Descent Documentation

Christian Sauer

COLLABORATORS			
	TITLE : Descent Documentation		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Christian Sauer	April 12, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Descent Documentation

1.1 Welcome to Descent I V1.5 r0.25

Descent I V1.5 r0.25

Port by Christian 'Ville9' Sauer

1.2 Some words from PARALLAX

```
Legal Stuff:


THE COMPUTER CODE CONTAINED HEREIN IS THE SOLE PROPERTY OF PARALLAX
SOFTWARE CORPORATION ("PARALLAX").  PARALLAX, IN DISTRIBUTING THE CODE TO
END-USERS, AND SUBJECT TO ALL OF THE TERMS AND CONDITIONS HEREIN, GRANTS A
ROYALTY-FREE, PERPETUAL LICENSE TO SUCH END-USERS FOR USE BY SUCH END-USERS
IN USING, DISPLAYING,  AND CREATING DERIVATIVE WORKS THEREOF, SO LONG AS
SUCH USE, DISPLAY OR CREATION IS FOR NON-COMMERCIAL, ROYALTY OR REVENUE
FREE PURPOSES.  IN NO EVENT SHALL THE END-USER USE THE COMPUTER CODE
CONTAINED HEREIN FOR REVENUE-BEARING PURPOSES.  THE END-USER UNDERSTANDS
AND AGREES TO THE TERMS HEREIN AND ACCEPTS THE SAME BY USE OF THIS FILE.
COPYRIGHT 1993-1998 PARALLAX SOFTWARE CORPORATION.  ALL RIGHTS RESERVED.

We make no warranties as to the usability or correctness of this code.

About the Source Code:

Included is almost all the source code to Descent 1, ver. 1.5.  We removed
all code to which we did not own the copyright.  This mainly involved low-
level sound and modem code.

Kevin Bentley, maintainer of the Descent Developer Resources and author of
Kahn, has agreed to support this code base through his website at
www.ladder.org/ddr.  If you have questions or comments about the code, please
go there.  Please do not contact Parallax Software, its employees or its
affiliates.

The following tools were used in the development of Descent:
Watcom C/C++, version 9.5
Microsoft Macro Assembler, version 6.1x
Opus Make, version 6.01

Have fun with the code!
Parallax Software Corporation
Jan. 12, 1998

Kevin Bentley’s notes:

As included in this zip file, the source should compile and run with the
following:

Watcom 10.6
Miscrosoft Macro Assembler 6.11
GNU (or compatible, such as Microsoft, or Borland) Make utility

To build the executable, you should be able to just run maked1.bat.
Descentr.exe should be built in the main subdirectory.

When you unzip the archive, make sure you use the -d option to expand  ↩
   subdirectories, or it will not
compile.

The compiled code will run fine with no sound, and no serial port functionality
```

(This means no VR equipment that uses serial interfaces, and no modem or serial play games.)

If you port this to another platform, please drop me a note kevinb@stargatenetworks.com , I would be happy to post a copy of your port on the web site. *PLEASE* make an attempt to keep the existing code compilable (ie. use #IFDEFS)), so other people can benefit from your work.

To build the editor, change the CCFLAGS variable in all the makefiles to "CCFLAGS = /dNETWORK /dEDITOR" and rebuild all files. You will need the files in editdata.zip (included in this archive) to run the editor. Also, I was only able to get the editor to run using the -nobm command line arguement.

Interview with Matt Toschlog and Mike Kulas regarding the release of the Descent source code:

Q: Why are you releasing the source code?

Mike: The main reason is we figured a lot of people aspiring to work in the game industry would like to see production "quality" code. There are two reasons for this. One, you can learn a lot by looking at working code. Two, people will see that you can write a decent game without writing beautiful code.

Matt: That's a good thing?

Mike: Not really, except that it might make people think writing production quality code isn't that hard.

Matt: That's a good thing?

Mike: Well, not really, unless they learn that they have to focus on designing a brilliant game, rather than writing brilliant code.

Matt: Ah, that's a good thing.

Mike: Yeah, I don't think I understood that until we started working on Descent. At first all I cared about was writing technically good code.

Matt: Then we ran out of money and all we cared about was finishing our game.

Mike: Right. Our code got ugly, but our game got done.

Matt: Writing code is easy. Finishing is hard.

Q: Is there any code you're particularly proud of?

Matt: No.

Mike: No.

Matt: I'm most ashamed of the vector intersection code.

Mike: Yeah, that's pretty bad.

Matt: Mike's AI code is the worst I've ever seen.

Mike: Only because you program with your eyes closed.

Matt: I'm proud we managed to finish without going bankrupt.

Mike: Yeah, finishing is all. Who said that?

Q: Seriously, what should people look at?

Matt: I really don't know. It's not like there's a lot of reusable code in there. Some low level stuff could probably be used. The vector-matrix library is probably fairly instructive.

Mike: The AI is an example of how not to write an AI system.

Matt: You're doing it the same way in FreeSpace, right?

Mike: Yeah.

Matt: The texture mapper is worth looking at, not that you'll be needing a software texture mapper much longer.

Q: You're including the editor, right?

Mike: Yeah, people will be able to use our editor. On the whole, I doubt it's any better than DMB2. And it doesn't run under Windows.

Q: What enhancements do you hope to see people make?

Matt: If an aftermarket sprouts for it, we'd be very happy. People ask for lots of little features that we just don't have the time to add.

Mike: People could probably roll in some of the D2 multiplayer features without too much difficulty.

Matt: I don't think we can guess what people will do. We had no idea how much would be done with third party levels.

Q: The license states that people can't use the code for commercial gain. What if some kid develops something and wants to recoup some of the cost through shareware?

Mike: We're not opposed to that in principle. They need to get written permission from us, though. And, it would have to be after the thing is done so we know what we're permitting to be commercialized, if you want to call it that.

Q: Any plans to release the D2 source code?

Matt: No definite plans. Though, I guess I don't see why all our source code wouldn't eventually get released.

Mike: Me, too. It loses commercial value in just a few years. And, releasing it brings us closer to our customers, which is a very good thing to do.

Q: Any final comments?

Mike: Yeah, have fun with the code.

1.3 What you need to run Descent

You'll need the following to be able to start Descent:

- 68020+ (no FPU required)
- AGA, CyberGraphX or Picasso96
- AmigaOS 3.0
- lowlevel.library V40+ (part of AmigaOS 3.1)
- asyncio.library V39+ (available on Aminet)
- about 8MB FastRAM or 2MB FastRAM & VMM
- Descent PC V1.5 REGISTERED ".pig" and ".hog" files

That's it.

1.4 What you can do with it

The current features include:

- Fully multitasks
- AGA support with fast C2P
- support for CyberGraphX/Picasso96 direct video access
- Support for different screen modes
- Virtual Memory support
- Keyboard support
- Fast asynchronous loading
- Debug support (only of interest when sending Bug-Reports)

See also
Future

1.5 How to install and use

First, get a copy of Descent I V1.5 for the PC. The CD-ROM Bundle "DESCENT I and II - THE DEFINITIVE COLLECTION"
is still available in computer game stores.

Copy the descent drawer on Disc one onto your HD and install the DescentAmiga executable over it.

That’s all - fairly simple, isn’t it?

1.6 How to use Descent

Descent can be started via Icon from Workbench or from Shell. Make sure, that the stack is set large enough,
as I haven’t compiled in any stack extension code. An amount of 300KB should be large enough (if the game crashes,
try to increase the stack).

One word about the used key mapping:

F11 is mapped to the DEL key and F12 is HELP. That’s because all Keypad keys are used by the game and I wanted to
have F11/F12 near F10... I hope, you can live with it.

Some of the key commands are:

```
<AMIGA><F>                      Enable the frame counter

<AMIGA><C>                      Show the cheat menu
```

There are no tooltypes yet. The currently added/tested commandline arguments are:

```
-fpu                            Enable FPU routines
        The default is off with a 68020-40 and on with a 68060

-monitor <monitorname>          Opens the screen with the specified monitorname.
        The default is to use ntsc.

        examples:
        -monitor cybergraphx
        -monitor p96-0:picassoiv
        -monitor multiscan
        -monitor pal

-directgfx                      Make use of direct accessing the videomem on a
        CyberGraphX/Picasso96 screen.

-nopointer                      Disable the mouse pointer

-320x100                        Open a 320x100 screen during game (preferably with NTSC)

-320x240                        Open a 320x240 screen during game

-320x400                        Open a 320x400 screen during game

-640x400                        Open a 640x400 screen during game

-640x480                        Open a 640x480 screen during game
```

-nodoublebuffer	Disables double buffering (default is on when using a screen bigger than 320x200)
-nodoscheck	Disables the AmigaOS version check.
-nomemcheck	Disables the memory checking
-notitles	Skip the titles screens
-autodemo	autoload the demo at start

1.7 What has changed

I started on 30th January 1998 with porting directly the PC source, as I don't want to make
a port of a port ...

0.00 - 0.05 12/02/98

- rewritten Makefiles for GNU make and changed directory structure a bit
- replaced all DOS/conio stuff with dos.library function calls
- moved x86-Assembly into C by peeking into the Mac-Source
- named every unnamed struct/union
- packed nearly every structure as this is the way Watcom C/C++ handles them per default
- added TONS of byteswappes, inlined them and turned to assembly
- removed byteswap of ID-Strings in IFF-loader code ;-)
- ported error handling routines
- ported debug mini-system (disabled the exec.library/Debug() call for Int3() in this release)
- ported timer interrupt code via lowlevel.library/TimerInt
- ported keyboard interrupt routines via lowlevel.library/KBInt
- ported compressed file IO routines and changed some other places to use asyncio.library
- quickported graphic setup code, it's nasty right now and I'm blitting directly into videomem
- inlined fixmul(), fixdiv(), fixmuldiv() and sqrt() for better performance
- replaced all rand()'s and RAND_MAX's with own routines returning short instead of int
- cleaned up code a bit and added a few missing prototypes
- a lot more changes in nearly every file, mainly cosmetic changes ...
- made an icon and created this documentation

0.06 - 0.07 17/02/98

- rewritten loader code and got rid of all __attribute__((packed))s
- ported critical error handling stuff
- ported main() program entry point
- added support for different screen modes (no AGA though)
- ported graphic setup code and fixed automap bug

0.08 - 0.10 20/02/98

- rewritten and ported most of the canvas/bitmap stuff to support AGA ;-)
- introduced new automap bug (please don't use it right now) :-(

- disabled temporarly direct Cybergfx

0.11 - 0.15 27/02/98

- replaced FPU instructions with fast 32Bit integer routines
- fixed some gfx bugs
- fixed an Enforcer hit caused by fix_fastsincos()
- added support for direct video on CyberGraphX/Picasso96 screens
- finally ported the ModeX stuff in automap mode
- made some small optimizations

0.16 - 0.20 06/03/98

- fixed a small accuracy bug in the palette code
- fixed a bug in the demo playback code
- made automap autoscrolling when no 320x400 mode is available
- added a switch to disable the mousepointer
- turned fixedpoint library to assembly, using 64Bit integer math on 68020-40 and FPU instructions on 68060

0.21 - 0.25 12/03/98

- oops, direct video access was never used during game - fixed
- fixed a hashtable bug in the automapper and changed the algorithm to the faster one used in the Mac port
- PC keyboards don't create scancode zero, but so do AMIGA keyboards. Thus, taking zero
  as an error code isn't such a good thing - fixed
- added c2p routines by Aki Laukkanen, modified for being able to c2p at odd addresses
- changed the window shrinking stepsize to 32, for better chipmem-alignment
- turned the vector/matrix library to assembly, again using 64Bit integer math on 68020-40
  and FPU instructions on 68060
- added extra lores mode
- lots of other small things I've forgotten

1.8 What time will bring

- make pre-V1.5 .pig/.hog files and the shareware files work with it
- make a special version for Virge (already in the work!)
- joystick and mouse support
- support for ECS
- sound/music
- networking and modem support
- turn 3D engine back to assembly (partially done already)
- port the editor
- make a version for PowerPC
- localize it
- many other things that might be implemented, like Descent in a WB-Window, Overlay, IGlasses!...

Where's the source? you might ask. Well, I might include it in one of the following versions,

as soon as it is in a bit better state (mainly remove all the terrible warnings when compiling and
remove all the unused PC-stuff...).

1.9 What, How, When...?

Q: Descent is crashing my computer when I start it - I have a graphicsboard installed!
A: Most unlikely. I think, that it does only open a NTSC-screen which can't be seen when using the monitor
on the graphicsboard. Use the -monitor option to use a different screenmode.

Q: Why is Descent so much slower when using -320x240?
A: Descent uses double-buffering on any screen bigger than 320x200. Use the -nodoublebuffer switch to
disable this.

Q: Descent exits with "Not Enough Strings" or something like that!
A: You're using pre-V1.5 .pig/.hog-files which currently don't work with Descent. A solution for this
is in the works.

Q: Why do the walls look so curious when I fly near them? Am I stoned?
A: No, you're not. ;-)
The texture mapper currently uses fast, but less accurate interpolation which causes this.
I'll fix this, when I turn it to assembly (still is in pure C).

1.10 About myself

If you find any bugs (for sure ... :-)), have new ideas or something else,
you can send an e-mail to

sauer@informatik.uni-wuerzburg.de

Sorry, no Web-Page yet.
The newest version of Descent will always be on Aminet.