

mk3d.doc

COLLABORATORS

	<i>TITLE :</i> mk3d.doc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 17, 2023	

REVISION HISTORY

<i>NUMBER</i>	<i>DATE</i>	<i>DESCRIPTION</i>	<i>NAME</i>

Contents

1	mk3d.doc	1
1.1	mk3d.doc	1
1.2	Introduction	1
1.3	The Author	2
1.4	The Documentation	3
1.5	Distributability	3
1.6	Disclaimers	3
1.7	The Program	3
1.8	Using mk3d	4
1.9	INPUT	5
1.10	OUTPUT	5
1.11	ERRORS	5
1.12	SIMPLE	6
1.13	The Input File	6

Chapter 1

mk3d.doc

1.1 mk3d.doc

```

                                X          X          X          X          X          X
8!!AK4Dm!!AK4Dm!!AK4Dm!!AK4Dm!!AK4Dm!!AK4Dm!!AK4Dm!!AK
ry=8>p+Ry=8>p+Ry=>pj+Ry>p"jy>' [jyD5>[jyD$5>J[jy$5>J[jy
sM?&E/)_M?&E/)_M?E/) j_?E/@) ?Ef@) ]g?f@a)g??ff@)Og?ff@)O
yY-PM-e>Y-PM-e>Y-M-e>Y-M-Be-M-Bw"e-M-B"e)-MW-"e4)MW-"e
\7.\<.Ok7.\<.Ok7.<.+k7z<.p+z<,p+MHZ,px+Hz{,gp+sH{,gp+s
2$!YZ,M|$!YZ,M|$!Z,pM|$Z,Sp$Zvjps$aXZjps$a[XZFjps[XZFjps
=D)<qGyuD)<qGyuD)<qGyuD)<qGyuD)<qGyuD)<qGyuD)<qGyuD)<q

```

So, ya wanna make 3-d pictures using random stereo-whatsis technology, but using a text format so your BBS friends can enjoy it. Well... this is the program for you.

Another fine product of The Cheese Olfactory Workshop.

Introduction

The Program

The Input File

1.2 Introduction

Believe it or not, the basic idea of how to make these images has been around for several years. According to Magic Eye "A New Way of Looking at the World" (Copyright 1993 by N.E. Thing Enterprises, ISBN: 0-8362-7006-1):

"During the 1960s, Dr. Bella Julesz was the first to use computer-generated 3D images made up of randomly placed dots to study depth perception in human beings. Because the dot pictures did not contain any other information, like color or shapes, he could be sure that

when his subject saw the picture it was 3D only!"

Today, however, the technology seems to have gone crazy. The book mentioned above contains several very nice images you can freak your eyes out over (the rose seems to be a particular favorite amongst most of the people I know).

Of course, trying to transmit something as detailed as one of THOSE pictures would require bundling the resulting stereogram as one of your favorite little file standards (pick a standard.. ANY standard) uuencoding the monster, then hoping the person on the other side has the software necessary to undo the mess.

Well.. that's fine.

This program doesn't come close to the detail you can get with a really nice program that actually uses real dots, instead of the synthetic ones we're going to be working with <snort>, but this software will let you transmit your little creation over E-mail, to a printer, through your cat's whiskers, and to Alpha-Centari. In short, this program can create an ASCII 3-d image. And we all know how interchangeable ASCII is. Don't we?

The Author

The Documentation

Distributability

Disclaimers

1.3 The Author

I don't have the slightest idea who originally wrote this program. I caught a post off Usenet one day gave away some C code for a program called 'mk3d' (which, at least in name, I stuck with). There was a binary with it called mk3d.exe, which meant, to me, that it had been compiled for MS-DOS <plegh>.

However, since I'm an anal-retentive programmer with too much time on my hands this vacation, and I don't feel like messing with my other pet projects, I decided to convert the C source to Amiga E, and add a bunch of nifty features (well, a few anyway). After becoming bald, I managed to get the damned images to pop up the way they should. E is a nice language, but I swear, trying to convert from C can be a damning experience.

As for myself, I'm merely a college student, 26 years old, working towards my Bachelor of Arts in Music (emphasis on Composition). My senior project and its recital will be due by 5 Feb 95, by which time, hopefully, I will have released the .MID versions for the general public to scorn me over (my project has to be a composition, the recital of which will be performed by an Episcopal church choir I've been working with, and their organist.. pretty good performers, really).

The computers thing is something I picked up as a kid, and never really quite let go of.

1.4 The Documentation

The documentation for this program will hopefully be more detailed than the documentation I got for the MS-DOS version of mk3d. I converted it over to AmigaGuide format for those who like that, and put in a whole bunch of BS (like what you are reading now) for those who are desperate to read stuff.

My e-mail address is jvanriper@uncavx.unca.edu (at least, until I finish school).

Special thanks to the author of text2guide (Stephan Sürken). I wish I could kiss you.

1.5 Distributability

This software is freely distributable. Even the source code has been released. I am returning it to the public, since I got it from the public.

Heed my example.

1.6 Disclaimers

All applicable disclaimers apply. You didn't pay for this, so don't come screaming to me that you've overwritten your s:startup-sequence file with a stereogram, or that your big proposal that you've been slaving over for five years doesn't look very pretty in 3-d. This software (as far as I can tell) is pretty stable, and should do its job provided you do your job in protecting your system from yourself and others.

In any event:

I, Joseph E. Van Riper III, and my company, the Cheese Olfactory Workshop, refuse to accept responsibility for the results of this program in any way, shape, or form. Use of this program assumes responsibility on the user's part.

1.7 The Program

mk3d was originally written in C, and compiled for MS-DOS. The code has been re-written for Amiga E, thus the binary is smaller and better than the

original C counterpart's binary.

Since this was written in E, you don't need to worry about stacksize, and arguments are handled according to the standard AmigaDOS way (ReadArgs). As far as I know, this program will run on any Amiga system.

Original .EXE executable for MS-DOS:

```
-----rw-d    18368  02-Sep-93 15:56:56  mk3d.exe
```

Amiga E executable for AmigaDOS (with added bells and whistles):

```
-----rwed     7616  08-Jan-94 15:05:46  mk3d
```

Now.. given these stats, would you want an Amiga or an IBM-PC clone?

Using mk3d

1.8 Using mk3d

As with any program using ReadArgs, using a question mark as an argument \leftrightarrow will call up a template that might help you figure out the option you're looking for:

```
-----
> mk3d ?
```

```
IN=INPUT/A,OUT=OUTPUT,ERR=ERRORS/K,S=SIMPLE/N/K:
```

```
-----
Upon getting this, you can type another question mark for more detailed instructions:
```

```
-----
IN=INPUT/A,OUT=OUTPUT,ERR=ERRORS/K,S=SIMPLE/N/K: ?
```

```
Usage: mk3d IN "filename" [OUT "filename"] [ERR "filename"]
        [S "number"]
```

IN specifies a mandatory input file to read for a template.
 OUT specifies an optional output file to write.
 ERR specifies an optional error file to write (instead of stderr).
 S specifies how simple the characters should be, by this chart:

- 0 = Only uppercase characters
 - 1 = Upper/lowercase characters
 - 2 = AlphaNumeric characters
 - 3 = AlphaNumeric characters with symbols (default)
 - 4 = Anything from value 1 to 255
-

For information about the IN file's format, please, read mk3d.doc.

NOTE: This program based on the same written for MS-DOS.

Modified somewhat heavily by Joseph E. Van Riper III
of the Cheese Olfactory Workshop.

Generally, those instructions should be enough.

INPUT

OUTPUT

ERRORS

SIMPLE

1.9 INPUT

Simply put, this refers to the file you want to read. This parameter MUST

be filled (although you don't have to actually type 'mk3d input fulu.stx'.. you could just say 'mk3d fulu.stx'). If this parameter is missing, you'll see the extra helps screen and a little error message. Read

The Input File
for more information.

1.10 OUTPUT

Simply put, this refers to the file you want to create. This parameter is optional.. if it isn't filled, stdout is used (therefore, you can redirect the output). This option doesn't really have to be spelled out.. it can be inferred as the name following the INPUT's filename (eg: you can type 'mk3d infile.stx outfile.asc' or 'mk3d infile.stx output outfile.asc').

1.11 ERRORS

This parameter lets you see whatever mk3d sees as a picture. This is in case you've got a bunch of funky weirdness you're trying to do with other characters that mk3d doesn't recognize, but you want to create a picture out of. Using this parameter shows what mk3d is using, changing unknown characters to spaces.

It's called 'ERRORS' because it originally redirected the stderr port. I decided that was a bad practice <grin>.

ERROR's default is NIL:.. You MUST use the keyword to change ERROR's

default (eg: 'mk3d infile.stx outfile.asc error CONSOLE:' is okay, but 'mk3d infile.stx outfile.asc CONSOLE:' will not work).

NOTE: CONSOLE: can be thought of as the stderr device.

1.12 SIMPLE

The SIMPLE parameter takes a number that will change the kind of characters that are used to generate the 3-d image. If you wish to use this parameter, you MUST specify it (eg: 'mk3d infile.stx outfile.asc s 0' but not 'mk3d infile.stx outfile.asc 0' -- note also, 's' is a shortcut for SIMPLE).

0 - Yields only uppercase characters. This might be handy for those potentially existent people who are limited to modems that cannot output lowercase characters... or for similiar reasons. Very quick execution.

1 - Yields upper/lowercase characters. Any character in the alphabet, whether lowercase or uppercase, may be used to create the images. Medium execution.

2 - Yields AlphaNumeric characters. Upper and Lowercase characters of the alphabet, as well as numbers, can appear. Slow execution.

3 - Yields as 2, but may also have various punctuation characters. Very quick execution. Also the default.

4 - Yields anything from character 1 to character 255. Very hard to display for most terminals. Definately would require a special program to read. In fact, this is almost completely unusable, but is included in case someone figures out a way to deal with it. Remember, at the end of each 'line' is a '10' character.. although the 10 character itself may be used as a character according to this setting. Your program would have to know when to wrap the new line according to how many columns you're supposed to have in the image.

The more variety in characters, the less likely a chance for ambiguity in the imagine process. That's why I even bother with an option 4.

NOTE: if you choose a number higher than 4, the program defaults to 3.

1.13 The Input File

If you want to create your own images, you have to follow a special format.

The first line contains a number that represents the 'Gramwidth', or the maximum number of columns to go across the screen (to put it simply). So, if you want something that can be viewed by most terminals, choose '79'. If you have a compressed type on your printer, and you want to print out a large 3d image, you can figure out how many columns are available on the printer and use that. Or, perhaps you have a nice font on your Amiga that

you can use.

However, the Gramwidth must be between 1 and 512. How someone thinks they could make a nice stereogram with only 3 columns is beyond me.

The next line is the Xdepth value. This number comes between 5 and 64, and must be less than half of the Gramwidth. Which, using math, means that you must really have a Gramwidth of 11 - 512... but I programmed this part according to the original code. Sloppy programmers...

I haven't really played with the Xdepth values, but with a Gramwidth of 79 its been recommended to use an Xdepth of 16. I suspect this would control how much depth there is between layers.

After this, you can actually create the image you want. The only characters that mk3d looks for, from this point on, are carriage returns (which, obviously, goes to the next line), and numbers from 1 to 9. Everything else is considered 'background' characters, and ignored.

The characters 1-9 represent different layers. A '1' will sit on the layer that's closest to the background, but just above it. '2' sits just above a '1'... and so on down the line, until '9' is the very top-most layer you can work with. Therefore, careful readers will note that you have 10 layers to work with! That's as deep as it goes with this program.. sorry.

Examine the different input files included with the archive to get a feel for how the program works.
