TREK SETUP INSTRUCTIONS

This document describes all sorts of nifty things you should know before you start to muck around with the trek source code. Please read them carefully.

1. MAINTENANCE

1. There are a number of shell files which you may use to maintain the system. "Prtrek" produces a copy of the source code. It pipes its output to lpr and runs in background. "Comp" compiles up to nine source modules and leaves them in .o files. "Compile" is the same as "comp" except that it loads after compiling. If stated without any arguments, it loads from .o files. "Compall" compiles all the .c files into .o files, but does not load. It redirects its output to the file "output". To recompile the entire system, type

```
compall
compile
```

2. Main.c contains a variable called "Mother". This is initialized to the result of the "getuid()" call for the maintainer of trek at your installation. Only Mother is allowed to set trace flags and run the game at other than the default priority.

3. Speaking of priorities, trek eats up a lot of system resources. Hence, it normally runs at a very low priority. This makes it almost impossible to play if the system is loaded. However, the -pN flag sets the priority to N, which makes it possible to debug when the system is loaded. The default priority is set by a #define of PRIO, which is set to 10 in the default system.

4. Trace information is provided which may be useful in debugging things in the system. If you are in a bad way for space, comment out the #define xTRACE which appears in trek.h. This will cause the trace stuff to not occur in the object.

5. The version of trek released to you is compiled with the -f flag (for no floating point) and should work without problems on your machine. You can edit out the -f flag in "compile" if you have floating point hardware on your machine so that it will take less space.

2. THE PORTABLE C LIBRARY

The portable C library was used to do I/O in trek. Unfortunately, the version which we had at Berkeley had a number of small bugs which caused trek to do bad things at times. For some unknown reason (temporary insanity perhaps) I rewrote the portable C library. This version is much smaller than the old version and has cleaner code. It also works right (???). However, there are a few minor differences which you should be aware of.

1. Scanf no longer ignores the noise characters "\n", "\t", and space in the format string; i.e., these characters now require a match in the input stream.

2. A variable f_log has been added which is the file descriptor of a "log" file. If f_log is greater than zero a copy of everything read from the standard input and written to the standard output is written in the file f_log.

3. DISCLAIMERS

1. Frankly, I am getting pretty sick of playing this game. Hence, the version which you get may have several bugs in it; I freely admit that it is probably buggier than some previous versions. Sorry about that.

2. Along with being buggy, the game never had quite everything implemented that was originally intended. If you see things that look weird, that may be why. There are even some features which I have taken out (like ghost starsystems) upon deciding that I didn't have the energy to implement them correctly.

4. REQUESTS

There are several things that I would like to ask of anyone who does work on the source code.

1. Please let me know of any bugs which you find in the code, and any fixes which you may have. Other copies will probably be going out to other people later, and it would be nice if those copies where less buggy. Also, I would be interested in hearing about any enhancements of the game which you might install.

2. Please note that I have a distinct coding style. I feel that it is cleaner and easier to read than a more casual style. If possible, please stick to it, especially if you end up sending tapes back to me. This goes along with my whole belief in clean code: I ask you to please avoid obscure code whenever possible. If you throw some in, please don't let me see it. It just depresses me.

3. Unfortunately, the game is huge. There are many neat things which could go in, if there were only enough space. However, I have specifically not gone to seperated I/D space. The main reason is that I would like future versions of the game to be 11/40 compatible.

5. SUGGESTIONS FOR THE FUTURE

If you happen to have more energy than I do, you may want to examine the following areas. These are things that I may get to, but don't hold your breath.

1. Frankly, making the portable C library work (even without bugs) was a bitch. I should have done the I/O in a more ad hoc manner. It is my intent to rewrite the I/O routines to bypass the portable C library entirely.

2. The routine "capture" is quite unclean. First, it should have a manner of selecting Klingons other than random, either selecting the most likely or asking the captain (probably best). It should either be fully implemented, which includes adding a "board" routine (half written, on some tapes as board.x) which sends a boarding party to forcefully take over the Klingon, or it should go out completely, which is probably what I will end up doing. When this happens, the transporter will go completely. It seems that the space may be better used for something which more directly enhances the game.

Well, that's about it. To get hold of me, write to:

Eric P Allman

Electronics Research Laboratory
University of California
Berkeley, California 94720

Happy trekking!!