

# So you think you want a Dev Box?



Interested in the new Amiga Developer's box but not sure what it is all about? Read on...

If you read last month's Amiga Active, you'll know a little bit about the developer's box, officially called the Amiga Reference Platform, but more commonly known as the devbox. However, if you want to know more about the use and purpose of the machine rather than the specs of the hardware, read on...

It is common for development for a software platform to be done on another platform. This may seem odd at first, but it makes a lot of sense. Obviously if you are writing for the Dreamcast, the Gameboy or the control software for a DVD player you aren't going to write it in a "self-hosted development environment" - in other words the same environment you plan on running the software under - because those machines simply don't have the ability. Thus you would develop on some other hardware

**Below:** Some of the AVE functions - text gadgets, a transparent blend window, font engine, the windowless clock, and a rather nice boingball that you can pick up with the mouse and throw around.

such as a Sun workstation, a Dec Alpha box or a PC, running Solaris, NT, Linux or something similar. The problem with this is that it locks out the end user, and thus means the platform will only be developed for by companies willing to pay the considerable sums of money development systems cost.

## Desktop advantage

One real advantage for a desktop solution is that it has the facilities to provide a self-hosted environment. There is less for the developer to learn, it makes the user experience more familiar to the developer, and perhaps most importantly it delivers a development environment that doesn't cost the earth.

You could call the Amiga devbox a semi-self-hosted environment. The final target of developers for the new Amiga will not be some specific box, but the Elate virtual processor, which will allow all next generation Amiga software to run on whatever hardware it pleases. The native environment is not really far enough along the desktop route yet to provide an ideal development environment on its own. However as the native target is a virtual processor, it can be run on top of Linux, providing the benefits of a self hosted environment - notably the low cost.

When Sony announced their Playstation 2 dev kit, plenty of people drooled over the casework, but no Playstation fans went out and bought them - at almost \$20k a pop they aren't intended for anyone who doesn't think they'll repay the costs and much

"...it delivers a development environment that doesn't cost the earth."

"One thing that really needs to be stressed about the devbox is that it is a developer's box."

more by selling cartloads of software. Elate's totally hardware agnostic approach means that developer's boxes can be pretty much anything. This is partly why Amiga and Tao are a such good partnership - the defining thing about the Amiga is how easy and welcoming it is to developers, and the degree to which it encourages new developers to give it a try. Using Elate, Amiga are able to offer an 'everything you need' development kit for way under the cost of a Playstation 2 devkit. Prices aren't fixed, but we are looking at the "cheap PC" price range.

Environment). This is the part of Elate that handles graphical and audio events, including GUI elements. It can be used to create custom interfaces or it can be used to create desktop environments, or a whole lot more. While most of the parts you need to make a desktop OS are in place, they aren't exactly in order. When you boot Elate up, you don't get a desktop OS, just a CLI. You can use this to launch AVE applications, but you'll be doing this from the shell not the desktop. This ain't a fully fledged computer yet.

## A word in your shell-like

Now you wouldn't want to use it for every day use, but you don't need to be hugely techie to get things going under Elate. For example, to launch an AVE instance with Quake running in it, you'd open a Linux shell, go to the Elate home directory and type:

```
sys/platform/  
linux/elate
```

This will transform your Linux shell into an Elate shell. This is a reasonably solid, well-featured shell, functionally similar to a ZSH UNIX shell. This shell exists within Elate's own filesystem, which is created

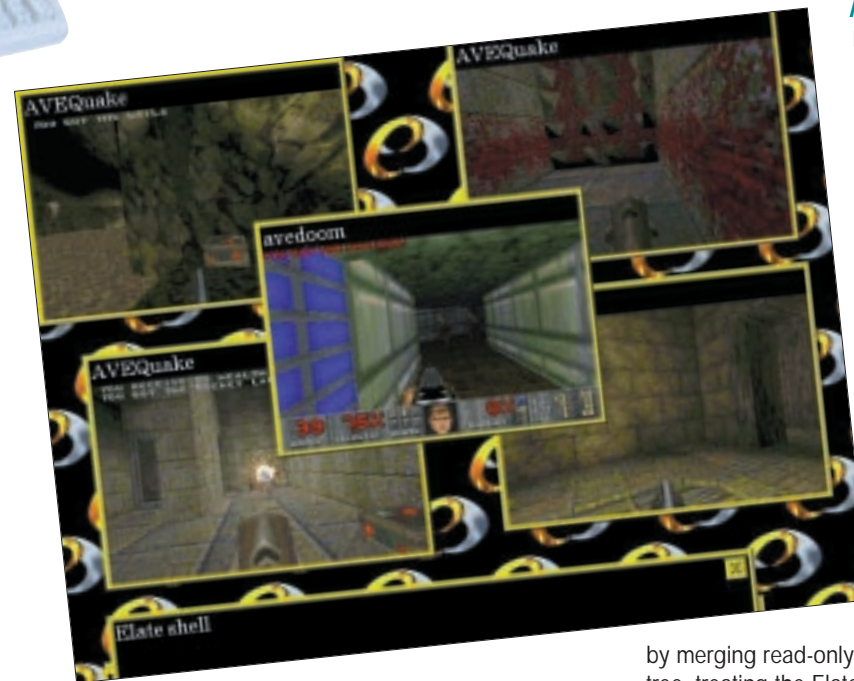
by merging read-only archives with the native OS file tree, treating the Elate directory in the host filesystem as root.

From here, to launch the Quake executable, you need to type the full path in (Yes shell fans, that is case-sensitive, miss out the capitals and it won't work. Yuck):

```
WinQuake/releaseelate/bin/avequake.00
```

This executable calls for an Elate window, so the window will pop open and Quake will start running on a window within that. An alternative executable, grquake.00, launches the game into it's own Window on the Linux desktop. If you didn't know better, you might think it was native Linux Quake.

"If you didn't know better, you might think it was native Linux Quake..."



**Above:** Quake isn't written in the most Elate friendly manner, but it can still run it four times simultaneously - and manage a great game of Doom at the same time.

## Not quite a desktop...

OK, so no Playstation fan is going to rush out and buy a PS2 devkit. But you're an Amiga fan, and that's different (OK, so plenty of you are probably Playstation fans too, but you get my point). Actually, it's not all that different - yet, at any rate.

One thing that really needs to be stressed about the devbox is that it is a developer's box. If you're not a developer, you probably don't want it. You might be thinking that you aren't a developer but you do like messing with computers, but be warned that messing with Elate is not like messing with AmigaOS at the moment. You'll have seen various screenshots of what looks like a desktop environment on these pages and elsewhere, but look more carefully and you'll see that there are no drive or file icons....

What you are looking at is the AVE (Audio Visual

## What you need

The devbox provides an all-in-one solution to developing for Amiga/Elate. The real advantage is in simplicity and support; you know it is going to work, and you know the hardware drivers are available. If you already own a Linux box you could, of course, install Elate over the top.

The base specs of the devbox are a 500MHz AMD k6-2 CPU with an AGP2 socket 7 motherboard, nVidia GeForce graphics card (although we understand this may change) and a Sounblaster 16 to 128 sound card. It will of course work with other machines - there's no great reason why you shouldn't be able to run on a 1GHz Athlon, for example. Multimedia drivers will remain an issue, but support for alternatives should appear.

The devbox comes with the latest version of RedHat Linux, with the Enlightenment window manager. It's a system which should be fairly comfortable for Amiga users, although if you prefer something else, the developer's system can run on other things. It'll run on Windows, if at an even greater speed penalty than under Linux, and there's no reason why it shouldn't run on other platforms as well, including Amiga PPC. Elate keeps as much as possible in virtual space rather than in native code, so many device drivers and even the compilers will run under Elate on any host (or, in theory, natively). In practise it would take a certain amount of time and effort to get things working on other systems, but where they are already in the works, pre-existing or in demand, development is tied to Elate, not Linux.

There will be devboxes from more than one source - Eyetech have already announced theirs. The Software Development Kit will also be sold separately on CD. We will bring you more information on each alternative as it becomes available.





### Using the box

A demo of Quake is one of the surprise packages you'll find inside the devbox. It, along with Doom, was roughly recompiled to VP (Virtual Processor) code by one of Tao Group's employees. Other than those, there isn't a whole lot to play with. However, Elate does boast one critical feature that does change things somewhat - it is a very impressive Java engine.

You could actually get real use of a devbox today by running Java applications on it. There are Java web browsers, Java graphics applications, Java games, Java business software... providing it

conforms to the Sun PersonalJava specifications, it should run on Elate, and run rather well (although note that PersonalJava doesn't cover everything). However, you're still stuck with the lack of a complete interface. The OS just isn't ready for general desktop use yet.

Well, you may still be tempted. As Amiga work at implementing the user interface stuff, there will be more and more of the complete system available to devbox owners, and it will get more and more usable over time. In the meantime, if you're not a developer already, you might want to consider becoming one.

I think that there is every chance that the thing that will really make Amiga is Java. Offering the world a Java that actually works the way it should, is stable, fast and lean, is an offer the world will find difficult to refuse. Java is actually a great base for the new Amiga - while we'll always want the option to code for speed, for many purposes Java is an ideal language. It is easy to learn and very quick to develop in.

It may not be AMOS, but it could do a similar thing for the Amiga. Developing Java applications is a lot easier than developing in C/C++ - if a little harder than basic. It allows you to write software that can be appreciated by the widest audience, and it runs best on an Amiga. What more could the amateur, student or small application programmer ask for?

### Develop your own

You don't need a devbox to develop Java applications, of course. Pretty much anything will do, even if your test target is rather slower than the devbox would have been. You can even write and compile Java on the Amiga, although the lack of a Java Virtual Machine means that you'll have to find a different machine to test it. Most developers are going to want to work in a development environment such as Jbuilder or even GoldEd anyway, but if you don't have a machine running a JVM, this wouldn't be a bad way to go.

For more heavyweight developers and applications, the devbox comes with all you need to develop C/C++ or VP assembly code as well. C and C++ code can be compiled from a command line GCC derivative compiler into VP source code, which is then assembled into the native VP binary format. Similarly, there is an assembler for dealing with VP code, which is a machine code assembly language with some almost C like abstractions. The assembler can handle both VP and native code compilation; this allows the developer to offer platform-specific tools where unique hardware might offer a benefit. Thus if Elate doesn't support certain CPU extensions, they can still be supported by working through the translation layer direct to the local hardware. Elate binaries use an identifier to select the target; binaries ending .00 are native VP, other codes indicate other, "real world" CPU targets.

### Complete package

There is pretty much whatever else you would expect in a package like this. You get a streamlined editor derived from Emacs, debugging tools, and most useful of all, documentation. A considerable amount of basic introductory material and reference material is supplied in PDF format for easy printing or on-screen viewing, and there is a wealth of help files in HTML format scattered throughout the Elate install and indexed together for easy reference. If not quite a beginner's guide, it does provide a thorough, ground-up reference.

The most important thing to keep in mind about the developer's box is that it's a device for the early adopter amongst the development community. There will be some problems developing for a system which is not yet complete and there are APIs yet to implement. However, the inherent nature of Elate encourages a modular, object approach to software development which will make these issues less of a problem than they might be in another system. The important thing is that it is enough to get started, to get to understand the nature of Elate, and to get a head start on the rest of the world. Elate is going to be a very powerful tool; it's worth getting to know it.

Andrew Korn **A**

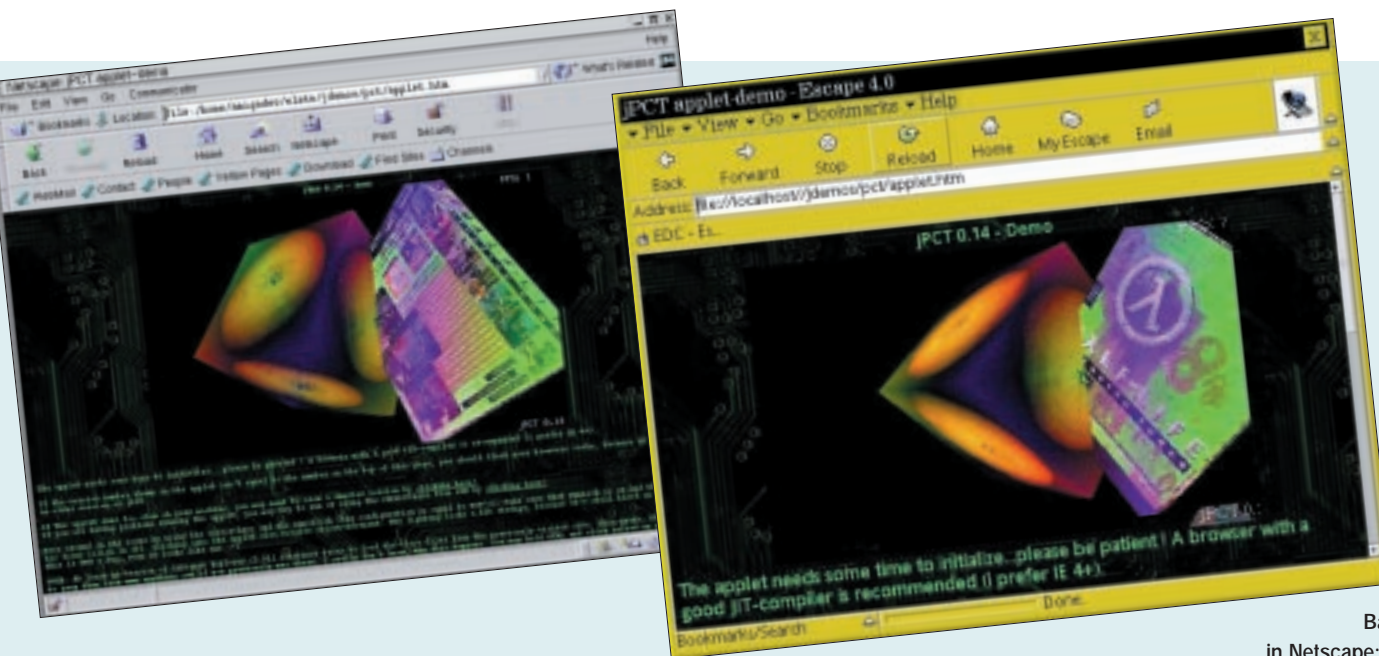
Right: Elate strutting its stuff. A Java game, a fully featured web browser, Doom and a pair of Quakes.



## Translated, Interpreted, hosted... can it work?

The brief way of explaining Elate is that it is an operating system, environment and tools which are written to run on a Virtual Processor (VP). The VP has to be written to work on the underlying platform. This can mean working directly with the underlying hardware (native Elate) or using an intermediary OS (hosted Elate). The code necessary to do this is isolated in modules called the Platform Isolation Interface and the CPU Isolation Layer. By switching these to suit a new hardware platform, everything else will run just the same without any changes. That means that an Elate application will run on a native Amiga / Elate machine, or any other supported platform. It also means a native machine can have almost any CPU. Indeed, it could change from one family of CPU to another without any applications breaking - imagine if all your current Amiga software just automatically ran (faster) on PPC.

One common question is how software translated from a foreign code to the code native to the hardware can be as efficient as native code. The trick is that Elate can translate at load time, in effect doing during that load period what on another system the developer does by compiling code for a specific platform. The significant point is that compilation is so fast it makes little



difference, so in most cases the compilation time is masked by load time. Elate dynamically binds code modules called 'tools' as needed - think of something like the Amiga shared library system but working on a individual function call level. The translated code is re-entrant and can be stored for re-use. This keeps Elate code pretty efficient. Of course this approach does to a significant extent demand the developer writes code which suits this method for optimum performance, but in use Elate seems to manage pretty well.

### Benchmarking

There's no easy way to benchmark Elate. For what it's worth, running 4 simultaneous instances of Quake and one of Doom was an insufficient load to render Quake (an unoptimised port) too slow to play. On its own in a standard timedemo1 test it managed 28fps in 320x200. The only thing this really says is that individual processes don't seem to suffer through multiplicity much more than is accounted for by polygon shifting alone.

Java is perhaps a more interesting measure of Elate's performance. Elate actually translates Java into native VP and it models well - it runs at JIT like speeds because it is, in a round about way, rather like a JIT compiler. Elate's efficiency means that it produces a far less bloated bytecode, requiring significantly less resources than JITs.

Again, it's difficult to measure performance. I tried loading a few applets into Espial's Escape Java browser running under Elate and Netscape (not the best competition, I know). Elate thrashed the Linux JVM used in Netscape so badly it should be ashamed of itself. A shaded

Battlezone style game was jerky but playable in Netscape; in Escape it was actually too fast. These screenshots (above left) show a very tricky applet with animated textures on rotating polygons. On Netscape, the frames per second counter rarely broke 2, while it went up to 8 on Escape. Running both side by side slowed Escape down to 3-4 fps, while Netscape's counter bottomed out on 1 - it was taking up to a couple of seconds a frame. While quick tests like this should be taken with a pinch of salt, they do impress - particularly when you call that this is running hosted on another OS, not natively.





# AMIGACTIVE MAGAZINE

*The article you have just read  
was first published in issue 9  
of Amiga Active magazine.*

*Individual issues are available direct from Pinprint Publishing.  
Visit our web site or call us for more information - details below.*

## *The future of Amiga computing!*

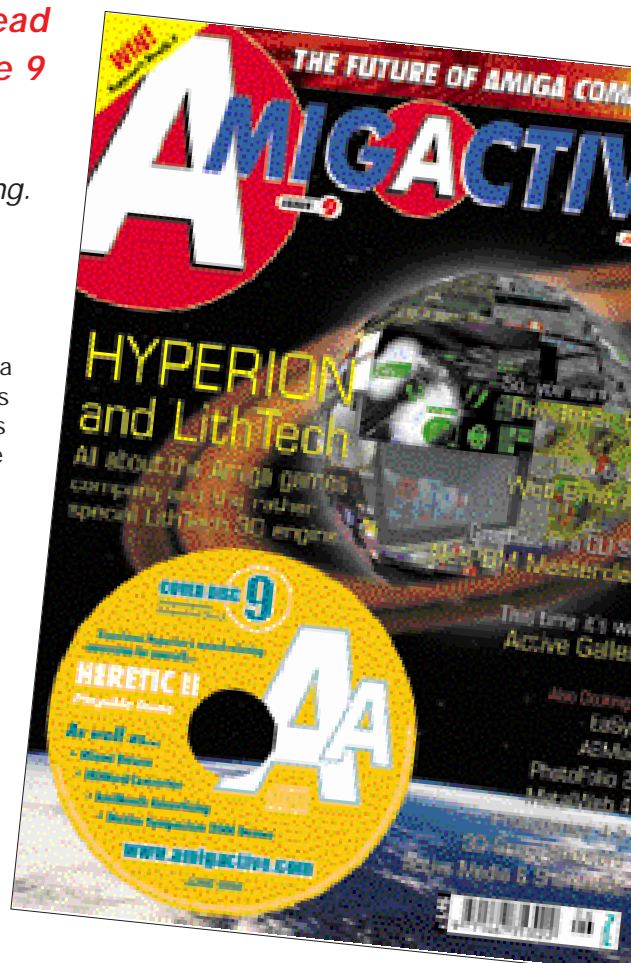
*Amiga Active is a monthly computer magazine targeted at the serious Amiga user. We take a fresh look at the Amiga, report the latest news in depth, review the best products, analyse the latest market trends with extensive features and provide tutorials on the latest hardware and software. Each issue of the magazine comes complete with a uniquely interactive cover-mounted CD-ROM which complements the content of each issue, providing our readers with a uniquely interactive experience.*

*Amiga Active magazine is produced by Pinprint Publishing Ltd. in Bournemouth, England and distributed worldwide via the newsstands and on subscription. Individual issues and subscriptions can be purchased online via our web site.*

**Amigactive Online:** [www.amigactive.com](http://www.amigactive.com)

For more information, please get in touch via e-mail, phone or fax:

**Email:** [info@amigactive.com](mailto:info@amigactive.com)  
**Phone:** +44 (0)1202 296293  
**Fax:** +44 (0)1202 296294



**ACTIVATE YOUR AMIGA !**