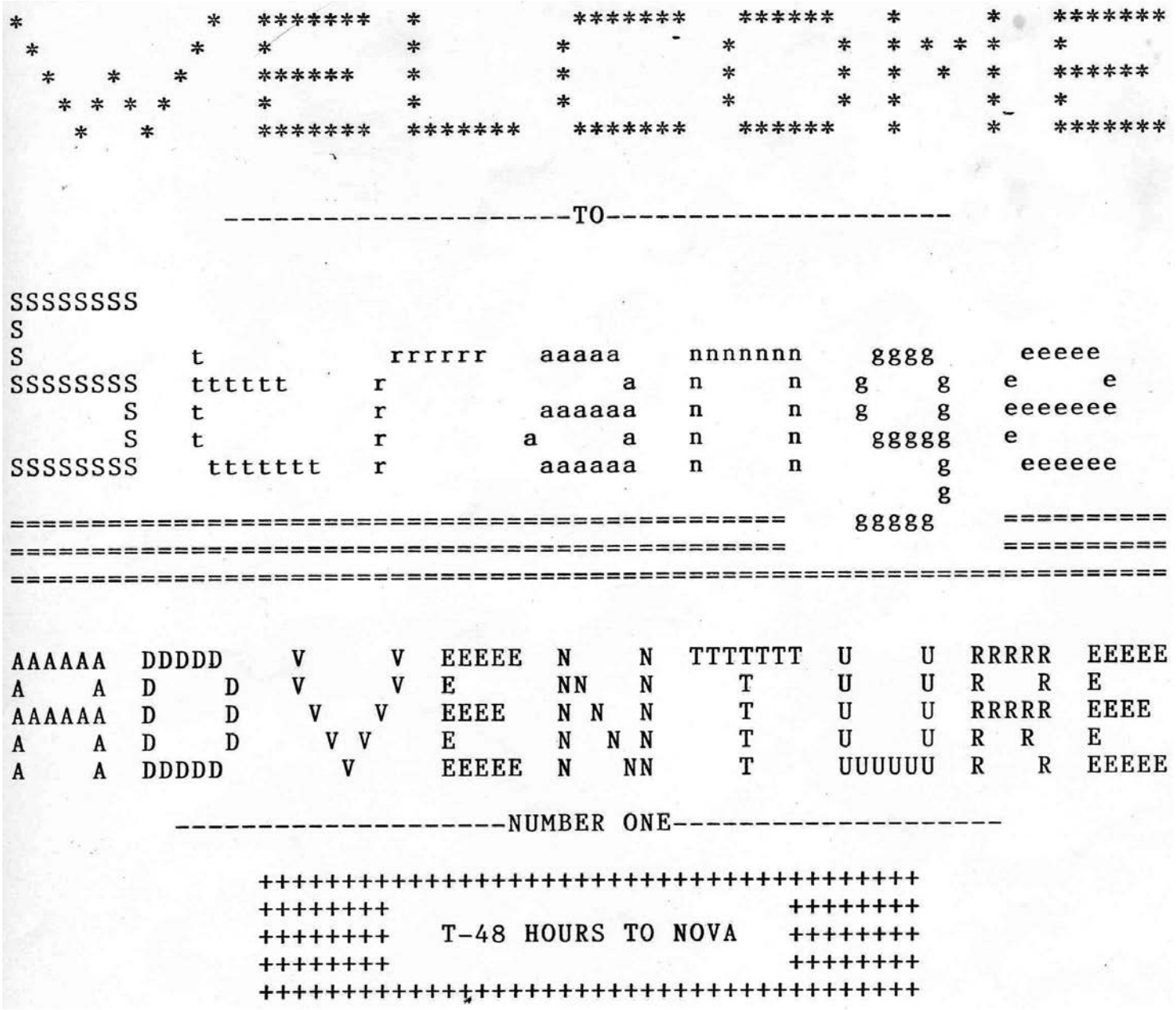

TO
SSSSSSSS
S
S t rrrrrr aaaaa nnnnnnn gggg eeeee
SSSSSSSS ttttt r a n n g g e e
S t r aaaaaa n n g g eeeeeee
S t r a a n n ggggg e
SSSSSSSS ttttttt r aaaaaa n n g eeeeee
g
ggggg
AAAAAA DDDDD V V EEEEE N N TTTTTTT U U RRRRR EEEEE
A A D D V V E NN N T U U R R E
AAAAAA D D V V EEEE N N N T U U RRRRR EEEE
A A D D V V E N N N T U U R R E
A A DDDDD V EEEEE N NN T UUUUUU R R EEEEE
NUMBER ONE
T-48 HOURS TO NOVA

SOME USEFUL INFORMATION

****

Please note: The disc supplied is 40 tracks.
The program will run on BOTH BASIC 1 AND BASIC 2.

1) - Select '40 track mode' if using one of those switchable drives
2) - Insert the disc into the drive and LOAD by pressing SHIFT & BREAK

The list below is of all the programs and files on the disc.

$.STARTER L - Starter program containing title screen, game instructions, and LOAD / SAVE overlay. (Listing 4)

$.ADVEN L - Main adventure program. (Listing 3)

D.GEN-1 L - Location descriptions file generation program. Creates the file called "DESCRIP". (Listing 1)

D.GEN-2 L - Object descriptions/routes/verbs/words etc. file generation program. Creates the file called "FILE2" (Listing 2)

$.DESCRIP L - Location descriptions file.

$.FILE2 L - Object descriptions/verbs/words/routes file.

$.PTRprog L - Uses BPUT# to create a *EXEC file of the users own PTR# values for his location descriptions. (Listing 5)

$.!BOOT L - !BOOT file - *EXECuted by OPT 4,3. It CHAINs the STARTER program.

$.CFILE L - This is a *EXEC file which contains the necessary commands to allow the main program to complete itself. When the main program is loaded and you get the "PRESS ANY KEY TO BEGIN" message, press ESCAPE, and type *EXEC CFILE (return). Now sit back and let the game complete itself!!

*DRIVE 2 - Contains EDWORD files which you can, if you wish, use for editing. If you have an EDWORD release of 2.0 or above, you will need to convert them by using *EDW1TO2 "filename" and answer Y when you get the over-write question. You will need to unlock them first - i.e. *ACCESS "filename"

$.TPAGE L - Title page : $.PROBLEM L - Hints and answers
$.SYNOP L - Game synopsis : $.LDINSTR L - running instructions
$.COMPSOL L - Complete solution : $.INFORMA L - This information
$.ARTICLE L - My Article on creating RAF files for adventure games
$.VARS, $.VARS2, AND $.VARS3 - PROCEDURE AND VARIABLE LISTS

I have taken the liberty to find out how much space the programs would take up on the disc.
If you just use the FILES i.e. the descriptions are already created then "ADVEN", "STARTER", "DESCRIP", "FILE2" and "PTRprog" would take up 178 sectors. That is, 17 tracks and 8 sectors.
If you included the above programs but with the file GENERATOR programs - i.e. "D.GEN-1" instead of "DESCRIP", and "D.GEN-2" instead of "FILE2" then they would take up 185 sectors (18 tracks 5 sectors.)

****
* T-48 HOURS TO NOVA *
****

THE GAME SYNOPSIS

No one knew the vaccine could be poisonous in large quantities....

The mission was so mundane, so boring, but it was nevertheless assigned top priority. According to new, more accurate estimates, scientists had discovered that the sun present in the Beta-Taros system was going to explode into a supernova in approximately forty-eight hours, taking with it its five planets. One of the planets was an Earth type planet, and stationed on it was a virology research outpost. The staff numbered just five people, and they would have to be evacuated before the star exploded. Thus the mission was to remove all the personnel and their supplies.

As the auto-nav performed the delicately intricate manoevers of piloting the Interstellar craft Horizon into one of the outpost's orbital docking births, the base was contacted, and it was confirmed that they were ready to be picked up by the ship's planetary shuttle, which had sufficient cargo space on its 'basement' deck to accommodate all the researcher's supplies.

The shuttle lowered itself to the outpost's landing pad with four high powered thrusters. As soon as it was down, hydraulically extended fuel feed lines and automatic coupling rods telescoped from an open hatch in the wall, and locked onto it, fuelling it to add that extra degree of safety for the short return trip.

It was quickly done. The supplies, stored in old-style cardboard boxes, were brought on board the shuttle by robot drones. The research personnel then walked on board, and clearance was given by the outpost's computer to commence departure manoevers, as it retracted the ground support-systems 'extremities.' Launch thrust was verified by the shuttle's FCS computers, and then it took off vertically until it reached a height of five thousand metres, where slightly more power was added to the forward thrusters, causing the front-end to rise more than the back-end. Suddenly, the main 'J2' engines came on line, and in moments the blue of the planet's sky was replaced by the inky blackness of space.

It was at that point that an unexpected incident occurred. One of the cardboard boxes broke free from its mooring straps and impacted with the floor. The sound of a container breaking was clearly heard, and a thick yellow-green vapour rose from the box. The scientists seemed rather calm, explaining that the vapour was originally a vaccine, for some virus which you have never heard of - 'Mextrapoline D prime, strain 113.' The vaccine was extremely perishable, and to extend its life it had to be kept at cryogenic temperatures, where molecular motion slows down. They explained that the vapour was the rapid transformation of the liquid vaccine into a gas, because the cabin temperature was much higher than the temperature at which the vaccine liquefies.

Upon boarding the Horizon, the scientists started showing signs of dizziness. Moments later, they collapsed to the deck, dead.

The ship's computer ran a bio-scan of the corpses, and revealed massive amounts of de-composed haemoglobin.

You almost die from shock when the computer prints out the suspected cause of the deaths. Apparently, it was caused by the vaccine. Accessing the MediBase to search for known cures, the search proves negative. You then quickly run a bio-scan of yourself, to see whether or not anything is wrong. You are somewhat calmed when the computer informs you that you are a fit, healthy individual, with no signs of trouble.

The ship was designed for up to five crewmembers. However, since the mission was only to evacuate the research outpost's personnel, it was deemed that only one person need go on the mission. The flight would be controlled by the computer.

After making your way to the bridge, you press the button to activate the engines and begin the voyage home, to Earth.

As soon as the button is pressed, however, the ship is overcome with siren noises. The voice of the computer calmly announces, "Emergency. Fusion Reactor circuits 9, 10 and 11A overloaded. Fuel Rods have been ejected. Radiation leak in aft section of ship. Auto bulkheads have closed at section 48 and will remain closed until decontamination measures have been completed."

At the same instant, a detailed report appears on the VDU of your bio-scan. Apparently, the virus from the vaccine is present in your body, but because you have a very rare blood type, the virus has taken longer to become active, during which time your body has developed anti-bodies. When you read the last line you breathe a sigh of relief: THERE IS NO DANGER.

But suddenly you realise the emergency of the situation. You are now immune to the virus, but it is carried on the materials of the ship. The supernova will, when it occurs, obliterate the ship and spread the virus for thousands of light-years, around the galaxy. What if, by some chance, the virus gets to any inhabited planet? It would cause thousands of deaths....

Hurriedly, you make your way to the emergency bulkhead, to try and override the safety mechanism which keeps it closed. To your very great disappointment, nothing happens.

THIS IS WHERE YOUR ADVENTURE BEGINS. YOU MUST FIND A WAY TO REPAIR THE NUCLEAR REACTOR, AND PILOT THE SHIP INTO THE SYSTEM'S SUN TO DESTROY THE VIRUS. BEFORE IMPACT, YOU MUST ESCAPE....

****
* T-48 Hours to Nova *
****

Problems arising and Hints/Answers to solve

It must be remembered that each and every problem has a logical solution - however difficult to solve it may seem at the time. If you're still stuck on a specific problem after a long time, then you may consult the appropriate hint or answer which is listed below, but only after you're sure that the problem is impossible to solve for yourself.

1) - I can't find the Allen Key.
   Answer 14
2) - I can't find the Latex rubber tubing.
   Hint 12 / Answer 31
3) - I can't get into the garbage disposal airlock.
   Answer 2
4) - The scissors are too blunt.
   Hint 13 / Answer 29
5) - I can't get into the planetary expedition shuttle.
   Hint 6 / Answer 23
6) - I can't find the magnetic card.
   Answer 11
7) - I can't free the emergency shuttlecraft.
   Answer 7
8) - I can't stop rotation of the centrifuge hub.
   Hint 16 / Answer 22
9) - I can't find the Acetylene cylinder.
   Hint 15 / Answer 25
10) - I can't get into the elevator car.
   Hint 1 / Answer 17
11) - I can't mix the adhesive.
   Hint 8 / Answer 18
12) - I can't open the bulkhead.
   Hint 4 / Answer 24
13) - I can't find the Oxygen Tank.
   Hint 9 / Answer 27
14) - I die from the lurch when I lauch the shuttle.
   Hint 3 / Answer 19
15) - I can't get out of the garbage disposal airlock.
   Hint 10 / Answer 28
16) - I can't find the scissors.
   Answer 21
17) - I can't find the battery charger.
   Answer 20
18) - I die in the Radiation lock - even when I'm wearing the suit.
   Answer 30
19) - I can't activate the ship's engines.
   Hint 26 / Answer 32
20) - I can't find the crowbar.
   Answer 5

HINTS AND ANSWERS
***** *** *******

1) - Find something else that will fit then
2) - EXAMINE FLOOR just South of the Nuclear Reactor-Observation and Control Centre.
3) - Make sure you're secure then
4) - If you can't open it, how else could you negotiate it?
5) - Try MOVE SHEET, whilst in the Captain's Quarters.
6) - It needs to be lowered into the launch position.
7) - You need to cut through the pistons with the Oxy-Acetylene Torch.
8) - What do people normally mix pots of glue with?
9) - Oxygen is an essential for life support.
10) - You need something to help you stick to the walls.
11) - You need to move the cargo storage module.
12) - Where do people normally hide things whilst asleep?
13) - Well sharpen them, then!
14) - Try cutting the master command seat, then REMOVING what you find.
15) - If it's not plainly visible, it must be lost in an assortment of other objects.
16) - Are you sure the circuit is conducting properly?
17) - You need to prise or lever the hatch with the crowbar.
18) - You need the plastic spatula.
19) - You need to wear the seat-belt.
20) - Try examining the console in the botanical laboratory.
21) - They are inside what you find after examining the ceiling in the aft section of the spin module.
22) - You need to fit the circuit board to the bridge console, then press the green button.
23) - A broken connection lying around somewhere! Maybe this could be soldered!
24) - No, you can't. You need to somehow go THROUGH it, and that's all the help you'll get!
25) - Try searching the garbage in the garbage disposal airlock.
26) - Are you sure the power source is operational?
27) - For a backpack to be life supporting, it must contain an Oxygen tank! Try dismantling it.
28) - Wouldn't it be a good idea to use the suction cups?
29) - Use the grinding machine to sharpen them.
30) - Something must be wrong with it then! Look more closely.
31) - Try moving the pillow whilst lying on the bunk.
32) - The fuel rods need to be inside their receptacle tubes in the nuclear reactor.

T-48 HOURS TO NOVA --- COMPLETE SOLUTION

****

S, E, S, S, S, W, EXAMINE WALLS, OPEN CUPBOARDS, GET TIN, S, MOVE TABLE, GET ROPE, N, W, GO BUNK, MOVE PILLOW, GET TUBING, D, W, MOVE SHEET, GET CROWBAR, E, E, E, N, W, W, W, N, N, W, GET BACKPACK, N, EXAMINE WALLS, OPEN DOORS, TIE ROPE, THROW ROPE, N, D, EXAMINE FLOOR, LEVER HATCH, D, GET BOARD, U, U, S, W, W, S, EXAMINE WALLS, OPEN FRIDGE, GET BOTTLE, GET SACHET, N, E, E, S, E, N, E, GET IRON, W, S, S, S, W, W, U, GO SEAT, EXAMINE CONSOLE, FIT BOARD, PRESS GREEN.

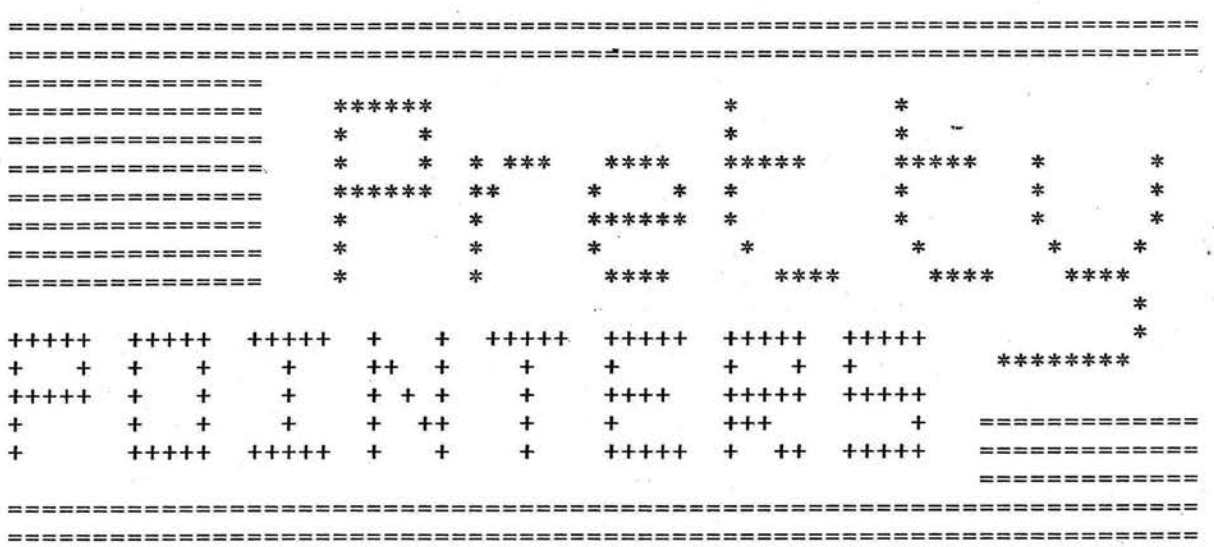
N, W, S, GET UNIT, N, E, E, S, GET ROLL, N, W, D, W, EXAMINE WALLS, REMOVE PANEL, GET SCREWDRIVER, W, U, U, GO MODULE, GET DOOR, E, MOVE MODULE, GET CARD, D, D, E, E, D, EXAMINE POD, FIT DOOR, EXAMINE FLOOR, REMOVE PANEL, D, EXAMINE PUMP, FIT TUBING, GET BATTERY, U, GO POD, EXAMINE PANEL, PRESS WHITE, E, EXAMINE CONSOLE, PRESS BLUE, E, GET EXPLOSIVE, W, U, E, E, E, E, E, N, N, N, EXAMINE CONSOLE, CHARGE BATTERY, FIT BATTERY, S, S, JUMP, USE SCREWDRIVER, W, GET SCISSORS, E, D, S, W, W, W, N, N, W, N, W, USE MACHINE, EXAMINE MACHINE, GET WIRE, MAKE BOMB, E, S, E, S, S, W, W, U, GO SEAT, CUT SEAT, REMOVE FOAM, GET KEY, N, D, E, E, E, E, E, N, N, W, N, DROP BOMB, PRESS RED.

N, N, N, U, EXAMINE WALLS, EXAMINE LOCKER, USE CARD, GET SUIT, D, S, W, W, N, W, W, GET MATCHES, W, W, W, N, N, E, U, GET VALVE, D, D, S, E, E, EXAMINE WALLS, EXAMINE PANEL, USE KEY, EXAMINE CIRCUITRY, E, E, D, E, GET SOLDER, W, U, W, W, SOLDER CONNECTION, E, N, E, MOVE TABLE, GET SPATULA, W, S, W, W, W, S, W, EMPTY SACHET, EMPTY BOTTLE, MIX ADHESIVE, CUT ROLL, STICK PATCH, WEAR SUIT, EXAMINE WALLS, PRESS ORANGE, S, GET RODS, OPEN HATCH, S, EXAMINE FLOOR, FIT RODS, N, N, PRESS ORANGE, E, E, S, D, D, GET SHOVEL, U, U, N, E, E, E, E, S, E, E, S, S, E, S, S, W, W, W, W, W, U, GO SEAT, PRESS YELLOW.

N, D, E, E, E, E, E, N, N, W, DIG, OPEN HATCH, D, GET CUPS, U, N, N, W, W, N, W, W, W, W, W, S, S, S, EXAMINE FLOOR, D, SEARCH GARBAGE, GET CYLINDER, USE CUPS, U, U, S, S, W, EXAMINE HYDRAULICS, DISMANTLE BACKPACK, MAKE TORCH, STRIKE MATCH, LIGHT TORCH, CUT HYDRAULICS, E, S, W, S, EXAMINE SEAT, WEAR SEAT-BELT, EXAMINE CONSOLE, PULL LEVER, PRESS BLACK.

-----THAT'S ALL FOLKS!

Martin Lee Briody / Tel:- [redacted]
[redacted] North Wales.
ARTICLE: "PRETTY POINTERS" PACKAGE TITLE: "T-48 HOURS TO NOVA"



Most people, when told that their micro comes with a glamorous 32K of memory, expect the full 32K to be available for programs. Unfortunately, this is not always the case. The Machine Operating System (MOS) takes up a sizeable chunk of it, and furthermore, if you have DFS fitted, even more memory gets gobbled up. In total you lose a staggering 6.25K.

The memory used for the screen memory map varies with different display modes, ranging from 20K to 1K. When we switch on the BBC, if PAGE=&1900 (i.e. DFS is fitted), we have only around 25K left for our programs and variable storage. Thus, it is impossible to write very long programs without encountering the dreaded "No room" error.

An example of a very long program is an adventure game, particularly because of the many arrays it needs for storing information such as the location descriptions, the game routes from one location to another, the objects the player is carrying and the game flags etc.

But by far the main 'space gobbler' in adventure games, is the text needed to describe the player's current location. The problem is not so pronounced with single line location descriptions such as those in games like 'Circus' and 'The Golden Baton', but in some of the better games like 'Wychwood', where more atmosphere is present, then it can cause serious problems.

Of course, there are some advanced text compression techniques which can be put into use to grab some memory back, but even with them, the concept of the adventure game cannot be explored to the full. Nevertheless, some excellent games are lurking around out there, in the midst of the software jungle.

Many people buy their micro as standard - that is with tape filing system only. This being the case, the majority of software writers write games using only the facilities of the TFS.

Upgrading to a disc drive, however, opens the door to many extra possibilities, including the ability to create adventure games with massive location descriptions, with each one being loaded in from disc when player movement occurs. The technique I am implying is known as Random Access Filing (RAF for short.)

Being disc users, this technique is immediately accessible to us, and all it takes is a little knowledge to put it into operation.

More ....

Martin Lee Briody / Tel:- [redacted]
[redacted] North Wales.
ARTICLE: "PRETTY POINTERS" PACKAGE TITLE: "T-48 HOURS TO NOVA"

So, what does Random Access Filing mean? Basically, it means that the computer will be able to instruct the disc drive to jump to and pick out any particular data item (in our case it is a location description) from a long file. It is so called 'Random Access' because it can jump to a data item anywhere on the disc.

The computer keeps track of which data item it is going to read next by use of a FILE POINTER. This pointer is assigned a numerical value by the computer and can be moved up and down the file by a special BASIC keyword - PTR#

Before we venture any further, it is important that I point out that although our file will be made of strings - and strings only, the end of one string does not signify the start of the next. Strings are stored in a special way in files. They always consist of the number of bytes in the string plus an extra 2, and the actual characters of the string occur in REVERSE order. The first byte is always written as &00. The second byte contains the number of characters in the actual string. This format is best illustrated by an example. If we take a typical location description, for instance 'You are in a forest", - would appear in the file as follows.

&00	&13	&74	&73	&65	&72	&6F	&66	&20	&61	&20
	-	t	s	e	r	o	f		a	
&6E	&69	&20	&65	&72	&61	&20	&75	&6F	&59	
n	i		e	r	a		u	o	Y	

There is nothing underneath code &20 because it is the code for a space. The second byte in the file tells us that the actual string's length is &13 (19) characters in length.

The syntax for using the PTR# command to set the file pointer to a specific value is 'PTR#<channel>=<value>', where the channel is the number allocated to the file when it was opened. The value can either be an actual number, or a numerical variable such as A or A%.

Now, we have a problem. It is easy to use the PTR# command to move the file pointer around and about the file at random, but we have to be careful to set it to the &00 preceeding all our strings, otherwise we will get a 'Type Mismatch' error. Also, we have to have all the PTR# values for each location description stored in an array in the adventure game. Going all through the file using *DUMP and noting down the values would be extremely monotonous and time-consuming. Fortunately, I am able to help. On this month's disc is a program called 'PTRprog' which scans a string file provided by you and *SPOOLs an ASCII file containing data statements of all the PTR values. To use it, follow the instructions below.

First, create your file of location descriptions in the normal way, i.e. by use of OPENOUT and PRINT#, then CHAIN 'PTRprog'. You will first be asked for the name of the *SPOOL file. Next, you will be asked for the name of the file containing all your location descriptions, and finally for the start-line number of your *SPOOL file. The program will then step through all your location descriptions and create a *SPOOL file of all the individual PTR# values. You can then *EXEC it back in. For more immediate access, I

More ....

Martin Lee Briody / Tel:- [redacted] North Wales.
ARTICLE: "PRETTY POINTERS" PACKAGE TITLE: "T-48 HOURS TO NOVA"

would recommend placing the values in an array, DIMensioned with the same number of cells as there are locations in your adventure.

Then, when you move location, the computer will look for the PTR# value contained within the array subscript which corresponds to the number of your new location, assign the file pointer value to what it finds, and loads in the new location description.

Simple.

Also on this month's disc, is a one-hundred location, futuristic mega-adventure, entitled "T-48 HOURS TO NOVA." The main program will take a few seconds to completely load, because all array data is contained within a file which is loaded at the start. This saves duplication of the array data from the basic DATA statement, which would be very wasteful on precious memory. The game is not a very difficult adventure - it was intended as something to 'while away the hours' - but there are nevertheless some taxing problems present.

GOOD LUCK ON YOUR MISSION, COMMANDER!

GETTING "T-48 HOURS TO NOVA" WORKING
****

(If you decide to include the file generator programs on the disc then the following note must be added to the end of my article, or in a 'blue column' down the centre of the page, as is quite common. If not then just issue the normal note about backup copying and the safety of the original etc.)

To get "T-48 HOURS TO NOVA" to work on your system, it will be necessary to create the data file for the location descriptions and the file for the other variables. To do this, follow the instructions below.

First, format a blank disc, then use the DFS *COPY command to transfer the following files to your new disc: "STARTER", "ADVEN", "D.GEN-1", "D.GEN-2", and "PTRprog."

Once copied, CHAIN "D.GEN-1". This is the location descriptions generator program. You will be prompted to enter a pre-formatted disc then press the space bar. Since there is already one present, you can do it straight away. The program will print out the number of each location description as it writes it to the file. The descriptions are coded in memory and each one has to be uncoded first. Creating the file takes quite a while so please be patient Once it has done, issue a *CAT command and somewhere in the directory should be a file called 'DESCRIP'. The generation program will have locked it by use of the *ACCESS command, to prevent accidental erasure.

You can now CHAIN "D.GEN-2". This program will create all the other data. Press a key when prompted to start creation. It makes the file called "FILE2", which will also be locked.

To be able to load the game automatically, you will need to *BUILD a !BOOT file. To do this, copy the following.

```
>*BUILD !BOOT (return)

      1 PAGE=&1900 (return)
      2 CHAIN "STARTER" (return)
      3 (escape)
```

Now type *OPT 4,3 to instruct the computer to execute the BOOT file when SHIFT and BREAK are pressed simultaneously.

All that remains to do is give it a fancy title e.g. ADVENTURE or SUPERNOVA.

You are then READY TO GO!

T-48 HOURS TO NOVA
****

MAIN PROCEDURES AND VARIABLES
****

Arrays:

OD$()	(78 subscripts)	- Object and examine descriptions
R$()	(100 subscripts)	- Route descriptions
V$()	(52 subscripts)	- Verb table
O$()	(80 subscripts)	- Word (noun) table
D$()	(1 subscript)	- location description
DPT	(104 subscripts)	- Descriptions PTR# values
L	(78 subscripts)	- Locations for objects
C	(28 subscripts)	- Carried objects
F	(64 subscripts)	- Game flags
EF	(50 subscripts)	- Examine flags (used to indicate whether something has become visible after EXAMining something else.)

Reals:

VB - Number of subscript to which a match was found between the entered verb and the ones held in array V$()

OB - Number of subscript to which a match was found between the entered word and the ones held in array O$()

LOB - Last subscript of correct word match (needed for use of IT and THEM)

RM - Player's current location

RM2 - In some rooms, performing an action (e.g. pressing a button) changes the description of the room. This variable is the 'mirror image' of the room the player is in - i.e. the changed location description, and is used by the Random access filing procedure.

YES - Indicates whether there is anything that 'You can also see:' in the player's current location.

Integers:

T% - The number of 'turns' the player has had.

X%, Y% & C% - Variables for OSCLI emulation.

String:

Q$ - The full length sentence of the player's command entry.

V2$ - Full length version of entered verb.

V$ - First four characters of entered verb.

W2$ & W$ - As for V$ & V2$ but used with the word (noun).

R2$ - See description of PROCtry

M$ - Computer's response message to player's command entry.

String: (continued)

U2$ - Contains two CHR$(11) characters & moves the text cursor up two lines.

NL$ - Contains a CHR$(13) and a CHR$(10) character & moves the text cursor to the start of the next line.

OK$ - "By your command."

DH$ - "You don't have it!"

NY$ - "I can't do that now."

FS$ - "I've found something!"

NS$ - "I see nothing special."

PROCEDURES

DEF PROCdescribe
==========

First prints the location description (held in D$(0)), then loops around to see whether there is any objects or 'props' in the location. If there is, it calls PROCalso, to print them out. Finally, it prints the exits from the location.

DEF PROCalso
==========

Loops through the Flag and Examine Flag arrays, to see if there are any props or objects currently held in that location. If so, it prints out the appropriate object description string.

DEF PROCg
==========

When the player moves location, this procedure assigns the PTR# (pointer) value to the value it finds in the subscript of array DPT() corresponding to the player's new room. Certain conditions are tested to see if the computer should load the alternate description (i.e. ones that are altered due to the player performing a said action in that location). The appropriate description is loaded into D$(0).

DEF PROCgetwords
==========

This is the word-splitting procedure. The player's full command sentence is entered into Q$. It then looks for a space and takes everything to the left of the space and puts it in V2$. It puts everything to the right in W2$. It then strips off any leading spaces from W2$ and re-records itself without them. The next thing it does is to make V$ and W$ equal to the first four characters of V2$ and W2$ respectively. It adds spaces if the length was less than four characters - e.g. 'GO' becomes 'GO '. Finally it converts both V$ and W$ to upper case characters.

DEF PROCtry(E$)
==========

This procedure is included to overcome a problem. The game is not a real 3D adventure, so up and down commands need to be converted to compass points. However, this would mean that both up and, say, north commands would go in the same direction, which we surely do not want. The format for storing route descriptions is to have the true direction the player should move after the up or down exit. So, if we take an example, say, "USDNE", the exits the computer prints out would be "Up, Down, and East", but the true exits would be South, North and East. This procedure is called twice from the GO subroutine. On the first call, the routine tests it against R2$, which contains the U and D directions. If the chosen exit is an Up or Down the routine jumps to the conversion part, which changes the numerical value of the player's chosen exit to one corresponding to either North, South, West or East. PROCtry is then tested against the routes contained in the array R$ and then will move location. If, however, the player tried to go the same way by just typing N, the first call is against R2$, and being "UDE", does not contain an N character, thus preventing him (or indeed her) from moving.

DEF PROCmessages
==========

This procedure tests for a match between the entered verb or word and their corresponding array look-up tables. If the word entered is 'IT' or 'THEM' the computer assigns variable OB to that of the last valid word entered. The program filters out any invalid entries and accounts for a bad verb/good word, good verb/bad word, just a verb entered and etc.

DEF PROCinit
==========

This is the initialisation procedure. It sets M$, Q$, D$(0), and V2$ to their maximum length (255 characters) to avoid problems from 'dead copies' of the strings caused by the computer's inability to perform 'garbage collection'. It DIMensions all the arrays, and loads in the data for them from disc. Frequently used strings are assigned - e.g. DH$="You don't have it ". The procedure finally loads in the description for the player's initial location.

DEF PROCexamine
==========

Reports to the player that he has been successful/unsuccessful in examining something and if successful sets the corresponding flag to 'visible.'

DEF PROCdead
==========

Informs the player that he has been killed, and asks him/her for another game.

DEF PROCcompleted
==========

Congratulates the player for finishing "T-48 HOURS TO NOVA", and prints out how many 'moves' it took him/her.

A+B Computing Submissions
6c Belgic Square
off Padholme Road
Peterborough
PE1 1XF

Tel 0733 53355

Martin Briody
[redacted]

18th August 1989

Program Status Note

Dear Contributor

Thank you for your enquiry about T-48 Hours to Nova. We are actively considering this program for inclusion in A&B Computing, and Fast Access, although we do not as yet have plans to put T-48 Hours to Nova on either.

Please find below a summary of your program as seen by our evaluation team. This is intended both for information, and to help you should you wish to make any modifications.

Compatibility

BBC_B_Plus	Master_128	BASIC_I	Electron
y	y	y	n

ADFS	TUBE	Special Hardware
n	y	n

Bugs Found n

Description:
Test adventure where you need to get away from a star that is near to going nova. Your spaceship needs to be repaired This makes up the whole idea of the game.

Documentation:
Comprehensive with programming synopsis etc.

Overall:
Interesting to play not too difficult but involved.

Yours Sincerely

Matthew Fifield

Disk User Editorial
6c Belgic Square
Off Padholme Road
Peterborough
Cambs.
PE1 1XF

14th October 1988

Dear Martin,

Thank you for sending us your adventure game 'T-48 Hours to Nova'. After a period of asessment we found the game to be of a very high standard. This surprised us considering your age. The game has lots of atmosphere due to the colourful descriptions of locations and objects. Playing the game was good fun but there was one thing we noticed, that the puzzles all revolved around technical gadgets etc. They all seemed to take a similar form (e.g. there are lots of buttons to be pressed). Also it felt a little lonely with only one character appearing in the scenario. Some alien creatures are always a point of interest in a space oriented game. Criticisms aside the program is very well written and contains all the features one would expect in commercially available adventures. If you continue developing your programming ability at the rate you have been then you are bound to be a very accomplished programmer.

We at Disk User are very interested in publishing software of this calibre and we hope to find and opening for it some time in the future. We will be in contact with you when this happens. We also would like to see any other programs you have or are writing. Keep up the high standard.

Yours faithfully

Matthew Fifield

Matthew Fifield (technical consultant)

FAST ACCESS Editorial
6c Belgic Square
PETERBOROUGH
PE1 5XF

Tel. (0733) 53355

20th March 1990

[redacted]

Re: T-48 Hours To Nova

Dear Martin,

Thankyou for your letter of the 15th of March. I am sorry that such a long time has passed before contacting you. I am afraid that having software published in a magazine is a real waiting game. Your program will be given priority over other adventures. I cannot give any firm details about when it will be used. I am currently putting together Issue 5 of FAST ACCESS. This will be published in June.

I have a working copy of T-48 Hours so there is no need to send another one. It may be used in Issue 5 or maybe Issue 6. When I have definite plans for it I will contact you with an offer of payment. Until then it is a matter of waiting patiently. Using adventures in FAST ACCESS is difficult because they are usually about 40 to 50 KBytes in length and I only have 200K to play with.

Yours faithfully

Matthew Fifield

Matthew Fifield

Technical Editor

6C Belgic Square
Off Padholme Road
PETERBOROUGH
PE1 5XF

[redacted]

2nd November 1990

Re: T-48 Hours to Nova

Dear Martin,

Please accept my apologies for not writing to you sooner about your adventure program. I had hoped to publish your program in issue 6 of FAST ACCESS but as it turned out there was not space for it to fit in among the other programs. It has been a very long time since you submitted the program. I can understand your frustration at not having had it published. Your program, along with many others, has been kept on file in the hope that there may be a suitable opportunity to use it. I would strongly suggest that you offer your program to all the other BBC micro magazines and adventure publishers to see if they would like to publish it. TOPOLOGIKA is one publisher who is still supporting the BBC Micro. Write to Brian Kerslake, PO Box 39, Stilton, PETERBOROUGH, PE7 3RL. If they would like to use it you are under no obligation to us, all that we ask is that you let us know if it has been used.

The adventure is quite original and I would like to use it but exactly when I simply cannot say. All I can do is promise that your program will be given priority over all other adventures. I will be in contact with an offer of payment when I have firm plans to use the adventure.

Yours Faithfully

Matthew Fifield

Matthew Fifield