

# IOCCC – Schmankerl

Was ist das denn? IOCCC steht für International Obfuscated C Coding Contest, was übersetzt in etwa heißt: Internationaler Unleserlicher C Programmier Wettbewerb.

Oder wie es Wikipedia beschreibt:

Der **International Obfuscated C Code Contest** (kurz **IOCCC**) ist ein [Programmierwettbewerb](#) für die am kreativsten [verschleierte C-Programme](#) (engl.: *to obfuscate*: von lat. *obfuscare*, dt.: *verdunkeln*).

**Link:** <https://www.ioccc.org/>  
[https://de.wikipedia.org/wiki/International\\_Obfuscated\\_C\\_Code\\_Contest](https://de.wikipedia.org/wiki/International_Obfuscated_C_Code_Contest)

## Weitere Links:

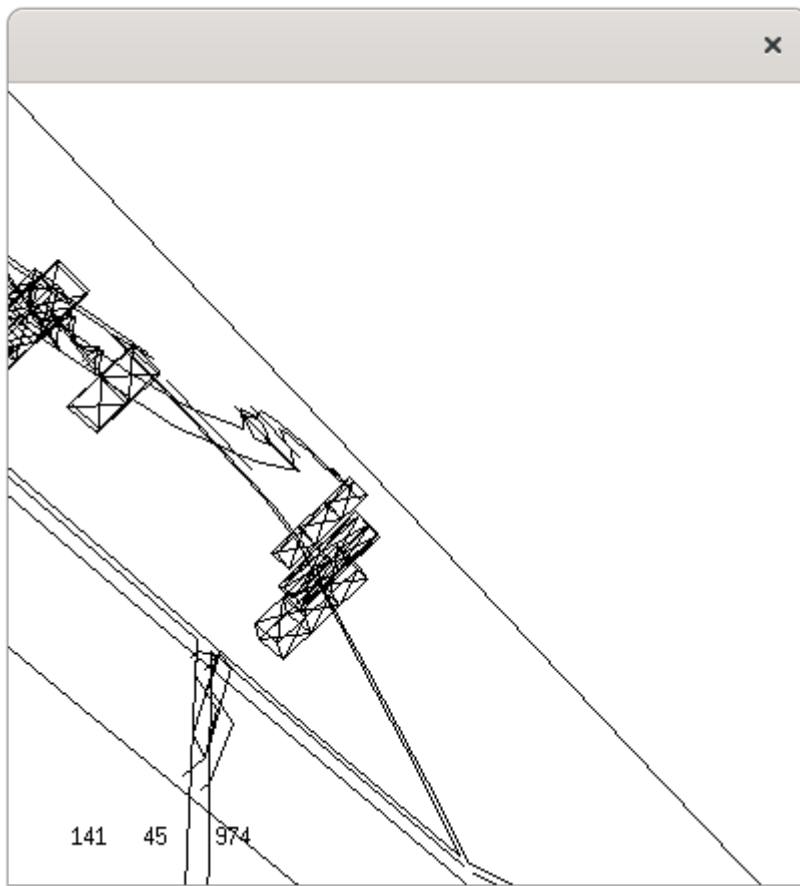
<https://medium.com/@sidneyriffic/deobfuscating-obfuscated-code-5e81029e3c4d>

Ausprobiert habe ich nachfolgende Programme unter Ubuntu 20.04LTS und Debian 11.

Generell gibt es dort viel interessantes und kurioses zu sehen.

Ich habe mir dort folgende Schmankerl ausgesucht zur Vorstellung:

## **banks - X11 Flight Simulator (1988)**



### **Author:**

Carl Banks  
Penn State Department of Aerospace Engineering  
232 Hammond Building  
University Park, PA 16802  
USA

<http://www.personal.psu.edu/users/c/w/cwb129/>  
<https://blog.aerofockey.com/post/iocccsim>

### **Programm erstellen:**

```
make banks
```

### **Ausführen:**

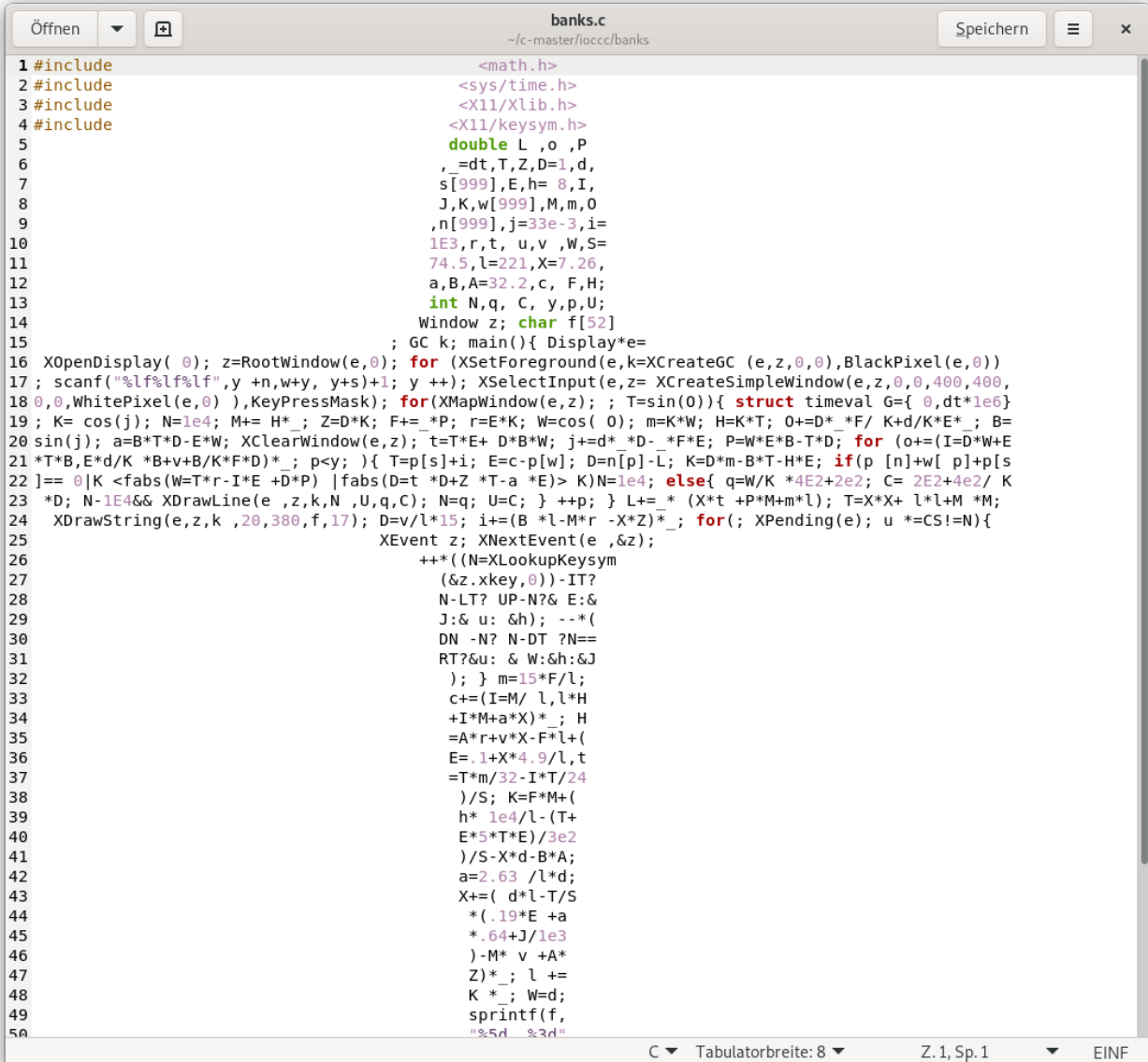
```
cat horizon.sc pittsburgh.sc | ./banks
```

Es handelt sich um einen Flug Simulator für X11 in 1536 Bytes! Cursor Links, Rechts, Hoch, Runter steuern das virtuelle Flugzeug.

Die Landschafts/Sceneriedaten sind in den \*.sc Dateien:

- **horizon.sc** – Nur ein Horizont. Sollte man immer mitnutzen.
- **mountains.sc** – Ein alternativer Horizont mit „Bergen“.
- **pittsburgh.sc** – Scene der Innenstadt von Pittsburgh (Zur rechten Seite).
- **bb.sc** – Einfache Scenerie mit Hindernissen.
- **pyramids.sc** – Virtuelle Ägyptische Scenerie mit Pyramiden.
- **river.sc** – Flusslauf vom Himmel.

Und aus reinem Interesse, so sehen z.B. solche IOCCC Programm im Source aus:



```
1 #include <math.h>
2 #include <sys/time.h>
3 #include <X11/Xlib.h>
4 #include <X11/keysym.h>
5 double L, o, P
6 , _dt, T, Z, D=1, d,
7 s[999], E, h= 8, I,
8 J, K, w[999], M, m, 0
9 , n[999], j=33e-3, i=
10 1E3, r, t, u, v, W, S=
11 74.5, l=221, X=7.26,
12 a, B, A=32.2, c, F, H;
13 int N, q, C, y, p, U;
14 Window z; char f[52]
15 ; GC k; main(){ Display *e=
16 XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
17 ; scanf("%lf%lf%lf",y +n,w+y, y+s)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
18 0,0,WhitePixel(e,0) ),KeyPressMask); for(XMapWindow(e,z); ; T=sin(0)){ struct timeval G={ 0,dt*1e6}
19 ; K= cos(j); N=1e4; M+= H*_; Z=D*K; F+= _P; r=E*K; W=cos( 0); m=K*W; H=K*T; O+=D*_F/ K+d/K*E*_; B=
20 sin(j); a=B*T*D-E*W; XClearWindow(e,z); t=T+E+ D*B*W; j+=d*_D*_F*E; P=W*E*B-T*D; for (o+=(I=D*W+E
21 *T*B,E*d/K *B+v+B/K*F*D)*_; p<y; ){ T=p[s]+i; E=c-p[w]; D=n[p]-L; K=D*m-B*T-H*E; if(p [n]+w[ p]+p[s
22 ]== 0|K <fabs(W=T*r-I*E +D*P) |fabs(D=t *D+Z *T-a *E)> K)N=1e4; else{ q=W/K *4E2+2e2; C= 2E2+4e2/ K
23 *D; N-1E4&& XDrawLine(e ,z,k,N ,U,q,C); N=q; U=C; } ++p; } L+=_ (X*t +P*M+m*l); T=X*X+ l*l+M *M;
24 XDrawString(e,z,k ,20,380,f,17); D=v/l*15; i+=(B *l-M*r -X*Z)*_; for(; XPending(e); u *=CS!=N){
25 XEvent z; XNextEvent(e ,&z);
26 ++*( (N=XLookupKeysym
27 (&z.xkey,0))-IT?
28 N-LT? UP-N?& E:&
29 J:& u: &h); --*(
30 DN -N? N-DT ?N==
31 RT?&u: & W:&h:&J
32 ); } m=15*F/l;
33 c+=(I=M/ l,l*M
34 +I*M+a*X)*_; H
35 =A*r+v*X-F*l+(
36 E=-.1+X*4.9/l,t
37 =T*m/32-I*T/24
38 )/S; K=F*M+(
39 h* 1e4/l-(T+
40 E*5*T*E)/3e2
41 )/S-X*d-B*A;
42 a=2.63 /l*d;
43 X+=( d*l-T/S
44 *(.19*E +a
45 *.64+j/1e3
46 )-M*_ v +A*
47 Z)*_; l +=
48 K *_; W=d;
49 sprintf(f,
50 "%5d _%3d"
```

Hier wird versucht den Sourcecode wieder lesbar zu machen:

<https://linuxgamecast.com/forums/topic/deobfuscate-the-carl-banks-flight-simulator-from-ioccc-1998-readable-clear-up-indented/>

## cable3 - PC Emulator (2013)

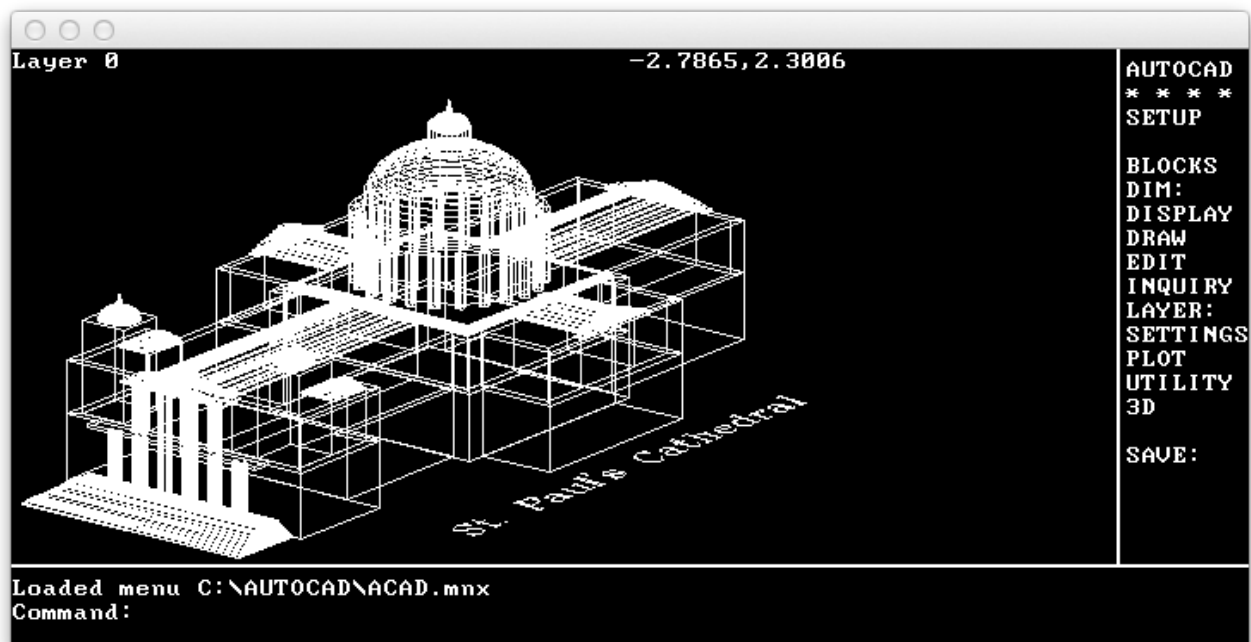
```
ich@ubuntu: ~/c-master/ioccc/cable3
FreeDOS kernel - SVN (build 2040 OEM:0x9d) [compiled Apr  7 2012]
Kernel compatibility 7.10 - WATCOMC - FAT32 support

(C) Copyright 1995-2011 Pasquale J. Villani and The FreeDOS Project.
All Rights Reserved. This is free software and comes with ABSOLUTELY NO
WARRANTY; you can redistribute it and/or modify it under the terms of the
GNU General Public License as published by the Free Software Foundation;
either version 2, or (at your option) any later version.
- InitDiskno hard disks detected

FreeCom version 0.82 pl 3 XMS_Swap [Dec 10 2003 06:49:21]

type HELP to get support on commands and navigation
A:\>
```

[A:\quitemu](#) beendet den Emulator. In hint.\* gibt es einen Link zu einem 40MB hd.img.



**Author:**

Adrian Cable

[adrian.cable@gmail.com](mailto:adrian.cable@gmail.com)

<https://github.com/adriancable/8086tiny>

<https://pushbx.org/ecm/8086tiny/doc.html>

<https://jaybertsoftware.weebly.com/8086-tiny-plus.html>

**Programm erstellen:**

(Benötigt SDL)

make cable3

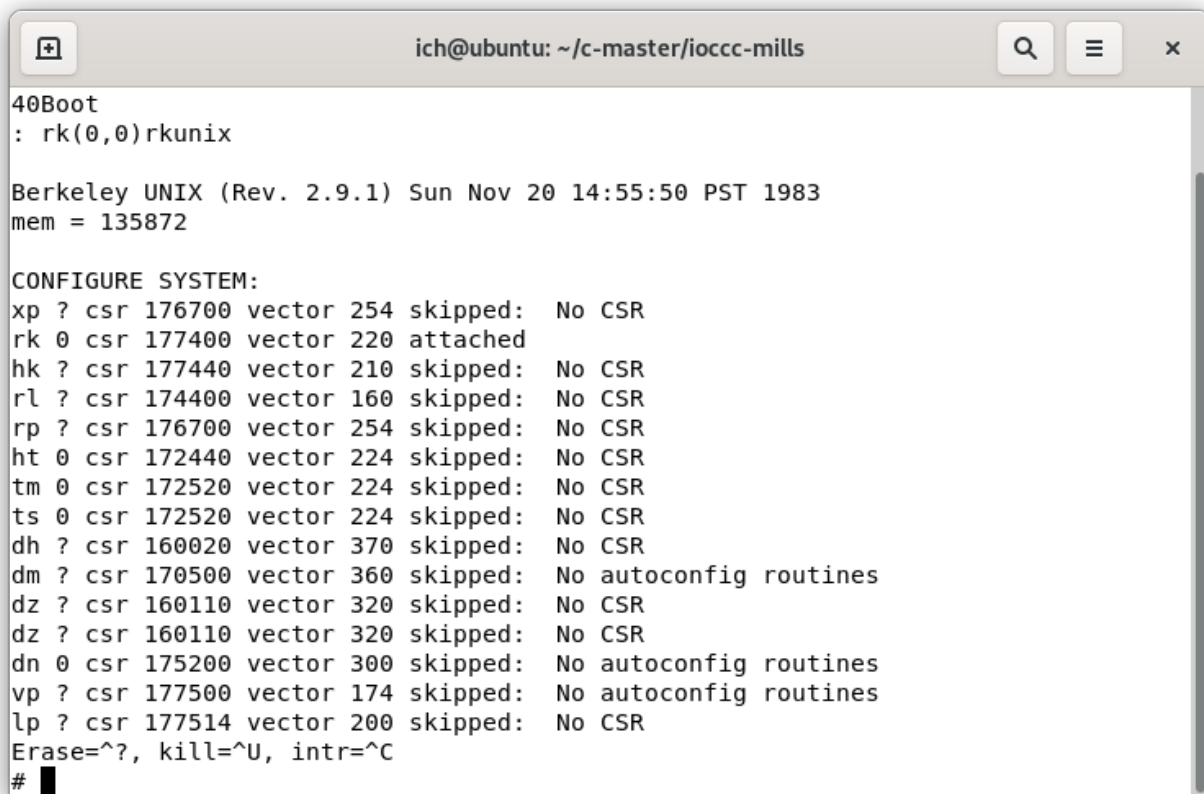
**Ausführen:**

```
./cable3 bios-image-file floppy-image-file [harddisk-image-file]
```

Nur 4043 Bytes Sourcecode und realisiert einen fast vollständigen IBM-PC aus den 1980 Jahren mit:

- Intel 8086/186 CPU
- 1MB RAM
- 8072A 3.5" fFoppy Disk Controller (1.44MB/720KB)
- HDD Controller (supports a single hard drive up to 528MB)
- Hercules Graphics Card with 720x348 2-color Graphics (64KB video RAM), and CGA 80x25 16-color Text mode support
- 8253 programmable interval timer (PIT)
- 8259 programmable interrupt controller (PIC)
- 8042 keyboard controller with 83-key XT-style keyboard
- MC146818 real-time clock
- PC speaker

## mills – lässt uralte Unix Systeme wieder aufleben (2018)



```
ich@ubuntu: ~/c-master/ioccc-mills
40Boot
: rk(0,0)rkunix

Berkeley UNIX (Rev. 2.9.1) Sun Nov 20 14:55:50 PST 1983
mem = 135872

CONFIGURE SYSTEM:
xp ? csr 176700 vector 254 skipped: No CSR
rk 0 csr 177400 vector 220 attached
hk ? csr 177440 vector 210 skipped: No CSR
rl ? csr 174400 vector 160 skipped: No CSR
rp ? csr 176700 vector 254 skipped: No CSR
ht 0 csr 172440 vector 224 skipped: No CSR
tm 0 csr 172520 vector 224 skipped: No CSR
ts 0 csr 172520 vector 224 skipped: No CSR
dh ? csr 160020 vector 370 skipped: No CSR
dm ? csr 170500 vector 360 skipped: No autoconfig routines
dz ? csr 160110 vector 320 skipped: No CSR
dz ? csr 160110 vector 320 skipped: No CSR
dn 0 csr 175200 vector 300 skipped: No autoconfig routines
vp ? csr 177500 vector 174 skipped: No autoconfig routines
lp ? csr 177514 vector 200 skipped: No CSR
Erase=^?, kill=^U, intr=^C
#
```

### Author:

Christopher Mills <mrxo@sonic.net>

Twitter: @MisterXopher

### Erzeugen:

make

### Benutzen:

./prog

(At the ">boot" prompt, type return)

(At the ":" prompt, type "rk(0,0)rkunix")

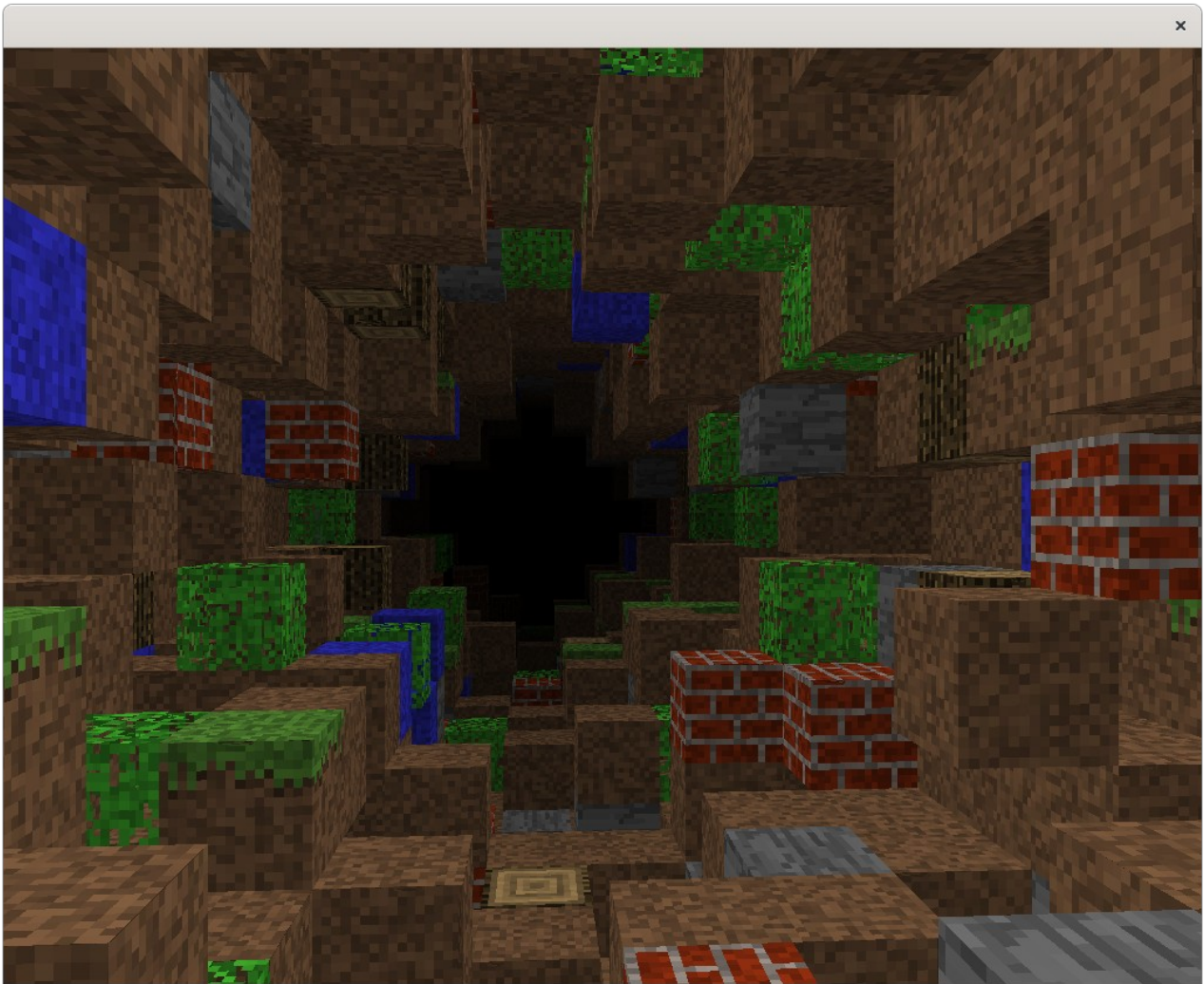
NOTE: To quit, press Control-E

Mit nur 3636 Bytes emuliert dieses Programm eine PDP-7! Damit und mit den beiliegenden HDD Images lassen sich emulieren:

1. UNIX v0 for the PDP-7 (circa 1969)
2. Research UNIX Version 6 (circa 1975)
3. BSD 2.9 (circa 1983)

Bitte den hint.txt lesen und beachten!

## dogon - X11 Labyrinth/Minecraft ähnliches Durchwandern (2015)



### Author:

Gil Dogon

<http://sokoban-gild.com>

### Erstellen:

make

### Ausführen:

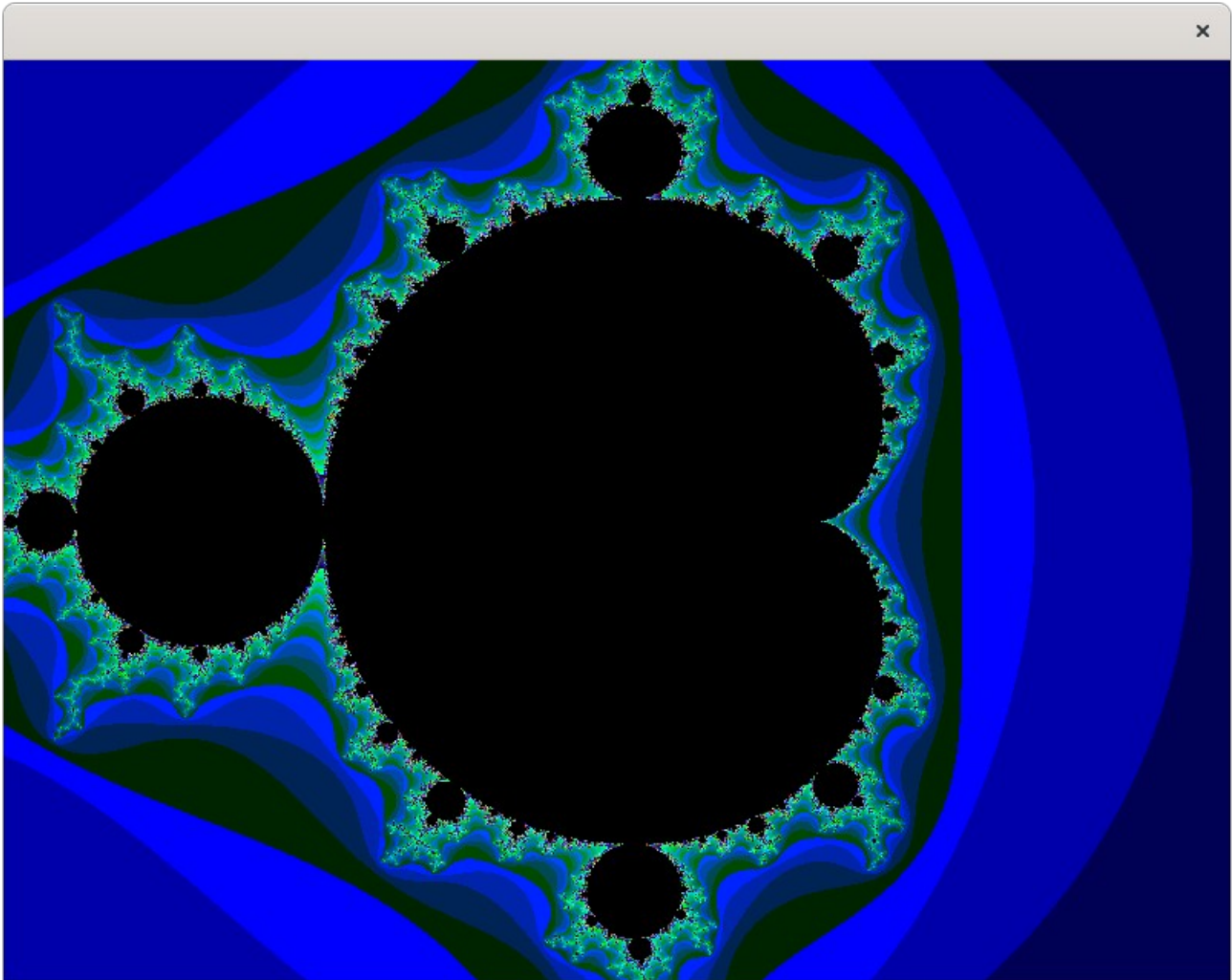
./prog

### Bedienung:

Linke Cursortaste drücken bis man um 90 Grad gedreht hat.  
Wenn ein langer Tunnel zu sehen ist, F2 drücken.  
Ansonsten Curser Hoch, Runter, Links, Rechts steuern die Richtung.  
F3, F4.. schneller. F1 Stop.  
ESC Ende.



## monge - X11 Formel Imager (2006)



### Author:

Maurizio Monge  
Department of Mathematics  
University of Pisa  
Pisa  
Italy  
maurizio.monge at gmail dot com

### Erstellen:

(Benötigt SDL)  
make monge

### Ausführen:

```
./monge "z = 0" "z = z*z*z + c; Abs2(z) < 4"
```

Hier bekommt man eine Maus gesteuerte Fraktal Erzeugung mit Zoom. Hint.txt lesen!