

Métodos de interacción de entrada e interfaces gráficas

Contenido

- Motivación
- Clasificación lógica de los dispositivos de entrada
- Ejemplos de una aplicación interactiva en OpenGL

Motivación

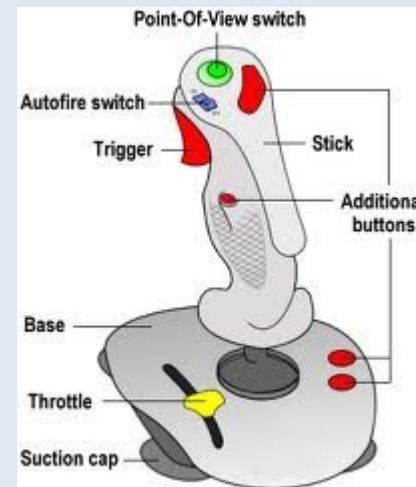
- Especificar input interactivamente
 - Cambiar la posición de un objeto apuntando la nueva posición con el mouse
 - Cambiar los parámetros de una simulación seleccionando el menú apropiado
 - Creación de dibujos ejecutada de manera interactiva

Dispositivos de entrada (input devices)

- La clasificación lógica estándar es:
 - **Locator**: un dispositivo para especificar una posición
 - **Stroke**: un dispositivo que especifica múltiples posiciones
 - **String**: un dispositivo para especificar strings (secuencias de caracteres)
 - **Valuator**: un dispositivo para especificar un valor escalar
 - **Choice**: seleccionar una opción de menú
 - **Pick**: un dispositivo para seleccionar un objeto de la escena

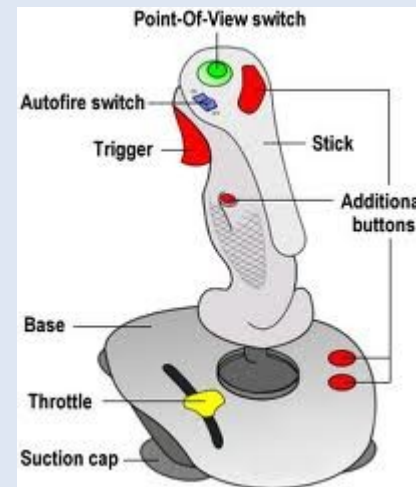
Dispositivos de entrada (input devices)

- Locator:
 - Ubicar el “cursor” en la pantalla
 - Dispositivos reales?: mouse, joystick, trackball, flechas del teclado, etc



Dispositivos de entrada (input devices)

- Locator:
 - Ubicar el “cursor” en la pantalla
 - Dispositivos reales?: mouse, joystick, trackball, flechas del teclado, etc



Rutinas de OpenGL para input-devices interactivos

- Funciones Glut para el mouse
 - `glutMouseFunc(mouseFcn)`: especifica que el procedimiento `mouseFcn` es el encargado de actuar cuando el mouse es presionado o soltado
 - MouseFcn debe tener 4 argumentos
 - `void mouseFcn(GLint button, GLint action, GLint xMouse, GLint ymouse);`
 - button puede ser: `GLUT_LEFT_BUTTON`,
`GLUT_MIDDLE_BUTTON`,
`GLUT_RIGHT_BUTTON`
 - action puede ser: `GLUT_DOWN` o `GLUT_UP`

Ejemplo: dibujo de puntos rojos

- En la posición del mouse, cada vez que se presiona el botón izquierdo, se dibuja un punto rojo de tamaño 3.

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

GLsizei winWidth = 400, winHeight = 300; // Tamaño inicial de la ventana
                                           // de visualización.

void init (void)
{
    glClearColor (0.0, 0.0, 1.0, 1.0) // Establecer azul como color ventana de
                                       // visualización.

    glMatrixMode (GL_PROJECTION);
    gluOrtho2D (0.0, 200.0, 0.0, 150.0);
}

void displayFcn (void)
{
    glClear (GL_COLOR_BUFFER_BIT); // Borrar ventana de visualización.

    glColor3f (1.0, 0.0, 0.0); // Establecer rojo como color de punto.
    glPointSize (3.0); // Definir tamaño de punto 3.0.
}
```


Ejemplo: dibujo de puntos rojos

```
void winReshapeFcn (GLint newWidth, GLint newHeight)
{
    /* Reinicializar parámetros de proyección y visor */
    glViewport (0, 0, newWidth, newHeight);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ( );
    gluOrtho2D (0.0, GLdouble (newWidth), 0.0, GLdouble (newHeight));

    /* Reinicializar parámetros de tamaño ventana de visualización. */
    winWidth = newWidth;
    winHeight = newHeight;
}

void plotPoint (GLint x, GLint y)
{
    glBegin (GL_POINTS);
        glVertex2i (x, y);
    glEnd ( );
}
```

Ejemplo: dibujo de puntos rojos

```
void mousePtPlot (GLint button, GLint action, GLint xMouse, GLint yMouse)
{
    if (button == GLUT_LEFT_BUTTON && action == GLUT_DOWN)
        plotPoint (xMouse, winHeight - yMouse);

    glFlush ( );
}

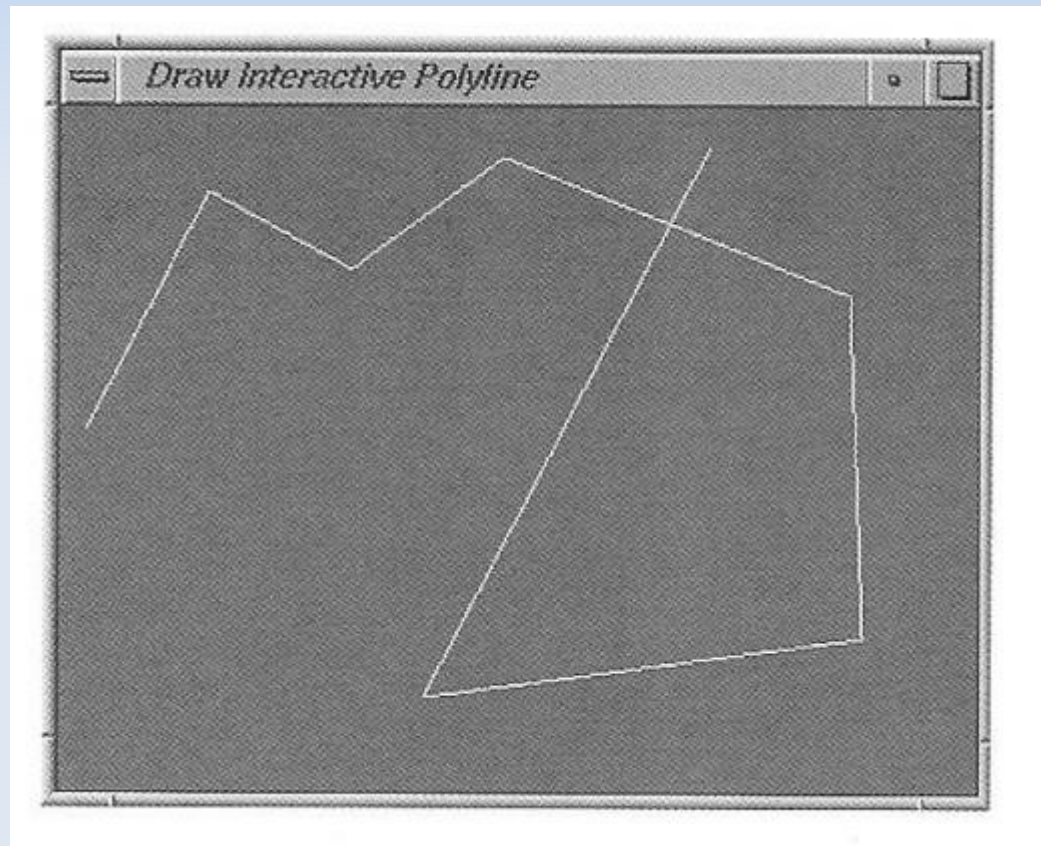
void main (int argc, char** argv)
{
    glutInit (&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition (100, 100);
    glutInitWindowSize (winWidth, winHeight);
    glutCreateWindow ("Mouse Plot Points");

    init ( );
    glutDisplayFunc (displayFcn);
    glutReshapeFunc (winReshapeFcn);
    glutMouseFunc (mousePtPlot);

    glutMainLoop ( );
}
```

Ejemplo: dibujo de una polilínea

Cada nueva posición seleccionada con el botón izquierdo dibuja una nueva línea entre la posición anterior y la actual.



Ejemplo: dibujo de una polilínea

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

GLsizei winWidth = 400, winHeight = 300; // Tamaño inicial ventana visualización.
GLint endPtCtr = 0; // Inicializar contador de puntos
// extremos de líneas.

class scrPt {
public:
    GLint x, y;
};

void init (void)
{
    glClearColor (0.0, 0.0, 1.0, 1.0) // Establecer azul como color ventana de
    // visualización.

    glMatrixMode (GL_PROJECTION);
    gluOrtho2D (0.0, 200.0, 0.0, 150.0);
}

void displayFcn (void)
{
    glClear (GL_COLOR_BUFFER_BIT);
}
```

Ejemplo: dibujo de una polilínea

```
void winReshapeFcn (GLint newWidth, GLint newHeight)
{
    /* Reinicializar parámetros de proyección y visor */
    glViewport (0, 0, newWidth, newHeight);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ( );
    gluOrtho2D (0.0, GLdouble (newWidth), 0.0, GLdouble (newHeight));
    /* Reinicializar parámetros de tamaño de la ventana de visualización. */
    winWidth = newWidth;
    winHeight = newHeight;
}
```

```
void drawLineSegment (scrPt endPt1, scrPt endPt2)
{
    glBegin (GL_LINES);
        glVertex2i (endPt1.x, endPt1.y);
        glVertex2i (endPt2.x, endPt2.y);
    glEnd ( );
}
```

Ejemplo: dibujo de una polilínea

```
void polyline (GLint button, GLint action, GLint xMouse, GLint yMouse)
{
    static scrPt endPt1, endPt2;

    if (ptCtr == 0) {
        if (button == GLUT_LEFT_BUTTON && action == GLUT_DOWN) {
            endPt1.x = xMouse;
            endPt1.y = winHeight - yMouse;
            ptCtr = 1;
        }
        else
            if (button == GLUT_RIGHT_BUTTON) // Salir del programa.
                exit (0);
    }
    else
        if (button == GLUT_LEFT_BUTTON && action == GLUT_DOWN) {
            endPt2.x = xMouse;
            endPt2.y = winHeight - yMouse;
            drawLineSegment (endPt1, endPt2);

            endPt1 = endPt2;
        }
        else
            if (button == GLUT_RIGHT_BUTTON) // Salir del programa.
                exit (0);

    glFlush ( );
}
```

Ejemplo: dibujo de una polilínea

```
void main (int argc, char** argv)
{
    glutInit (&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition (100, 100);
    glutInitWindowSize (winWidth, winHeight);
    glutCreateWindow ("Draw Interactive Polyline");

    init ( );
    glutDisplayFunc (displayFcn);
    glutReshapeFunc (winReshapeFcn);
    glutMouseFunc (polyline);

    glutMainLoop ( );
}
```

Nota: winReshapeFcn se llama cada vez que un usuario de la aplicación modifica el tamaño de la ventana