Interfaz de usuario: Tema 3

Máster en Programación de Videojuegos

Ignacio Martínez Rodríguez

Curso 2013-2014

Contenido

Tema 3: Ratón.

- Descripción
- Obtener características
- Métodos de lectura
  - Mensajes
  - Polling

Descripción

- Un ratón típico tiene 3 botones, una rueda de scroll, y dos ejes de movimiento X-Y.

- Podemos leer todos estos valores para controlar nuestro juego.

- En Windows, se pueden leer los valores con los mensajes del sistema operativo, de manera similar a como vimos con el teclado o por “polling” (excepto la rueda).

Características

- Para saber si el ratón está presente se puede usar la función ***GetSystemMetrics***, con el parámetro ***SM_MOUSEPRESENT*.**

```
if( GetSystemMetrics( SM_MOUSEPRESENT ) )
{
  //... Ratón presente, hacer lo que sea
}
```

Obtener características

- Podemos averiguar el número de botones del ratón, usando la misma función, con el parámetro **SM_CMOUSEBUTTONS.**

```
int numButtons = GetSystemMetrics( SM_CMOUSEBUTTONS ) );

if( numButtons == 1 )
{
  printf( “Un ratón de un Mac????” );
}
```

Obtener características

- Para saber si el ratón tiene rueda de scroll, usamos de nuevo la misma función con el parámetro ***SM_MOUSEWHEELPRESENT*.**

```
if( GetSystemMetrics( SM_MOUSEWHEELPRESENT ) )
{
  //... Perfecto, no tenemos que reinventar la rueda
}
```

Mensajes

- Cuando ocurre un evento relacionado con el ratón (pulsaciones de botones y movimientos) se envía un mensaje, y junto con él, las coordenadas del punto activo del cursor en el parámetro ***lParam***.

- Para extraer estos valores usaremos la macro ***MAKEPOINTS*** *con* ***lParams,*** lo cual nos devolverá una estructura ***POINTS*** con las coordenadas (x, y)

Mensajes

- El parámetro ***wParam*** contiene información sobre si ciertas teclas o botones del ratón están pulsados.
  - Flags de ***wParam***:

MK_CONTROL	Activo si la tecla CTRL está pulsada.
MK_LBUTTON	Activo si el botón izquierdo del ratón está pulsado.
MK_MBUTTON	Activo si el botón central del ratón está pulsado.
MK_RBUTTON	Activo si el botón derecho del ratón está pulsado.
MK_SHIFT	Activo si la tecla MAYÚSCULAS está pulsada.

Mensajes

- Mensaje ***WM_MOUSEMOVE*** se recibe cada vez que el usuario mueve el ratón.
- Las coordenadas son con respecto a la esquina superior izquierda de área de cliente (la ventana).

```
bool butLeft, butCenter, butRight;
POINTS point;
...
case WM_MOUSEMOVE:
    point = MAKEPOINTS( lParam );
    butLeft = ( wParam & MK_LBUTTON );
    butCenter = (wParam & MK_MBUTTON);
    butRight = (wParam & MK_RBUTTON);
...
```

Mensajes

- Mensajes de botones:

WM_LBUTTONDOWN **WM_LBUTTONUP**	Botón izquierdo
WM_MBUTTONDOWN **WM_MBUTTONUP**	Botón central
WM_RBUTTONDOWN **WM_RBUTTONUP**	Botón derecho

Mensajes

- Ejemplo:

```
...
case WM_LBUTTONDOWN:
    player->Shoot();
    break;
case WM_RBUTTONDOWN:
    player->Jump();
    break;
...
```

Mensajes

- Capturaremos el mensaje ***WM_MOUSEWHEEL*** para conocer los movimientos de la rueda de scroll. Siempre nos dará valores múltiplos de 120. Para conocer el valor usaremos la macro ***GET_WHEEL_DELTA_WPARAM(wParam)***

```
case WM_MOUSEWHEEL:
    int delta = GET_WHEEL_DELTA_WPARAM( wParam );
    camera->ChangeZoom( delta );
    break;
```

Polling

- También podemos leer los valores por “polling”.
  - Posición del puntero:

```
GetCursorPos( _Out_ LPPOINT lpPoint );
```

Ejemplo

```
void Cursor::Update()
{
  POINT point;

  GetCursorPos( &point );
  ScreenToClient( m_hWnd, &point );

  m_sprite->SetPosition( poin.x, point.y);
}
```

Polling

- El estado de los botones se lee igual que el teclado, con ***GetAsyncKeyState:***

```
if( GetAsyncKeyState( VK_LBUTTON ) & 0x8000 )
{
    m_player->Shot();
}
```

Práctica

- **Práctica 1**

  Mover un píxel blanco por la pantalla manejándolo con el ratón.

  Si se pulsa el botón izquierdo el píxel será de color verde.

  Si se pulsa el derecho será de color rojo.

  Si se pulsa la tecla “s” reproduciremos un sonido del tipo beep.