Frodo

Christian Bauer

COLLABORATORS

	TITLE : Frodo		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Christian Bauer	February 7, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Frodo

1.1   Frodo Manual

```
                                      Frodo
                      Version 1.3

          What's new?
                                A Commodore 64 emulator
              for Commodore-Amiga computers

           © Copyright 1994-1995 Christian Bauer
                   Freely distributable




          Overview
           Why another C64 emulator?

          Installation
           How to install Frodo


          Start/quit
           There and back again

          Settings
           You only have to configure...

          Keyboard layout
           Where the hell is the "any" key?

          File access
           A 64 with hard disk

          Printers
           Using C64 printers


          The SID card
           Do you want the real sound?
```

1.2   What's new?

The most important changes to version 1.0:
· Frodo can now use the "6581sid.library" for sound emulation over the
  Amiga hardware
· There is a new, very fast monochrome Amiga screen type and a normal
  one for CyBERgraphics
· Amiga modes now use double buffering
· The sprite collisions can now be turned off resulting in a significant
  speed increase (depending on the type of program running)
· Now you have the possibility to have, for instance, only every second
  frame of the C64 graphics calculated, likewise gaining an immense
  amount of speed
· Frodo is now multilingual
· The floppy emulation has wildcards (*,?)

Details are given in the
              revision history
                 .

Note: The joystick polling has changed. The joysticks are now polled
only once per frame instead of every raster line (as before). I don't
think this would be a problem for any game (who can move a joystick
that fast?), but if it is, please tell me. However, I've tried
"Spindizzy" and managed even tricky parts with the new emulation, so
it should be ok.

1.3 Overview

  "Those days, the Third Age of Middle-earth,
   are now long past, and the shape of all lands
   has been changed."

Frodo is a multitasking C64 emulator for the Commodore Amiga and
the world's first C64 emulator not bearing a "64" in its name. :-)
(No, it has absolutely nothing to do with frodo.hiof.no, that's a pure
coincidence.)

Frodo was especially developed to reproduce the graphics of games and
demos better than the existing emulators for the Amiga. Therefore Frodo
needs much more computing power than other emulators. Even using a
25MHz 68040 with a graphics card gives only slightly more than half of
the 64's original speed. But on the other hand, Frodo can display raster
effects correctly that only result in a flickering mess under "A64".

An Amiga with (at least) OS2.1, a 68020 processor and 1MB RAM is
required. Frodo can use the built-in Amiga graphic capabilities as well
as access EGS graphics cards and the "Picasso" directly to vastly
improve performance. So the recommended configuration is an A4000/040
with a Zorro-III graphics card. It is possible to run it on an A1200,
but you won't have any fun with it. There is, however, a special
monochrome mode that has no color but is a lot faster than the full
emulation.

The program is not a 100% emulation (is there such?), the main missing
things are:
 · Only line-by-line raster effects possible
 · No sprite-graphics priorities
 · No sprite-background collisions
 · Neither Amiga printers nor RS232 usable

But the next version is coming real soon now... :-)

1.4 Installation

            The archive contains the following files:
  README   - Short description
  Frodo   - Main program
  Frodo.info   - Its icon
  Frodo.guide   - The german version of this manual

```
  Frodo.guide.info   - Its icon
  Frodo_E.guide      - This manual
  Frodo_E.guide.info - Its icon
  Frodo Prefs        - Preferences file
  Frodo Prefs.info   - Its icon
  FrodoSIDCard.lha   - Archive with information on the
              SID card
                Basic ROM.info    - Icon for Basic ROM
  Kernal ROM.info    - Icon for Kernal ROM
  Char ROM.info      - Icon for Character ROM
  Catalogs/#?        - Language files
  64prgs/#?          - Some sample programs
  src/#?             - Source
```

All you have to do is to unpack the archive somewhere onto your hard
disk. The files in "Catalogs" don't need to be copied to LOCALE:, they
can stay where they are. Frodo doesn't need any assigns, but copies of
the original C64 ROMs are required. For legal reasons these are not
included in the archive and you have to get them elsewhere.

Frodo looks for three ROM files named "Basic ROM", "Kernal ROM" and
"Char ROM" in the same directory Frodo is in. If you already have
ROM files for "A64 V2.0" (not the *.data files, these are not the
real ROMs), you can simply rename them and use them with Frodo.
Otherwise, they can be read from a real C64 by means of the following
program:

```
  10 OPEN 1,8,2,"BASIC ROM,S,W"
  20 FOR X=40960 TO 49151:PRINT#1,CHR$(PEEK(X));:NEXT
  30 CLOSE 1
  40 OPEN 1,8,2,"KERNAL ROM,S,W"
  50 FOR X=57344 TO 65535:PRINT#1,CHR$(PEEK(X));:NEXT
  60 CLOSE 1
  70 POKE 56334,0:POKE 1,51
  80 FOR X=16384 TO 20479:POKE X,PEEK(X+36864):NEXT
  90 POKE 1,55:POKE 56334,1
 100 OPEN 1,8,2,"CHAR ROM,S,W"
 110 FOR X=16384 TO 20479:PRINT#1,CHR$(PEEK(X));:NEXT
 120 CLOSE 1
```

Unfortunately, Frodo has no function to get these files saved on a
C64 disk to the Amiga, but you can transfer them via a null-modem
cable or using A64 or any of the 1541-to-Amiga transfer utilities.
(Frodo can indeed access C64 disk drives, but to start Frodo you
must already have the ROMs %-).

There is, however, a way to get the ROMs via a mailserver. Write to
"duck@pembvax1.pembroke.edu" with subject "MAILSERV" and "send roms.zip"
in the body of the mail. You'll receive a uuencoded .zip archive
containing a lot of commodore ROMs. The ones needed for Frodo are in
the directory "64": "basic", "kernal.3" and "charset". Simply rename
them and copy them into Frodo's directory.

For sound emulation, Frodo uses the "6581sid.library" which is part
of A64 V3.0 and likewise not included. The demo version of A64 on the
Aminet contains this library so you can get it from there.

1.5 Start/quit

After a double cklick on the icon, the window for the emulation

settings
 apprears.

The actual emulation is started by a click on "OK". Then, at first
a black screen should appear followed by the C64 startup message
after a short moment.

Clicking the right mouse button will quit the emulator at any time.

1.6 Settings

The settings window apprears directly after starting Frodo, or by
pressing F9 in the running emulation.

"Cycles per line (CPU)" and "Cycles per bad line (CPU)" control the
relation of graphics speed and processor speed. The higher the values,
the faster the emulated CPU, but the slower the graphics refresh speed.
If a program is showing flickering lines or graphical flaws you should
try to slightly alter both values.

With "Cycles per line (CIA)" you can control the speed of the CIA
timers. Entering a higher value increases the frequency of cursor
blinking and key repeat. Some programs don't run correcly with the
default value (e.g. "Ballblazer" which needs a value of 65).

The settings for the three "Cycles" coming closest to an original PAL
C64 are (63,23,63).

Under "Every (n)th frame" you can select if Frodo should skip frames
when displaying the C64 graphics. The normal setting is "1", that is,
every frame (every simulated raster beam sweep) is recalculated. If
you change this to "2", for example, then only every second frame is
calculated, immensely speeding up the display, though some raster
effects may look a bit jerky. The setting for an A4000/040 without
graphics card that gives the original C64's speed in most cases is "5".

"Joystick on Port 1/2" relates to the Amiga ports. Normally, you will
want to have port 1 switched off if you have your mouse connected there.
With "Swap joysticks" you can swap the assignment of the Amiga ports to
the C64 ports without having to plug out and in your joysticks. E.g. if
a C64 game is using a joystick on port 1 you can simply activate "Swap
joysticks" and use a joystick in port 2 to play the game.

With "Use SID card" you specify if the sound emulation should be done
over the

                                        SID card
                          or using the "6581sid.library" (i.e. over the Amiga
sound channels). If you don't have a SID card, you should leave this
gadget switched off to prevent illegal memory accesses.

The keyboard layout normally matches that of a german keyboard (Y and Z
swapped). With "American keyboard" you can prevent the swapping
(preferable, if you are not using a german keyboard).

Programs using many sprites but not requiring sprite collision detection
can be made running much faster when disabling the "Sprite collisions"
gadget. Besides, this setting can be used as a "trainer" to become
invincible in some games (sadly, your enemys are likely to become
invincible, too :-/.

With the setting "Fast Reset" you can disable the memory test on C64
reset. On the Amiga, the memory test is not necessary and the reset
procedure (F10) becomes much faster when it is disabled.

The setting "Screen type" determines the way Frodo controls the
graphics. The choices are "Amiga" (for using the built-in Amiga
graphics), "EGS" (for EGS graphics cards), "Picasso" (for the Picasso
graphics card), "CyBERgraphics" (for CyBERgraphics cards) and
"Amiga Mono", a particularly fast monochrome mode without sprite
collisions (useful for "Elite", for example). After choosing the screen
type, you have to select an appropriate resolution using "Screen mode".
The display on graphics cards is faster than the Amiga graphics (besides
monochrome mode), but CyBERgraphics is a bit slower than EGS and
Picasso. These settings can not be altered in the middle of the running
emulation.

Next to "Drive 8" to "Drive 11" you give the paths of directories
(relative to the directory Frodo is in) that are used for the simulation
of disk drives 8, 9, 10 and 11 (see
                        file access
). Clicking on the symbols
to the right of the string gadgets will open a requester to select the
directory more user friendly.

The four checkmark gadgets under "IEC" control whether the belonging
disk drive is to be accessed over the
                        IEC cable
                         as real C64 devices.
If you have, for example, a real 1541 with drive number 8 connected to
your Amiga, you have to activate the "IEC" checkmark for drive 8 to use
the 1541 (instead of the simulated one).

When the symbol "Use IEC for other devices" is active, Frodo uses the
IEC cable also for device numbers 4 to 7 and 12 to 15. So, if you have
connected a
                        C64 printer
                         with device numer 4, you should click on this
gadget to use the printer. If this gadget is not activated, all device
numbers except 8 to 11 give a "device not present error".

With "'/' -> '\' in file names" you control whether the '/' in C64
filenames will be translated to '\' and vice versa. The '/' character

is used to access subdirectories on the Amiga, but as the C64 doesn't
have subdirectories, it's a valid part of a C64 file name. This is a
problem if a program wants to create a file with '/' in it as AmigaDOS
would interpret the part before the '/' as a directory name and, finding
no such directory, would return an error and the open would fail. Now
simply activate this gadget and all '/'s will transparently be
translated into '\', so in directory listings the '/' will still
appear. If you turn off this gadget, you can of course use the '/' to
access files in Amiga subdirectories from the C64.

By clicking "Save" the settings are saved on disk to be reloaded next
time you start Frodo. "OK" will start the actual emulation (resp. return
to it) and "Cancel" will discard your changes to the settings and quit
Frodo (resp. discard the changes and return to the emulation).

1.7 Keyboard layout

The keyboard layout closely resembles that of a C64, but the "Y" and
"Z" keys are normally swapped (german keyboard layout). This can be
prevented by clicking on "Amerik. Tastatur" in the settings window.

The individual rows of the keyboard are mapped as follows:

```
<- 1 2 3 4 5 6 7 8 9 0 + - £
   Q W E R T Z U I O P @ *
   A S D F G H J K L : ; =
  ^ Y X C V B N M , . /
```

In addition, the following key are used:

```
 ESC   - RUN/STOP
 <-    - INS/DEL
 Del   - CLR/HOME
 Help  - RESTORE
 Amiga - C=
 F1-F8 - F1-F8
 F9    - Opens the settings window
 F10   - Resets the C64
```

So the famous key combination RUN/STOP-RESTORE corresponds to ESC-Help
on the Amiga keyboard. But you don't have to thrash the Help key the
same way as the RESTORE key on the original C64. :-)

The cursor (arrow) keys can be used as on the Amiga. I.e. "cursor up"
corresponds to "Shift-cursor down" on the C64. The same applies to the
function keys F2, F4, F6 and F8.

1.8 File access

Frodo can simulate 1541 drives with the drive numbers 8, 9, 10 and ←
11

in Amiga directories as well as access real C64 drives by means of an

                    IEC cable
                    . So you can store your C64 programs on your Amiga's hard disk
and LOAD and SAVE them as usual from Frodo. The paths to the
directories of the simulated drives are given in the settings window
under "Laufwerk 8" to "Laufwerk 11".

You can even load the directory with LOAD"$",8. All files marked as
executable are displayed as "PRG", all the others as "SEQ". The file
type "DIR" marks an Amiga subdirectory. To get into it you have to
open the settings window and change the path for the drive, as the
operating system of the C64 doesn't know about subdirectories.
Though, it is possible to use a "/" in the C64 file name to access
these subdirectories (e.g. LOAD"GAMES/ELITE",8), unless the '/'
translation setting is turned on.

For the opening of files, the file types "PRG" and "SEQ" and the access
modes "R", "W" and "A" are supported. Wildcards (*,?) can be used, but
you have to remember that files in Amiga directories have no specific
order and the result of LOAD"*",8 is rather random. Files are always
overwritten even if they are not opened with "@:". Floppy commands,
relative files and direct block access are not implemented, you can
however read the error channel.

For connecting real C64 drives you need a special cable to hook them
to your Amiga. Then you have to activate the symbol "IEC" for the
corresponding drive number in the settings, so Frodo will know that
this drive number belongs to a device on the IEC cable. Drives
connected this way can be used like from the C64, but speed loaders
are most probably not working.

1.9 Printers

                    Frodo can't use Amiga printers, but you can connect a C64 IEC ←
                       printer
using the
                    IEC cable
                     that is accessed as usual under the C64. You have
to activate the gadget "IEC für andere Geräte verwenden" in the

                    settings window
                     to use the printer.

1.10 The SID card

Though Frodo has a sound emulation over the Amiga's hardware (by means
of the "6581sid.library") it can access a real C64 sound chip (SID) on
a slot card in the Amiga for the real sound experience. This card
(a unique example for now) maps the SID into the address range starting

at $a00001 (odd addresses).

The schematics and additional information can be found in the archive
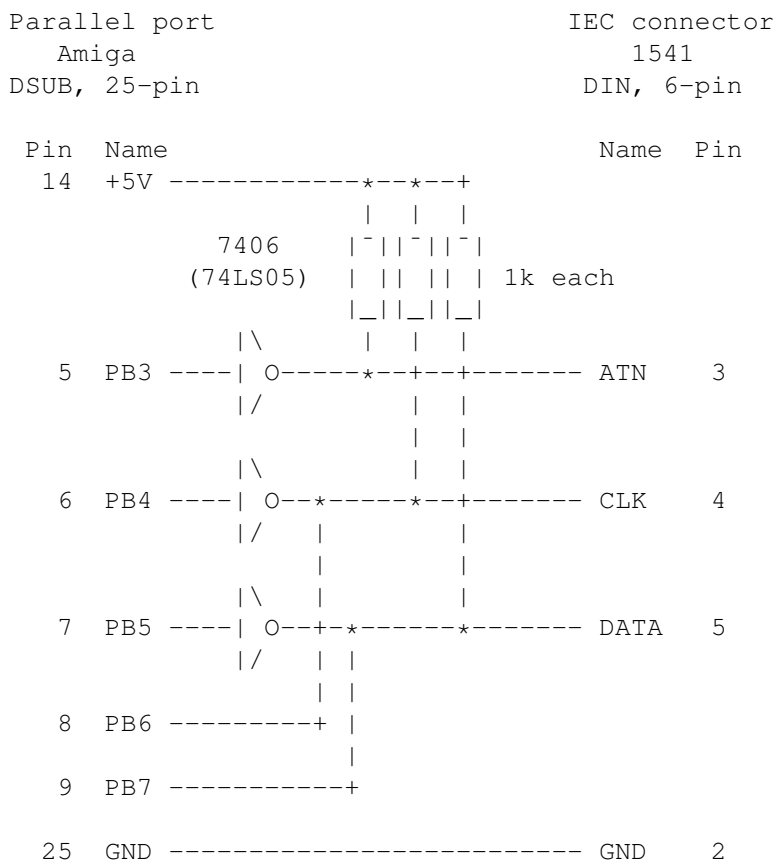"FrodoSIDCard.lha".

1.11 The IEC cable

"This is Ohm of Borg.
 Resistance is voltage divided by electric current." :-)

The best way to transfer your old C64 software to the Amiga is to
connect a real C64 disk drive. To do this, you need a special cable
with a small circuit that, together with Frodo's software, will
expand your Amiga by the so-called "IEC" bus system that is used on the
C64 for connecting disk drives and printers.

If you already have such a cable for the "A64" you most probably can
also use it with Frodo (untested). The cable described in the german
"Amiga Magazin" issue 8/88, p.36 is not usable with Frodo as it has
inverted output lines.

Building an IEC cable is fairly simple, it connects the parallel port
of the Amiga with the round 6-pin connector on the C64 devices.

```
 Parallel port                      IEC connector
    Amiga                              1541
 DSUB, 25-pin                       DIN, 6-pin

  Pin  Name                           Name  Pin
   14  +5V ------------*--*--+
                       |  |  |
             7406     |¯||¯||¯|
            (74LS05)  | || || | 1k each
                      |_||_||_|
             |\        |  |  |
    5  PB3 ----| O-----*--+--+------- ATN    3
             |/           |  |
                          |  |
             |\           |  |
    6  PB4 ----| O--*-----*--+------- CLK    4
             |/   |         |
                  |         |
             |\   |         |
    7  PB5 ----| O--+-*------*------- DATA   5
             |/   | |
                  | |
    8  PB6 --------+ |
                    |
    9  PB7 ----------+

   25  GND ------------------------- GND    2
```

The 7406 (a 74LS05 will do as well) must be connected to +5V and GND
of course. The trained technician will notice that this is the same

circuit found inside the real C64.

[Important note: If you plan to build and connect such a cable, please
remember that you do it on your own risk. I will not take any
responsibility if there is blue smoke coming out from the back of your
computer!]

1.12   Sample programs

The directory "64prgs" contains some little sample programs to show
the capabilities (and weaknesses) of Frodo. They are all loaded with
LOAD"<name>",8 and started with SYS49152. To do so, you have to type
'64prgs' in the settings for "Laufwerk 8" so Frodo will find the
programs.

If you like, you can try these programs on other emulators or on
a real C64.

A short description of each program:
```
  3fff     : Opens the top and bottom border and displays swinging
             letters ("Zyklen pro Zeile (CPU)" should be set to 60
             for this one)
  colorbars: Flickering colors
  dycp     : Scrolling with different Y character position
  fld      : Demonstrates the FLD effect (flexible line distance)
  stretch  : One thing Frodo is not (yet) able to emulate: variably
             expanded sprites
  tech-tech: Horizontal scrolling with large amplitude (use joystick
             in port 2 to control)
  text26   : Displays (nearly) 26 lines of text
  monitor  : A simple machine language monitor
```

1.13   Technical info

```
  "Wissen?" wiederholte Gandalf.
  "Ich weiß vieles, was nur die Weisen wissen, Frodo."
```

In contrast to A64 that is biased towards the Amiga hardware, Frodo
tries to exactly imitate C64 hardware features. Now the 64's hardware
(esp. the graphics chip "VIC") has a rather simple design resulting in
many of the internal processes coming to the "outside". So there are
lots of "undocumented features" you can do effects with the designers
never dared to dream about.

Frodo uses a line-by-line emulation, i.e. the function of the VIC
and the processor (6510) are emulated for one raster line of the
C64 screen at times. In practice, Frodo runs VIC and 6510 alternately
for 63 simulated cycles each (corresponding to one raster line).
At first, it emulates the processor for 63 cycles, then switches
over to the VIC that paints one pixel row to the screen, then again
63 cycles processor, and so on... With the exception of monochrome
mode, all graphics is internally calculated for a chunky mode display.

The Amiga mode uses a chunky->planar conversion to display the chunky
pixels on screen.

Even though this is a heavy simplification of the processes in a
real C64, it lets you perfectly emulate many graphical effects possible
on the 64, e.g. FLD, FLI, DYCP, hyperscreen and many more. But this
method has one big disadvantage: Changes made to VIC registers by the
processor in the middle of a raster line will only take effect at the
start of the next line. E.g. you can't change the border color in the
middle of a line, the color change takes place in the next line.
Therefore, very sophisticated techniques depending on the exact
position of a register change can't be emulated. For instance, it is
no problem to open the top and bottom border, but opening the left
and right border is impossible (and therefore not implemented in the
emulation).

The next step above line-by-line emulation would be to resolve VIC and
6510 into single cycles. The processes in the VIC and the 6510 for each
cycle are quite well known, so there is no theoretical problem in the
realization. The only real problem is speed. You would have to switch
between VIC and 6510 after every cycle, instead of after 63 cycles as
Frodo does in the moment. This would mean abandoning the register-based
6510 emulation, as the overhead for storing and retrieving the complete
register set for every switch would be too high.

The present opcode dispatch model would have to be replaced by some
kind of "cycle continuation pointer" pointing to the subroutine for
the next cycle of the current opcode (or should the opcode dispatch
table be extended by 8 entries for each opcode for the single cycles?).
As a first estimation, the speed would go down by a factor of 3-4,
so before the availability of faster Amigas there is no hope for
cycle based emulation. Though you could emulate effects like AFLI and
the VC and sprite counter confusions of the VIC precisely.

1.14   A note on speed

Ok, the emulation is very slow, but why is this so?
Some little calculations:

The overscanned display area of the C64 is 411$\times$284 pixels, 368$\times$272 $\hookleftarrow$
    of
which are displayed by Frodo, giving a total of 100096 pixels. In
chunky mode each pixel needs one byte, so at a frame rate of 50Hz we'd
have to transfer 5MB/s into the display memory. The bandwidth of the
Zorro III bus on an A4000/040 is 6.6MB/s...

Considering that the emulator moreover has to read the graphics from
somewhere and has to emulate the processor and I/O "by the way", you'll
discover that there are theoretical limits for the speed of the
emulation. By using Amiga bitplane graphics the amount of data could be
halved, but the gain is largely absorbed by the lower chip RAM bandwidth
(4.5MB/s) and the more complicated processing required.

So what can be done about it? The most obvious idea would be to remove
the chunky->planar conversion and to have the VIC write directly into

the Amiga bitplanes using optimized routines. The monochrome mode is
the first step in this direction, but at full color emulation the
results were disappointing, even when using the blitter asynchronously.
They were nevertheless not that disappointing that I would have
abandoned this idea completely.

The chunky mode routines don't look very efficient at the first glance,
but I have not yet succeeded to make them significantly faster. I have
tried to replace the pixel color conversions by table lookups, but this
even got slower.

So, if the graphics can't be made faster, how about the processor
emulation? I have also thought about this and tentatively replaced the
opcode jump table model by a model with opcode routines of 256 bytes
each which are called directly using the opcode value as index. But
although this model allows fetching the second instruction byte without
additional overhead, it was slower than the current emulation that
seems to be using the cache more efficiently. CPU transcription is
impossible because of the intensive use of self-modifying code on the
C64.

In my opinion, accuracy is more important than speed, so removing
features out of the emulation would be out of the question.

1.15 The source

For those confused by the numbers: A small description of all modules:

```
  1541.asm      - File access and IEC routines
  6510.asm      - The processor
  6526.asm      - The CIAs (in/output, keyboard, timers)
  6569.asm      - The VIC (graphics)
  6569mono.i    - Routines for monochrome graphics
  6581.asm      - The SID (sound)
  c2p4.asm      - Chunky->planar conversion
  ExpTable.bin  - Included by 6569mono.i
  FrodoPrefs.#? - GadToolsBox user interface of settings window
  Frodo_rev.#?  - Version information
  Main.asm      - The main module
  Strings.#?    - For localization, created by CatComp
  Boopsi.o      - An object file from GadToolsBox to be linked in
```

The calls for assembling and linking are:

```
  PhxAss Main.asm INCPATH Include: OPT !
  PhxAss 6510.asm INCPATH Include: OPT !
  PhxAss 6569.asm INCPATH Include: OPT !
  PhxAss 6526.asm INCPATH Include: OPT !
  PhxAss 6581.asm INCPATH Include: OPT !
  PhxAss 1541.asm INCPATH Include: OPT !
  PhxAss FrodoPrefs.s INCPATH Include: OPT !
  PhxAss c2p4.asm INCPATH Include: OPT !
  PhxLnk Lib:RXstartup.obj Main.o 6510.o 6569.o c2p4.o 6526.o 6581.o
    1541.o FrodoPrefs.o Boopsi.o Lib:small.lib TO Frodo
```

Note that PhxAss V4.0 or later must be used for assembling.

1.16 Copyright and distribution

The program "Frodo", this manual and the source code may be freely
distributed as long as they remain unchanged (archiving and packing is
allowed) and all files are included. You must not make any profit by
selling Frodo, especially the price of a disk containing Frodo may not
exceed US$ 5,- (or equivalent amounts in other currencies). Please feel
free to distribute Frodo over bulletin board systems and networks and
as part of shareware/freeware CD-ROMs.

Anyone using this program agrees to incur the risk of using it for
himself. In no way can the author be held responsible for any damage
directly or indirectly caused by the use or misuse of this manual
and/or the program.

The rights on the source code remain at the author. It may not - not
even in parts - used for commercial purposes without explicit written
permission by the author. Permission to use it for non-commercial
purposes is hereby granted als long as my copyright notice remains
in the program. You are not allowed to use the source to create and
distribute a modified version of Frodo.

The user interface of the program was designed with GadToolsBox
© Copyright 1991-1993 Jaba Development.

Parts of the program are © Copyright 1992-1993 Jaba Development.

1.17 Bug reports

If you find a bug or a misfeature in Frodo, or have an idea how to
make some things better, then please drop me a note so I'll be able
to improve Frodo in the future. My address can be found
here
.

I don't think the emulation can be made much faster without changing
the concept but I'll be happy to find someone showing me how to do it.

Besides, I'm looking for translations of Frodo.catalog in other
languages (esp. finnish and swedish). An elvish version would be
appropriate, too. :-)

1.18 Credits

The following persons deserve special thanks from me as they made a
significant contribution to the development of Frodo.

- Frank Wille <phoenix@komet.guru.sub.de>, whose PhxAss was used for developing Frodo
- Jaba Development, whose GadToolsBox developed Frodo's user interface
- Peter McGavin <peterm@maths.grace.cri.nz>, who wrote the chunky-to-planar conversion
- Andreas Boose <a.boose@ldb.han.de> and Marko Mäkelä <marko.makela@hut.fi>, who provided me with precious information on the VIC and on the C64 in general
- Markus Winklbauer <sysop@ng-box.wwb.sub.de>, for the docs for the Picasso card
- Teemu Suikki <Teemu.Suikki@lut.fi> for the idea of skipping frames
- J.R.R.Tolkien for the suggestion for the name of the emulator
- All the people who have sent me suggestions and comments

1.19   The author

```
              My address is:

  Christian Bauer
  Langenaustr. 65
  56070 Koblenz
  Germany

EMail:
  cbauer@mzdmza.zdv.uni-mainz.de  (More reliable but polled
                                   less frequently)
  cebix@ng-box.wwb.sub.de         (Polled daily, but slower connection)

Questions, criticism, suggestions and
                bug reports
                  are always
welcome. EMail is preferred.

Frodo is not a shareware program, but I won't reject any
gifts. :-)
```

1.20   History

```
This emulator started as a player routine for C64 music (therefore
the SID card) until I saw A64 V3.0 and was so disappointed of it
that I began to write a "real" C64 emulator (around christmas 1994).

V0.4  05.Jan.1995
  - First released version

V0.5  10.Jan.1995
  - Starting Frodo on a 68000 correcly gave a requester but then
    the computer crashed. I told you this only works on 68020+ :-)
  - Removed two enforcer hits in the memory configuration routine
  - Sprite cycles are now calculated correctly
  - Switching off a sprite in the middle of its display blanked it
```

immediately. Now it is drawn completely.
  – Graphics speed for EGS/Picasso improved by means of a line
    buffer (20% faster), other small optimizations to the graphics
  – VIC interrupt flag is now cleared on every write access to $d019
    (a hack for RMW commands doing two write accesses)
  – CIA timers B are now capable of counting underflows of timer A
    (for "Ballblazer" :-)
  – PLP and RTI clear the break flag
  – Completely new 1541 emulation, 4 drives
  – CPU cycle counter heavily optimized (again 25% gain in speed)

V1.0 25.Jan.1995
  – PHP now sets the break flag on stack
  – HandleRESET erroneously set the I flag (don't trust a data sheet,
    even if it's from the manufacturer...)
  – A RESET will set the VIC bank to 0
  – All illegal opcodes implemented
  – New algorithm for the VIC's VC that is more close to the real VIC
  – The VBlank routines now jump back to Periodic6569 as there was
    a non-existant -1st line
  – TOD clocks implemented
  – ECM+BMM, ECM+MCM and ECM+BMM+MCM now correctly display a black
    screen
  – Access to SID card is now disable-able (!? :-)
  – Some bugs in 1541 emulation corrected
  – Use of IEC devices over an adaptor cable is now possible
  – Selecting the directories for the 1541 emulation can now be done
    with a directory requester
  – Turning off the screen using bit 4 of $D011 is working
  – Setting for "raster frequency" removed
  – Interrupt handling completely rewritten
  – Some internal optimizations
  – During a reset, the CBM80 signature is now only overwritten if
    it is present
  – Frodo now works on non-AGA machines
  – Sprite-sprite collisions implemented
  – Swapping of Y and Z keys is now a settings item

V1.1 was an internal test version with bitplane graphics
     (removed again in V1.2)

V1.2 09.Feb.1995
  – Average graphics speed is now higher, especially in multicolor and
    bitmap modes
  – New, very fast monochrome mode
  – Sprite routines are more efficient. No processor time is wasted
    when all sprites are off.
  – Sound emulation with 6581sid.library when SID card is disabled
  – Minimal screen resolution is now 384$\times$272
  – Sprite collisions can be turned off
  – 1541 emulation now handles sequential files and access modes and
    ignores drive numbers and the '@'
  – Frodo is now multilingual by means of "locale.library"
  – Overscan resolutions can now be used
  – Modes like PAL:Lowres are now selectable under OS2.x
  – Option to translate '/' in file names to '\' and vice versa
  – Joysticks are now only polled in the VIC vertical blank routine

  – Amiga graphics routines now also use double buffering
  – Option to accelerate the memory test on reset

V1.3 22.Feb.1995
  – Bug in Close1541 corrected (could crash on quit)
  – CyBERgraphics support
  – 1541 emulation now handles wildcards
  – Option to skip frames implemented

1.21 The future

These are some of the things on planned for the next versions:
· Faster graphics on Amiga chips (theory is ready, but the code needs
  to be heavily optimized to compare with the current method)
· A built-in machine language monitor that pops up by hotkey
· Possibilities to save and restore the state of the emulation
  (memory and register snapshot)
· Direct access to .d64 files (is this urgent?)
· Sound emulation using "PlaySID"
· Usage of Amiga printers and modems (rather unlikely, as Frodo has
  not been written to run PrintShop on it :-)
· Cycle based emulation. I have ideas for that and hopefully the speed
  won't go down as drastically as I expected at first. This would of
  course be switchable or there would be two versions of the emulator.