PC DEMOS FAQ

Version 2.02

The official FAQ of the Usenet Newsgroup comp.sys.ibm.pc.demos

This FAQ was last modified on Thursday, 12-Mar-1998 23:15:37 CST

New Since Last Version -- PLEASE READ:

I now have my own website dedicated to old PC resources and nostalgia. The permanent home of the PC Demos FAQ is now:

http://www.oldskool.org/demos/pc/pcdemos.faq.html

Another important note: I drift in and out of the demoscene every so often, so about 15% of the information in this document is probably outdated. Also, I want to add new questions to the FAQ, but I have much less time than I did three years ago when I started this FAQ along with Houman. The end result? This document is probably not going to get updated any time soon unless someone else wants to volunteer maintaining it. If you'd like volunteer, email trixter@hornet.org.

Finally, if some of the formatting looks a bit odd in the text version, then that's because the source is HTML and the text version is a conversion of it.

Introduction:

This is a list of Frequently Asked Questions regarding PC demos. (Not game or product demos, but the "hacker" or "underground" type of demo.) The information contained in this FAQ is geared mainly toward people with IBM PCs or compatibles with Internet access, although much of the information is useful to people asking questions about demos in general. All questions are answered by the maintainer (trixter@hornet.org), although corrections and additions by other parties are encouraged.

At times, Trixter will use and/or reprint information obtained directly from individuals on the newsgroup comp.sys.ibm.pc.demos to answer questions more effectively. (This is known as "contributing to the FAQ", even if you didn't know you were doing it. :-) A list of everyone who's contributed to the FAQ in some way is available at the end of the FAQ. Authors of contributions are given where specified; if you don't see your name listed in conjunction with a passage of this FAQ, please email me to get it corrected.

Other Sources For This FAQ:

If you prefer viewing this FAQ via the World Wide Web, you may do so at:

http://www.oldskool.org/demos/pc/pcdemos.faq.html

If you're interested in Amiga demos, this isn't the FAQ you should be reading. That FAQ is the alt.sys.amiga.demos FAQ for Amiga demos by Hollywood/AXIS (hollywood@kosmic.org).

If you're looking for something specific and you're not viewing the WWW version (see above for URL), use your file viewer's case-insensitive search function to find all occurances of it.

Index:

General:

How to create demos:

Demos and the Internet:

Sound Questions:

Graphics Questions:

Slang/Glossary/Terms:

Demo Groups:

Demo Parties/Compos:

Operating Systems and Platforms:

Technical/Programming Info:

Misc. Information:

FAQ information:

General:

(1.0) What is a Demo?

A Demo is a program that displays a sound, music, and light show, usually in 3D. Demos are very fun to watch, because they seemingly do things that aren't possible on the machine they were programmed on.

Essentially, demos "show off". They do so in usually one, two, or all three of three following methods:

- They show off the computer's hardware abilities (3D objects, multi-channel sound, etc.)
- They show off the creative abilities of the demo group (artists, musicians)
- They show off the programmer's abilities (fast 3D shaded polygons, complex motion, etc.)

Demos are an art form. They blend mathematics, programming skill, and creativity into something incredible to watch and listen to.

(1.0.1) Where did/do demos come from?

Demos started as loaders for cracked games (a loader is a small program that was used to identify who had cracked the game you were currently playing.) This gradually expanded into being programmed just for fun, or as a way for the programmer to show off.

You can find more expanded information on the history of demos at:

http://www.cdrom.com/pub/demos/hornet/html/demo_history.html

Also See FAQ 9.3.

(1.1) How/Where do I get a demo?

Many demos can be found on local BBSes, but if you have Internet access, you can easily get them off Internet sites. You need to have ftp access to a couple of ftp sites. Here are some listings:

- ftp://ftp.hornet.org/pub/demos: A huge site in the U.S; home base of the Hornet demogroup's demo archive, the largest in the world. Carries diverse programs related to music, graphics, magazines, programming info/source code, and, of course, demos. If you have access to a web browser, do yourself a favor and visit Hornet's official web interface to this archive, at http://www.hornet.org/pub/demos. Snowman has spent months creating a custom search engine to the archive; you can search for demo names, group names, ratings, parties, and more.

Some other sites are: (some of these are mirrors of ftp.hornet.org)

- ftp://ftp.arosnet.se
- ftp://ftp.luth.se
- ftp://ftp.sun.ac.za
- ftp://ftp.telepac.pt
- ftp://ftp.uni-paderborn.de
- ftp://ftp.uwp.edu
- ftp://ftp2.mpoli.fi

- ftp://hagar.arts.kuleuven.ac.be
- ftp://wuarchive.wustl.edu

Demos are usually in a subdirectory similar to /demo, /demos, /pub/demos or /pub/msdos/demos.

If you don't have access to an FTP site directly, but you do have email, you can use FTP sites via email. See http://www.instantweb.com/~tech/topics/ftpmail.html for more information.

(1.2) What are the best Demos?

Since demos are an art form, no single person is qualified to say which ones are the best. Since here are plenty available, you can choose. Here's a list of some of the most praised demos overall (the "classics", if you will), and are highly recommended to beginners to the demo scene:

Name	Demogroup	Sound Blaster?	Gravis Ultrasound?
Unreal	Future Crew	Yes	Yes, in v. 1.1
Panic	Future Crew	Yes	No
Second Reality	Future Crew	Yes	Yes
Crystal Dream I	Triton	Yes	No
Crystal Dream II	Triton	Yes	Yes
Show	Majic 12	No	No
Verses	EMF	Yes	Yes
Dope	Complex	No	Yes
Stars	Nooon	Yes	Yes
Caero	Plant & EMF	Yes	Yes
Inside	CNCD	Yes	Yes
Contrast	Oxygene	Yes	Yes

These are considered the "classics", and should be viewed to get a wide understanding of what goes into a demo. There are thousands more out there for you to view, of course.

These demos are available in the ftp sites listed in the above FAQ 1.1, "How/Where do I get a demo?". You can also grab them directly off of PC Demos Explained (see FAQ 3.2) on the World Wide Web.

For continuing reports on what people think is a good or bad demo, you can monitor what people say on the Internet (see FAQs 3.0-3.3). The semi-weekly newsletter DemoNews (see FAQ 3.3) also rates demos on a five-star system.

Another way to find good demos is to check the compo (competition) results from various Parties. The largest party is called The Party, held each year around Christmas. Another big party is Assembly, held in the summer and reputed to show the best demos. Here's the top three from each (as of November, 1996):

The Party

TP93 PCDemo

1. Untitled by Dust
2. The Good, The Bad, and The Ugly by S!P
3. Cardiac by Infiny

TP93 PCIntro

1. Cyboman by Gazebo
2. Symbology by Admire
3. Blackzone by Masque

TP94 PCDemo

1. Project Angel by Impact Studios
2. No by Nooon
3. Contagion by The Coexistence

TP94 PCIntro

1. Cyboman 2 by Complex
2. Peripheral Vision by Valhalla
3. Finkel by Jamm

TP95 PCDemo

1. Caero by EMF & Plant

TP95 PCIntro

1. Lasse Reinbong by Cubic Team

2. Dream by Jamm	2. Illumination by Yodel
3. Reanimator by Rage	3. Sea Robot of Love by Orange

Assembly

ASM93 PCDemo	ASM93 PCIntro
1. Second Reality by Future Crew	1. Eclipse by EMF
2. Optic Nerve by Silents	2. Tangle by Epical
3. Elements by Xography	3. Debut by Darkzone
ASM94 PCDemo	ASM94 PCIntro
1. Verses by EMF	1. AirFrame by Prime
2. Holistic by Cascada	2. Space Jam by Fascination
3. Heartquake by Iguana	3. Fyvush by Jamm
ASM95 PCDemo	ASM95 PCIntro
1. Stars by Nooon	1. Drift by Wild Light
2. Juice by Psychic Link	2. Stickman's World by Coma
3. DX Project by RealTech	3. Bill G Force by Complex
ASM96 PCDemo	ASM96 PCIntro
1. Machines of Madness by Dubius	1. Blind by Eufrosyne
2. Toasted by Cubic Team	2. We Go by Complex
3. Vivid Experiment by Doomsday	3. Nation Zero 2 by Jamm

(1.3) I can't get this Demo to run! What can I do?

First of all, the problem may be your operating system or environment. You can expect to have some problems if you are running:

* Windows 3.1
* Windows 95
* OS/2

In the case of Windows 3.1, exit windows via Program Manager's File|Exit menu; if you're running Windows 95, shut down to "MS-DOS mode"; if you're running OS/2, try the demo in a full-screen session with "Vertical Retrace Emulation" turned OFF.

If that doesn't work and you're not running actual DOS, then you must reboot your machine and boot DOS 3.3 or later. For instance, you cannot expect demos to run under:

* Windows NT
* SoftPC (Macintosh MS-DOS emulator)

This is because demos sometimes rely on hardware tricks to achieve their effects, and these operating systems do not allow direct access to the hardware. In fact, these operating systems must emulate DOS, so the demos, if they did run, would run twice as slow anyway. If you boot actual DOS and it still won't run, you might be hitting the old Protected Mode issue: Many complicated demos (high-speed specialized graphics, for instance) usually work best when they take over the entire machines' hardware. This can complicate running the demo on different machines ("It worked on my friend's machine, why not mine?" and so on). Furthermore, since a couple of years ago, demo programmers started to use more and more of their own protected mode programming routines instead of using the ones already running on the system. So usually, if you have QEMM386, 386MAX, or EMM386 installed (protected mode memory managers), those demos would not run. (This does not mean all demos won't run on today's memory managers--all Future Crew demos, for instance, run just fine.)

So, the solution is to boot your machine without a protected-mode memory manager. Personally, we suggest you do a clean boot by hitting F5 when you see the message "Starting MS-DOS" (or F8 if you're running Windows 95), or to make a multiple config menu system that includes a "bare-bones" config.sys and autoexec.bat configuration.

Here's what a "bare-bones" configuration looks like:

Your CONFIG.SYS should look like:

```
     DEVICE=C:\DOS\HIMEM.SYS
     REM Some programs require HIMEM in conjunction with their custom
     REM protected mode memory managers
```

Your AUTOEXEC.BAT should look like:

```
     SET BLASTER=A220 I5 D1 H5 T6
     REM if you have a Sound Blaster sound card; make sure you use the
     REM right settings, the above is an example

     SET ULTRASND=240,7,6,7,5
     REM if you have a Gravis Ultrasound card; make sure you use the
     REM right settings, the above is an example

     C:\ULTRASND\ULTRINIT
     REM if you have a Gravis Ultrasound card
```

...and any other sound card initialization programs. (These will depend on your sound card, of course.)

That's it--nothing else. Try running the demo; with nothing in its way, it should work.

If you absolutely must have some other programs in your CONFIG.SYS or AUTOEXEC.BAT in order for your computer to run, then try having them take up as little memory as possible, if they provide the option.

Note: Some programs need EMS to run. For that, you need a memory manager. To do this, add this line after the "HIMEM.SYS" line in the above CONFIG.SYS example:

```
     DEVICE=C:\DOS\EMM386.EXE RAM
```

Also, a demo might complain about not having enough free DOS memory. To fix that, add this line after the "HIMEM.SYS" line in the above CONFIG.SYS example:

```
     DOS=HIGH
```

(1.3.1) What's the best PC system to run a demo?

Ironically, a slower system used to be better for appreciating demos, because you can really see how good the code is. One of the great things about the early demo scene was how 3D shaded objects were displayed at 60 or 70 frames per second--on a 16 MHz machine. You just couldn't ignore how excellent the code was.

Nowadays, the best machine for appreciating demos is probably a Pentium 90. It's slow enough so that you can appreciate the coding, but fast enough so that the newer generation of demos won't be laboriously slow.

If compatibility is an issue, then get a video board based on the Tseng Labs ET4000, Cirrus Logic or S3 chipset. FAQ 5.0 has information on the fastest video boards.

If you want to appreciate the sound, one of the Gravis Ultrasound family of cards (see FAQ 4.0) is required, as over 90% of all demos support it, and many only support it. (As to why many only support it, see FAQ's 4.1 to 4.3.)

When it comes to RAM, the more the better. :-) But 8 MB should probably be your minimum.

How to Create Demos:

(2.0) What Compilers do I use?

In order to make a demo, you need to know how to program. (Self-explanatory, I guess, but we still needed to address it. :-) You need to know a common language that lends itself to programming DOS programs, like Assembler, Pascal, or C/C++.

Some assemblers are:

* Turbo Assembler 3.1 or above ("TASM")
* Microsoft Assembler ("MASM")

Some C/C++ compilers are:

* Watcom C++ 10.x, 11.x
* Borland C++ 3.1, 4.0, 4.5
* Visual C++
* Turbo C/C++
* GNU C for DOS (this is a FREE 32-bit C compiler! You can get it at ftp://ftp.idsoftware.com/djgpp/v2 or http://www.rt66.com/~brennan/djgpp/

Some pascal compilers are:

* Turbo Pascal 6.0 or 7.0
* Borland Pascal 7.0

(2.1) Are there any programming references I can read?

For demos specifically? Only one, but if you get at least one book, get Michael Abrash's Zen of Graphics Programming, 2nd Edition, published by The Coriolis Group (ISBN 1-883577-89-6). It's about $44.99 in the USA and comes with Zen of Assembler in electronic form on the included CDROM. It covers everything from low-level VGA hardware programming, to fast animation techniques, to 3-D algorithms, BSP trees, and Quake. As many democoders would agree, this is this shit. :-) We all wish we had access to this book six years ago, but of course back then, all of us were discovering these techniques at the same time.

Here's some more suggestions:

ASSEMBLER:

* Assembly Language for the IBM-PC 2nd Edition, by KIP R. IRVINE
* Mastering Turbo Assembler by Tom Swan (Hayden Books, ISBN 0-672-48435-8).
* PC System programming, Abacus 1990, by Michael Tischer
* Zen of Code Optimization, by Michael Abrash

C/C++:

- Programming in C, Third Edition Kelley/POHL

VGA:

- EGA / VGA a programmer's reference guide, 2nd edition, Bradley Dyck Kliewer
- Programmers guide to EGA and VGA cards, 3rd Edition, Addison Wesley, by Richard Ferraro

Graphics:

- Computer Graphics: Principles and practice, 2nd edition, Foley, Van dam, Feiner, Hughes
- The "Graphics Gems" series, currently volumes 1 through 5 (first one edited by Glassner). These books are full of useful tricks and algorithms, for all types of graphics related stuff from 3D to image processing

Demo Programming:

- PC INTERDIT (French version), PC Underground (English version) by Boris Bertelsons and Mathias Rasch from Micro Application/Abacus/Data Becker

Magazines:

- Dr. Dobb's Journal
- Game Developer
- PC Techniques
- You can also consult The Programmers' Booklist, which is a list of books, magazines, emags, digital sources, WWW pages, etc. of programming resources. This can be found at http://intranet.ca/~sshah/booklist.html and ftp://ftp.intranet.ca/usr/synapsis/progbook.lst

---

(2.2) Where can I find some example Source Code?

You can find plenty of source code for graphic effects, sound routines, and other examples in the following ftp sites:

- ftp://ftp.hornet.org/pub/demos/code -- Included are subdirectories which are categorized depending on the nature of the effects. Most of the interesting stuff is under graph and demosrc.
- ftp://x2ftp.oulu.fi/pub/msdos/programming -- Same as above, but different subdirectories, of course. Also a fantastic wealth of programming information.
- ftp://ftp.co.iup.edu/code -- This is a mirror of the Hornet Archive's /code directory.

As for CDROM's, the Nightowl CDROM series, Software Vault CD's & Emerald CD's have been suggested as a source of source code.

(2.2.1) I'm no idiot--Where can I find some REAL programming info?!

Okay, okay--you don't have to yell. :-) The stuff listed above is fairly complete, but some readers of this FAQ have suggested some other things to look at if you didn't know about them:

- Hardware descriptions of different hardware, like Intel processor optimizations, are at http://www.intel.com/IAL/processr/ap500ovr.html

- "Real-time" methods of Phong shading can be found at ftp://ftp.hornet.org/pub/demos/code/graph/phong
- Graphics algorithms can be found in the usenet newsgroup comp.graphics.algorithms and in their FAQ.
- The PC Games Programmers Encyclopedia has lots of juicy tricks, and can be found at ftp://x2ftp.oulu.fi/pub/msdos/programming/gpe/pcqpe10.zip
- Mode X programming information and libraries can be found at ftp://x2ftp.oulu.fi/pub/msdos/programming/xlib
- The PC Games Programing Encyclopedia is a good collection of documents on VGA programming, as well as other issues of game and demo programming. There are two versions available:
  - DOS executable version: ftp://x2ftp.oulu.fi/pub/msdos/programming/gpe/pcgpe10.zip
  - Windows Help format version: ftp://x2ftp.oulu.fi/pub/msdos/programming/gpe/wpcgpe10.zip

There are also many other technical documents on the Internet, including:

- DOSREF34.ZIP : Programmer's technical reference DOS by Dave Williams
- INTDOS44.ZIP : MSDOS Interrupt List by Ralph Brown
- HELPPC21.ZIP : Help PC by David Jurgens

(2.3) Where can I find some painting programs?

Most demo coders use DeluxePaint][e by Electronic Arts. This program is out of print for the IBM, but you can find it in some bargain bins or you can purchase it directly from Electronic Arts. It is also rumored that if you have the original Rebel Assult CDROM (a game made by LucasArts), there is a hidden directory with some development tools used for the game, and DeluxePaint is among them. :-) If you own an Amiga, you can easily purchase Deluxe Paint IV for the Amiga and use that (although many argue that Brilliance is a better product).

Another set of DOS users argue that Animator PRO 1.3a by Autodesk is the best bitmap illustration program to use. For photographic bitmap editing, Adobe Photoshop is easily one of the best programs; for "natural bitmap" creation, Fractal Design Painter is the best.

There is a freeware program called Satan Paint, which is by the same guys as Abuse (cool shareware game) - you might find it at ftp://ftp.hornet.org/pub/games/abuse.

There are some good shareware painting programs as well, such as Paint Shop Pro 3.11 for Windows, and Neopaint for DOS. You can find them at:

- Paint Shop Pro: ftp://ftp.cica.indiana.edu/pub/pc/win3/desktop/psp30.zip
- Neopaint: ftp://x2ftp.oulu.fi

GD / Hornet also added: "There's a new MSDOS paint program on hornet called "Grafx 2"... I have heard people talking about it on #trax. If you want to check it out, I believe the filename to search for is "gfx2b".. then the rest of the filename is the exact version number."

(2.4) Where are some music composition programs? Which one should I use?

For music, most demo groups use "music modules", a format that originated on the Amiga. MODS vary greatly in formats on the PC, however; the standard formats right now are S3M, IT, and XMs; some older formats still being used are MOD and MTM.

In order to create a module, you need a program called a "Tracker". The best ones so far are:

Name	Loads:	Saves:
Fast Tracker II v.2.06 by Triton	MOD, XM, S3M	XM
Scream Tracker v3.21 by Future Crew	MOD, S3M, STM	MOD,S3M
MultiTracker v1.01 by Renaissance	669, MOD, MTM	MTM
Impulse Tracker v2.10 by Pulse	MOD, S3M, IT, MTM, XM	S3M, IT
Velvet Studio by Velvet	MOD, S3M, AMS, XM	AMS, XM

You can find these trackers in ftp://ftp.hornet.org/pub/demos/music/programs/trackers

Notes:

- FastTracker, MultiTracker, Impulse Tracker, and Scream Tracker are fully functional trackers, with the following exceptions:
  - FastTracker is shareware
  - The author of IT will send you a disk writer device driver if you contribute some money to the Impulse Tracker project.
- Velvet Studio is crippleware; when you purchase it, you get full functionality.
- IT is the only tracker as of 1/1/1997 that natively supports the extra quality of Interwave cards, like the GUS PnP.

Now, as to which one you should use, that depends completely on why you're composing music in the first place. If you're just composing for fun, then use any tracker you feel comfotable with. If you want the most flexibility, you should probably use either Impulse Tracker or FastTracker II, since they have a ton of options; FT2 also has a built-in sampler and sample editor, just like the good old Amiga days. If you're composing music for a demo, however, then always make sure you're using a tracker that your coder can support. You're always safe with Scream Tracker 3, since the .s3m file format is the 2nd-most supported format for players (MOD being the first). Also, ST3 has an Amiga MOD editing mode, so you can compose proper MOD format songs in that tracker as well.

For more info on the different types of PC MOD formats, check out PC Demos Explained--specifically, at this URL:

http://www.cdrom.com/pub/demos/hornet/html/demo_reference.html

---

(2.4.1) How can I play music modules from my own code?

There are several libraries for playing music on several demo sites. One place to get them is ftp://ftp.hornet.org/pub/demos/code/sound; in there, you'll find many libraries, including:

Name	Filename	Formats	Language
MIDAS v0.40	mdss*.zip	Most formats, except XM	Assembler, Pascal, C
MIDAS v0.5x	mdss*.zip	Most formats	C
Mikmod v2.xx	mik*.arj	Most formats	C
Bells, Whistles, and Sound Boards v.1.2x	bwsb*.zip	Most formats, except XM	Assembler, Pascal, C, BASIC
CapaMOD v3.xx	cmod3*.zip	MOD, S3M, XM	Assembler, Pascal, C
FMODDOC2 by FireLight	fmoddoc2.zip	Most formats	C

Any of the above libraries are more than enough to play modules on the GUS, and some of them support many other sound cards (such as Sound Blaster/Pro/16, Pro Audio Spectrum/+/16, etc.). FMODDOC2 contains fantastic and complete information on file structures, period values, information, etc. It's highly recommended if you want to write your own player.

(Note that these libraries are for playing MODs or MOD-related formats, and do not play MIDI music. For that, you could try the widely available Miles Drivers; or, if you only have a GUS, you could use UltraMID or the GUS SDK. See the GUS FAQ for more info.)

Demos and the Internet:

(3.0) Is there a place on the Internet I can learn more about demos?

There are several, actually:

- For files or information: ftp://ftp.hornet.org/pub/demos
- For discussions or news, you have two choices, Usenet and IRC:
    - IRC: #coders (programming talk) or #trax (music talk)
    - Usenet: comp.sys.ibm.pc.demos
- For General Info on the WWW, you can always try: http://www.cdrom.com/pub/demos/hornet/html/demos.html

(Note: Although this newsgroup has "demos" on the end of it, it does not talk about subjects such as game demos. For those, look in the newsgroup series comp.sys.ibmpc.games.*)

comp.sys.ibm.pc.demos is a nice place to ask questions about demos in general, so if you have any general questions concerning demos, ask here first. As such, we'll talk about comp.sys.ibm.pc.demos first, then IRC and the WWW later in this document.

(3.0.1) What can I talk about on comp.sys.ibm.pc.demos?

People post various things. Some concerning Demos, others about technical questions, such as:

- "How do I code Gouraud Shading?"
- "How do I code Plasma/Copper bars, etc.?"
- "Does anyone know how to contact xxx?"
- "What did you think of demo xxx?"
- "Where is the next party being held in Germany?"

Other misc. postings are about new programs coming out. For example, if you coded a demo or made some of your source code public, let us know about it here. Essentially, if it's demo-related, post. :-)

(3.0.2) What can I *NOT* talk about on comp.sys.ibm.pc.demos?

Some posts come up so frequently that many of the members of the newsgroup are getting tired of hearing them. Here's a list of subjects to avoid:

Subject	Why you should avoid posting this
"GUS is better than SB!"	These arguments should be posted on a sound card group like comp.sys.ibm.pc.soundcards.advocacy, not here.
"SB is better than	

GUS!"	
"What's the fastest PutPixel?"	This is asked way too many times. (It is covered in this FAQ as question 10.0, BTW)
"mY dEm0 t0TalLy rUleZ!"	"Elite" postings are usually the mark of a "lamer", and are ignored.
"PC SUCKS!"	Usually, these are posted by people who have left their terminal unattended :-), or by Amiga/Atari advocates.
"mydemo.zip, part 1 of 23"	Do not post binaries in comp.sys.ibm.pc.demos! Post them in alt.binaries.demo-scene.ibm-pc instead.

These kinds of posts usually bring lots of replies--and they're not friendly ones!

Also, comp.sys.ibm.pc.demos is NOT a binary newsgroup! Many people in third-world or poor countries get this newsgroup the only way they can--delivered directly to their home computer. A binary file in comp.sys.ibm.pc.demos can easily cost them a lot of money! So, please do not post any binaries. If you want to post a binary, please do so in alt.binaries.demo-scene.ibm-pc instead. If you want to upload a binary file where many other demo people can get at it, look in the above section about ftp sites--you can upload there as well, usually in /incoming directories, like ftp://ftp.hornet.org/pub/demos/incoming/demos

----

(3.0.3) I'm new at this... How can I post a message without sounding like a total fool?

- Be yourself.
- Use common sense; sometimes a reply to the original sender would be much better than posting it into comp.sys.ibm.pc.demos if there's already a ton of replies.
- Flaming (sending mean messages to) other people will get you nowhere.
- Avoid long signature files.

Another good practice to follow when posting is to edit the original message and answer appropriately; i.e. don't just reply back with all the text of the previous senders already shown. It is usually a hassle to go through 5 pages of old replies, and then 1 sentence of answer.

(3.0.4) Can I erase my old postings?

ONLY IF YOUR NEWSREADER PROVIDES THE OPTION TO DELETE POSTS. Otherwise, you can't. Once you've posted something, it is posted to every newsgroup it was bound for, and some people may see it before you cancel it. So, think before posting a message. If you post something that you realize was a mistake, cancel it (if you can) and post another one, right away, and write in it what you really meant. That usually stops people answering another 50 messages to the (incorrect) original.

(3.1) What's IRC? Are there any IRC channels established for demos?

IRC stands for Internet Relay chat. You can find more info about it in newsgroups such as alt.irc, etc. With IRC, you can enter "channels" (like a CB Radio) and discuss things with people located all around the world in real time. It is a good place to stop by and ask questions.

As of this writing, the main two channels concerning demos are:

- #coders - Concerning PC Demo coders
- #trax - Concerning PC Music Makers

Note: #trax moved to Anothernet, so if you join #trax and nobody is there, type "/server neato.org" or the name of another Anothernet server, like irc.phonet.com, and then try again. AnotherNet has a home page that you can check out if you want more info, at http://www.another.net. Also, #trax has it's own home page, at http://www.spaz.com/trax

There are other channels dedicated to demos and democoding:

- #daskmig - Lots of chat about demos
- #nlcoders - The Dutch and Belgian scene
- #demofr - French chats about demos
- #codersnl - Dutch chats about democoding
- #amigascne - Amiga demoscene chat
- #suomiscene - Finnish demoscene chat

The above channels are on the Euro-Efnet. A good USA server for the EuroEfnet is irc.stealth.net

Also, lots of groups have their own channel while they're online, like #massive and #natives. Finally, some parties have their own channel during the party, like #asm95, #naid, etc. Feel free to join at any time... it is fun, and fairly addicting!

For those of you who have been having problems connecting to AnotherNet (the IRC network that #trax has moved to and #coders is trying to move to) recently, I finally managed to get my hands on the little script that fixes the problem.

Here's what to do:

1. Start your IRC client (this script is written for ircII).
2. Load the script.
3. Connect to the server.

Here's the script:

```
# Script to fix irc to work with AnotherNet "initping".
on ^raw_irc "PING :%" {quote PONG $1-}
on ^raw_irc "% PING :%" {quote PONG $2-}
```

Just take those 3 lines, and save them as "initping.irc" ... and add a line to your .ircrc file. Should work perfectly then.

(3.2) What's the World Wide Web (WWW)? Are there any places to learn about demos on the WWW?

Created by university professors in CERN (Switzerland), WWW is a network of hypertext documents that can connect to other hypertext documents. Accessible by WWW "browsers", such as Mosaic and Netscape, it allows you to connect to WWW pages, where you can read articles, look at pictures, download files, etc. In short, if you haven't tried the WWW yet, you're missing a whole new aspect of the Internet.

If you're looking for a browser, try one of these three:

- Mosaic, the original WWW browser, can be found in ftp://ftp.ncsa.uiuc.edu and exists for Windows, Mac and Unix.
- Netscape, a very nice browser that supports additional layout commands, is available in ftp://ftp.netscape.com

   * Lynx is a text-mode browser that you can use on any terminal.

Many demo groups in the demo scene have their own WWW home pages. It would be too long here to list all of them, however Trixter has a homepage which explains about PC demos and lists many demo resources. You can access this page at:

     http://www.cdrom.com/pub/demos/hornet/html/demos.html

(3.3) Are there any newsletters about demos?

Other than the irregularly-released diskmags (see FAQ 6.0), there are two semi-weekly newsletters produced for the demo scene: DemoNews and TraxWeekly.

DemoNews is both a list of new files uploaded to ftp.hornet.org (Hornet's home base and the largest demo site on the Internet), news, interviews, and articles that pertain to the demo scene. DemoNews was started by Dan Wright (Pallbearer / Toxic Zombies & Hornet) and is continued by the Hornet demogroup.

TraxWeekly is similar to DemoNews, but is dedicated to the music scene. It's called "TraxWeekly" because it stemmed from the IRC channel #trax (see FAQ 3.1).

Here's how you can subscribe to either magazine (the following is an excerpt from TraxWeekly):

```
 _____How to subscribe to TraxWeekly

 TraxWeekly subscriptions are free, and can be requested in this
 matter:

 Send mail to:                listserver@unseen.aztec.co.za
 And put in the message body: subscribe trax-weekly [your real name]

 If you want to unsubscribe to the list, mail the same address and write:

                              unsubscribe trax-weekly

 TraxWeekly is also available on ftp.hornet.org:

 /pub/demos/incoming/info/  for the most recently uploaded version, or
 /pub/demos/info/traxw/     for all of the back issues.
```

DemoNews is published approximately once a week, and can be emailed directly to you. According to each issue of DemoNews, here's how you can subscribe:

```
 _____How to subscribe to DemoNews

 Mail to : listserver@unseen.aztec.co.za
 Body    : subscribe demuan-list [first_name] [last_name]

 The listserver will send DemoNews to your e-mail's return address.

 _____Back Issues

 Older issues of DemoNews can be located under /pub/demos/info/demonews.
 Newly released issues of DemoNews are posted to /pub/demos/incoming/info.
```

Finally, someone pointed out that I didn't mention the RAW mag on-line. This is because I don't have information on it; I expect you can search the World

Wide Web for it.

(3.4) How can I contact demo people?

The easiest way to contact a demo person is to email them; it costs the both of you little money, and avoids huge time differences. If you can't find a person's email address, then you can try finding them on IRC (see FAQ 3.1), posting a message looking for them on comp.sys.ibm.pc.demos (see FAQ 3.0), or checking the "demo contact list" that is maintained by SCouT/SuccesS, at http://utopia.knoware.nl/users/rolando. Snowman also maintains a list at http://www.hornet.org/ha/pages/email.cgml. There was an older list that was maintained by rob@span.com, although it is at least 18 months out of date.

(3.5) Are there any Demo BBS's on the Internet?

There used to be a great one called Digital Horizons at dragon.axs.net that has since gone down. The only other two I know is BlueWater at
http://www.Bluewater.ch and LooK BBS at
http://www.geocities.com/siliconvalley/Heights/8481/.

Sound Questions:

(4.0) What are the different types of GUS sound cards?

There are four currently available, and their major differences are mostly in their recording abilities:

Card	Record	Playback
Gravis Ultrasound	44.1KHz, 8-bit stereo	44.1KHz, 16-bit stereo
Gravis Ultrasound MAX	48KHz, 16-bit stereo	48KHz, 16-bit stereo
Gravis Ultrasound ACE	(cannot record)	44.1KHz, 16-bit stereo
Gravis Ultrasound PnP	48KHz, 16-bit stereo	48KHz, 16-bit stereo

The Gravis Ultrasound is not manufactured anymore, but you can still order it from many places for cheap, like around $50 or so. The MAX retails for about $179 US, and the ACE retails for about $99 US. Street (actual) prices are usually 20% to 40% lower than retail prices in the US.

The Gravis PnP is a Plug-and-Play card mainly designed for Plug-and-Play operating systems, like Windows 95, but it will work with non Plug-and-Play operating systems as well. For programs that don't support it directly, it acts just like a normal GUS if RAM is added to it. The PnP has no RAM on it (but has the expansion capability), and the PnP Pro has 512K RAM. Also, unlike the previous GUS cards, the PnP is based on the Interwave sound chip from AMD, and has General MIDI capability built into the card. With RAM added, you can also load General MIDI instruments off of disk, just like the old GUS cards. The PnP also overcomes some of the previous GUS limitations, like 1MB of RAM (you can have up to 8 MB) and the quality drop-off after 14 channels (there is no drop-off at all if used in native Interwave mode). Finally, the GUS PnP is more Sound Blaster-compatible than previous GUS cards.

The PnP retails for $179.

(4.1) Why is the Gravis Ultrasound (GUS) sound card supported more than the Sound Blaster in demos?

Several reasons:

- The GUS sounds much better than the Sound Blaster because it can play back 32 channels of stereo digitized sound at once, while the Sound Blaster Pro/16 series can only play back 2 channels.
- Because the GUS can play back multiple channels, there is no need for mixing several channels into one (like playing MODs on the Sound Blaster series), so much less CPU time is utilized for playing music. This means more CPU time is available for graphical effects, calculation, etc.
- Some demo coders consider the GUS easier to program.
- The software development kit for the GUS was available for free as early as 1994 (and Renaissance's UltraDox was available before that) and contained lots of helpful source code. The Sound Blaster SDKs, by comparison, have either cost money, or provided binary libraries without any useful source code.
- Many demo scene members have gotten free GUS cards in exchange for programming demos/games/utilities that specifically use the GUS.

(4.2) Why is the Gravis Ultrasound sound card supported more than General MIDI?

While General MIDI has much higher sound quality instruments, the number of instruments (and the instruments themselves) are fixed in nature; that is, you cannot change any of them. (If you don't like the sound of a particular piano instrument, for example, you're stuck with it.) MODs, and MOD-like formats (see FAQ 2.4) allow the composer to use whatever instruments he feels like.

Also, General MIDI boards differ greatly in price and sound/instrument quality, and a song on one GMIDI board might not sound the same on another.

Finally, General MIDI boards usually cost over $200 for a good quality one--which is usually unattainable, given the budget of most demo scene members.

(4.3) Has MIDI been used in any demos?

Believe it or not, it has. Superunknown by Five And Then Some (TG '94) used MIDI files and custom GUS patches for their music (the demo requires a GUS to hear the music properly, however; the demo will not play the music on a GMIDI board). And The Phony Coders only supported the Roland MT-32 in their demos. As for FM synthesis, it's used in a lot of places, but it's neither MIDI, nor composed in a MIDI sequencer; there are custom FM trackers for that.

(4.4) Why is the Gravis Ultrasound sound card supported more than the Sound Blaster AWE32?

Even though the SB Awe32 is very similar (and in some ways, superior) to the GUS, it is not being accepted well by the demo community, mainly because for a long time low-level information about the card was not available for free. Also, up until recently, the SB AWE32 cost almost 50% more than the GUS (this has changed, however). But since the GUS was around over 3 years more than the SB AWE32, the GUS has much more "market saturation" and existing free source code. (Although, to be fair, Renaissance originally disabled the drivers to find out what was going on, and Gravis took the hint and released a development kit.)

Some coders have reverse-engineered the drivers and come up with C source code than can not only drive the SB AWE32, but play MODs and XMs on it, even with echo, chorus, and reverb effects. To find this info, visit ftp://ftp.hornet.org/pub/demos/code/sound. There are also some trackers that support the AWE32: The Ultimate Tracker and RamTrack. You can find these from ftp://ftp.lysator.liu.se and other sites.

(4.5) Can I emulate the GUS with my Sound Blaster?

No. The GUS performs so many functions over the Sound Blaster that writing an emulator would be extremely difficult. Couple this information with the fact that many demos take over the hardware entirely, and you realize it makes writing an emulator impossible.

Well, that's not entirely true. It IS possible with DPMI and IO permission bitmaps. It shouldn't take up more than 20% CPU time on a Pentium [for 32-channels at 44kHz], but most people would consider this unacceptable.

Of course, a hardware emulator for the GUS exists. :-) It's called the GUS ACE (see FAQ 4.0), and was designed to co-exist with your existing sound card. They're only about $80 in the US.

(4.6) Why does my Sound Blaster 16 not output stereo with demos?

If the demo supports the SB16 directly, you usually get stereo and have no problems at all. The lack of stereo usually happens when the demo supports the SBPro only, and it's not the demo's fault, it's Creative Lab's (the manufacturer of the Sound Blaster series). The SBPro had two DAC's for each channel, with a bit on the on-board filter controlling if the output of those DACs was mixed into stereo or mono. The SB16's mixer does not acknowledge this stereo/mono filter bit--it just doesn't exist on its circuitry. So you get mono sound.

This phenominon is usually only present with older demos that support SB/SBPro only.

Graphics Questions:

(5.0) Which video card is best for viewing demos?

The nature of demo programmers dictates that they program the coolest stuff with the crappiest hardware. In other words, most (if not all) demos require a register-compatible VGA card. For the best experience, you need as fast a video card as possible, because most video cards are the main bottleneck when displaying fast graphics (the PC's memory is much faster than the video card's memory). Here's a general guideline:

* If your PC has a VESA Local bus (VLB) or PCI bus (PCI), then make sure you're using a VLB or PCI card, as the speed of these cards' memory is at least twice as fast as a standard ISA card.
* Try not to buy a video card unless you've tested it yourself on the store's computer--preferably with a demo or action game, which update the video card's memory over 30 times a second.
* If you can't do that, then generally, get a video card based on either the Cirrus Logic, S3, or Tseng ET4000 or ET6000 chipset. These chipsets are very fast. (Many older demos use special hardware tricks that might

only work on an ET4000; the ET4000 is considered the demo "standard".)

The Hercules Stingray uses the ARK-xxxx series chips. The Hercules Dynamite Pro/Power use the Tseng ET4000 W32p chipset. They are probably the fastest implementation of that chipset. The Stingray w/ it's ARK chipst is a tad faster under DOS and a bit more noticeably faster under windows than the Dynamite series.

If you really must have the lastest chipset then the et6000 is the one to go for. On a GrafixStar 600, it does around 100mb/s a second (!!). However it's about 7% slower in windows then the Matrox Millenium--only comes with one channel of multi-bank ram (wait for a card which has 2 channels as it means nearly 1 gig per second on the card) and the RamDAC isn't the greatest.

Finally, several people have recommended the newer revision of the Matrox Millenium, not only because of it's fast DOS performance, but because it implements the VESA 2.0 specification in hardware, which several hi-res demos are starting to support. If the card doesn't have VESA 2.0 in BIOS, it can usually be emulated by a general-purpose VESA program, like UNIVBE by Scitech Software (see FAQ 5.3).

The ATI 3D Expression+PC2TV card also has VESA 2.0 in BIOS, also with low-res modes and a true linear framebuffer. Plus, it has TV output--you can see many demos on your TV set or record them to videotape. :-) However, to do so, the board forces the screen refresh rate to 60Hz, which can screw up demos pretty badly if they're synced to a different refresh rate.

For a current list of video benchmarks, you might want to inquire on the newsgroup comp.sys.ibm.pc.hardware.video.

(5.1) What's Mode X?

Mode X (and its variant, Mode Y) is the slang term given by Michael Abrash to the two most common unchained 256-color VGA video modes used in demos. "Unchained" comes from the method of "unchaining" the video memory in order to access it all--normal mode 13 (320x200x256 colors) is "chained". Mode X is 320x240 and Mode Y is 320x200.

Unchaining the video memory has a slight disadvantage: The memory organization is different, and is harder to program for (it is organized in "planes of bytes", which is harder to work with than the standard linear format of normal mode 13). However, the advantages are numerous; when you unchain video memory, you get:

* More video modes on a standard VGA, from 320x240 (square aspect ratio) to 360x480 (high res with 256 colors on any monitor) and others
* The ability to pan across all of video memory, as if it were a big virtual page
* Horizontal split screen, with the lower half not affected by the panning of the upper half
* 2 or more video pages (by "panning" to an area of video memory one screen length down)
* The ability to write up to four pixels at once with just one write (great for polygon filling)

In the "old days" of IBM PC demos, Mode X programming was the only way to get any speed out of the slow ISA bus and slow video cards of the time. Nowadays, normal mode 13 is making a comeback because of two things:

* Texture mapping, "Phong" shading, and other modern effects are faster to do in normal mode 13 than in Mode X because the polyfiller/effect/etc. rarely writes out 2 or more pixels of the same

color in a small area.
- Local-bus video cards (VLB and PCI) can move video memory much faster than before, and are becoming quite common. So, it's easier/lazier to just write to an off-screen buffer, then blit the buffer to video RAM.

Here's some background on programming Mode X:

When you unchain, you are given access to all 256k of VGA RAM. Unfortunately, only 64k is addressable by A000:0-A000:FFFF. The VGA then utilizes the plane method of color selection. Due to some weird hardware "features", instead of one bit in each plane controlling each pixel's color, one byte in one plane controls the color of a pixel. This is arranged according to the following formula:

```
     if Mode13h_offset = y*320 + x, then ModeX_offset=Mode13h_offset / 4,
     and ModeX_plane=Mode13h_offset % 4.
```

So:

```
     (0,0) would be at plane 0, offset 0
     (1,0) would be at plane 1, offset 0
     (2,0) would be at plane 2, offset 0
     (3,0) would be at plane 3, offset 0
     (4,0) would be at plane 0, offset 1
     (5,0) would be at plane 1, offset 1
     (6,0) would be at plane 2, offset 1
     (7,0) would be at plane 3, offset 1
     (8,0) would be at plane 0, offset 2
     .
     .
     .
     (0,1) would be at plane 0, offset 80
```

As mentioned in question 2.2.1, excellent Pascal and C libraries for programming ModeX are at ftp://x2ftp.oulu.fi/pub/msdos/programming under all the "xlib" directories. You can also pick up the Mode X FAQ, maintained by Zoombapup // CodeX. While it's not known if this is being maintained anymore, you can pick up a version of it at http://www.depaul.edu/~trixter/docs/modex.faq.

(5.2) Why is Mode 13 sometimes faster than Mode X?

People are sometimes told that blitting is faster than flipping pages in Mode X. That depends completely on how intelligent your drawing routines are and how fast your video card is.

For example: If your flat-shaded polyfiller is written to utilize Mode X, it can fill up to four pixels directly into video memory with a single write. But if you draw completely to an off-screen buffer, then move that buffer to display memory on a VLB or PCI video card, then yes, a simple rep movsd will be faster than copying to Mode X, because with Mode X, you have to (in a simple case) copy 16K, then write a couple of bytes to VGA to switch planes, then copy another 16K, etc. until all 64K is done.

Here's some quick (Pascal, real-mode) examples for blitting:

```
Procedure move_repmovsd(src,dst:pointer;size:word); assembler;
{for moving off-screen buffers to Mode 13 display memory for VLB or PCI}
asm
  push ds
  les di,dst
  lds si,src
  mov cx,size
  shr cx,2
  db $66; rep movsw
  pop ds
end;

Procedure copyvscreentox(sourceseg,destseg,count:word);assembler;
{blits 64K buffer from RAM to unchained video memory; thanks to Jussi L. for help
asm
  push ds
  mov ds,[sourceseg]
  mov es,[destseg]
  sub si,si
  sub di,di
  mov bl,1

@loop1:
  mov dx,3c4h
  mov ah,bl
  mov al,2
  out dx,ax
  mov cx,[count]
  shr cx,3

@loop2:
  mov al,[ds:si]
  mov ah,[ds:si+4]
  mov [es:di],ax
  add si,8
  add di,2
  dec cx
  jnz @loop2

  sub si,[count]
  inc si
  sub di,di
  shl bl,1
  cmp bl,16
  jne @loop1

  pop ds
end;
```

So you can see how much more is involved with copying Mode X screens. There is still an advantage to doing this, though: You can copy to a hidden video page, wait for verticle retrace, then simply switch pages. This ensures that the visible page updating will be flicker-free.

Like I said, it completely depends on what you're doing. If you're doing a starfield, you should use neither of the above methods, since a couple of port writes for each star is just a silly waste of cycles. For that, you'd wait for retrace, then erase all previous stars directly in video memory, then draw. Even better, you wouldn't have to wait for retrace if you were erasing each star as you plotted the new one, as the flicker generated by that is almost imperceptible.

(5.3) This demo requires VESA or UNIVBE. Where can I get it?

Take your pick: http://www.scitechsoft.com or ftp://ftp.scitechsoft.com

UNIVBE was officially renamed the Scitech Display Doctor, BTW.

Slang/Glossary/Terms:

(6.0) What's a diskmag?

A diskmag is an electronic magazine or newsletter that is distributed on a semi-regular basis. It usually has demoscene news, reviews, party reports, and general rambling. :-) Diskmags are similar to musicdisks (see FAQ 6.1) in that they are executable programs that come with their own reader, which itself usually has a nice selection of music and interface graphics.

You can find many diskmags at ftp://ftp.hornet.org/pub/demos/mags

---

(6.1) What's a musicdisk?

A musicdisk is a collection of songs put out by a single music or demo group (or, in rare cases, multiple groups--see the multi-group musicdisks Chromatiks and Epidemic, for example) or musician whose distribution medium is a single diskette. A musicdisk has a custom player built just for the musicdisk--it plays only the songs on the disk, and usually has a nice graphical interface or specific information from the composers on their songs. Musicdisks are put out whenever the group feels it has enough music to put in them, so they're released irregularly and infrequently.

You can find many musicdisks at ftp://ftp.hornet.org/pub/demos/music/disks

---

(6.2) How is a musicdisk different from a music pack?

A music pack is very similar to a musicdisk, but with some important differences. Here's a list of how music packs are different than musicdisks:

* Music packs are usually released from musicians or music groups only (groups who only do MOD/S3M-like music).
* Music packs are usually released on a regular basis, like once a month.
* Music packs have no custom player/interface (some include Future Crew's MDP player, however, which is a generic player that can have its menu customized).

The point is that, traditionally, music disks fit onto a single disk and have their own player, as per the Amiga or C64. So, ultimately, if it doesn't fit on a disk or come with it's own player, it's generally not proper to call it a musicdisk.

Many people disagree with the above viewpoint, but most of those people are pure music scene people, not demoscene people. Since this FAQ is for comp.sys.ibm.pc.demos, I stick to the demo viewpoint.

Demo Groups:

(7.0) How can I create or join a demo group?

Advertise. Posting a message on your local demo-oriented BBS or on comp.sys.ibm.pc.demos is probably a good way to start. You can also talk to anyone on the usual IRC channels (see FAQ 3.1), or search demogroup's WWW pages (see FAQ 3.2). You could even contact individual groups directly through email.

(7.1) Whatever happened to Future Crew?

(As of 5/10/96:)

As a group, nothing official (ie., they haven't disbanded officially yet). Henchman has hinted that they will probably do a demo for Windows 95 as soon as the Pyramid3D chip hits the market.

Commercial:

* Psi, Trug, and Trug's brother designed and created the Pyramid 3D chip, which is supposed to be the fastest 3D accelerator chip in the PC world. Boards using this chip should be ready by December 1996. You can get more info on it at http://www.tritech-sg.com/mm/pyra/index.htm.
* Most of the members of Future Crew formed Remedy Inc., a game company. Their first game, Hispeed (a racing game), has already been picked up by Apogee and should come out sometime in 1996. Gore studied business in school so that he could lead the company.
* Purple Motion has done music for Bloodhouse, an Amiga game company, and continues to write music for game companies. It is also rumored that he is cutting a techno album.

Demos:

* Purple Motion has contributed music to EMF since 1991, and continues to contribute to productions by Plant, a group formed from members of both EMF and Sonic PC.
* Pixel is in the army as of this writing, but continues to draw for other groups.

Personal:

* Some members are in the army as part of the mandatory Finnish army service.
* Skaven is at a Disney cartooning school in Ireland, but it's rumored that he will return before completing his four years there.

(7.2) Whatever happened to Triton?

Triton has just about switched from demos to commercial games. They keep their name, but don't produce demos any more and don't plan to.

(As of 5/10/96:)

Commercial:

* They are finishing up a game called Into The Shadows that offers some of the best 3D graphics in a PC game. It's also one of the last games to be programmed in pure assembler. It is supposed to be much faster than ID's Quake.

Demos:

* Lizardking occaisionally enters compos and lends his music talents to other groups.
* Triton also continue to maintain and update FastTracker 2.0x (see FAQ 2.4), one of the best trackers ever created.

Personal:

- Vogue, Mr. H, and Lizardking were in California for a bit finishing up the game.

(7.3) Whatever happened to Cascada?

Cascada has gone through a couple of changes since 1994: First they split up into Cascada VR, a commercial venture, and the remaining members joined Imphobia for a while, although that didn't last.

(As of 5/10/96:)

Commercial:

- Cascada VR reportedly has a very nice virtual reality 3D engine that they are trying to sell for the use of a game company.

Demos:

- Zodiak continues to compose music, both in general and for Cascada demo productions.
- New members of Cascada created an intro for TP95 (which had really nice shadows, BTW).

Personal:

- Nothing known at this time. Anyone?

(7.4) Whatever happened to Twilight Zone?

Twilight Zone gained early popularity with funny demos that were the first to support the GUS (Jungly Kitchen and Monty Python). Since then, they've formed a game company.

Commercial:

- Twilight Zone is now a game company with their first game, Interpose, published by Webfoot Technologies. You can download the shareware version of their game from http://www.webfootgames.com

Demos:

- Inactive.

Personal:

- Nothing known at this time. Anyone?

(7.5) Whatever happened to Byterapers?

Byterapers are arguably the oldest surviving demogroup, starting out on the C64 and Amiga, and currently doing projects on both the C64 and PC. As for their current status, they put out a press release, which, ah, speaks for itself:

Byterapers, Inc., a demogroup celebrating this year their
10th anniversary, has entered the PC demoscene on 1st weekend
June 1996, by releasing the demo "Drill Me, Please Me"
in Abduction'96 demoparty in Oulu, Finland.

This first PC-release ever took part in Abduction's democompetition, and won the first place by a groovy margin of almost twice as much votes as 2nd or 3rd participants.

The winning of a democompetition with Byterapers' first PC-release is a great way to show that this 10 years old group is still alive and active, and continues our already established line in Abduction'96 demoparty in Oulu, Finland.

Products like the World of Code -series (WOC, WOC 2, WOC 3) and Extremes are top notch releases on C-64 -demoscene, and our three latest entries in democompetitions have gained two wins and one second place, which are nicely supplemented by "Drill Me, Please Me"'s instant success.

Byterapers' history spans from the year 1986, when four little would-be fellas started using name "Byterapers" on their small releases. In two years group had grown into multi-national and rather famous, starting 1988 on Amiga. In 1989 the group stopped C-64 activities and concentrated on making career on the Amiga, making rather well there as well. Around 1991-1992 group started a long sleep, resting from earlier efforts and doing a lot of own inner stuff, picnics, ByteParties etc but never finishing any products.

From this six years period Byterapers, known in short as (B), had released around 150 demos on C-64, not counting intros or other small stuff, a countless number of cracks and all-time high position as NO:9 in world's cracking-group charts (an activity which was later banned and stopped completely), seven musicdisks on Amiga, about 20 Amiga-demos not counting intros and other small stuff, bunch of weird fun-releases an uncontested fame of heavy partydudes always drunk, noisy, doing strange stuff, looking strange, wandering from bar to bar but overall being quite harmless.

(B) made comeback in 1994 by releasing the World of Code -demo in spring'94. This resulted in many scene-awards like:
-comeback of the year
-demogroup of the year
-demo of the year
-coder of the year
As voted in _several_ leading C-64 disk-publications.

As looking into 1996 and forward, (B) plans to exist and booze, and release occasinal demo on C-64 and PC when feasible.

Byterapers' side company, the "Byterapers Movie Productions" s also finalizing the 4th Byterapers world-class, high quality home movie known as "Splattered Death II - River Runs Red", hopefully hitting audiences this summer is we manage to scrape couple bucks to pay the camera rents to they'll let us edit the film.

Earlier Byterapers Movie Productions Big Money Great Quality whole-evening all-family no-blood films are:
Splattered Death
Fochrybenis
(B) Files

the latter being the hugely-successful and famous X-Files parody. Followup coming this autumm! On plans also "Byterapers goes Doom".

Byterapers - a pain in your arse
Byterapers - the reason for your erection

Official warning bit:
"Byterapers is dangerous for your brain. Booze kills braincells.
Official Byterapers membership-requirement is to be able to consume
huge amounts of alcohol."
"Byterapers is no demogroup. It's is way of life."

(B) memberstatus:

NAmn (Loc) E-Mail Kone*WasMachts

(B) Pielavesi
Nico - Amiga*gfx
Lasse - -*hitman
Proton - -*hitman, wall
Enema - *hitman, king fartTampere
Mr. Sex b150542@proffa.cc.tut.fi C64*kooder,laatta
Dr. Dick t136344@proffa.cc.tut.fi C64/Amiga*kooder,hw-huru-ukko,guru
Birra jsoini@vip.fi PC*hitman, myy halvalla, sheepish partyguard

(B) Savon SNTL
Reznor - -*hitman,kiljudoctor, movie master
Hazard hazi@finhost.fi Amiga*hitman,teknokko,gamemaster,warlock,hirmu
Jazz jazz-b@freenet.hut.fi -*muzza, party, Her Royal Cuteness
Grendel jkauppin@muikku.jmp.fi C64/Amiga/PC*hitman,hirmu
Kasper jarkko.sonninen@lut.fi C64/PC*kooder,hitman
Professor Fate PC*hitman
Lanttu U.S.Army on PC*kooder,hw-huru-ukko,hitman,official mascot
T.o.B - PC*muzza,hitman
Sivu - PC*kooder,hirmu

(B) Muualla
Icemann Vantaa - PC*hitman, expensive partyguard (batman)
TP Helsinki/Somero pomppuju@atki.helbp.fi Amiga/PC*kooder
Mike Jyväskylä - C64*gfx
Jate Jyväskylä ? PC*gfx
Micron Raahe mhyvonen@ratol.fi on PC*hitman
Albert Helsinki ? Amiga*hitman
Kemu Oulu mysti@stekt.oulu.fi C64/Amiga*Design,juoppohullu
Kraku ? cracker@freenet.hut.fi PC*kooderRuåtti/Tukholma
Dr. Star ? ?*?
CSA f93-caa@nada.kth.se ? ?*?
Speed-Head damaskus@algonet.se amigamuzzak,3d,unix ja nettaus

This press-release has been written while drinking Heineken-beer
by Grendel/Byterapers, the founder of whole stupid group.
Have fun.

Ps.
Availability of the demo will be accounced, maybe party-organizers
will spread it bu twe'll make a fixed version.

grendel/(B)

```
--
Jukka O. Kauppinen  Mail: Sankarinkatu 9A3,74100 IISALMI,FINLAND
Journalist          E-Mail: jkauppin@muikku.jmp.fi
MikroBITTI          Tel/fax +358-77-24225
  Byterapers Inc.   Amiga 500/1200(030),PC486,CD32,C64,Spectrum
```

(7.6) Whatever happened to Vibrants?

Vibrants was one of the most famous and all-around liked groups on the C64, with many famous musicians who have done work for various video games on various platforms; they also dabbled in PC Adlib music, producing arguably the best Adlib music and Adlib tracker ever. They are now fully migrating to the PC, as this press release illustrates:

Vibrants has now started on the PC.

If you need music for a gameproject, feel free to contact Vibrants at this adress :

Vibrants
Co. / Thomas Egeskov Petersen
Merkurvej 2
6600 Vejen

Vibrants started on the 64 quite a while ago, and has over the years emerged to a varity of platforms. We work mostly on the PC nowadays, in form of mods, midi, or CD (Which ofcourse plays on any CD-Rom based platform)

Vibrants Member are :

Thomas Mogensen	(aka. Drax)
Torben Hansen	(aka. Metal)
Jesper Olsen	(aka. JO)
Jens Christian Huus	(aka. JCH)
Thomas Egeskov Petersen	(aka. Laxity)

Feel free to contact us, in case you're interested !

Demo Parties/Compos:

(8.0) What's a Demo Party? What's a Demo Compo?

A Demo Party is just that--a party celebrating demos. People bring their computers, show off effects, watch demos, track music, etc. They're usually on the small side; no more than 50 people.

A Demo "compo" (Euro slang for "competition"), on the other hand, is a competition where prizes are given to the best demos and intros. While parties are small, compos are large. Compos are usually over 400 people (the largest are The Party and Assembly, which usually host over 3500 people), and are mostly held in schools because they're easy to rent and don't cost a lot of money. Activites are held all 24 hours of the day, so it's difficult to get sleep and not miss something cool. As a result, many people simply don't sleep during a compo. :-)

After a party or compo, some groups put out "party reports", where you can see pictures taken at the party/compo and read text describing the results of the compo, and see pictures of events and people at the compo. One place you can find party reports is http://ftp.hornet.org/pub/demos/party

----

(8.1) When and Where are Demo Parties held?

Much to the chagrin of North American 'sceners, most demo compos are held overseas, in Europe. Only one recurring demo party, NAID, is held in North America (Montreal, Quebec, Canada), although it is rumored that after 1996 there will not be any more; Eclipse '97 is rumored to take over.

Some Demo compos are held every year at roughly the same time:

- Assembly: August
- The Party: Christmas
- The Gathering: Easter

aschlud@autelca.ascom.ch (Denis Schluchter) was nice enough to write up this schedule for the FAQ:

DEMO EVENTS '95:

Bizarre	9/10 September	Etten-Leur, Holland
Wired	3/4/5 November	Mons, Belgium
The Party	Winter	

DEMO EVENTS '96:

The Partyplaces may change! (I took those from last year ...)

X	Spring	Utrecht, Holland
The Gathering		
NAiD		Canada
Abduction		
Somewhere in Holland	Summer	Roosendaal, Holland
Juhla		Iisalmi, Finland
Bushparty		
The Summer Encounter		Denmark
Assembly		Helsinki, Finland
Gasp		Montpellier, France
Wired	Fall	Mons, Belgium
The Party	Winter	

Phoenix / Hornet offered this more up-to-date (hey, it's his job :-) list as well:

Other Regular Demo Parties:

				code on hornet.org
Spring:	X	Holland	since 1995	X9?
	Cache	Hungary	since 1994	CAC9?
	Saturne	France	since 1994	SAT9?
	Scenest	Hungary	since 1995	SCE9?
	Mekka/Symposium	Germany	since 1996	MEK9?/SYM9?
Summer:	Abduction	Finland	since 1994	ABD9?
	Enlight	Russia	since 1995	ENL9?
	Euskal	Spain	since 1995	EUS9?
	Icing	Sweden	since 1995	I9?
	Remedy	Sweden	since 1995	REM9?
	Summer Encounter	Denmark	since 1995	SE9?
	The Scene	Singapore	since 1995	TS9?
Fall:	Bizarre	Holland	since 1994	BIZ9?
	Movement	Israel	since 1995	MOV9?
	Wired	Belgium	since 1994	WIR9?

Other than this list, you can find more demo-related events at http://hagar.arts.kuleuven.ac.be/~sdog/party/ which lists other WWW pages dedicated to demo parties and demo compos.

(8.2) How are Demo Compos judged?

Good question. First, a point system has to be agreed on. There are many ways of doing this; for example, at NAID'95, each demo was given up to 5 points for graphics, programming, music, and overall design. In this case, the demo with the highest total point value won, the 2nd higest came in 2nd, etc. Second, if there are way too many entries, all entries go through a preselection process, where an informal vote disqualifies substandard entries. (This step only applys to compos with a judging panel; see below.)

Now, who actually does the judging? Usually, there are two different methods: A public vote, and a juding panel.

- Public Vote: The people attending the compo vote on the demos shown. Pros: The public can't complain about the judging, since they were the ones who voted. Cons: The public is usually way off. While the crowd consists of true professional programmers, musicians, and artists, they usually vote highest for demos that have "flash" and "pizazz", ignoring the actual quality or design (or lack thereof).
- Judging Panel: A panel is created, usually with a cross section of programmers, musicians, and graphicians. The panel can not have anyone in it that's involved in some way with the demos being voted upon. Pros: The judging is usually fair. Cons: The public usually disagrees with some of the judge's decisions, especially during the preselection process.

Please note that any of these methods is not necessarily better than any other. Sometimes they're combined in some way, like NAID'95, where there was a public vote that determined who got the "public favorite prize", and a judging panel, who determined the "offical" 1st-10th place order and top three prizes.

Operating Systems and Platforms:

(9.0) Why aren't there any Windows/Windows 95 demos?

Probably for the simple fact that Windows does several things to restrict the typical democoder, including inhibit system performance, and restrict direct access to hardware. These limitations have workarounds, but it's so much easier to just dive in under DOS with assembly language that nobody wants to make one.

Windows is generally looked down upon in the demo community, because of it's slow speed and huge size; many joke that it's one of the lamest demos ever written. :-)

Windows 95, on the other hand, has some possibilities for being a demo platform, with it's DirectX API. DirectX has direct interfaces to hardware--DirectSound, DirectDraw, etc.--so it's possible to circumvent the limitations listed above. Still, the API is overhead, so by default it can't be as fast as raw DOS, and as such is still looked down upon by democoders.

If you're interested in coding demos for Windows 95, you can order the Win95 Game SDK directly from Microsoft completely free-of-charge, or download it from their web site at http://www.microsoft.com. If contacting them, you just need to tell them that you are "porting your games which you have under development to the Windows 95 DirectX platform". The Game SDK (GDK) includes DirectDraw, DirectSound and DirectPlay.

But in order to fully develop Windows 95 apps, you might need the Win32 SDK too, which comes with the "Microsoft Developer Network (MSDN) level 2". This costs around $500US per year and it includes all MS' SDKs (about 30 CDs), and it comes with a quarterly update service too.

If you want to learn more about DirectX, consult the following links:

* http://ourworld.compuserve.com/homepages/stant/ - Cutting-Edge Direct3D Programming
* http://www-scf.usc.edu/~permadi/gamelink.htm - Floating Point's Game Developers Resource List
* http://www.employees.org/~mnorton/ - Spells of Fury: Direct X Programming
* http://www.mathematik.uni-marburg.de/~rusi/mikdx/mikdx.html - Mikmod For Directsound Homepage
* http://www.microsoft.com/mediadev/ - Microsoft Developer Network - Interactive Media
* http://www.readmedotdoc.com/1572313307.htm - DirectX Game Programming for Microsoft Windows 95 w/CD, by Leavens, Alex
* http://www.rosenmader.com/lar/DirectX/DirectDraw.html - DirectDraw Programming
* http://www.silvercrk.com/directx/ - DirectX Developer's Page
* http://www3.hmc.edu/~tsemple/games/win95.html - Hot Win95 Game Programming

Luc-Eric Rousseau had this to contribute:

Ok, here are my comments (as a Windows developper).
The Direct X allows programmer everything they do in DOS, but doesn't save you much work. It is just meant as a clean and safe way to use the hardware directly under Windows 95. As such, demo coders can write the code the way they like -- all assembler, if they like, and they _will_ gain much performance by doing that. These demos _will_ however always run slower than in DOS, of course, but they will be more user-friendly (no clean reboot, compatibility will all sound cards, ect).

Demo coders will have no interest in Direct3D - they benefit from custom-made 3d routines, and anyway that's the whole fun of writting demos. Direct3D might be used by art-oriented groups.

Direct sound is like programming a very easy to program new audio card. I'm sure it's possible to port a DOS mod player over to windows in less than one week. Even without it, taking a look at Mod4Win tells you there's a lot of music to be played under Windows.

The only problem is that documentation is hard to understand for medium-level Windows programmers . But it's a bit like VESA, it has been a gourou thing for long, but (really) not anymore.

The DirectX SDK is less than 6 month old, it will take at least a year for people to be at ease with it. Many groups will jump over to Linux instead, but the audience just isn't there. Of course, DOS will always work! But debugging ease in other environment will move people to other environment. It just isn't fun to debug a DOS program - unless you're well equipped with a dual-monitor setup/

My prediction is than in one year, you'll see as many demos under Windows95 as you see VESA demos now : still a thing for programmers with plenty of experience. The first few ones will get plenty of attention, though.

(9.1) Why aren't there any Linux demos?

There's one, actually: Hard Rox by Skal. But considering that Linux is a fast 32-bit multitasking operating system and that good optimizing compilers are free under Linux, you might think that many more demos exist for it. But while some screen savers that use Linux's SVGAlib might be considered demos, "true" demos haven't caught on (although many claim they are porting their demos over to Lunix).

One possible reason for this is that the multitasking nature of the OS means that the CPU cannot be dedicated to any one critical task--for example, the inner loop of a texture-mapping engine, or vertical-retrace interrupts, or anything else requiring exclusive use of the CPU. Another possible explanation is that even if dedicated CPU attention was possible, it is difficult to do so without writing a device driver, something that many democoders are not willing to do. Still, it seems a bit more likely that demos will be written for Linux before they are written for Windows, if only for the fact that Linux is free and fast.

(9.2) Are there any demos that run on PC platforms that aren't DOS-based?

Only two that I know of: An OS/2 demo has been written for OS/2 3.0 (OS/2 Warp, with MMPM/2 extentions installed) called Peripheral Evolution OS/2 by Ethos. It also comes with its own source code. A demo for Linux also exists, Hard Rox by Skal. It won the Wired '96 demo competition, and it runs in DOS as well.

There is a ongoing project to create an operating system exclusively for writing demos, called the DemOS project. You can find information about it at three possible places:

* http://www.cdrom.com/pub/demos/info/DemOS/
* http://www.ifi.uio.no/~dag-erli/demos/
* http://www.skolesjefen.oslo.no/skole/persbraten/hjemmesider/demos/

The first URL listed (cdrom.com) is the fastest for people in North America.

(9.3) I don't own a PC. Are demos written for other computers?

Several, actually. Demos didn't originate on the PC; they've been around for over a decade, having grown from the cracktros that software pirates used to tack onto their releases. Nowadays, demogroups are not related to cracking groups anymore; in fact, most PC demogroups were never related to a cracking group.

Demos exist/have existed for the following platforms (listed in approximate chronological order):

* Apple][(in a cracktro form only, to my knowledge)
* C64 (where the demo scene was truly born)
* Amstrad CPC (it was very popular in Europe, especially in England and in France, at the same period as the C64, and there have been good demos for it. Check ftp://ftp.nvg.unit.no/pub/cpc or ftp://ftp.ibp.fr/pub/amstrad for the emulator and software...)
* Commodore Plus/4
* Spectrum 128
* Apple][gs
* Atari ST
* Amiga (where the demo scene flourished into new territory)
* Atari Falcon
* Macintosh (Harassment, Erotomania, Frivilous Exodus, and Mirror Image are a few; I believe they require a 68030)
* IBM PC
* Arcon Archimedes (This is a RISC PC. There have been demos for that computer at the Symposium Party held in Hamburg, Germany.)

Some demos exist for game consoles! No, I'm not lying. There are some demogroups for the Super NES, for example, that connect to the SNES through a "cart copier" connected to an Amiga, and they program the machine that way. The SNES has a decent sound chip and texture-warping built into the machine, so I guess some people got curious. :-) Check out http://www.futureone.com/~damaged/Consoles/SNES/demos.html for some examples.

(9.4) Are Amiga groups migrating to the PC?

With the death of Commodore and the future of the Amiga no longer clear, some groups have decided to move partially or fully to the PC. In recent months the PC demo scene has seen music, graphics, or entire demos by members or former members of CNCD, Oxygene, Bomb, Silents, Scoopex, Byterapers, etc.

Nooon has had contributions from some Sanity members (Moby and RA); many members of Pulse are from the Amiga scene as well (DreamDealers).

Oxygene is actually ex-Atari and an ex-Amiga group. :-) Oxbab and company wrote one of the best demos for old Atari ST machines.

(9.5) Are there any Java demos?

As of the moment, nothing serious demo-related exists for Java, for a couple of reasons:

* Java, as a language in general, specifically prohibits you from modifing the hardware directly. This makes it difficult to program your own graphics or sound routines with any degree of speed.
* Java on personal computers is usually an interpreted or p-code language, which does not deliver the speed that demos demand. This could change when Sun comes out with the Java chip and Java computers, however.

Many people are trying, however. http://www.hornet.org/pub/demos has a link to Demo Java that lists some demo effects on the web. There are other sources as well:

* http://www.uni-koblenz.de/~cohnen/SIMPLE.HTM - Phong shading
* http://www.dur.ac.uk/~d405ua/Plasma.html - Plasma, with frames per second display
* http://www.assembly.org/javacomp/ - Assembly Organizing is hosting a Java demo competition at this URL.

(9.6) Why aren't there any BeBox demos?

There are no BeBox demos because the platform is too new, and hardly anyone owns one. However, the Bebox seems like one of the best new machines to come along for demos. As far as fast graphics are concerned, the Be product demo has a part where it plays 6 true-color video feeds onscreen simultaneously, without choppiness. Plus, it has a standard sound interface.

For more info, check out the Be homepage (http://www.be.com/) and the Be newsgroup (comp.sys.be).

Technical/Programming Info:

(10.0) What's the fastest PutPixel?

There are two basic version: One that uses a lookup table, and one that doesn't. The lookup-table version is the fastest.

Non-lookup-table version:

```
Procedure PutPixel(X,Y:Word; c:Byte); Assembler;
Asm
  mov AX,0A000h
  mov ES,AX
  mov AH,Byte Ptr Y
  mov BX,X
  add BX,AX
  ShR AX,2
  add BX,AX
  mov AL,c
  mov ES:[BX],AL
End;
```

Lookup table version:

```
Var
  lut : Array[0..199] Of Word;

Procedure PutPixel_LUT(X, Y : Word;  C : Byte);  Assembler;
{ code from  Jannie Hanekom  }
{ optimized by  Andreas Jung }
Asm
  mov  BX, Y
  add  BX, BX
  mov  AX, 0A000h
  mov  ES, AX
  mov  BX, Word Ptr lut[BX]  { Note:  BX not changed within 2 cycles }
  mov  CX, X
  add  BX, CX
  mov  AL, C
  mov  Byte Ptr ES:[BX], AL  { Again 1 cycle before memory move }
End;
```

Just make sure you fill the lookup table before you use the procedure, with something like for i := 0 To 199 Do lut[i] := i*320.

Kneebiter has provided 386 protected-mode equivalents:

```
;eax - color
;edi - y
;ebx - x
;
; The fastest (?!?) non-table version

    shl    edi, 6                ; ebx *= 64                    3 clocks
    lea    edi, [edi*4+edi]      ; ebx *= 5   (64*5 = 320)      1 clock
    mov    [edi+ebx+0a0000h], al ;                              memory ref

; The fastest (?!?) table version

    mov    edi, [edi*4 + table]  ;                              memory ref
    mov    [edi+ebx+0a0000h], al ;                              memory ref
```

What many people don't realize is that the real fastest putpixel is the one inside your own code, polyfiller, or whatever, because the time it takes to push the registers on the stack and call the function takes more time than actually plotting the pixel! So, try to incorporate these methods into your own program instead of making a function you call out for.

(10.1) How do I program a Vertical-Blank Interrupt?

A VBI can be done with something like this:

```
proc INT8 ; VBI
  SetVGAStartAddress ;optional
  WaitForVRT
  SetPalette ;optional...
  ReprogramTimer ;with a rate FASTER than the screen refresh rate
  EOI
end
```

This can be expanded further into a system that reaches any scan line, like this, for instance:

```
proc INT8_part1 ; VBI
  WaitForVRT
  ReprogramTimer ; make the int occur in the middle of the screen
                 ; or any scanline you want
  SetInt8Vector(INT8_part2)
  EOI
end

proc INT8_part2 ; some scan line X -interrupt
  WaitForHRT
  DoSomeFancyStuffWithTheScreen
  ReprogramTimer ; make the int occur before the VRT
  SetInt8Vector(INT8_part1)
  EOI
end
```

Just add as many parts to this thingy! For a good example, check the productions of White Shadow/Renaissance (no longer in Renaissance, though). There are some of them in ftp.hornet.org. For a start, take a look at DoWhackaDo and DeadBeat BBS-intro.

Here's a protected mode assembler example. The main thing that will change for other asm based languanges is the interrupt hooking.

```
.386p
code32  segment para public use32
        assume cs:code32, ds:code32

include pmode.inc

public  VBI_function
public  frame

; DATA

DELAY   EQU             16805

old_pmode_irq0_vect     dd      ?
old_rmode_irq0_vect     dd      ?
irq0_buffer             db      21 dup (?)
old_pmode_irq1_vect     dd      ?
old_rmode_irq1_vect     dd      ?
```

```
irq1_buffer             db      21 dup (?)
VBI_function            dd      ?
frame                   dd      ?

; CODE

_main:
        sti

        mov     VBI_function, offset _ret       ; notice that I initialize
                                                ; VBI_function before hooking
                                                ; the IRQs
        call    hookirq1
        call    hookirq0

        ; blah, blah, blah...

        call    unhookirq1
        call    unhookirq0
        jmp     _exit

hookirq1:                                    ; hook the keyboard
        mov     bl, 1
        call    _getirqvect
        mov     old_pmode_irq1_vect, edx     ; in pmode
        mov     edx, offset newirq1
        call    _setirqvect
        mov     edi, offset irq1_buffer
        call    _rmpmirqset                  ; and in real mode, just in case.
        mov     old_rmode_irq1_vect, eax
        ret

unhookirq1:
        mov     bl, 1
        mov     eax, old_rmode_irq1_vect     ; unhook real mode kb IRQ
        call    _rmpmirqfree
        mov     edx, old_pmode_irq1_vect     ; unhook pmode kb IRQ
        call    _setirqvect
        ret

newirq1:                                     ; just completely ignore the keyboard
        push    eax
        in      al, 60h
        mov     al, 20h
        out     20h, al
        pop     eax
        sti
        iretd

hookirq0:                                    ; see comments from hookirq1
        cli
        mov     bl, 0
        call    _getirqvect
        mov     old_pmode_irq0_vect, edx
        mov     edx, offset newirq0
        call    _setirqvect
        mov     edi, offset irq0_buffer
        call    _rmpmirqset
        mov     old_rmode_irq0_vect, eax

        ; I assume that the timer is in its normal mode of operation at program
        ; startup and I have the ISR reprogram the timer for me.  That's also
        ; why I do the hooking early in the program set up, to make sure that
        ; atleast one timer interrupt goes off.  I'll synch the timer to the
        ; routines later in setup.

        sti
        ret

unhookirq0:
        cli
        mov     bl, 0
        mov     eax, old_rmode_irq0_vect
        call    _rmpmirqfree
        mov     edx, old_pmode_irq0_vect
        call    _setirqvect

        mov     al,  00110110b          ; timer 0, 16-bit counter, mode 3
        out     43h, al                 ; resets the timer.  What about
        xor     ax, ax                  ; reseting the time based on the
        out     40h, al                 ; CMOS clock?  Their time will be
        mov     al, ah                  ; wrong.
        out     40h, al

        ; this resets the timer to its normal mode of operation

        sti
        ret

newirq0:
        pushad
        cli

        call    wait_retrace   ; retrace is 1/70th of a sec, timer is usually 1/18
                               ; of a second.  No problem before first setting the
                               ; timer of having recursion.  After that the timer
                               ; set for one shot, so there's no way the ISR goes
                               ; off twice.
                               ;
                               ; The CLI instruction just before doesn't hurt eith
                               ; :-)

        ;
        ; reprogram the timer here...
        ;

        mov             al,  00111000b   ; timer 0, 16-bit counter, mode 4 (one sl
        out             43h, al
        mov             ax, DELAY        ; the delay is just under 1/70th of a sec
        out             40h, al          ; want to wait too long and miss the retr
        mov             al, ah           ; a little room for interrupt latency and
        out             40h, al          ; that run nearer to 68 Hz refresh.

        call    [VBI_function]   ; put your blitting routine here?

        ; poll the music player here.
        ;
        ; makes life a little easy if the music is at a certain BPM

        inc     frame            ; frame count
        mov     al, 20h
        out     20h, al          ; EOI
        popad                    ; restore stack
        sti                      ; set interrupts
        iretd                    ; return

code32  ends
end
```

(10.2) How long is one "clock cycle" on a particular machine?

It's the simple speed of the computer (xxMhz). So, for a program that syncs with the vertical retrace, if the program is run on a 486/33 for example, then for each retrace there is only 33000000/70 = 471428 clock cycles to spare.

(10.3) How do you do environment mapping?

If you've done a texture mapper, you're not far away from an environment mapper. Say you want to environment map one face of an object. For each vertex, get the vertex normal (it's just the average of all the normals of the connected faces), then convert that into spherical coordinates (Rho, Phi, and Theta; although Rho doesn't matter since normals are length 1)... well Phi and Theta are now the spherical direction of that vertex. What you do is, you use these two angles as the lookups into a giant spherical texture map that theoretically surrounds the object. There are many ways to do the "wrapping" of the sphere... I just took a 256x256 image, centered it at the front, and curved the corners towards the back (which would be 128, plus/minus phi in one direction, plus minus theta in another; you get the idea... I can post the exact formulas if you can't figure 'em out, but they're very simple). With this, you just use those two angles, process the values into your wrapped texture to find the texture U,V at that vertex. Do that for all four vertices and texture map as usual. Do it for all faces, and you've got your envmapping.

Note though, that for all sphere objects (and note that a cube is just a 6-sided sphere) with the map in the center, the vertex normals ARE the normalized vertices, so it will give some strange results... it's better to use objects that don't have the normals equal to the vertices themselves, just so the envmapping looks legit. :-)

(10.4) How can I do a DMA Putpixel or DMA Mem-to-Mem copy?

You shouldn't bother. Not only is the DMA controller limited to 64K segments, but only on the slowest PC's (we're talking 8086 here) is it faster than doing a REP MOVSW.

Still, DMA memory transfers take no CPU time at all, but you still shouldn't use them because they're just too slow on modern PC's. If you really want to do it anyway, just to see if it works, here's a small article from 1989 by Patrick & David O'Riva that can help you. Remember, in 1989 on a 286, this was faster than REP MOVSB. By the 386, however, a REP MOVSD was faster, so this is reprinted for historical reference only.

DMA Transfers for FAST Memory Moves

If you ever have the need for super fast moves of blocks of memory and you can put up with a lot of restrictions in order to gain the speed there is a method in the PC environment to move bytes at the rate of one every 3 clock cycles. Compare that to the clocks for a rep movsb and then say that isn't fast. This isn't for every programmer and it certainly isn't for every time you need to move a few bytes.

The DMA chip is a four channel device. Channel 0 is reserved for refresh of memory that requires at least 256 consecutive bytes be read some- where in memory at intervals less than about 4ms. The other three channels are available for such things as transferring information to and from the disk drives or the ports. Interesting trivia so far, but not very useful in moving memory around. It gets worse. The 8259 DMA doesn't know anything about segments. It only knows a 64k universe. This is where the Page registers come in. The page registers decide which page (on 64k boundaries) the 8259 will look at for any operation. There are not, as you might guess, 4 page registers,but only 2 plus a default. If it is not channel 1 or 2 then it uses the default register programmed as channel 3.

A careful reading of the data sheet of the 8259 discloses that it is capable of doing a memory to memory transfer but only between channels 0 and 1. That is why this method is a little tricky to use. In order to set up your own parameters you have to disable the timer from asking for a DMA from channel 0 every x milliseconds and reconfigure the 8259 and assume the responsibility for doing the memory refresh. It actually sounds worse than it is. The configuring and re configuring of the 8259 doesn't take all that long, so the time is made up after only moving a few tens of bytes, and if you move at least 256 CONSECUTIVE bytes the memory refresh requirement is met for another 2 or 3 milliseconds. The page registers are taken care of by setting channels 1 and 3 to the same value.

Given below is an example of a program I wrote just to test the idea. A lot of the setup is too complex to explain in this short article, but if you are interested in checking it all out you will need a data sheet on the 8259. This worked nicely on my machine and should on most compatibles just the way it is. With the not-so-compatible it may very well not. I hope this listing is well enough commented so you can figure it out and make use of it sometime.

` DMA SOURCE

```
PAGE 60,132
TITLE DMA MEMORY TO MEMORY

DMA         EQU 0

STACK       SEGMENT PUBLIC 'STACK'
            DB  32 DUP('STACK')
STACK       ENDS

DATA        SEGMENT PUBLIC 'DATA'
SOURCE      DW   08000H
TARGET      DW   09000H
NUMBER      DW   800H
INCDEC      DB   0
PAGER       DB   0BH    ;PAGE (O TO F)
FILL        DB   0      ;2 IF A FILL OP
DATA ENDS

ASSUME CS:CODE,DS:DATA,ES:DATA
CODE SEGMENT PUBLIC 'CODE'

START:
     MOV AX,DATA
     MOV DS,AX
     MOV AX,0B800H
     MOV ES,AX

PAGE:
     MOV AL,PAGER       ;PAGE TO OPERATE IN
     OUT 83H,AL

UNDMA:
     OUT 0DH,AL         ;MASTER RESET OF DMA
     MOV DX,03D8H
     MOV AL,1
     OUT DX,AL
     MOV AX,SOURCE     ;WHERE IS IT COMING FROM
     OUT 0H,AL
     MOV AL,AH
     OUT 0H,AL
     MOV AX,TARGET     ;WHERE TO PUT IT
     OUT 2H,AL
     MOV AL,AH
     OUT 2H,AL
     MOV AX,NUMBER     ;HOW MANY
     OUT 3H,AL
     MOV AL,AH
     OUT 3H,AL
     MOV AL,009H       ;ENABLE M TO M,COMPRESSED
     OR  AL,FILL       ;WILL BE 2 IF FILL OP
     OUT 8H,AL
     MOV AL,088H       ;BLOCK MODE, INC, READ
     OR  AL,INCDEC     ;WILL BE 20H IF DEC
     OUT 0BH,AL
     MOV AL,85H        ;BLOCK MODE, INC, WRITE
     OR  AL,INCDEC     ;WILL BE 20H IF DEC
     OUT 0BH,AL
     MOV AL,4          ;THIS IS THE REQUEST
     OUT 9,AL          ;DO IT
     MOV AL,9
     OUT DX,AL

RESET:
     OUT  0DH,AL       ;THIS IS A MASTER RESET
     OUT 0CH,AL        ;RESET F/L F/F
     MOV AL,01
     OUT 0,AL
     OUT 0,AL

REINIT:
     MOV AL,0
     OUT 83H,AL        ;MOVES REFRESH TO BASE PAGE
     MOV AL,0FFH
     OUT 1,AL
     PUSH   AX
     OUT 1,AL
     OUT 0BH,AL
     INC AL            ;MODE CHAN3

     OUT 0BH,AL
     PUSH   AX
     POP AX
     POP AX
     POP AX
     MOV AH,4CH
     INT 21H

CODE ENDS
END  ART
```

Kneebiter / Threesome had this to add: "If you're doing it to put a pixel, it takes just as long to write to the DMA buffer as it takes to write to a video buffer. If you want to use it to blit memory screens, you can get about 15 frames in a second at maximum and the user can see the DMA doing the copying. That, and it's a pain in the butt as you need to use 2 DMA channels to do it, very specifically channels 0 and 1. Channel 0 is not guaranteed to be free. Ever wonder why demos don't support DMA channel 0 for sound?"

Finally, some new mainboards (the Shuttle 433, for example) don't support DMA mem->mem transfer anyway.

Are you getting the feeling you shouldn't ask about this any more? :-)

(10.5) What's protected mode?

For the purpose of talking about demos and PC's, protected mode is an operating mode of all Intel 80386 cpu's and higher that allows several benefits for the coder over the older real mode, including:

- The 1-cycle performance penalty is gone when accessing 32-bit registers (EAX, EBX, etc.) and extra registers (FS, GS, etc.) because you don't need the prefix override.
- All the memory in the computer is available to the programmer, in large chunks of any size (not limited to 64K).
- Memory is protected, so exceptions (accesses to memory a program doesn't own) can not only be trapped, but handled.

http://207.239.225.146/articles/pmbasics/tspec_a1_doc.html is one place that has more information on protected-mode programming.

(10.6) How do you do bump mapping?

Bump mapping isn't much harder than environment mapping: you just precalculate a map which perterbs the normals (from phong/gouraud) by a given angle. Combine it with a texture map to fit the perturbances, and it looks bumpy.

Mr. P / Powersource adds more:

Bump mapping applies almost entirely to Phong shading only. Applying it to other shading models is much more difficult because of the way it works.

What we do is step through the maps like texture maps, but instead of reading a color, we read an X displacement and a Y displacement. We then add these values to the surface normal at that pixel (now it should be obvious why Phong is the best suited to this), calculate the intensity, and store it.

Again, if you want, you can use any coordinate system you want. I chose a 2 dimensional displacement because that is basically how it is done with environment mapped Phong.

Rex Deathstar sums it up like this: envmap[x+bumpx][y+bumpy] :-)

(10.7) What's a fast way to calculate a square root?

My buddy JARE/Iguana was kind enough to donate this source code:

```
/*
 * A High Speed, Low Precision Square Root
 * by Paul Lalonde and Robert Dawson
 * from "Graphics Gems", Academic Press, 1990
 * Modified by Javier Arevalo to avoid some unnecessary shifts.
 * Seems like this adds some precision too, but I'm unsure.
 */

/*
 * SPARC implementation of a fast square root by table
 * lookup.
 * SPARC floating point format is as follows:
 *
 * BIT 31   30  23  22  0
 *     sign exponent    mantissa
 */

#include

long sqrttab[0x100]; // declare table of square roots

void build_table(void) {
    unsigned short i;
    float f;
    unsigned int *fi = (unsigned*)&f;   // To access the bits of a float in
                                        // C quickly we must misuse pointers
    for (i = 0; i <= 0x7f; i++) {
        *fi = 0;

            // Build a float with the bit pattern i as mantissa
            // and an exponent of 0, stored as 127
        *fi = (i << 16) | (127 << 23);
        f = sqrt(f);

            // Take the square root then strip the first 7 bits of
            // the mantissa into the table
        sqrttab[i] = (*fi & 0x7fffff);

            // Repeat the process, this time with an exponent of 1,
            // stored as 128
        *fi = 0;
        *fi = (i << 16) | (128 << 23);
        f = sqrt(f);
        sqrttab[i+0x80] = (*fi & 0x7fffff);
    }
}

    // fsqrt - fast square root by table lookup, original C version
float fsqrt(float n) {
    unsigned int *num = (unsigned *)&n; // to access the bits of a float in C
                                        // we must misuse pointers

    short e;        // the exponent
    if (n == 0) return (0); /* check for square root of 0 */
    e = (*num >> 23) - 127; /* get the exponent - on a SPARC the */
                            /* exponent is stored with 127 added */
    *num &= 0x7fffff;   /* leave only the mantissa */
    if (e & 0x01) *num |= 0x800000;
                /* the exponent is odd so we have to */
                /* look it up in the second half of  */
                /* the lookup table, so we set the high bit */
    e >>= 1;    /* divide the exponent by two */
                /* note that in C the shift */
                /* operators are sign preserving */
                /* for signed operands */
        // Do the table lookup, based on the quaternary mantissa,
        // then reconstruct the result back into a float
    *num = ((sqrttab[*num >> 16])) + ((e + 127) << 23);
    return(n);
}

// ------------------------------------------------------------------------
// Optimized Watcom inline version, 9-cycle? (Pentium), tested with 10.6.
// By Javier Arevalo, A.K.A. Jare/Iguana in 1996.
// E-mail me at jarevalo@ran.es for comments.

extern float VSSQ_Sqrt(float a);
#pragma aux VSSQ_Sqrt modify nomemory [EBX EDX] parm [EAX] value no8087 = \
"   MOV     EDX,EAX              " \
"   AND     EAX,0x007FFFFF       " \
"   SUB     EDX,0x3F800000       " \
"   MOV     EBX,EDX              " \
"   AND     EDX,0xFF000000       " \
"   SAR     EDX,1                " \
"   AND     EBX,0x00800000       " \
"   ADD     EDX,0x3F800000       " \
"   OR      EAX,EBX              " \
"   SHR     EAX,16               " \
"   MOV     EBX,OFFSET sqrttab   " \
"   DB      0x8B, 0x04, 0x83     " \
"   ADD     EAX,EDX              "

// Note that DB ... means MOV EAX,[4*EAX+EBX] which WC won't accept :(
// Unluckily it won't accept "MOV EAX,[4*EAX+sqrttab]" in DB either. :((
// It doesn't generate the relocation info for DD OFFSET sqrttab.
// Yes Watcom inline ASM sucks bigtime, but it's better than nothing.
// Will 2 AGIs be generated by both EBX & EAX being assigned before the
// address generation? I don't know but I can't fix this for inline code.

#include

//#define TRY(a) (printf(#a ": sqrt() == %f, fsqrt() == %f, error == %f\n",
sqrt(a), fsqrt(a), fabs(sqrt(a)-fsqrt(a))))
#define TRY(a) (printf(#a ": sqrt() == %f, fsqrt() == %f, error == %f\n",
sqrt(a), VSSQ_Sqrt(a), fabs(sqrt(a)-VSSQ_Sqrt(a))))

main() {
    build_table();
    TRY(1.0);
    TRY(2.0);
    TRY(4.0);
    TRY(1.522756);
    TRY(12341.1345);
    TRY(2346.522756);
    TRY(0.004522756);
}
```

Jyrki Alakuijala has come up with another way:

Integer SQRT seems to be a never ending thread. Here is an evolution version of an approach I presented here some two-three years ago. It is based on a different approach: generated C++ code. That means that you have to compile it and run it. Then you have to compile the result it produces to stdout and use that to calculate square roots. The method uses lookup tables of values and function pointers, fixed point piecewise linear approximation, and binary search (huh). I used to have a version that I checked to have +-1 accuracy for the whole range (0 - (2^32 - 1)). I suspect that the version I am posting has an accuracy of +-2.

The code run 3-4 times faster on my machine than the code recently presented in Dr Dobbs. However, this code has little real value. The generated code may be too big for most purposes, BUT it demonstrates a technique that can be used to model arbitrary functions very efficiently.

I cannot claim that this is "very" fast. Still, it is the fastest I have found. However, I feel that it should be possible to speed it up by 50% by using computed gotos, some clever asm tricks and, perhaps, by avoiding the integer multiplication.

BTW, make the actual function (isqrt) inline.

```
-----CLIP------
#include
#include
#include

#define FUNCNAME "i_sqrt"
#define FUNCTABLESHIFT 24
#define FUNCTABLESIZE 256
#define SMALLSQRTTABLE 256

#define NESTING() {int i; for (i = n; i--;) printf (" ");}

void muladd(double a0, double b0) {
#define SCALERA 0.3000
#define SCALERB 0.3333
  double a = (1.0 - SCALERA) * a0 + SCALERA * b0; /* hilliton kluge */
  double b = (1.0 - SCALERB) * b0 + SCALERB * a0; /* hilliton kluge */
  double mul = (sqrt(b) - sqrt(a)) / (b - a);
  int shift = (int)(- (log(mul) / log(2.0))); // round down
  int jappe = (1 << shift);
  unsigned int imul = (unsigned int)((mul * jappe * (1<<16)) + 0.5);
  if (sqrt(b0) > 65530.0) {
    printf ("{R US((%u+(((int)(val-%u)>>%d))*%d)>>16);}",
      (unsigned long)(sqrt(a) + 0.5) * (1 << 16),
      (unsigned long)(a + 0.5 - (0.4 * (1 << shift))),
      shift, imul);
  } else {
    printf ("{R US((%u+(((int)(val-%u)>>%d))*%d)>>16);}",
      (unsigned long)(sqrt(a) + 0.5) * (1 << 16),
      (unsigned long)(a + 0.5 - (0.5 * (1 << shift))),
      shift, imul);
  }
}

void recursion (double a, double b, int n) {
  if ((unsigned long)b <= (unsigned long)SMALLSQRTTABLE) {
    printf("\n");
    NESTING();
    printf("{return " FUNCNAME "SmallTable[val];}");
    return;
  }
  double midp = 0.5*a+0.5*b;
  double error = sqrt(midp) - (0.5*sqrt(a)+0.5*sqrt(b));
  if (error > 0.15) {
    printf("\n");
    NESTING(); printf("if(val<%u)", (unsigned long)((a + b) / 2.0 + 0.5));
    recursion (a, midp, n + 1);
    printf("\n");
    NESTING(); printf ("else");
    recursion (midp, b, n + 1);
  } else {
    muladd(a,b);
  }
}

void main () {
  int i, j;
  printf("unsigned char "FUNCNAME"SmallTable[] = {\n");
  for (j = 0; j < SMALLSQRTTABLE; j++) {
    printf("%d, ", (int) (sqrt(j)));
    if ((j % 16) == 15)
      printf("\n");
  }
  printf("};\n\n");
  printf("#define T(j) unsigned short j(register unsigned long val)\n");
  printf("#define R return\n");
  printf("#define UL (unsigned long)\n");
  printf("#define US (unsigned short)\n\n");

  for (i = 0; i < FUNCTABLESIZE; i++) {
    double start = i * (double)(1 << FUNCTABLESHIFT);
    double end = ((i + 1) * (double)(1 << FUNCTABLESHIFT));
    printf("\nT(%s%d){", FUNCNAME, i);
    recursion (start, end - 1.0, 1);
    printf("}");
  }
  printf("\nunsigned short (*sqrtFuncTable[])(unsigned long) = {\n");
  for (i = 0; i < FUNCTABLESIZE; i++) {
    printf("%s%d,", FUNCNAME, i);
    if ((i % 7) == 6)
      printf("\n");
  }
  printf("};\n\n"
    "unsigned short isqrt (register unsigned long val){\n"
    " return sqrtFuncTable[val >> %d](val);\n"
    "}", FUNCTABLESHIFT
  );
}
```

-----CLAP------

Have fun!

--Jyrki Alakuijala

(10.8) What's Karl/NoooN's famous 40-byte sinus generator?

From the ACE CD #2:

--(sin.txt)--

Here's some explanation about the sinus table generator in the ACE BBS advert.
The method used is a recursive sinus sythesis. It's possible to compute all sinus values with only the two previous ones and the value of cos(2ã/N) , where n is the number of values for one period.

It's as follow:

Sin(K)=2.Cos(2ã/N).Sin(K-1)-Sin(K-2)

or

Cos(K)=2.Cos(2ã/N).Cos(K-1)-Cos(K-2)

The last one is easiest to use , because the two first values of the cos table are 1 & cos(2ã/n) and with this two values you are able to build all the following...

Some simple code:

the cos table has 1024 values & ranges from -2^24 to 2^24

```
build_table:          lea    DI,cos_table
                      mov    CX,1022
                      mov    EBX,cos_table[4]
                      mov    EAX,EBX
@@calc:
                      imul   EBX
                      shrd   EAX,EDX,23
                      sub    EAX,[DI-8]
                      stosd
                      loop   @@calc

cos_table             dd     16777216     ; 2^24
                      dd     16776900     ; 2 ^24*cos(2ã/1024)
                      dd     1022 dup (?)

                             Enjoy  KarL/NoooN
```

---(sin.txt)--

(10.9) Where can I get more info on programming VESA?

http://www.vesa.org is a good place to start; you can also get info from SciTech (see FAQ 5.3).

(10.10) I'm having trouble getting a VESA 2.0 Linear Frame Buffer working... any tips?

Use the "Get Mode Info" function. It will fill in a struct, and one of the members is a physical address that is a pointer to the linear frame buffer. To access the physical address, you need to call DPMI function 800h to map it to a linear address that you can make use of. When you load UNIVBE, it should say something like linear frame buffer located at 4096 MB or something like that, which is referring to the physical address, so you can double-check with that when testing your code. Note, however, that the physical address will be different on most systems, so you will have to get it from the VESA Mode Info function.

(10.11) Where can I find a list of Intel 80x86 opcodes?

Try http://www.quantasm.com for starters.

Misc. Information

(11.0) What CDROMs exist with demo-related material on them?

For info on CDROMs with demos and demo-music files on them, email Dan Wright at dmw@gate.net to get a more-or-less complete list of them. If Dan is unreachable, you can get a copy of this list at http://www.depaul.edu/~trixter/docs/scenecds.txt

Walnut Creek CDROM published Hornet Underground Volume 1, a CDROM of all the demos on the Hornet archive as of May 1996. Hornet Undergound volume 2 is reportedly underway as well; visit http://www.cdrom.com for more info.

The following is a mini-review of some of Assembly Organizing's party CD's submitted by Hugo Habets:

> On the cover of the CD-box is a picture of a monster and a boy. The monster is crying out and the boy is crying. At the top we can read "This CD is full of the best demos, music and graphics for PC and Amiga". At the bottom we can read (in bigger letters) "an ASSEMBLY Organizing production". At the back of the cover it's the same picture as on the front of the cover, only now it's a bit red-like with some text in small font and 6 pictures from the compos covering it.
>
> Very suprisingly, the first of those 6 pictures again is the picture of the monster with the boy. I don't recognize the second picture (not much time to search for it on the CD).. the third picture is from the CD menu (the Party 4 CD menu is very much like the Asm'94 CD menu, but with some improvements). The 4th picture shows the sea (probably from the graphics compo), the 5th picture is from the Valhalla intro. The last picture shows a face, a chess-board, mountains and a tree and is probably from the graphics compo also. The publisher seems to be 'Romware'
>
> The front picture can be taken out of the CD cover and then magicly turns into a little book. Most of it is about the menu (or the GUI as it is called in the book).

page@ie2.u-psud.fr reminded me of this:

```
> I know another Audio CD :
>    Dr Awesome and Fleshbrain's "hoBbiTs & SpACesHipS"
>    by Bjorn A. Lynne and Seppo Hurme
>    from DENS DESIGN/CRUSADERS and Audio Visual Magic
>    (c) 1992
> with 12 songs and a time of 71'02".
>
> The ACE Demo Collection Vol. 1 & 2 costs 99 FF.
> And the number 3 will arrive in december 1995, and the 4th in july 1996.
```

Rene V. C. (Zteel of Diffusion) also had this to add:

```
>Here are some of the ones I can remember.
>
>Assembly '94
>The Party '94
>ACE #1
>ACE #2
>
>Btw: for info on the ACE cd's, write to yvon@sept.fr
>
>Anyway, the 'The Party CD' costed 120FIM at Assembly, filled with all
>releases from The Party 1994, and the best from Abduction and Juhla I
>think it was. The Assembly CD, well, has everything from Assembly
>1994, along with a bunch of movies and animations.
>
>The ACE's costs aprox. 80F .. (french franc)
```

(11.1) I want to create my own demo or demo-music CDROM... What's involved?

Usually, most demo coders don't want their productions put on a CDROM without their permission, because they feel that the person selling the CDROM's makes profit off of their work. So, the only true demo CDROMs that have been produced so far are by Dan Wright (dmw@gate.net), who manufactures and sells them -- while all the time *breaking even*. This is important: He doesn't lose money, but doesn't make any money either, so no one feels taken advantage of. What a cool guy. :-)

Dan Wright wrote an excellent article on creating your own CDROM; this article is part of the "demobook" on the Freedom CD. You can also email him to get a copy.

As to how to create an audio CD of mod-style music, Dan Wright offers:

> "We took the modules and pumped them through the Gravis Ultrasound...grounded the computer to the stereo, funneled the sound through a "DSP" that added reverb(hall), flange, and all that other stuff I got a few complaints on. From there the signal made its way through an equalizer, then the master receiver and finally to the SONY DAT player. Once the DAT was made I paid someone $175 to create a CD-R with all the time stamping and music equalization."

Another way to do it is to just use the "direct-to-disk" mode of Mod4Win, or the "Disk Writer" device of Cubic Player. Instead of playing the module, these methods write out a 44.1KHz 16-bit stereo .WAV file of the song, which is the format audio CDs use. Many CD-ROM writer software programs take .wav files and burn them directly into CD audio tracks.

(11.2) What are some of the more interesting quotes said in the demoscene?

Trixter maintains (more or less :-) a list of interesting things heard in the PC demo scene. You can get this list from http://www.depaul.edu/~trixter/docs/demoquot.txt

----

(11.3) What are some of the hidden parts in demos?

Phoenix / KFMF (vossa@rpi.edu) has compiled a list of hidden parts in demos. For the most recent version of the list, request it from Phoenix directly, or view it on the WWW at http://www.rpi.edu/~vossa/secret.txt.

(11.4) Where did the "I am not an Atomic Playboy" sample come from?

(This sample is used in a part of Second Reality's soundtrack composed by Purple Motion.)

There are many theories, actually. One is that the words "I am not an Atomic Playboy" come from the song "Atomic Playboys" by the American rock group called Atomic Playboys (don't ask). The band was fronted by Steve Stevens, the guitarist from the Michael Jackson video "Diry Diana". (Stevens was also responsible for the excellent Top Gun anthema with Harold Faltermayer from the Top Gun soundtrack.)

Another is that the "I am not an Atomic Playboy" sample was from a speech by Richard Nixon (at one time, president of the US).

Yet a third was suggested by IOR / Hornet:

> I don't think it was Nixon, but I can tell you exactly where the first use of "I am not an atomic playboy" was... or at least where you can find it now. And it's not from an American rock group. Or from a techno song.
>
> It's from a documentary on Nuclear Proliferation and Nuclear Tests on Binkini Atoll, called "Radio Bikini"... The entire quote is this:
>
> "I am not an atomic playboy, as my critics have called me."
>
> Of course, I doubt that PM took the sample from Radio Bikini, but regardless, that's the first place it showed up.

Of course, I don't think that IOR thought about the possibility that the speech from Radio Bikini was Richard Nixon. :-)

---

(11.5) Where did "Is everybody in?" sample come from?

The sample "Is everybody in? Is everybody in? The ceremony is about to begin." used in Verses / EMF came from the movie "The Doors" starring Val Kilmer.

---

(11.6) Are there really 23,800 faces in the bee object in Stars / NoooN?

Aumury Aubel speculates:

Nooon's bee is optimised to hell, that's for sure. And it is very likely the original 3DS model is made up of 23,800 faces. How did they manage though?

As someone pointed out, the bee moves along a single axis. That reduces drastically the number of multiplications. And obviously there are not 23,800 faces simultaneously displayed...

It doesn't move a lot around, which means it is pretty easy to get away with the hidden face test. In my opinion, there's no test at all. I've seen this done by some friends of mine and it works perfectly:)

There is also a wide-spread rumour saying that they didn't actually compute all the bee's positions: every eighth frame and used linear interpolation inbetween. That *could* work but I haven't tried it myself, nor have the people I know. I've heard someone say this had been done in an old amiga demo though.

The third possible trick is to avoid any divide. Who said you need to perform divides to perspective project a vector? A divide is nothing but a multiply. Think it over:)

An even beter solution is to remove all multiply operations as well. This is feasable, I've seen it done folks! And it runs a lot faster and without any large precalculation tables...Nooon might have done something similar even though I doubt it.

Once you removed the rotation/projection part and the hidden faces test, what remains to be optimized? the rendering...Nooon's rendering algorithm is thoroughly optimized. Just check out their various toruses:)
But even so, I bet that it's the part on which MOST OF THE TIME IS SPENT.

Finally, they may have used some other tricks. Who said they use actual vectors. The bee's wings could be easily replaced by sprites...Sure it would not be very noticeable...And keep in mind it's a demo, in other words, it is

NOT actual real-time.

Last but not least I've met Karl twice and he's a damn good coder. That accounts for most of their superb effects!

Amaury.

(11.7) How many people read comp.sys.ibm.pc.demos?

Makrus was surfing one day and found out the statistics:

I was just surfing and I found a site (tile.net) which contains statistics about different newsgroups. This is what was said about c.s.i.p.d:

"Actual readers of this group: 29000
 Average number of messages per day: 30
 Kilobytes per day: 163k
 Percentage of Internet sites who receive this group: 70%
 Crossposting: 13%"

Personally I was amazed by the number of readers (it may be very inaccurate however!). What do you think (is 29000 more than you've ever imagined?) about this? And what could be the number of demosceners then (all sceners don't read the news)?

And just for comparison I checked the statistics of alt.sys.amiga.demos:
11000 / 6 / 10K / 56% / 10%
--
Makrus
Abd'96 MO

While I think 29,000 is a bit high, I wouldn't be surprised if the number was over 10,000.

Contact Information:

(100.0) Who's contributed to this FAQ?

The creator was Houman Ghahremanlou, who wrote answers to questions 1.0 to 3.0, with some small help from Trixter / Hornet's PC Demos Explained page. Trixter then reformatted the FAQ, added the rest, became the maintainer, and continues to add to it.

Once released, many people submitted some contributions. Trixter would like to thank the following people for submitting to the FAQ, or to comp.sys.ibm.pc.demos with valuable information that has been included:

"Luc-Eric Rousseau" (l.e.rousseau@usa.net)
"RC Hoeft" (rchoefz1@ulkyvm.louisville.edu)
"Warren E. Downs" (downwa@wwc.edu)
Alain Chardonnens <100124.1007@compuserve.com>
Amaury AUBEL (amaury@cln46fw.der.edf.fr)
Andréas Kühne (andreas.kuhne@mailbox.swipnet.se)
Anssi.Saari@lmf.eua.ericsson.se (Anssi Saari)
Bastiaan_Zapf@p77.secobox.leine.de (Bastiaan Zapf)
Ben Shelton (diablo@wcarchive.cdrom.com)
Blake Kadatz (blake_kadatz@mindlink.bc.ca)
CMSLHES1@vaxc.livjm.ac.uk (Liam the lemming)
Canard@ax.com (Canard)
Captain Hook / THI (pjscorreia@telepac.pt)

Charles Scheffold (daredevi@amanda.dorsai.org)
Chris Hargrove (kiwidog@vt.edu)
Christian Cohnen (cohnen@uni-koblenz.de)
Cobra@aloha.com (cobra)
Dan Wright... enough said. :-)
Diogo 'Spellcaster' Andrade (l42686@alfa.ist.utl.pt)
Frans Bouma (perseus@xs4all.nl)
GD / Hornet (gd@ftp.hornet.org)
Gary D Stowasser (gary+@andrew.cmu.edu)
Hamid Moazed (hamid@rs.com)
Heikki Ylinen (flap / Capacala) (flap@flap.pp.fi)
Houman Ghahremanlou (ghahrema@bowler.dacc.wisc.edu)
Jason Maas (clueless@WPI.EDU)
Jason Nunn (root@superr.topend.com.au)
Jesse Rothenberg (jroth@mailhost2.csusm.edu)
Johannes Lehtinen (jle@cs.hut.fi)
Jordan Phillips (jordanp@cent.com)
Jyrki Saarinen (jsaarinen@kone.fipnet.fi)
Kemal Bayram (omeec@westminster.ac.uk)
Kenneth Foo Chuan Khit (kenfoo@techm.pl.my)
Kilian Hekhuis (jal@hacom.nl)
Kim Davies (kimba@it.com.au)
Lars Troen (Lars.Troen@colargol.idb.hist.no)
Lee Chun Kwok (h9504367@hkueee.hku.hk)
Lewis Berrie (hd66@dial.pipex.com)
Lewis Sellers (bcannon@usit.net)
Luca D'Ambros (micro@comedia.it)
Mark Edward Hardwidge (hardwidg@tvd0002.urh.uiuc.edu)
Michael Seow (Eaglehawk@c031.aone.net.au)
Morten Eriksen (MORTENER@sofus.hiof.no)
Niklas Kring
Paul Bragiel (bragiel@students.uiuc.edu)
Paul Hsieh (qed@xenon.chromatic.com)
Pekka Aakko (pehu@icon.fi)
Peter Kendell (P.Kendell@bra0119.wins.icl.co.uk)
Petteri.Kangaslampi@research.nokia.com (Petteri Kangaslampi)
Phil Jones (fil@muon.demon.co.uk)
Praveen \"Ripclaw\" Reddy (praveen@elf.udw.ac.za)
Quantum Porcupine (jshagam@nmsu.edu)
Renaud Guerin (renaudg@club-internet.fr)
Rene Vinding Christensen (rvc@vision.auc.dk)
S M Carless (S.M.Carless@durham.ac.uk)
Sam (Samuel.Marshall@durham.ac.uk)
Samuli Syvahuoko (gore@clinet.fi)
Sietze Dijkstra (sdijkstr@cs.ruu.nl)
Simon Bostock (Simon@simjohn.demon.co.uk)
Sleeping Dog / The Natives
Snowman / Hornet (r3cgm@hornet.org)
Stephen Takacs (takacs@eng.usf.edu)
Sumaleth@starfury.apana.org.au (Rowan Crawford)
T H Pineapple (thp@cix.compulink.co.uk)
THE ROPESTER (mjfi3sjr@umist.ac.uk)
TORNERO@LAMBDA.UPC.ES (TORNERO GARCIA, JOSE)
Tero Pulkkinen (terop@kotka.cs.tut.fi)
Todd M Zimnoch (tz26+@andrew.cmu.edu)
Toni Lindroos (tonilind@netti.fi)
Totty (totty@mi.net)
XtaC (jwalther@smartt.com (XtaC))
ac@interaccess.com (Andrew Carlson)
adia@egnatia.ee.auth.gr (Alejandros Diamandidis)
alang012 (dsteg838@student2.uwsp.edu)
ammo@xs4all.nl (ammo)
andreas.kuhne@mailbox.swipnet.se (Andréas Kühne)
aronchce@sp.zrz.TU-Berlin.DE (AiRoN)

```
aschlud@autelca.ascom.ch (Denis Schluchter)
azure@people-s.people.de (Tim Boescke)
beppu@rigel.oac.uci.edu (John Beppu)
beren@infolink.no (Niklas Saers)
blake@widomaker.com (Blake Patterson)
bq689@freenet.carleton.ca (Anis Ahmad)
centero@mail.ddnet.es
chaos@wiloyee.shnet.org
chuck@freeside.fc.net (Chuck Walbourn)
clef@suburbia.net (Phil Sweeney)
dagsm@infolink.no (Finrod / Ewox)
daredevi@dorsai.org (Charles Scheffold)
davidm@them.com (David Mandala)
deathstr@singnet.com.sg (Rex Guo Yiwei)
dennisc@community.net (dennis courtney)
devine@cs.ualberta.ca (Michael Devine)
devries@cam.org (Mike DeVries)
dhk_fj@p10.nix.fido.teuto.de (Christian Kram)
dmw@gate.net (Out There!)
dominion@ripco.com (Michael Chisari)
ekallion@cc.Helsinki.FI (Esa J Kallioniemi)
fc@fee.uva.nl (Frank Compagner)
fischerj@Informatik.TU-Muenchen.DE (Juergen Fischer)
fmah@morse.ecn.purdue.edu (Frederick Y Mah)
francois.baligant@ping.be (Francois Baligant)
fuller (fuller@hap.arnold.af.mil)
fuzz@ionline.net (Arlo Gingerich)
gberigan@cse.unl.edu (Greg Berigan)
gerald@parker.EECS.Berkeley.EDU (THE Gerald)
ghahrema@bowler.dacc.wisc.edu (Houman Ghahremanlou)
grosje_s@epita.fr (Le Fongus jaune)
gruel@hondo.cyberverse.com (Nick)
gt4148b@prism.gatech.edu (Stephen Carter Morgan)
habets@worldaccess.nl
hb@cafu.fl.net.au (Nicholas Vinen)
ithomson@mortimer.com (Ian Thomson)
j.fenkes@public.ndh.com (Joachim Fenkes)
jakarppi@raita.oulu.fi (Jari Karppinen)
jarevalo@sip.es
jarno heikkinen (jmag@cOmPLeX.math.jyu.fi)
jean-marc.leang@ping.be (JML)
jeanmarc.leang@ping.be (marmelade)
jim@kd3bj.ampr.org (Jim Paris)
jisidoro@acs.bu.edu
joemess@mail.utexas.edu (helpless boy)
jroth@coyote.csusm.edu (Jesse Rothenberg)
jsno@amigar.apana.org.au
jth@jth.ping.de (Jens Theisen)
jukkak@dns.mikrobitti.fi (Jukka O Kauppinen)
kimmy@planet.fi (Kimmy/PULP)
kiwidog@mail.vt.edu (Chris Hargrove)
larsen@lal.cs.utah.edu (Steve Larsen)
larstr@colargol.idb.hist.no (Lars Troen)
lcs@lysator.liu.se (Martin Blom)
lodder@cuci.nl
matthewp@netcom.com (Matt Pritchard)
maurala@cc.hut.fi (Markus Aurala)
mdaniel@zeus.polsl.gliwice.pl (Milosz Danielewski)
millen3@alum01.its.rpi.edu (Neal W. Miller)
mithril@ict.org (Jeremy Tavan)
mithril@ng.netgate.net (Jeremy Tavan)
more@stekt.oulu.fi (Jyrki Alakuijala)
mrp@spartan.pei.edu (Mr.P / Powersource)
mrytkola@tor.abo.fi (Markus Rytk|l{ INF)
```

mud@merlin.sedona.net (Justin Frankel)
mystical@inet.uni-c.dk (Asbjorn Andersen)
page@ie2.u-psud.fr
perseus@xs1.xs4all.nl
plexus@plexus.seanet.com (James B. Johnson)
porat@ibm.net (Hand of Fate)
ppsloan@buzzworm.cs.utah.edu (Peter Sloan)
prsam1@MFS02.cc.monash.edu.au (Paul Sampson)
rawvibes@ix.netcom.com
rbarnhar@freenet.niagara.com (Robert Barnhardt)
rbeath@julian.uwo.ca (Stephen Beath)
rcskb@minyos.xx.rmit.EDU.AU (Kendall Bennett)
resimmon@students.uiuc.edu (simmons russel evan)
rimbo@ccwf.cc.utexas.edu
rob@span.com
rolando@knoware.nl (Scout/SuccesS)
rvc@vision.auc.dk
ryan.mahoney@tssbbs.com (Ryan Mahoney)
ryston@login.cz (Martin Pilny)
s106275@cs.tut.fi (Anssi Saari)
s9106065@student.utwente.nl (Sparcus / Nostalgia)
sci-slb@groper.jcu.edu.au (Stephen Banhuk)
slmyv@paradise.declab.usu.edu (Denys Larry)
smh@europa.com (smh)
src@cray.tuug.utu.fi (Saracen / EMF)
sschaem@teleport.com (Stephan Schaem)
sshah@intranet.ca
tedjones@voyager.co.nz (Oliver Jones)
then@superpallo.cs.hut.fi (Tomi Holger Engdahl)
thomasep@funcom.com (Thomas Egeskov Petersen)
tom@halls1.cc.monash.edu.au (Tom PATON - Tom)
trixter@hornet.org (Trixter / Hornet)
tst@dcs.ed.ac.uk (Tristan Tarrant)
tw@wile.thetech.org (Tod Weitzel)
vossa@matisse.its.rpi.edu (xproject)
whippet1@quiknet.com (Scott Tyson)
yrmafa@utu.fi (Yrj| Fager)
yvon@sept.fr (christophe yvon)
zsazS (kfpeters@artsci.wustl.edu)

(100.1) How can I contact you to submit something?

Anything that makes this FAQ bigger and better is welcome, so please mail me your suggestions at trixter@hornet.org

----

FAQ Greets:

Respect goes to Dan Wright, who started the whole "demos on the Internet" thing in 1992. Of course, thanks go to Houman for starting this thing; greets also to Snowman, for being a good friend and bringing me back into the scene when I was going to quit in 1995. Finally, thanks to all past and present comp.sys.ibm.pc.demos members and lurkers.