YESTERDAYS NEWS

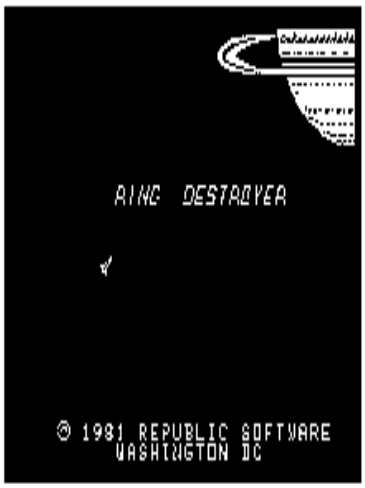
VOLUME 1 NUMBER 7 Established 2016 OCTOBER 2016

VAST TI99ERS

INSIDE INFORMATION

In The Beginning...

REVIEWED



RING DESTROYER

BY

REPUBLIC SOFTWARE

Reviewed by D. G. Brader

Message to the scout ship squadron leader - "Aliens have invaded the Solar System ! Massing their forces in the Rings of Saturn,they have already raided and destroyed a number of outposts on Saturn's satellites,and the attacks are spreading. Your mission is to penetrate the rings and destroy any ring fragments large enough to interfere with our main invading forces. Caution, it is probable that enemy ships will attempt to harass you, but permission is granted to destroy them. Good luck, Commander."

The preceding dialog sets the scenario for Ring Destroyer, an arcade-style game from Republic Software.

This game really impressed me the first time I played it. The screen turned black and a beautiful graphic presentation of Saturn and its rings appeared. There was simulated motion in the rings and a space craft was shown moving from the foreground toward Saturn. As the spacecraft approached the planet, it decreased in size to give the feeling of traveling a great distance. I got a kick out of restarting the game several times just to admire this simulation!

You are given 3 ships to start the game; the first is placed in the center of the screen. The current total of points scored is displayed in the right-hand upper corner. The number of points scored on a particular hit is determined by what is destroyed. Small ring fragments score less than large ring fragments; enemy ships, logically enough, are worth the most.

During play, moving the joystick to the left rotates your ship counter-clockwise . Moving the stick to the right rotates the ship clockwise. Pushing the stick forward applies acceleration in the direction your ship is pointed. Just as with a real spaceship, you must rotate and decelerate to slow down. Pulling the stick back sends your ship into hyperspace and brings it back at a random location on the screen. Pressing the fire button launches a torpedo from the ship's nose.

Ring Destroyer is delivered with two language versions resident on the media. The first is in Extended Basic and requires the Extended Basic module plus joysticks . The second version in Assembly Language requires the Extended Basic module and Expansion Memory (either joysticks or keyboard will work). If you have the TI Disk Drive Controller and a disk drive, order the game on diskette. This version will automatically maintain the past high scores with player's initials for you - just like the games in the video arcades do. The disk version can also automatically decide which of the two language versions your system configuration can run!

The graphics in the Extended Basic version are not very pleasing because they are so large. This probably wouldn't have bothered me very much if I had not played the super Assembly Language version first.

I think you will like this game but it may be hard to get a chance to play the game if you have "video Kids"at home.

STORY OF THE URBITE

by
Paul
Urbanus

I would like to clarify and correct some of the statements made in the original posting of the below article. To help substantiate my comments, I will start with a history of my relationship with the TI 99/4A Home Computer.

I was hired by TI in December of 1979 as a coop student, and spent six months working for the Home Computer Divison of TI in Lubbock. During this time, I did such diverse things as gather statistics on the distribution of data in the various GROM chips and program an HP test station to verify that the RF modulators were meeting FCC and TI specifications. In my spare time, I purchased a surplus IBM terminal keyboard (a really nice one) and interfaced it to the 99/4. This was the first REAL keyboard for the 99/4. I even added logic to do auto-repeat and mapped the IBM cursor keys to function correctly. My coop term expired, the fun ended, and I went back to school (New Mexico State University) for a year.

I returned to TI in Lubbock in the summer of 1991 to serve a second stint as a coop student (and earn some money!). In my absence, the TI99/4 had undergone puberty and blossomed into the TI99/4A. In addition to the new keyboard, there was also a new video chip. The TMS9918 had been replaced by the TMS9918A. At this point several things were happening, and the confluence thrust me into a rather unique position at TI.

My first assignment was to perform some testing of the new video chip and plot a chart of chip operation versus supply voltage and temperature. While waiting for the temperature chamber to stabilize during these tests, I was reading the detailed chip specification and came to a startling discovery - there was a new graphics mode in this chip which would allow neat new applications. At the same time, the Editor/Assembler (E/A) cartridge was in the early stages of alpha (internal) testing. I used the E/A cartridge to play around with the new graphics mode (Graphics Mode II). One of the first programs I wrote was a simple line-drawing program which I called "Lines". This is the same program which was bundled with the Mini Memory module.

After I wrote the Lines program, management moved me from the hardware to the R&D group and suggested that I collaborate with Jim Dramis on a new game. I thought this was better than sex (oK, I WAS Kinda naive) - getting paid to write a video game. Just for reference, Jim had written some of the best TI games available at that point - Car Wars and MunchMan. We quicKly agreed that we wanted to write a space game and we wanted to have smooth horizontal scrolling to give the illusion of flying over the surface of a planet. As some of you may Know, there is NO hardware support for scrolling the screen on a pixel basis in the 99/4A video chip. After lot's of pondering, I hit upon the solution - copy the inner loop of the scroll code into the fast 16-bit RAM of the 99/4A. Since this code is responsible for 80% of the execution time of the scroll loop, substantial speed gains were made by moving the loop to fast RAM. In today's world of 486s and Pentiums, this RAM would be referred to as cache RAM. I then handed this code off to Jim so he could incorporate it into the game.

The next thing I wanted to do for the game was to come up with some really neat sound effects. Since the sound chip on the /4A was only capable of generating square waves, I wanted to use the speech chip. The speech chip operates by using a model of the human vocal tract, and I reasoned that if people could maKe really strange noises, then so could the speech module. After studying the speech chip specification, I made an important discovery: the speech chip didn't need new data very often (it sure helps to understand hardware when writing software). This fact could be combined with one of the new features in the 99/4A software architecture - the User video interrupt. The net effect of this combination was that the speech chip could be used while the game was going on. When I went to the software folKs with my discovery, they told me that "you couldn't do that". Only after I showed them did they believe.

I created the asteroids in Parsec in TI Logo. I wrote a small Logo program to animate them, and iterated the shapes until they were satisfactory to me. Then I wrote an assembly program to convert the asteroid bitmap from the binary TI Logo data file to ASCII data statements for use with the 9900 assembler.

All of the above programming was done on the 99/4A using the Editor/Assembler packAge. EVERYTHING I wrote for the 99/4A was written using the Editor/Assemble cartridge. I liKed it much better this way, because I could worK at home, and I could fix the /4A system if it went down, unliKe the 990 minicomputers.

As Jim continued to progress with Parsec (we brainstormed on ideas, but he did most of the game flow implementation), the Mini Memory cartridge was developed. However, there was no software available to maKe it do anything useful. So I suggested that this would be a great tool for letting people experiment with assembly language without having to have any peripherals other than a cassette recorder. The Line-by-Line Assembler was a derivative of the code used in a TI single board computer which had been developed for microprocessor courses at the university level. This single board computer was called the University Board (model no. 990/189). When I returned to school after my first coop session, I had borrowed one of these from TI and it was an excellent learning tool for me so I assumed that a similar capability on the /4A would also be good

We were able to get the source code for the assembler from another TI group. All the I/O routines expected a dumb terminal, and so they had to be converted for use with the /4A Keyboard and screen. I also added a routine to dump the symbol table. In retrospect, the code could have been a lot cleaner and more compact, but I can probably say that about any program I write today after I have finished it. We decided to include the Lines program as an example of how to program the new video chip, as well as instant gratification for Mini Memory customers.

My final tasK was in this coop session was to go out to La Jolla, California and worK with Control Data Corporation and educate and support them in their efforts to port the Plato series of computer-based courseware to the 99/4A. I spent about a month out there, and in that time I wrote the graphics and disK I/O packAge for the Plato interpreter. A byproduct of this worK was an intermediate tool, DISKO, which was used for debugging the disK I/O packAge. I understand this program eventually made it into the public domain. For those of you familiar with this tool, there is a whimisical menu choice, "Resign/Go to BlacK's Beach". BlacK's Beach is a nude beach in La Jolla :)

I returned to school in the fall of 1992, but only lasted for one semester. At that time I joined with my Parsec partner, Jim Dramis and the author of TI Invaders, Garth Dollahite, along with two business types and we formed a company called SofMachine. Our charter was to author, produce and marKet game cartridges for the TI 99/4A. Kind of liKe the TI version of Activision. While Sofmachine

was in existence, we wrote three games of our own and converted two games for Atarisoft. The games we wrote during that time were:

Title	Author	Company
Spot-Shot	Jim Dramis	Sofmachine (ourselves)
Barrage	Garth Dollahite	Sofmachine (ourselves)
Jumpy	Paul Urbanus	Sofmachine (ourselves)
Pole Position	Dollahite/Urbanus	Atarisoft
Jungle Hunt	Dramis/Urbanus	Atarisoft

Because our business partners were unsuccessful at securing the required venture capital funding, combined with TI's exit from the home computer market, we were unable to manufacture and market our (Sofmachine's) three games. However, due to a sequence of events beyond our control the Sofmachine games were pirated and eventually freely exchange around the TI 99/4A community. A valuable lesson was learned: NEVER trust anyone with your own livelihood. Lesson number two: Don't believe what a "business" guy tells you just because they're the business guys and you're the technical guys, ESPECIALLY if it goes against your gut instincts.

A number of years later, the Sofmachine games were released in a cartridge form by Databiotics under license by Sofmachine.

All games programmed by Sofmachine used the TI 99/4A as the development platform, along with the Editor/Assembler cartridge. Two of us programmers purchased a Myarc 10 MByte hard drive for $1800 EACH! I just sold it about 6 months ago for $100. OUCH!

Wow! I intended for this to be a brief history for the purpose of lending credence to my following comments on the John Phillips article. Sorry for the long windedness, but the recollection of these times is good and I almost couldn't stop typing.

Please see below for my rebuttal to specific points.

Thanks and regards,

Paul Urbanus

the "Urbite" in Parsec

*** Warning - - - Urbite ships attacking! ***

*ORIGINALLY PUBLISHED IN LIMA NEWSLETTER APRIL 1993

*JOHN PHILLIPS
*by Bill Gaskill

* I would venture a guess that most people who have owned a TI-99 for more than a couple of years have run across the name John Phillips before. He is a near legend in the TI-99/4A cartridge and assembly language programming community and can claim authorship, co-authorship or significant involvement in over a dozen cartridge programs produced for the 99/4A, not to mention numerous articles written about the inner workings of the 4A's architecture.

Original article content omitted here.

* HOPPER - Michael Archuleta and John Phillips co-wrote Hopper, which was the only cartridge developed entirely on the TI-99/4A Home Computer, using the Editor/Assembler cartridge for all of the programming. All of the other TI-99 cartridge software programs were developed on a TI Mini, not the 99/4 or 4A.

All of the programs for the Mini Memory cartridge were programmed exclusively using the 99/4A and Editor/Assembler cartridge. As noted at the beginning of this article, all 5 of the titles developed by Sofmachine were also programmed using only the 99/4A. I suspect that many of the third-party games were also programmed in this manner, but I can't say for sure.

* MINI MEMORY'S LINE-BY-LINE ASSEMBLER - Phillips claims responsibility for its development, but I am not sure exactly what that means.

I developed the Line-By-Line Assembler excusively. John Phillips was not even working in the Home Computer division at that time. And I can't claim responsibilty for writing all of the code, only for porting it from the 990/189 University Board single board computer. Also, I wrote the Lines program. I am not aware of ANY contributions which John made to the programs in the Mini Memory.

* MOONMINE - Programmed by John Phillips from a design by Bob Hendren. You may remember that Hendren was also the project engineer behind Parsec and the person who recruited Aubree Anderson to do the voice for the Parsec game.

Bob Hendren had ABSOLUTELY NOTHING to do with the development of Parsec in regards to content, playability or technical direction. With respect to his role in the recruitment of Aubree Anderson, I really don't know about

that. Parsec was strictly a collaboration by Jim Dramis and myself. Parsec was not directed or defined in any way by management or anyone else. We were merely instructed to "..get together and see what you can come up with...". Although we received much input from our coworkers as they played with Parsec during the development process, almost all of Parsec was what came from our imaginations. I think I can speak for Jim when I say we are still pleased with our efforts more than 10 years later. We both hope that everyone who has played Parsec has enjoyed it as much as we enjoyed both writing AND playing it.

Original article continues from here.

SUBROUTINES RETRIEVING AND REUSING

By George Steffen
from MICROPENDIUM Dec. 1985 page 49

You know how it is: you've got a 200 line program and you'd really like to save 15 lines of it as a subroutine but you wish there was an easier way than retyping the 15 lines or deleting the other 185 lines one by one.

Fortunately there is, according to George Steffen of the Los Angeles 99ers user group. Writing in the group's newsletter, Steffen provides a six-line program that does the job so well that you may decide to go back and extract subroutines from an entire library of programs just to make up for all the tedium you've had to put up with in the past.

The program is meant to be MERGEd into the program--which is the reason the program uses such low line numbers--you wish to extract the subroutine from, so, after saving it as a program, asve it again in the MERGE format: SAVE DSKX.FILENAME,MERGE. Now, load the programfrom which you want to extract a subroutine--any group of consecutive program lines will do--and then MERGE the subroutine extractor into it: MERGE DSKX.FILENAME. Enter RUN. You will be prompted for the starting and ending lines you wish to extract. Having done so, the program will do its job.

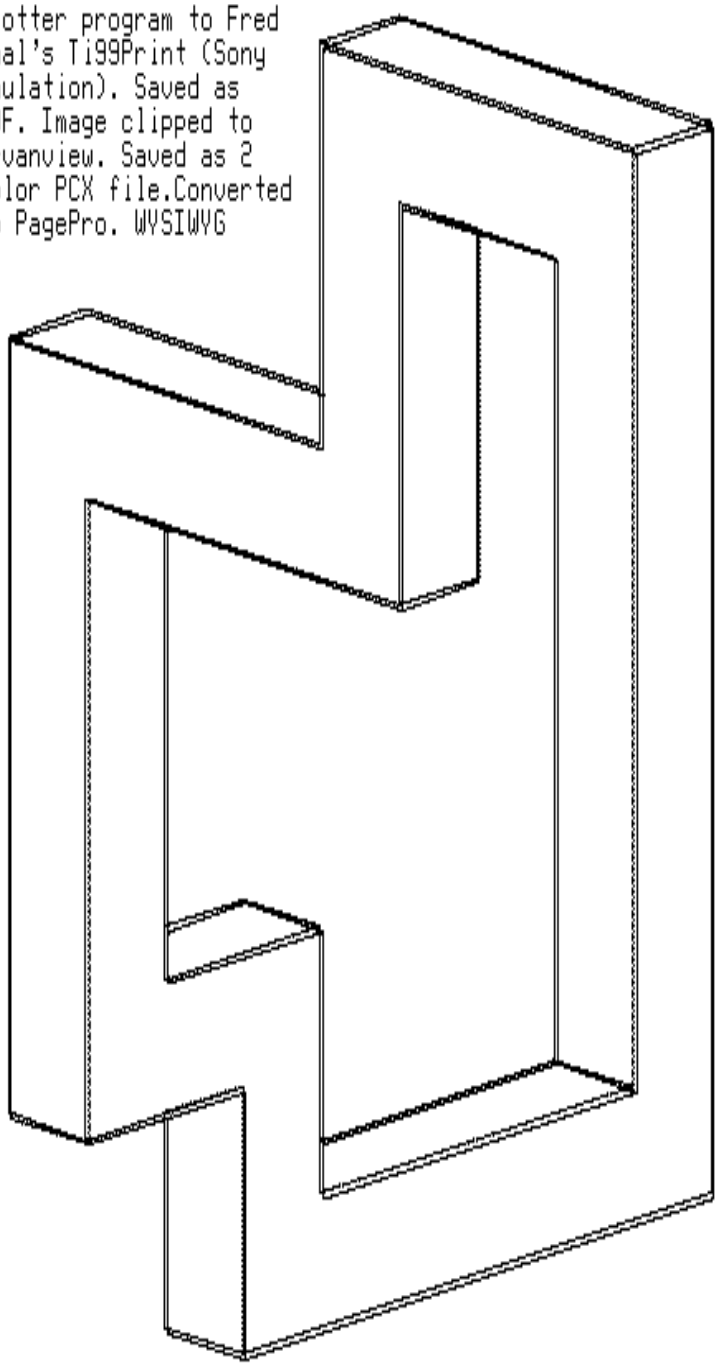
The proof of the job comes after the "READY" sign appears. List the program. You should see only the lines that you want to preserve. Now, save these to disk.

One caveat: it is suggested that you save the preserved lines in a MERGE format to start, because the lines that you sought to delete actually are still there. However, we found that the extracted lines can also be saved as a program. The deleted lines did not reappear in either case.

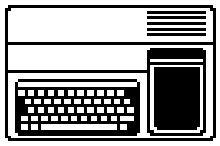
This program requires Extended BASIC, a disk system and memory expansion.

```
1 !SUBROUTINE EXTRACTOR by G
eorge F. Steffen. SAVE in ME
RGE format. MERGE into anypr
ogram (with line # starting
above 8). RUN to extract
2 !selected lines. Deletes i
tself. Then be SURE to SAVE
the selected lines in MERGE
format because the remaining
 lines are still in memory!
3 CALL CLEAR :: CALL INIT ::
 INPUT "LINE NUMBERS OF ROUT
INE TO BE SAVED: FIRST,LAST?
 ":L,M :: G=256 :: CALL PEEK
(-31952,H,I,J,K)
4 C=INT(M/G):: D=M-C*G :: F=
(J-G)*G+K :: FOR E=(H-G)*G+I
 TO F STEP 4 :: CALL PEEK(E,
A,B):: IF A=C AND B=D THEN 6
5 NEXT E :: PRINT :"LINE";O;
"NOT FOUND!" :: STOP !@P-
6 H=INT(E/G):: I=E-(G*H):: H
=H+G :: C=INT(L/G):: D=L-C*G
 :: FOR E=E+4 TO F STEP 4 ::
 CALL PEEK(E,A,B):: IF A=C A
ND B=D THEN 8 !@P-
7 NEXT E :: PRINT :"LINE";N;
"NOT FOUND!" :: STOP !@P-
8 E=E+3 :: J=INT(E/G):: K=E-
(G*J):: J=J+G :: CALL LOAD(-
31952,H,I,J,K):: STOP !@P-
```

From a 7 line TI-99/4A plotter program to Fred Kaal's Ti99Print (Sony emulation). Saved as PDF. Image clipped to Irvanview. Saved as 2 color PCX file.Converted to PagePro. WYSIWYG



Yesterdays News Information



Yesterdays News is a labor of love offered as a source of pleasure & information for users of the TI-99/4A & Myarc 9640 computers.

TI-99/4A HARDWARE

Black & Silver computer
Modified PEB
WHT SCSI card with SCSI2SD
Myarc DSQD FDC
Myarc 512K Memory Card
Horizon 1.5 meg RamdisK
TI RS232 card
Corcomp Triple Tech Card
1 360K 5.25 floppy drive
1 360K 3.50 floppy drive
1 720K 5.25 floppy drive
1 720K 3.50 floppy drive
80K Gram KracKer
Samsung Syncmaster 710mp

TI-99/4A SOFTWARE

PagePro 99
PagePro Composer
PagePro FX
PagePro Headline MaKer
PagePro Gofer
TI Artist Plus
GIFMania

PC HARDWARE

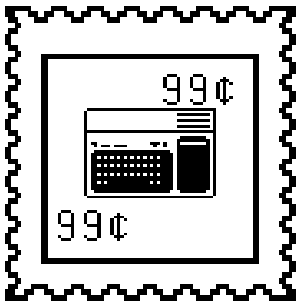
Compaq Armada 7800 Notebook
Compaq Armadastation
Samsung Syncmaster 710mp

PC SOFTWARE

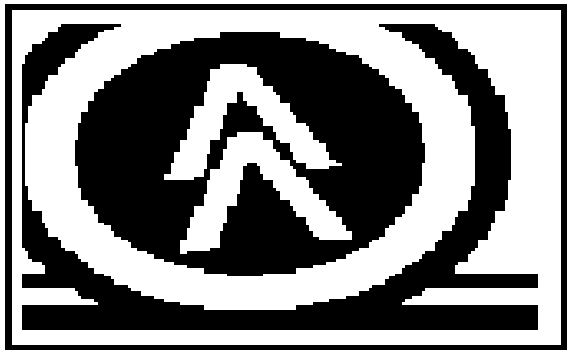
Dead,Dead,Dead Windows 98se
FileCap
prn2pbns
Irfanview
Adobe Distiller
Adobe Acrobat

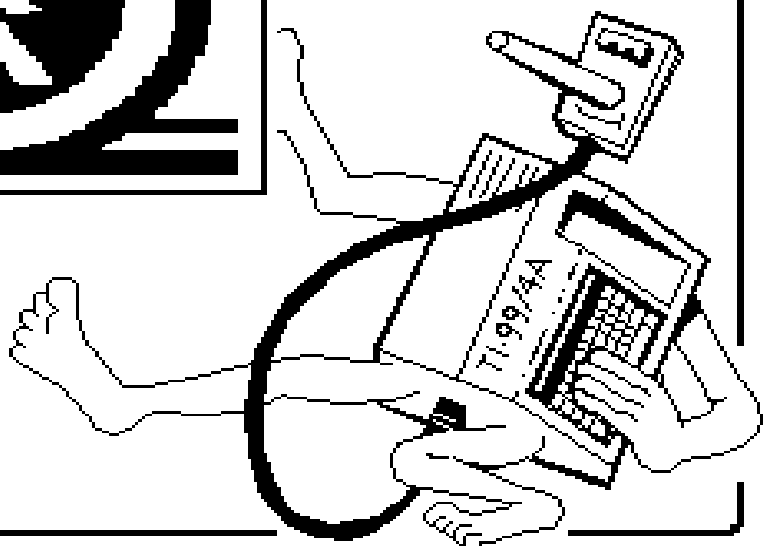
Yesterdays News is composed entirely using a TI-99/4A computer system. It consists of 11 PagePro pages which are "printed" via RS232 to PC to be published as a PDF file.

Yesterdays News
c/o SparKdrummer
AtariAge forum
Phoenix, AZ. 85027



FIRST CLASS MAIL





FIRST CLASS MAIL