

iMaze

# killing with a smile

VON CHRISTIAN PERLE

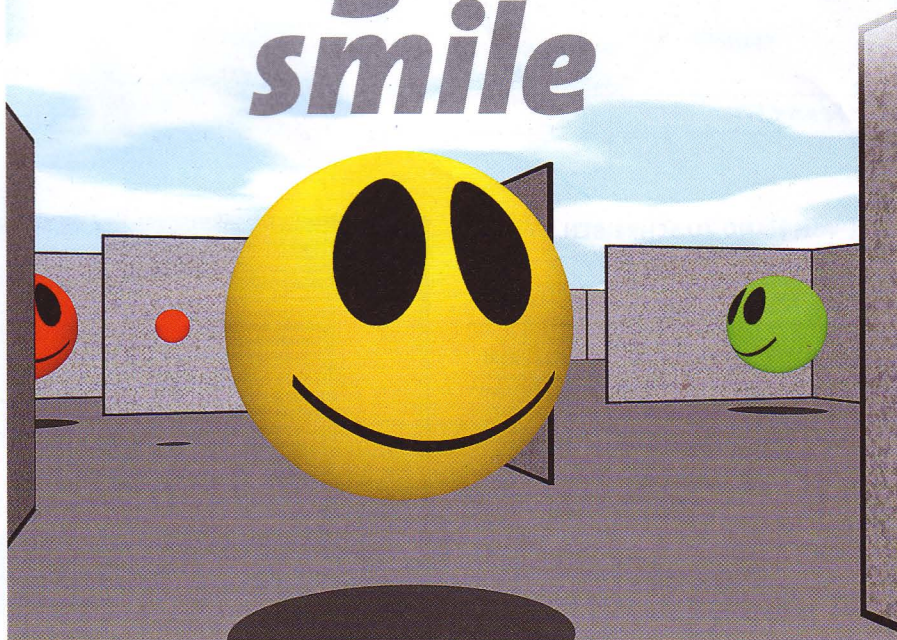
EIN VERGESSEN GEGLAUBTES SPIEL

WURDE IM DEZEMBER 2001 ZU NEUEM

LEBEN ERWECKT. FREUNDE DES ATARI-ST-

KLASSIKERS MIDI MAZE WERDEN DAS

SPIELPRINZIP KENNEN.



Im Jahr 1987, lange bevor Doom & Co. den Begriff *Ego Shooter* prägten, sorgte in der Atari-Szene ein Netzwerkspiel für schlaflose Nächte: *MidiMaze*. Für die Vernetzung bediente sich das Programm der **MIDI**-Schnittstelle, die fester Bestandteil des Atari ST war und mit billigen DIN-Kabeln auskam.

Die Spielidee war einfach: Die Spieler liefen als Smilies durch ein dreidimensional dargestelltes Labyrinth und schossen sich gegenseitig ab, wobei die Wände als Deckung dienten. Pro Abschuss gab es einen Punkt. Die Totgegläubten wachten jedoch nach kurzer Zeit an einer zufälligen Position wieder auf. Der Witz am Spiel: Computergegner können sich niemals so schön ärgern wie menschliche Mitspieler...

Im Rahmen eines Softwarepraktikums machten sich Hans-Ulrich Kiel und Jörg Czeranski von der Technischen Universität Clausthal 1994 daran, ihr Lieblingsspiel unter Unix als Client-/Server-Version zu implementieren. So entstand

*iMaze*, das fortan die Studenten im Clausthaler Rechenzentrum von der Arbeit abhielt. Im Gegensatz zum Vorbild *MidiMaze* ist *iMaze* nicht nur mit einer lokalen Vernetzung, sondern internetweit spielbar. Für die nötige Bandbreite reicht bereits ein 28800-bps-Modem aus. Nachdem die Entwicklung seit 1996 brach lag, gibt es nun eine neue Version 1.4.

## Halbautomatisch

Um die Installation etwas zu vereinfachen, stellen wir auf der Heft-CD das **Shell-Skript** *iminst.sh* zur Verfügung, das die meisten Arbeitsschritte selbst durchführt. Für die Installation benötigen Sie den **Tarball** *imaze-1.4.tar.gz*, den Sie unter <http://home.tu-clausthal.de/student/iMaze/> oder auf der Heft-CD finden. Kopieren Sie diese Datei und das Skript *iminst.sh* in ein gemeinsames Verzeichnis, und geben Sie folgende Befehle ein:

```
su (root-Passwort eingeben)
sh iminst.sh
exit
```

Stellen Sie vor dem Aufruf sicher, dass der C-Compiler *gcc* und die nötigen **Header-Dateien** (*glibc-dev*, *xlib-dev*, *xaw-dev*) installiert sind. Wenn alles klappt, befindet sich *iMaze* nach Ablauf des Skripts in der */usr/local*-Hierarchie.

## Wo sind die Server?

Beim Start des *iMaze*-Clients mit *imaze &* will dieser zunächst wissen, welcher Server das Labyrinth verwaltet (Abbildung 1). Außerdem legen Sie hier die Nachricht fest, die andere Mitspieler erhalten, wenn sie von Ihnen getroffen werden.

Falls auf dem vorgeschlagenen Server *imaze.rz.tu-clausthal.de* kein anderer Mitspieler zu sehen ist, können Sie alternativ einen eigenen Server für Ihr lokales Netz starten. Dazu geben Sie den Befehl

```
imazesrv /usr/local/lib/imaze/la2
bs/doors.lab &
```

TABELLE 1: TASTENFUNKTIONEN IM IMAZE-CLIENT

Taste	Funktion
[Cursor hoch]	Bewegung nach vorn
[Cursor runter]	Bewegung nach hinten
[Cursor links]	Drehung nach links (kann mit Bewegung kombiniert werden)
[Cursor rechts]	Drehung nach rechts (kann mit Bewegung kombiniert werden)
[Shift], [Alt] oder [Leertaste]	Schießen
[Strg-s]	Pause (Smily wird vorübergehend aus dem Spiel genommen)
[Strg-q]	Spiel fortsetzen (Smily erwacht an zufälliger Position)
[Tab]	Sofort um 180 Grad drehen (nur bei Server-Option -Q)

## OUT OF THE BOX

Es gibt tausende Tools und Utilities für Linux. „out of the box“ pickt sich die Rosinen raus und stellt pro Monat ein Programmchen vor, das wir für schlichtweg unentbehrlich oder aber zu Unrecht wenig beachtet halten.



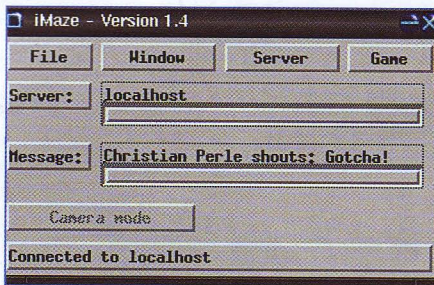


Abb. 1: Das iMaze-Menüfenster

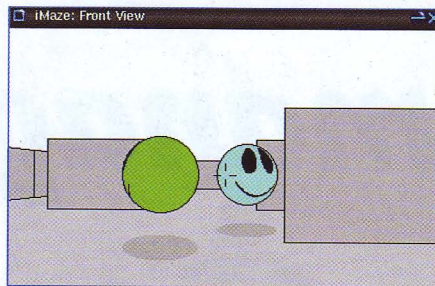


Abb. 2: Verfolgungsjagd im Labyrinth

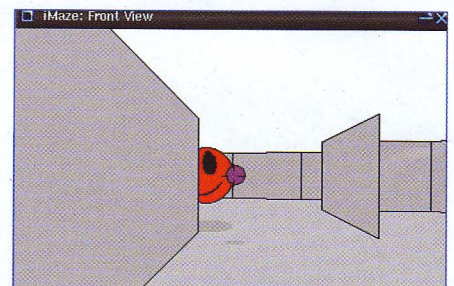


Abb. 3: Verstecken gilt nicht

ein. Außer *doors.lab* gibt es noch weitere Labyrinthdateien, die Sie im gleichen Verzeichnis finden. Als Server-Namen tragen Sie im Client *localhost* ein. Andere Mitspieler im lokalen Netz müssen im Menüfenster hinter *Server:* den Namen oder die IP-Adresse Ihres Rechners angeben.

Ein eigener Server gibt Ihnen auch die Möglichkeit, sich mit der Steuerung des Clients vertraut zu machen, ohne gleich abgeschossen zu werden. Die zur Verfügung stehenden Tastenfunktionen (Tabelle 1) sind zwar schnell zu lernen, aber Kurvenfahrten erfordern eine gewisse Routine, wenn man dabei nicht an den Wänden hängen bleiben will.

### Ballern mit Stil

Wer bei iMaze einfach sinnlos drauflosballert, wird schnell merken, dass er kaum jemanden trifft. Das liegt daran, dass der iMaze-Server nur einen Schuss pro Spieler gleichzeitig verwaltet. Schickt man einen neuen Schuss los, bevor der alte sein Ziel erreicht, wird der alte aus dem Spiel genommen und nur der neue beachtet. Man sollte also eher wenige, jedoch gezielte Schüsse abfeuern. Da iMaze keine Funktion zur Seitwärtsbewegung bietet, laufen Sie stattdessen rückwärts um die Kurve, um gegnerischen Geschossen auszuweichen.

Eine gute Ergänzung zum *Front View* (Abbildungen 2 und 3) bieten die Fenster *Map* (Karte), *Compass* und *Rear View* (Rückspiegel), die Sie über das *Window*-Menü erreichen (Abbildungen 4 und 5).

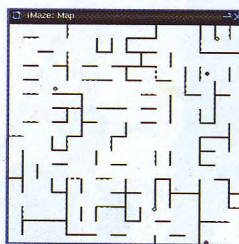


Abb. 4: Karte für den schnellen Überblick



Abb. 5: Wohin laufe ich?

In Labyrinth wie dem besagten *doors.lab* gibt es außer den grauen noch diverse eingefärbte Wände. Diese sind von der farbigen Seite aus durchlässig, aber nicht notwendigerweise von beiden Seiten! Solche „Türen“ lassen sich gut für Verfolgungstaktiken benutzen.

### Elektronische Pappkameraden

Findet sich partout kein anderer Mitspieler oder sind es einfach nur zu wenige für die Größe des aktuellen Labyrinths, so können Sie sogenannte *Ninja*-Prozesse starten, die sich wie ein normaler Client zum Server verbinden und die Gänge unsicher machen. Besonders schlaue sind sie zwar nicht, aber wo die Klasse nicht hilft, hilft die Masse... Sie starten die elektronischen Nervensägen mit

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

..., auf Ihrem eigenen Server-Rechner also mit *ninja -H localhost &*. Wollen sie keine Ninjas mehr im Labyrinth sehen, geben Sie in der Shell *killall ninja* ein. Falls sie Ihre Ninjas mal vergessen, ist das kein

Beinbruch: Nach maximal zwei Stunden beenden sie sich ohnehin von selbst.

Bei mehreren gestarteten Ninjas ist es empfehlenswert, die Option *-m* zu benutzen, um den einzelnen Prozessen spezielle Abschussnachrichten zuzuordnen. Das auf der Heft-CD enthaltene Shell-Skript *imaze\_demo* startet einen Server und sechs Ninjas. Sie brauchen nur noch den Client starten und sich mit dem Server *localhost* verbinden: *imaze -H localhost &* heißt der entsprechende Befehl.

### Big Brother und die Spezial-Server

Wer auf einem belebten Server „nur mal gucken“ will, kann im Menüfenster vor dem Verbinden den *Camera mode* aktivieren. Der eigene Smiley ist dann für die anderen Mitspieler nicht sichtbar und darf durch alle Wände laufen; allerdings fehlt ihm auch die Möglichkeit zu schießen – ein neutraler Beobachter eben.

Ein Blick in die Manualseite des iMaze-Servers mit *man imazesrv* verrät noch einige Spezialoptionen, die man beim Starten mit angeben kann. Die Option *-Q* ermöglicht allen Spielern das schnelle Drehen um 180 Grad mit der [Tab]-Taste, die Option *-R* lässt alle Geschosse von den Wänden abprallen, und *-F* macht die Spieler „gesichtslos“, so dass man nicht erkennen kann, in welche Richtung sie blicken. Zu guter Letzt haben Sie mit den Programmen *genlab* und *xlabeled* Werkzeuge, um eigene Labyrinth zu erschaffen und zu modifizieren. In diesem Sinne: angenehmes Ballern. (pju)

### Glossar

**MIDI:** „Musical Instruments Digital Interface“, ein Standard zur Steuerung elektronischer Musikinstrumente.

**Server:** Ein Programm, das einen bestimmten Dienst anbietet, den „Client-Programme“ nutzen können, wenn sie sich mit dem Server verbinden. Beispiele für solche angebotenen Dienste sind WWW, ssh und iMaze.

**Shell-Skript:** Ein Skript wird direkt von einem „Interpreter“ (etwa *bash* bzw. *sh*

für Shell- oder perl für Perl-Skripte) ausgeführt und muss vor der Ausführung nicht übersetzt werden.

**Tarball:** tar ist ein unter Unix übliches Archivierungstool. Eine damit zu einem File zusammengepackte Dateiansammlung, im Slang Tarball genannt, trägt meistens die Dateierendung *.tar.gz* oder *.tgz*. Diese rührt daher, dass solche Archive zuerst mit dem Programm *tar* zusammengefasst und dann mit dem Programm *gzip* komprimiert werden.

**Header-Dateien:** In Header-Dateien (auch „Include-Dateien“ genannt) sind die in einer Bibliothek verfügbaren Funktionen nebst Parametern aufgelistet. Der C-Compiler braucht diese Information beim Übersetzen eines Programms. Bei den gängigen Distributionen trägt ein Header-Paket zu einer Bibliothek üblicherweise den Zusatz *dev* oder *devel* im Namen.



iMaze 1.4, iminst.sh, imaze\_demo  
LinuxUser/ootb/