

PRACTICA Nº 2. INTERACCIÓN CON EL RATÓN Y EL TECLADO

2.1 Dibujo de puntos introducidos por el ratón

Se trata de dibujar puntos en pantalla en la posición que ocupa el ratón en el momento de pulsar el botón izquierdo. A continuación se da parte del código a utilizar.

```
#define A1 400
//Función que gestiona el ratón
void mi_Raton(int boton, int estado, int x, int y)
{
    if (boton == GLUT_LEFT_BUTTON && estado == GLUT_DOWN)
        dibuja_Puntos(x, A1-y); //llama a la función
    else if (boton == GLUT_RIGHT_BUTTON && estado == GLUT_DOWN)
        glClear(GL_COLOR_BUFFER_BIT);
}
```

Nota: dado que el valor de la coordenada y que toma es respecto a la esquina superior izquierda, es necesario referirla a la esquina inferior izquierda..

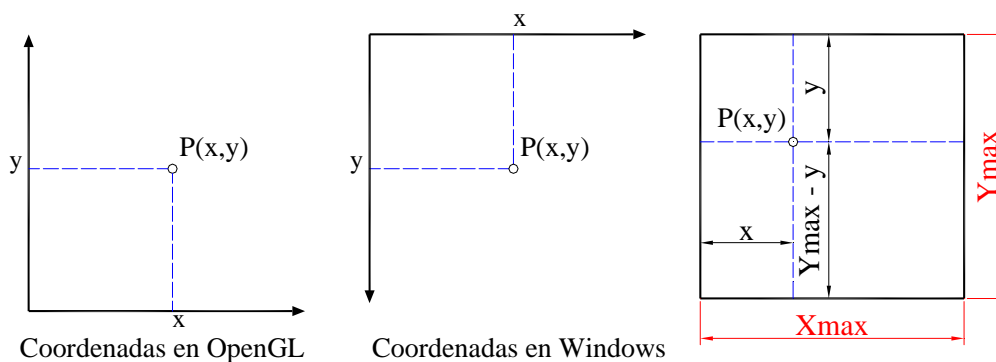


Figura 1. Sistemas de Coordenadas de OpenGL y Windows

2.2 Especificación de un rectángulo con el ratón

Implemente una función que sea capaz de dibujar un rectángulo definido por líneas dando las coordenadas de los vértices que lo definen con el ratón. Las coordenadas de cada rectángulo no tienen porque ser almacenadas, por lo tanto cada rectángulo nuevo reemplaza las coordenadas del previo. La pantalla se debe limpiar al pulsar el botón derecho del ratón. La parte de código que se da a continuación dibujará un rectángulo relleno.

```
#define A1 420
void miRaton(int boton, int estado, int x, int y)
{
    static GLintpoint corner[2];
    static int numCorners = 0; //inicia el valor a cero
    if (boton == GLUT_LEFT_BUTTON && estado == GLUT_DOWN)
    {
        corner[numCorners].x = x;          corner[numCorners].y = A1 - y;
        numCorners++;
        if (numCorners == 2){
            glRecti(corner[0].x,corner[0].y,corner[1].x,corner[1].y);
            numCorners = 0; //vuelve a 0 la esquina }
        }
    }
    else if(boton == GLUT_RIGHT_BUTTON &&state == GLUT_DOWN)
        glClear(GL_COLOR_BUFFER_BIT); //limpia la ventana
    glFlush();
}
```

2.3 Controlando la figura de Sierpinski con el ratón

Es simplemente una extensión de la rutina realizada en la práctica 1_4, en este caso el usuario debe poder especificar los tres vértices iniciales con el ratón. El siguiente código muestra como realizarlo, los tres puntos se almacenan en una matriz `corners[]`, y una vez introducidos los tres puntos se pasan como argumento a la función `Sierpinski()`.

```
#define A1 400
void myMouse(int boton, int estado, int x, int y)
{
    static GLintPoint corners[3];
    static int numCorners = 0;
    if (boton == GLUT_LEFT_BUTTON && estado == GLUT_DOWN)
    {
        corners[numCorners].x = x;
        corners[numCorners].y = A1 - y;
        if(++numCorners == 3)
        {
            Sierpinski(corners); //dibuja el gráfico de Sierpinski
            numCorners = 0; //vuelve a 0 la esquina
        }
    }
}
```

2.4 Dibujar una polilínea usando el ratón

Se trata de dibujar una polilínea de tal forma que los puntos sean introducidos con el ratón. En este caso los puntos introducidos se almacenarán en una lista para poder utilizarlos posteriormente. Si la matriz reservada se completa ya no se aceptarán más puntos. La polilínea se borra y se vacía la lista al pulsar el botón derecho del ratón.

```
void myMouse(int button, int state, int x, int y)
{
    static GLintPoint List[NUM];
    static int last = -1; //inicializa el índice para utilizar
    int AnchoPantalla = 400;
    //Test para el botón del ratón y para rellenar la matriz
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN && last < (NUM - 1))
    {
        List[++last].x = x;
        List[last].y = AltoPantalla - y;
        glClear(GL_COLOR_BUFFER_BIT); //limpia la pantalla
        glBegin(GL_LINE_STRIP);
        for (int i = 0; i <= last; i++)
            glVertex2i(List[i].x, List[i].y);
        glEnd();
        glFlush();
    }
    else if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
        last = -1; // para vaciar la lista
}
```

2.5 Dibujo a mano alzada con el ratón

Supongamos que se quiere crear una curva mediante el barrido del ratón a lo largo de una trayectoria sosteniendo pulsado el botón de éste. Además queremos que el aspecto de la trayectoria del dibujo tenga aspecto de cuadro. Esto es posible utilizando el siguiente código.

```
void MoverRaton(int RatonX, int RatonY)
{
    GLint x = RatonX;          //Graba la posición del ratón
    GLint y = A1 - RatonY;     //Cambio de referencia
    GLint brushSize = 5;       //Tamaño de la brocha
    glRecti(x,y, x+brushSize,y+brushSize);
    glFlush();
}
```

En main debe incluirse la función `glutMotionFunc(MoverRaton);` ver apartado 18 cap. 2 página 30.

```
// MAIN
void main (int argc, char** argv)
{
    ....
    ....
    glutMouseFunc(miRaton);
    glutMotionFunc(MoverRaton);
    ....
}
```

2.6 Interacción con el teclado

El programa tendrá interactividad con el teclado, de tal forma que al pulsar la tecla 'p' dibuje puntos en la posición del ratón, al pulsar 'l' dibuje líneas desde el origen a dicho punto, y al pulsar 'S' termine el programa.

La función de rellamada `myKeyboard()` se registra cuando el evento ocurre a través de `glutKeyboardFunc (myKeyboard)`. La función tiene de prototipo:

```
void myKeyboard(unsigned int key, int x, int y);
```

Los valores del teclado corresponden a los valores ASCII. Los valores de x e y corresponden a la posición del ratón en el momento en el que ocurre el evento.

```
void miTeclado(unsigned char tecla,int RatonX,int RatonY )
{
    static GLintPoint List[NUM];
    static int last = -1;
    GLint x = RatonX;          //Graba la posición del ratón
    GLint y = A1 - RatonY;     //Cambio de referencia
    switch(tecla)
    {
        case 'p':
            dibuja_Punto(x,y); // Dibuja un punto en la posición del ratón
            break;
        case 'l': List[++last].x = x; //añade un punto
            List[ last].y = y;
            glClear(GL_COLOR_BUFFER_BIT);
            glBegin(GL_LINE_STRIP);
            glVertex2i(0,0);
            glVertex2i(List[last].x,List[last].y);
            glEnd();
            glFlush();
            break;
        case 'S':
            exit(-1); //Termina el programa
        default:
            break; //No hace nada
    }
}
```

2.7 Dibujar una circunferencia dada por tres puntos mediante el ratón

Introducir tres puntos con el ratón y dibujar el círculo que pasa por dichos puntos. Represente los puntos introducidos con distinto color. (ver tema 4 apartado 4, páginas 68 y 69).

2.8 Parametrización de figuras

Dibuje una casa similar a la mostrada en la figura 2, de forma paramétrica. Los datos de entrada serán los puntos de coordenadas (x_1, y_1) y (x_2, y_2) que serán introducidas con el ratón.

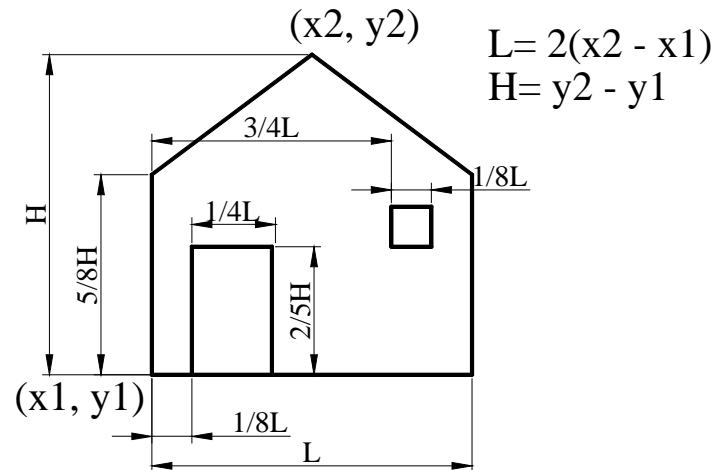


Figura 2. Datos de entrada de la práctica 2.8