

HowToCode7

COLLABORATORS

	<i>TITLE :</i> HowToCode7		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 6, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	HowToCode7	1
1.1	Introduction to HowToCode	1

Chapter 1

HowToCode7

1.1 Introduction to HowToCode

This file has grown somewhat from the file uploaded over Christmas 1992. I've been very busy over the last two months, so I'm sorry that I haven't been able to update this sooner. It started as an angry protest after several new demos I downloaded refused to work on my 3000, and has ended up as a sort of general how-to-code type article, with particular emphasis on the Amiga 1200.

Now, as many of you may know, Commodore have not released hardware information on the AGA chipset, indeed they have said they will not (as the registers will change in the future). Demo coders may not be too concerned about what is coming in a year or two, but IF YOU ARE WRITING COMMERCIAL SOFTWARE you must be.

Chris Green, from Commodore US, asked me to mention the following:

"I'd like it if you acknowledged early in your text that it IS possible to do quite exciting demos without poking any hardware registers, and that this can be as interesting as direct hardware access. amiga.physik.unizh.ch has two AGA demos with source code by me, AABoing and TMapdemo. These probably seem pretty lame by normal demo standards as I didn't have time to do any nifty artwork or sound, and each only does one thing. but they do show the POTENTIAL for OS friendly demos."

I have seen these demos and they are very neat. Currently you cannot do serious copper tricks with the OS (or can you Chris? I'd love to see some examples if you can...), for example smooth gradated background copperlists or all that fun messing with bitplane pointers and modulus. But for a lot of things the Kickstart 3.0 graphics.library is very capable. If you are in desperate need for some hardware trick that the OS can't handle, let Chris know about it, you never know what might make it into the next OS version!

Chris mentions QBlit and QBSBlit, interrupt driven blitter access. These are things that will make games in particular far easier to write under the OS now.

Chris also says "Note that if I did a 256 color lores screen using this document, it would run fifty times slower than one created using the OS, as

you haven't figured out enhanced fetch modes yet. A Hires 256 color screen wouldn't even work."

There are some new additions to the AGA chapter that discuss some of this problem, but if you want maximum performance from an AGA system, use the OS.

Remember that on the A1200 chipram has wait-states, while the 32-bit ROM doesn't. So use the ROM routines, some of them run faster than anything you could possibly write (on a A1200 with just 2Mb ram).

The only drawback is again documentation. To learn how to code V39 OS programs you need the V39 includes and autodocs, which I'm not allowed to include here.

Perhaps, in a later release, I'll give some highlights of V39 programming... Get Chris Green's example code, it's a good place to start.

Register as a developer with your local Commodore office to get the autodocs and includes, it's relatively inexpensive (£85 per year in the UK). You can now buy the includes/autodocs and Rom Kernal manuals in AmigaGuide from Commodore US on a CD-ROM (developers only), the CATS CD Edition 2. It's excellent!

Most demos I've seen use similar startup code to that I was using back in 1988. Hey guys, wake up! The Amiga has changed quite a bit since then.

Thanks to everyone who has replied. Any more questions, queries, or "CJ/CH, you got it wrong again!" send your email to the following address:

comradej@althera.demon.co.uk
