



## No pain, no gain

**Vibrant 3D images can cause file management headaches. Benjamin Woolley looks at ways of dulling the pain, and dips a toe in the water of the 3D interface with DIR3D.**

This spring, Silicon Graphics announced its development environment for creating Web content, "Cosmo". Among the suite of fabulously sophisticated and glamorous tools (including tools for integrating VRML and Java) was a rather uninteresting-looking fellow called "MediaBase". All it apparently did was help you organise your files. What a dull job.

Well, unfortunately, that dull job turns out to be one of the most important in generating any sort of media-rich content and this applies, squared, to 3D. Think of it as the bureaucracy of beauty (if that is not too tortuous): to get those wonderful, colourful, incandescent, textured images, you will need a lot of files — and you will need to know where they all are and what to do with them.

I have yet to encounter a 3D graphics

package, or even a utility for the PC, that takes all the pain out of file management (if anyone knows better, I would love to hear from them). But I would easily rate dear old DOS-based 3D Studio (3DS) as one of the worst.

When you install 3DS on your hard drive, it creates a series of directories for each of the constituents that are likely to make up a 3D project: meshes (the actual geometries for 3D models), materials (for materials libraries), lofts and shapes (both for building 3D models out of 2D shapes), fonts, images, maps (for texture maps, though sometimes these are to be found in the images sub-directory), processes (containing the IPAS routines, or "plug-ins" as they are better known in the rest of the graphics universe) — the list scrolls on forever.

Supposedly, saving a file as a "project" overcomes the problem of having to deal with all these different file types (materials, 2D shapes and other elements are stored in the one .PRJ file), but you still have to remember where all the texture maps are, the lofts and shapes you may have used in some earlier version of the project, and the clip models you may want to merge into the scene. Also, any large project is likely to comprise a number of smaller ones merged together.

Windows packages (such as TrueSpace, Visual Reality, Extreme 3D and Ray Dream Studio) overcome a few of these problems because you are better integrated, with a friendlier operating environment. You have the Registry and Explorer on hand to help. Also, with Windows 95 and NT, you can use long file names — an advantage you should exploit to the utmost.

### Avoiding problems

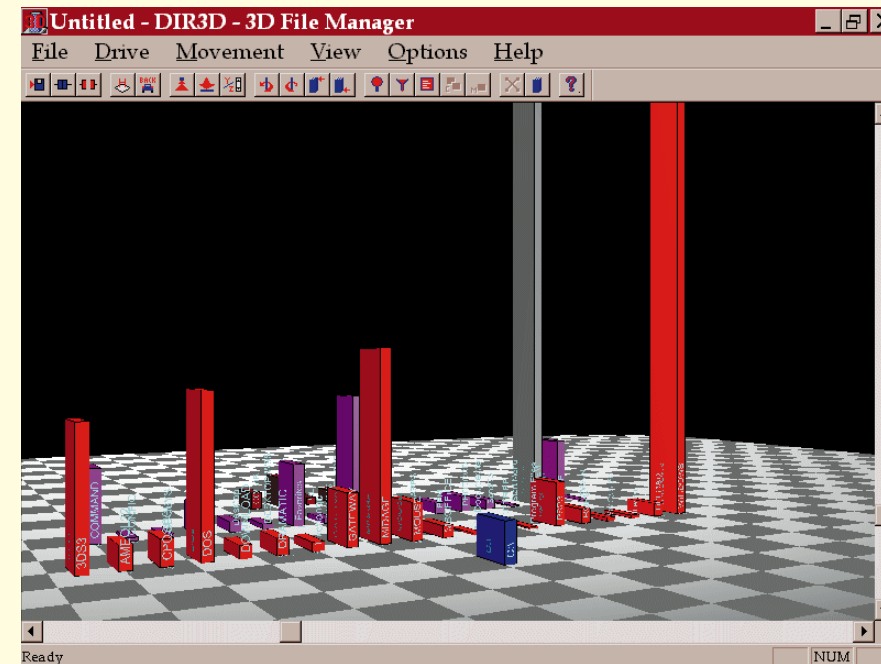
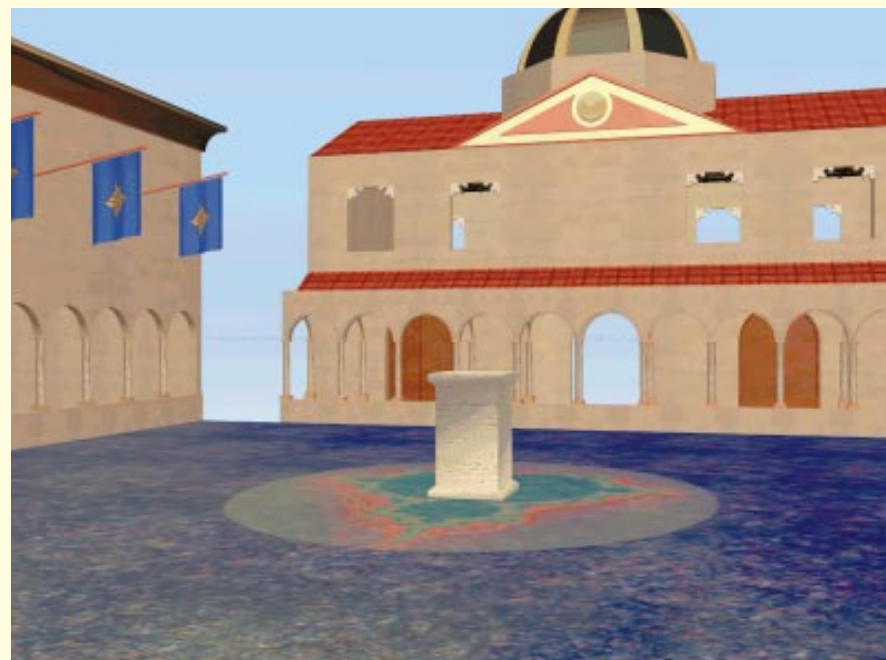
Nevertheless, no matter how disciplined you are, problems will still arise, so here are some ways of avoiding them.

Firstly, there is the obvious trick of creating a single sub-directory for each project. However, it is often better to add some sort of structure to this directory, so texture maps are in a sub sub-directory called "maps", and so on.

Secondly, there is the less obvious trick of trying to work out in advance what types of files you will need to use. This depends on the sort of package you are using. Are there separate formats for 2D geometry, for example? Do texture files need to be converted into a particular proprietary format (as they do in Extreme 3D)? Can you make the conversions in advance and do you need to keep the originals? The answer to the latter is yes, if you cannot reconver.

Thirdly, you need to think about texture and bump maps. These files, which will be bitmaps (in some cases, including video sequences and animations) are the ones that cause the most problems. The reason? They are often huge and you are

*It took a total of 23 separate files to make up this scene, including some project files for individual elements (e.g. the flags), more project files to render up textures (the stars on the flag material), images generated by other programs (the Mandelbrot used in the floor material, from a fractal generator), texture maps from clip libraries (the marble finishes), 3D mesh files containing objects used to cut out the arch shapes, and so on*



*The Manhattan skyline, as produced by DIR3D's view of my local hard disk drive. The two World Trade Center towers are, inevitably, my overloaded Windows and System sub-directories: see how they dwarf the 3D Studio sub-directory on the left of the picture. The toolbar contains navigation buttons*

likely to be getting them from all over the place: from a clipart library, off the Internet, from your own image directories, from scans and from other rendering projects.

Worse, you will often accumulate several versions: a high-resolution colour version for the texture map, a 32-bit version for Alpha channel data, a low-resolution greyscale version for the bump map, maybe a traced version to form the basis of a 2D shape for lofting. You have to discipline yourself to performing a regular cull of these files, printing out (or writing down) scene details for each project so you know what you have used.

A good tip is to keep an offline backup (on tape, say) called something like "originals" where you lodge one good high-res copy of every image you use, when you first use it. Then you can afford to delete online files when you reckon you have no further use for them.

It is tempting to believe that such problems will not arise as long as you bung all the required files into your new project sub-directory and sort out the mess later (my usual strategy). If you have a spare gigabyte or two of disk space this might work, but in the real world you will soon find yourself having to make room for new materials the whole time, deleting and moving files on the fly, hoping you have

kept all you need but never quite knowing whether that "bricktmp.bmp" was really just a temporary scratch file for the brick surface you used in an earlier version of the project, or the one you ended up using.

Unfortunately, being an anal-retentive is the only solution to the file organisation problem until some clever company produces a version of Cosmo MediaBase for the PC market. It is not a very glamorous product category at a time when everyone is wanting to be the next Netscape, so don't hold your breath.

### Beyond the GUI

When Alan Kay and his cohorts at Xerox's Palo Alto Research Center came up with the design for the graphical user interface, it was but a short logical step from a two-dimensional space (a "desktop") into a three-dimensional space.

Researchers at PARC itself have toyed with this idea, producing proposals for what they called the "Information Visualizer... a user interface paradigm that goes beyond the desktop metaphor to exploit the emerging generation of graphical personal computers and to support the emerging application demand to retrieve, store, manipulate, and understand large amounts of information."

How, then, would one go "beyond the desktop metaphor"? You could have a 3D representation of an office with a 3D desktop, a 3D filing cabinet with 3D drawers full of 3D files, a 3D waste paper basket (wow!) and, down the corridor, doors leading into the 3D "offices" of other users in your network neighbourhood.

You could, borrowing from the metaphor used in the interfaces such as

Apple's eWorld online service, have a door leading out into a street with buildings representing different services: point and click at the library, and you are offered a series of information services; point at the bank and you get financial services, and so on. It sounds quite seductive but so far nothing much has come of the idea. Nevertheless, things may be about to change.

One modest first step into the realm of the 3D interface is DIR3D, a Beta version of which I downloaded from the Web site of the program's authors, Regnoc (www.regnoc.com). It's nothing more than a version of the Windows 95 Explorer or 3.1/NT File Manager, in which the contents of local and network drives are represented as a 3D bar chart with the height of the bars showing the size of the directory.

Regnoc prefers a more glamorous urban metaphor in its description of the program, calling each directory a "building", and each file within it known as "floors". Hierarchy (the relationship of directory to sub-directory, to sub sub-directory and so on) is represented by the z axis: the root directory is at the front of the scene, the next level of directories behind it, their sub-directories behind them.

You use DIR3D by moving around the city, finding the building (i.e. directory) and then the floor (i.e. file) you want. When you click on the floor, it slides out — the urban metaphor is beginning to collapse here. This selected floor can then be subjected to any of the usual file operations that you would use with Explorer: copy, move and delete. You can right-click on the floor to get the associated file's properties and run it (assuming the file type is registered).

As implemented in DIR3D, the 3D interface idea seems to be little advanced but the program demonstrates a couple of interesting things. First, it shows a potential use for OpenGL, the 3D renderer built in to Windows 95 and NT. OpenGL works efficiently on Pentium systems, enabling programs like DIR3D to create pretty solid-looking 3D environments on the fly (not Doom standards, but that will come). Secondly, DIR3D suggests some possible ways of using VRML.

As most people now know, Microsoft is planning to integrate Web browsing into Windows 95. It might be possible to integrate 3D browsing too, so the interface to your system could be a VRML scene populated with 3D shortcuts to local files as well as remote resources.

### PCW Contacts

**Benjamin Woolley**, writer and broadcaster, can be contacted at [woolley@illumin.co.uk](mailto:woolley@illumin.co.uk). His home page is [www.illumin.co.uk/woolley/](http://www.illumin.co.uk/woolley/)