

0.1 User Inputs

In an era before Microsoft harnessed all inputs under DirectInput API with Windows 95, developers had to write drivers for each input type they wanted to support. This involved talking directly to the hardware in the vendor's protocol on a physical port. The keyboard is plugged into a PS/2 or AT port, the mouse to a serial port (DE9), and the joystick to a game port (DA-15).

0.1.1 Keyboard

As the keyboard is the standard and oldest input medium, it is fairly easy to access. When a key is pressed, the interrupt is routed to an ISR in the Vector Interrupt Table. The engine installs its own ISR there.

```
#define KeyInt      9 // The keyboard ISR number

static void INL_StartKbd(void) {

    IN_ClearKeysDown();

    OldKeyVect = getvect(KeyInt);
    setvect(KeyInt, INL_KeyService);

    INL_KeyHook = 0; // Clear key hook
}

static void interrupt INL_KeyService(void) {
    byte k;
    k = inportb(0x60); // Get the scan code

    // Tell the XT keyboard controller to clear the key
    outportb(0x61, (temp = inportb(0x61)) | 0x80);
    outportb(0x61, temp);

    [...] // Process scan code.
    Keyboard[k] = XXX;

    outportb(0x20, 0x20); // ACK interrupt to interrupt system
}
.
```

The state of the keyboard is maintained in a global array `Keyboard`, available for the entire engine to lookup.

```
#define NumCodes 128
boolean Keyboard[NumCodes];
```

0.1.2 Mouse

A driver has to be loaded at startup for the mouse to be accessible. DOS did not come with one. It was usually on a vendor provided floppy disk. MOUSE.COM (or MOUSE.SYS) had to be added to config.sys so it would reside in RAM. It was usually stored in DOS folder.

```
C:\DOS\MOUSE.COM
```

The driver takes almost 5KiB of RAM. With the driver loaded all interactions happen with software interrupt 0x33. The interface works with requests issued in register AX and responses issued in registers CX, BX and DX. With Borland compiler syntactic sugar it is easy to write with almost no boilerplate (notice direct access to registers thanks to _AX and co special keywords).

```
#define MouseInt 0x33
#define Mouse(x) _AX = x,geninterrupt(MouseInt)

static void INL_GetMouseDelta(int *x,int *y) {
    Mouse(MDelta);
    *x = _CX;
    *y = _DX;
}
```

Request	Type	Response
AX=0	Get Status	AX = FFFFh : available. AX Value = 0 : not available
AX=1	Show Pointer	
AX=2	Hide Pointer	
AX=3	Mouse Position	CX = X Coordinate, DX = Y Coordinate
AX=3	Mouse Buttons	BX = 1 Left Pressed, BX = 2 Right Pressed, BX = 3 Center Button Pressed
AX=7	Set Horizontal Limit	CX=MaxX1 DX=MaxX2
AX=8	Set Vertical Limit	CX=MaxY1 DX=MaxY2
AX=11	Read Mouse Motion Counters	CX = horizontal mickey ¹ count, DX = vertical mickey count

Figure 1: Mouse request/response.

