

4 Input Devices

4.1 Aims

DirectInput supplies a variety of human-input interfaces. This module explores how to use DirectInput to tap into the keyboard and the mouse.

4.2 DirectInput Overview

DirectInput supports input devices in the same way that DirectDraw supports graphics devices; it gives you raw, quick access to all kinds of hardware in a uniform fashion. Setting up DirectInput is a little different to DirectDraw, in that there is often more than one device that you will wish to use.

4.2.1 Get the Interface

Use `DirectInputCreate()` to get the `IDirectInput` interface. You can get the `IDirectInput2` interface using `QueryInterface()` from here as you did with `DirectDraw`.

4.2.2 Create the Devices

For the keyboard and mouse, call `CreateDevice()` using the built-in GUIDs `GUID_SysKeyboard` and `GUID_SysMouse`. If you want to use a joystick or a gamepad, you'll need to query for the device GUIDs, which gets a little messy.

4.2.3 Set the Cooperative Level

Use `SetCooperativeLevel()` in a similar fashion to `DirectDraw` to let Windows know how much control (if any) you are leaving for it. The principle choices are between whether you want access to the input devices whether the application is in the foreground or the background; and whether you want exclusive access to the device or whether Windows gets to play, too. For the keyboard, you'll want to use foreground, non-exclusive access to allow other applications to use keyboard, and to allow the task switching keys to still work. If you set the mouse as exclusive access, you'll have to create your own cursor and track it manually. You'll want it to have foreground access to allow other applications to use it if you task switch. Most of your joysticks will want exclusive access, and the

foreground/background access is largely irrelevant as other applications are likely to want access.

4.2.4 Set the Data Format and Properties

The `SetDataFormat()` method allows you to decide how you get your data. You can choose from one of the pre-defined formats or create one of your own. The pre-defined formats for the keyboard and mouse are `c_dfDIKeyboard` and `c_dfDIMouse` respectively. We'll get to the actual format itself later.

You can also set the properties of devices like joysticks using `SetProperty()`.

4.2.5 Acquire the Device

Now you are ready to tell `DirectInput` that you want to attach the device(s) to your application using `Acquire()`. That's it – no tricks here.

4.2.6 Get some Data

Now you're ready to grab the data from the device. You can get the data from the device using `GetDeviceState()` in a similar fashion to `GetAsyncKeyState()`. For the keyboard, the data are returned in an array of 256 chars, where each char represents the state of a key. The index into the array is defined in `DINPUT.H`.

For the mouse, the data is a data structure of type `DIMOUSESTATE` which gives the displacement along the x and y axes, as well as wheel data and button state.



Recode your tennis game from last week to use `DirectInput` to get the data from the keyboard.