 삼각형을 그림
  - 같은 방향의 정점들로 삼각형이 구성됨



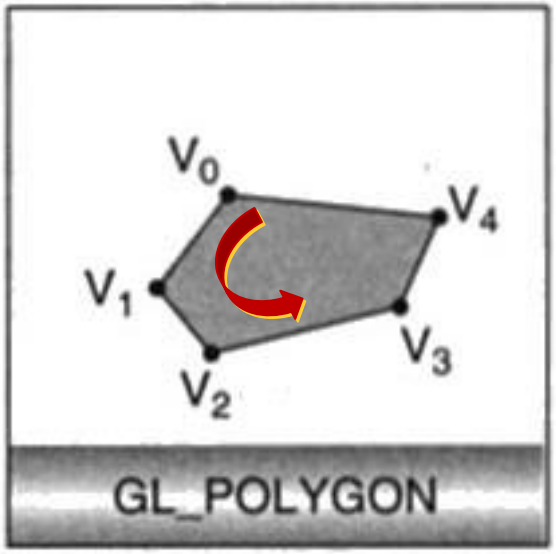
도형 그리기: GL_QUAD_STRIP

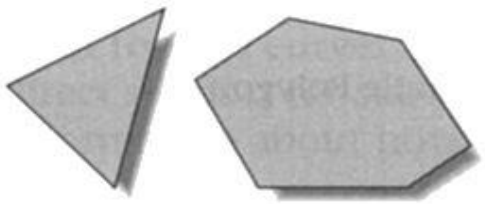
- GL_QUAD_STRIP
  - 연속된 정점들을 연결하여 사각형들을 그려나감. V0, v1, v2, v3, v4, v5, v6, v7로 구성된 정점들에 대해서, v0, v1, v3, v2로 사각형을 그리고, v2, v3, v5, v4로 사각형을 그리고, 마지막으로 v4, v5, v7, v6의 순서로 사각형을 그림
  - 같은 방향의 정점들로 사각형이 구성됨

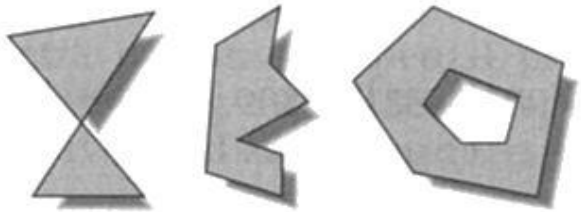


도형 그리기: GL_POLYGON

- GL_POLYGON
  - 연속된 정점들을 시작점을 기준으로 순서적으로 연결하여 다각형을 그림
  - OpenGL의 polygon의 edge들은 서로 교차 불가하며 Polygon들은 convex이여야 함
  - Convex: 어떠한 두 정점들이 polygon의 안에 있으면 두 정점을 잇는 직선 역시 polygon의 안에 있어야 함







실습 Code: 03_2 라인 그리기

```
/*
 *  lines.c
 *  This program demonstrates geometric primitives and
 *  their attributes.
*/

#include <GL/glut.h>
#include <stdlib.h>

#define drawOneLine(x1,y1,x2,y2)  glBegin(GL_LINES);  \
   glVertex2f ((x1),(y1)); glVertex2f ((x2),(y2)); glEnd();

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

void display(void)
{
    int i;

    glClear (GL_COLOR_BUFFER_BIT);

    /* select white for all lines  */
    glColor3f (1.0, 1.0, 1.0);

    /* in 1st row, 3 lines, each with a different stipple  */
    glEnable (GL_LINE_STIPPLE);

    glLineStipple (1, 0x0101);  /*  dotted  */
    drawOneLine (50.0, 125.0, 150.0, 125.0);
    glLineStipple (1, 0x00FF);  /*  dashed  */
    drawOneLine (150.0, 125.0, 250.0, 125.0);
    glLineStipple (1, 0x1C47);  /*  dash/dot/dash  */
    drawOneLine (250.0, 125.0, 350.0, 125.0);

    /* in 2nd row, 3 wide lines, each with different stipple */
    glLineWidth (5.0);
    glLineStipple (1, 0x0101);  /*  dotted  */
    drawOneLine (50.0, 100.0, 150.0, 100.0);
    glLineStipple (1, 0x00FF);  /*  dashed  */
    drawOneLine (150.0, 100.0, 250.0, 100.0);
    glLineStipple (1, 0x1C47);  /*  dash/dot/dash  */
    drawOneLine (250.0, 100.0, 350.0, 100.0);
    glLineWidth (1.0);

    /* in 3rd row, 6 lines, with dash/dot/dash stipple  */
    /* as part of a single connected line strip         */
    glLineStipple (1, 0x1C47);  /*  dash/dot/dash  */
    glBegin (GL_LINE_STRIP);
        for (i = 0; i < 7; i++) glVertex2f (50.0 + ((GLfloat) i * 50.0), 75.0);
    glEnd ();

    /* in 4th row, 6 independent lines with same stipple  */
    for (i = 0; i < 6; i++)
    {
        drawOneLine (50.0 + ((GLfloat) i * 50.0), 50.0, 50.0 + ((GLfloat)(i+1) * 50.0), 50.0);
    }

    /* in 5th row, 1 line, with dash/dot/dash stipple    */
    /* and a stipple repeat factor of 5                  */
    glLineStipple (5, 0x1C47);  /*  dash/dot/dash  */
    drawOneLine (50.0, 25.0, 350.0, 25.0);

    glDisable (GL_LINE_STIPPLE);
    glFlush ();
}

void reshape (int w, int h)
{
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D (0.0, (GLdouble) w, 0.0, (GLdouble) h);
}

/* ARGSUSED1 */
void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 27:
            exit(0);
        break;
    }
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (400, 150);
    glutInitWindowPosition (100, 100);
    glutCreateWindow (argv[0]);
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}
```

주요 함수 설명

- void glPointSize(GLfloat size);
  - 점의 크기를 설정하는 함수
  - size 인수는 점을 감싸는 원의 직경을 지정
- void glLineWidth(GLfloat width);
  - 선의 굵기를 지정하는 함수
  - width 인수는 선의 굵기를 지정
- void glLineStipple(GLint factor, GLushort pattern);
  - 여러 가지 선을 그릴 수 있는 함수
  - pattern은 이진수로 표현한 선의 모양
  - 하위 비트부터 선의 앞쪽 부분의 점 모양을 지정
  - 대응되는 비트가 1인 자리만 점이 찍힘
  - factor 인수는 비트 하나가 점 몇 개에 대응될 것인가를 지정

Stippled line

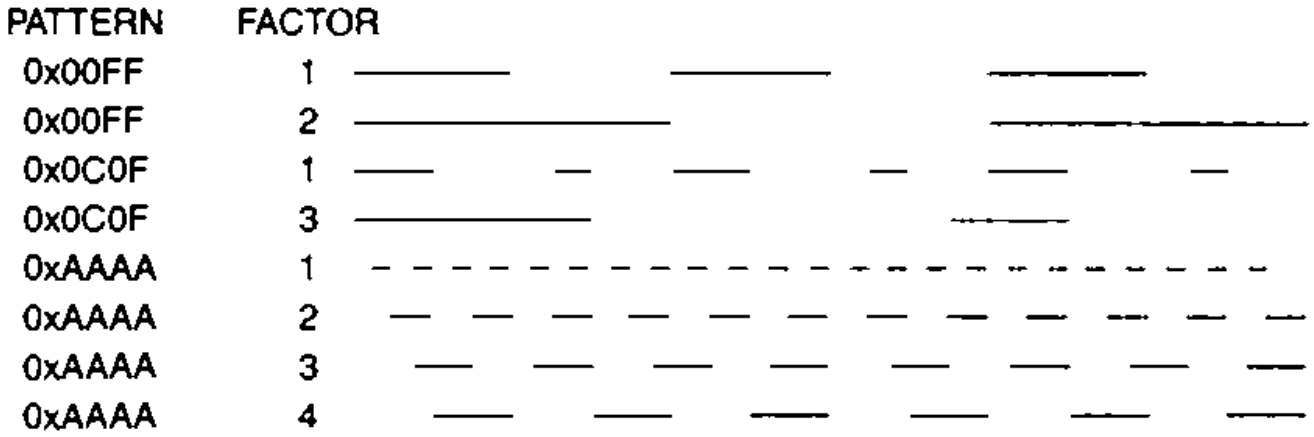


Figure 2-8 Stippled Lines

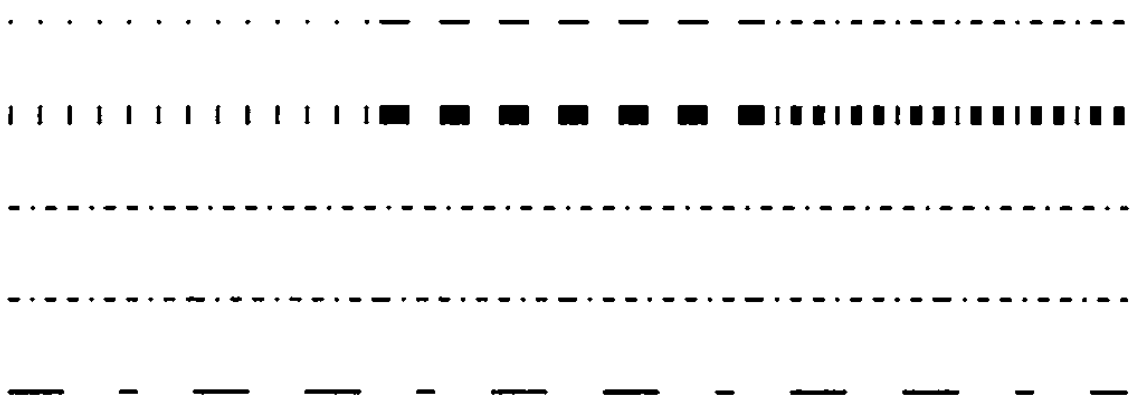


Figure 2-9 Wide Stippled Lines