HowToCode7

COLLABORATORS			
	TITLE : HowToCode7		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 28, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

HowToCode7

1.1 HowToCode: Input

```
                Handling Input for Amiga Demos
                ------------------------------

  1  Keyboard Timings
  2  Bogus Mouse Clicks
  3  Hardware differences
```

1.2 Keyboard Timings

```
Keyboard Timings
----------------

If you have to read the keyboard by hardware, be very careful
with your timings. Not only do different processor speeds affect
the keyboard timings (for example, in the game F-15 II Strike Eagle
on an Amiga 3000 the key repeat delay is ridiculously short, you
ttyyppee lliikkee tthhiiss aallll tthhee ttiimmee. You use
up an awful lot of Sidewinders very quickly!), but there are differences
between different makes of keyboard, some Amiga 2000's came with
Cherry keyboards, these have small function keys the same
size as normal alphanumeric keys - these keyboards have different
timings to the normal Mitsumi keyboards.

Use an input handler to read the keyboard. The Commodore guys
have spent ages writing code to handle all the different possible
hardware combinations around, why waste time reinventing the wheel?
```

1.3 Bogus Input

```
Beware bogus input falling through to Workbench
-----------------------------------------------

If you keep multitasking enabled and run your own copperlist remember
```

that any input (mouse clicks, key presses, etc) fall through to the
workbench. The correct way to get around this is to add an input
handler to the IDCMP food chain (see - you *do* have to read the
other manuals!) at a high priority to grab all input events before
workbench/cli can get to them. You can then use this for your
keyboard handler too (no more $bfexxx peeking, PLEASE!!!)

Look at the sourcecode for Protracker for an excellent example of
how to do the job properly. Well done Lars!

1.4 Hardware Differences

Hardware Differences

The A1200 has a different keyboard to older Amigas. One of the
side effects of this is it appears that older hardware-hitting
keyboard read routines are not able to register more than one
keypress at a time. I currently do not know whether this is a
limitation in hardware and if it is possible to read multi-key
presses (excepting the obvious CTRL/ALT/SHIFT type combinations)
at all... A bit annoying for games writers I would think.

If you now are using you own hardware and Interrupts to read
the keyboard on faster computers, make sure you ALWAYS have
the given time-delay for ALL keyboards you want your program
to work with (The delay between or.b #$40,$bfee01 and
or.b #$40,$bfee01)! Don't trust delay loops since the
cache can speed those up rather drastically!
I have seen too many, even commercial, programs that
just ignores this and have NO delay code or just a
simple dbf-loop. After about 15 keypresses your keyboard
is dead and there is no code to reset it.
If you can - skip having your own keyboard routines, since
they mostly fail anyway.