

Out of the box

# KILLING WITH A SMILE



Friends of the Atari-ST classic MidiMaze might recognise one of the latest additions to Linux gaming. Christian Perle gets his fingers twitching with the rediscovered classic, iMaze

**C**In 1987, long before Doom and its compatriots coined the term “first-person shooter”, there was one game on the Atari scene that used to keep people up at night. Its name? MidiMaze. The program used the **MIDI** interface for networking, as this was a fixed component of the Atari ST and could manage cheap DIN cables.

The idea behind the game was very simple: the players – as Smilies – would run through a 3D labyrinth shooting at each other, with the walls serving as cover. One point was awarded for shooting someone and after taking a hit, players would soon respawn at a random location. The joke of the game was that computer opponents can never be as annoying as human players...

In 1994, under the guise of a practical software exercise, Hans-Ulrich Kiel and Jörg Czeranski of the Technical University of Clausthal set themselves the task of implementing their favourite game under Unix as a **client/server** version. This is how *iMaze* came about, which henceforth prevented the students in the Clausthal computing centre from doing much work. Unlike MidiMaze, *iMaze* can be played not only with local networking, but also via the Internet – all you need is a 28,800bps modem. Although development was halted in 1996, it has now been resumed with version 1.4.

## Semi-automatic

To simplify the installation process we’ve provided the **shell script** `iminst.sh` on the cover CD, which performs most of the work steps itself. For the installation you will need the **tarball** `imaze-1.4.tar.gz`, which you can find at <http://home.tu-clausthal.de/student/iMaze/> or on the cover CD. Copy this file and the script `iminst.sh` into a shared directory, and enter the following commands:

```
su enter root password
sh iminst.sh
exit
```

Before you start make sure that the C compiler `gcc`

## Out of the box

There are thousands of tools and utilities for Linux. “Out of the box” takes the pick of the bunch and each month suggests a little program, which we feel is either absolutely indispensable or unduly ignored.

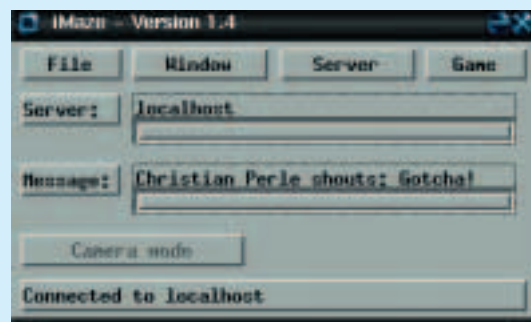


Figure 1: The iMaze menu window

and the necessary **header files** (`glibc-dev`, `xlib-dev` and `xaw-dev`) are installed. If everything works, *iMaze* will be in the `/usr/local` hierarchy once the script is completed.

## Where are the servers?

When starting the *iMaze* client with `imaze &` the first thing it wants to know is which server governs the labyrinth (Figure 1). Here you will also define the message other players will receive when they get hit by you.

If there are no other players to be seen on the proposed server, `imaze.rz.tu-clausthal.de`, you can also start your own server for your local network. To do so, enter the command:

```
imazesrv /usr/local/lib/imaze/labs/doors.lab &
```

As well as `doors.lab` there are also some other labyrinth files, which you will find in the same directory. As server name, enter `localhost` in the client. Other players in the local network should enter

**MIDI** Musical Instruments Digital Interface; a standard for the control of electronic musical instruments.

**Server** A program which offers a specific service, which “client” programs can use when they connect to the server. Examples of the services offered are `www`, `ssh` and `iMaze`.

**Shell script** A script is executed directly by an “interpreter” (such as `bash` or `sh` for shell, or Perl for Perl scripts) and does not have to be compiled before being executed.

the name or IP address of your computer in the Server field.

If you have your own server this also gives you the option of becoming familiar with controlling the client without being shot down straight away. The available key functions (Table 1) may be quickly learnt, but cornering requires a certain finesse if you don't want to end up wrapped around the scenery.

### Shooting away in style

Anyone playing iMaze who simply starts shooting away at random will soon notice that they hardly hit anyone. This is because the iMaze server only manages one shot per player at a time. So if you fire a new shot before the old one reaches its target, the old one will be taken out of the game and only the new one will be taken into account. It's therefore better to fire off fewer but better-aimed shots. Since iMaze offers no function for sideways movement, you will instead need to move backwards around a corner in order to avoid being shot by your opponents.

Some good additions to the Front View are offered by the windows Map, Compass and Rear View, which you can reach via the Window menu (Figures 4 and 5). In labyrinths, such as *doors.lab*, in addition to grey walls you'll also come across coloured ones. You can pass through these walls via the coloured side, but not the other way round. Such one-way doors can be very useful for pursuit tactics.

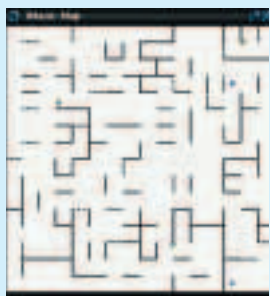


Figure 4: Map for a quick overview



Figure 5: Where am I running to?

### Electronic shadow targets

If there aren't any other players to be found, or you simply want to bump up the number of available opponents in a game, you can start so-called "Ninja" processes, which link to a game and play like a normal human client. They may not be especially clever, but sometimes brawn beats brain. You can start these electronic pains in the neck with:

```
ninja -H Server-Name &
```

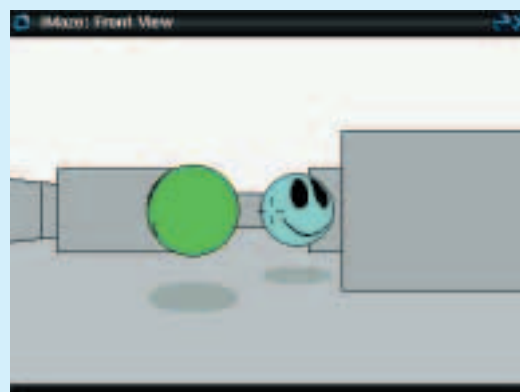
thus on your own server computer with:

```
ninja -H localhost &
```

If you don't want to see any more Ninjas in the

Table 1: Key functions in the iMaze client

Key	Function
Cursor up	Move forwards
Cursor down	Move backwards
Cursor left	Turn left (can be combined with movement)
Cursor right	Turn right (can be combined with movement)
Shift, Alt or Space	Shoot
Ctrl+S	Pause (Smiley is temporarily taken out of the game)
Ctrl+Q	Continue play (Smiley awakes in a random position)
Tab	Immediately turn through 180 degrees (only with server option -Q)



Chase through the labyrinth

labyrinth, enter *killall ninja* in the shell. If you ever forget your Ninjas, it's not the end of the world: after a maximum of two hours they will shut down on their own.

When several Ninjas have been started, it is advisable to use the option *-m* to assign special shoot-down messages to the individual processes. The *imaze\_demo* shell script, on the cover CD, starts a server and six Ninjas. All you need to do is start the client and link to the server localhost:

```
imaze -H localhost &
```

### Big Brother and the special server

Anyone who just wants to take a peek at an active server without taking part in the action can select Camera mode in the menu window before connecting. Your own Smiley is then invisible to the other players and can run through all walls; but it does not have the ability to shoot – making you a passive observer.

A look at the manpage of the iMaze server with the command *man imazesrv* reveals a few more special options, which can be specified when starting. The option *-Q* enables all players to rotate rapidly through 180 degrees with the Tab key, the option *-R* makes all shots ricochet off the walls, and *-F* makes players "faceless", so that you can't tell which direction your opponents are facing. Last of all, with the programs *genlab* and *xlabled* you have tools to create and to modify your own labyrinths. Happy shooting.

**Tarball:** tar is an archiving tool common under Unix. A collection of files packed together into one file by it, referred to in slang as a *tarball*, usually bears the file ending *.tar.gz* or *.tgz*. This is because such archives are first amalgamated with the program *tar* and then compressed with the program *gzip*.

**Header files:** In header files (also called "include files") there is a list of the functions available in a library plus parameters. The C compiler needs this information when compiling a program. In the most common distributions a header package for a library usually includes *dev* or *devel* in its name.