

COMMODORE 64

TIME * LOST

A COMPUTER ADVENTURE



que™

Timelost

(Commodore 64 Version)

**Kris Austen Andrews
Arlan Keith Andrews, Sr.
Joseph C. Giarratano**

**Que Corporation
Indianapolis**

**Series concept: Joseph C. Giarratano and
Arlan Keith Andrews, Sr.**

Story and Artwork: Kris Austen Andrews

Programs: Joseph C. Giarratano

Story consultant: Arlan Keith Andrews, Sr.

Lettering: Andy Andrews

ISBN 0-88022-074-0

©1984, Andrews & Giarratano

About the Authors

Arlan Keith Andrews, Sr., is a mechanical engineer at AT&T Consumer Products Laboratories in Indianapolis. He received his Doctor of Science in Engineering degree from New Mexico State University and is a member of Science Fiction Writers of America. Kris Andrews, born in 1967 at Las Cruces, New Mexico, is currently a junior at Carmel High School, Carmel, Indiana. He is an honor student who desires to be a theoretical physicist. He is also a science fiction fan and likes role-playing games and computers.

Joseph C. Giarratano received his B.S. and M.S. degrees in physics from California State University at Los Angeles and his Ph.D. in physics from the University of Texas at Austin. He is the author of: *Foundations of Computer Technology; Modern Computer Concepts; BASIC: Fundamental Concepts; BASIC: Advanced Concepts; the Timex/Sinclair 1000 User's Guide, Volumes 1 and 2; the Timex/Sinclair 1000 Dictionary and Reference Guide, the Texas Instruments 99/4A BASIC Guide, and the Commodore 64 BASIC Guide.*

Acknowledgments

The authors would like to thank the following for their work on this book:

Tim Renshaw, who managed the software programs and did considerable work on them

Steve Nye, who worked on Rockfall and Escape

Ted Powell, who did Battle at Stonehenge

Eric Benson, who worked on Attack of the Slime Creatures and Peril of the Pitdemons

Kirk Ostby, who wrote Caverns of Carnage.

Special thanks also to Kirk for all the debugging he did.

Kathleen Plunkett, for assistance in lettering

Other *Timelost* Books by Que

Book:	ISBN No.	Date Available
<i>Timelost</i> (Timex/Sinclair 1000 Version)	0-88022-030-9	Currently
<i>Timelost</i> (TI-99/4A Version)	0-88022-053-8	Currently
<i>Timelost</i> (Atari Version)	0-88022-056-2	Currently
<i>Timelost</i> (VIC-20 Version)	0-88022-054-6	Currently

Dedications

To my family:
Jane, Jenna, Melissa, and Anthony Giarratano

—*Joseph Giarratano*

To the family:
Joyce, Patty, Andy, Mandy, Sean,
Vivian Andrews, Arlie Andrews,
and Mrs. Lizzie Van Ness

—*Kris and Arlan Andrews*

. . . and my mom, Helen Andrews

—*Kris Andrews*

The characters in this book are completely fictional. Any similarity between these characters and any actual person, living or dead, is unintentional and purely coincidental.

Note to the Reader

Timelost is designed so that it can be used in different ways. The comics can be read straight through as an episodic adventure story. Or, if you wish, you can choose the games that appeal to you the most, key the programs into your computer, and just play those games. Each game is self-contained. You don't have to read the comics section to understand how to play the game.

You may get the most enjoyment, however, by reading one episode and then playing the corresponding game at the back of the book. If you want to use the book this way, look for the "Play the game here" notation in the bottom right-hand corner of certain pages. At that point, simply refer to the Contents at the beginning of the book to find the page on which the appropriate game begins.

All game program listings have been thoroughly tested by the programmer(s) and work properly if typed correctly. If a problem occurs in playing any of the games, check carefully your entry of the listing.

Contents

Introduction

The Mystery of Stonehenge	1
The Slime Creatures	14
The Pitdemons.....	28
The Cavern Monsters.....	46
The Dungeon of the Lord-Wizard ..	59
Cave-In	72

Games

Battle at Stonehenge	85
Game Program Listing	85
How the Program Works	92
How You Can Change the Program	94
Attack of the Slime Creatures	96
Game Program Listing	97
How the Program Works	101
How You Can Change the Program	103
Peril of the Pitdemons.....	104
Game Program Listing	104
How the Program Works	109
How You Can Change the Program	112

Games *(cont'd)*

In the Caverns of Carnage	113
Game Program Listing	113
How the Program Works	119
How You Can Change the Program	122
Escape	124
Game Program Listing	124
How the Program Works	129
How You Can Change the Program	134
Rockfall!	135
Game Program Listing	136
How the Program Works	141
How You Can Change the Program	147

Introduction

Welcome to the Universe of THE TIMELOST! In this new and exciting series, you'll follow the adventures of John Moore, a teenage computer whiz; his little sister, Erin; and their mysterious friend from the future. Join them as they battle against the evil Wizard, a Warlord of the Universe.

These books combine a story with programs that you can enter and run on a standard Commodore 64 computer with 64K bytes of memory. You follow the story and then get into the action with the game programs. All of the games were written especially to follow the adventures in the book.

These game programs are designed to be fun and educational. If you don't know how to program, you'll enjoy playing the games and may wish to learn about programming. If you do know how to program, or are learning how, the detailed explanations will show you how the games work and how you can change them and improve them.



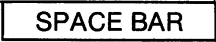

The games in this book are designed to illustrate action and cover many different

sound, color, and animation effects you can achieve with your computer. We know they'll be fun!

So, enter the fun world of computer games and the mysterious worlds of THE TIMELOST...

Timelost Listing Conventions

Some symbols used in this book for special functions may not be familiar to you. PLEASE NOTE THAT ANY TERM IN BRACKETS, [], IS AN ABBREVIATION FOR SOMETHING ELSE. The brackets and the term inside should NOT be entered into your game program. The following list shows which keys should be pushed when entering these special functions. (When they appear on your screen or dot-matrix printer, they will have the shapes shown in your *Commodore 64 User's Guide*.)

ABBREVIATION	ACTUAL KEY PUSHES
[HM]	
[SC]	
[SPACE]	
[22 SPACES]	 22 TIMES

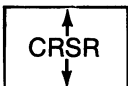
[12 SPACES]

PUSH SPACE BAR

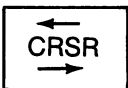
12 TIMES

(Ignore blank lines
inside the brackets.)

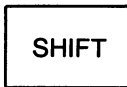
[CD]



[CR]

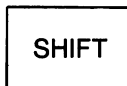
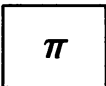


[CL]

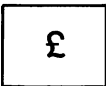


[PI]

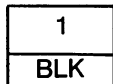
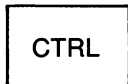
(Greek letter)



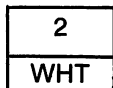
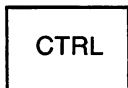
[ENGLISH POUND]



[BK]



[WH]



[RD]

CTRL

3
RED

[CY]

CTRL

4
CYN

[PR]

CTRL

5
PUR

[GR]

CTRL

6
GRN

[BL]

CTRL

7
BLU

[YL]

CTRL

8
YEL

STANDING STRONG AND SILENT,
TOWERING OVER THE GRASSY PLAINS
OF SALISBURY, ENGLAND -- THE
MYSTERIOUS STONE CIRCLES OF
STONEHENGE!

IT IS HERE, IN THIS PLACE OF
ANCIENT WONDER, THAT THE JOURNEY
BEGINS... A JOURNEY FRAUGHT WITH
SCIENCE AND SAVAGES, MAGIC AND
MAYHEM, SPANNING THE FARTHEST
REACHES OF TIME AND SPACE.

FOR IT IS HERE THAT THE PRESENT
AND THE FUTURE SHALL MEET AND THE
UNSUSPECTING SHALL BECOME...

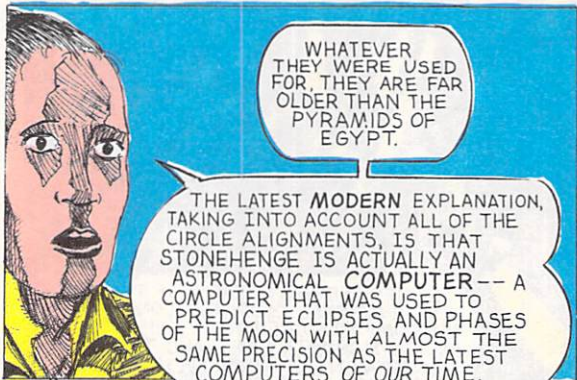
TIME! LOST!



LADIES AND GENTLEMEN OF THE
INTERNATIONAL COMPUTER CONFERENCE:
WELCOME TO STONEHENGE. A VISIT TO OUR
ANCIENT MONUMENT WILL ADD TO THE
ENJOYMENT OF YOUR TECHNICAL
DISCUSSIONS AT THE UNIVERSITY.



WE ARE STANDING AT THE EDGE OF THE LARGEST OF SEVERAL CIRCLES OF STONE... "MEGALITHS", WE CALL THEM. THE TALLEST STONES ARE TWENTY FEET HIGH, IN A CIRCLE 108 FEET IN DIAMETER. FURTHER OUT, SMALLER STONES LINE UP WITH THESE CIRCLES. SOME SCIENTISTS SAY THAT THE ALIGNMENTS POINT OUT THE BEGINNINGS OF THE SEASONS, AND THE RISING OF THE STARS.



WHATEVER THEY WERE USED FOR, THEY ARE FAR OLDER THAN THE PYRAMIDS OF EGYPT.

THE LATEST **MODERN** EXPLANATION, TAKING INTO ACCOUNT ALL OF THE CIRCLE ALIGNMENTS, IS THAT STONEHENGE IS ACTUALLY AN **ASTRONOMICAL COMPUTER**-- A COMPUTER THAT WAS USED TO PREDICT ECLIPSES AND PHASES OF THE MOON WITH ALMOST THE SAME PRECISION AS THE LATEST COMPUTERS OF OUR TIME.

IN THE TOUR GROUP, DR. MARY AND PETER MOORE, COMPUTER SCIENTISTS, LISTEN INTENTLY, UNAWARE OF THE IMPENDING FATE OF THEIR CHILDREN...



THERE IS A LESS PLAUSIBLE, BUT MUCH MORE ROMANTIC, LEGEND WHICH CLAIMS THAT IN KING ARTHUR'S TIME, THE STONES WERE PUT IN PLACE BY A MAGICIAN...

... TEENAGER JOHN MOORE, HIGH SCHOOL COMPUTER WHIZ...



... A SORCERER WE KNOW AS MERLIN.

...AND SEVEN-YEAR-OLD ERIN MOORE, DEVOTEE OF COMPUTER VIDEO GAMES.



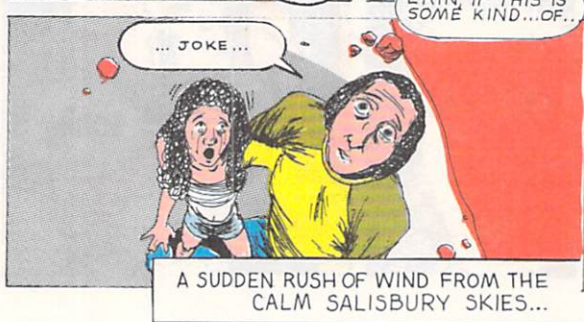
MERLIN-? WAS HE LIKE JOHN...



...PREDICTABLY BECOMES
AN UNBEARABLE IRRITATION
TO THE EARS OF A
MODERN ADOLESCENT.









... EXPLODES INTO A FLURRY OF SCREAMING, DIVING CHAOS!!

WHAT ARE THESE BLASPHEMIES OF LIFE? DEMONS, SENT FROM THE NETHER REALMS, TO FILL A DAILY QUOTA OF SOULS?

OR PERHAPS FOR A MORE VORACIOUS PURPOSE?

WHATEVER THEIR ORIGIN MAY BE, THE GIBBERING FURIES SWOOP STEADILY

CLOSER...



IT WOULD SEEM TO BE A
DAY FOR SURPRISES!



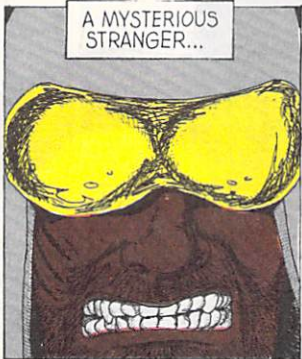
THE MAN CAME FROM
NOWHERE, AND
WIELDS A GUN LIKE
NONE JOHN HAS
EVER SEEN BEFORE!



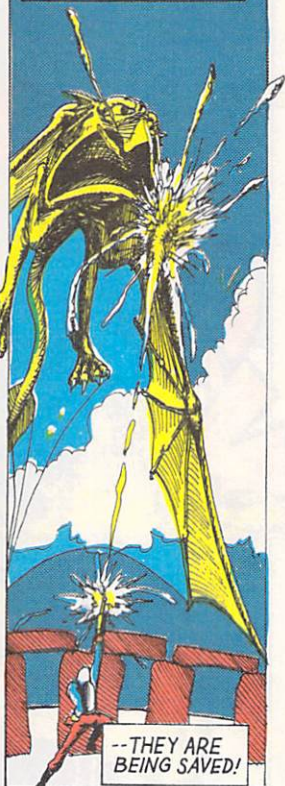
A NIGHTMARE
ATTACKER...



A MYSTERIOUS
STRANGER...

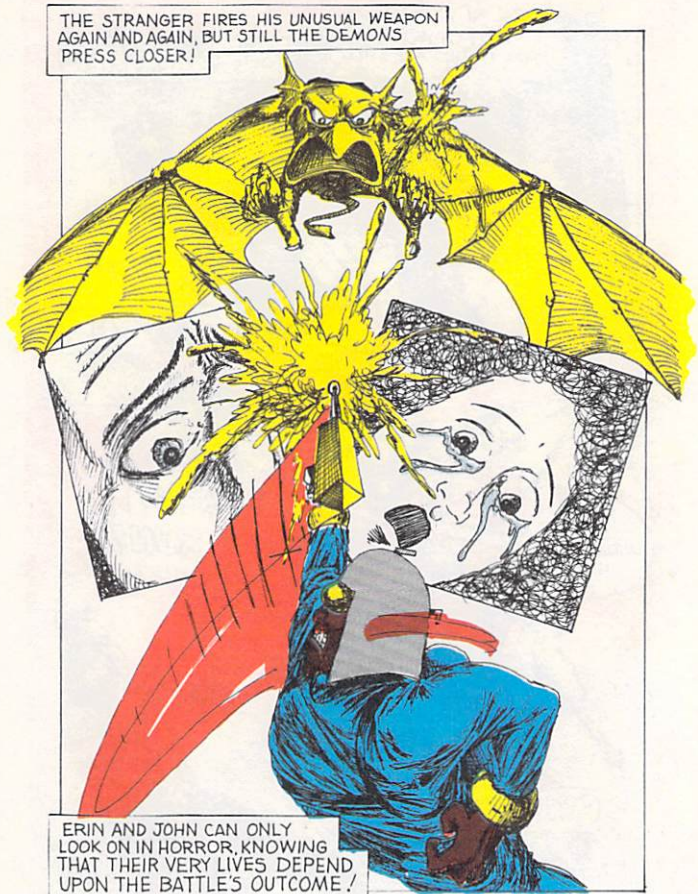


CONFUSION RUNS
RAMPANT, BUT ONE
THING IS CLEAR---



--THEY ARE
BEING SAVED!

THE STRANGER FIRES HIS UNUSUAL WEAPON AGAIN AND AGAIN, BUT STILL THE DEMONS PRESS CLOSER!



ERIN AND JOHN CAN ONLY LOOK ON IN HORROR, KNOWING THAT THEIR VERY LIVES DEPEND UPON THE BATTLE'S OUTCOME!





Play the game here.

IT IS ANOTHER PLACE, ANOTHER TIME. THE DEMON IS GONE; THE MYSTERY OF STONEHENGE SEEMINGLY FAR BEHIND, REPLACED BY YET A DEEPER ENIGMA:

HOW, AND MORE IMPORTANTLY WHY, WERE THEY BROUGHT TO THIS RANK AND DISMAL PLACE IN A FORGOTTEN CORNER OF TIME?

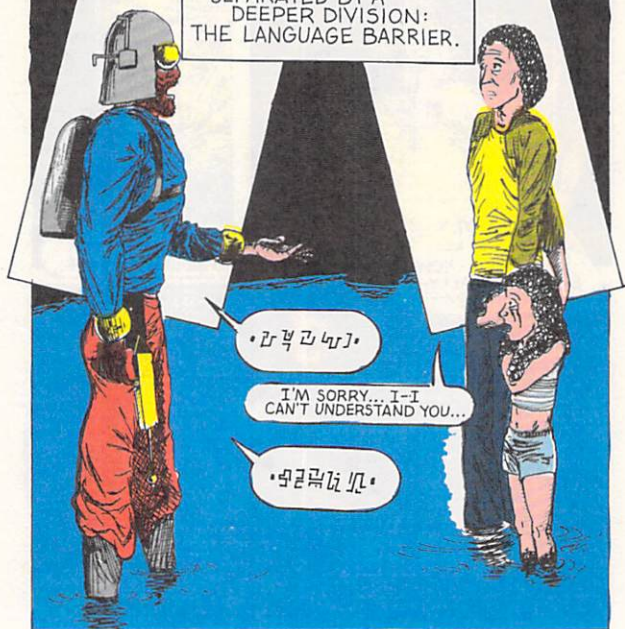


THE SEETHING SWAMP SEEMS TO STRETCH FARTHER THAN THE YOUTHS' STUNNED MINDS CAN ACCEPT. AS FOR THE STRANGER, HIS THOUGHTS ARE HIS OWN.



THE PRESENT FACES THE FUTURE:
EACH A STRANGER IN THIS UNKNOWN
LAND, BROUGHT TOGETHER BY A
BIZARRE STROKE OF FATE --- OR
PERHAPS SOMETHING MORE....

BUT WHILE THE
TIME BARRIER SEEMS TO
HAVE BEEN BROKEN, THE
TRAVELLERS ARE YET
SEPARATED BY A
DEEPER DIVISION:
THE LANGUAGE BARRIER.

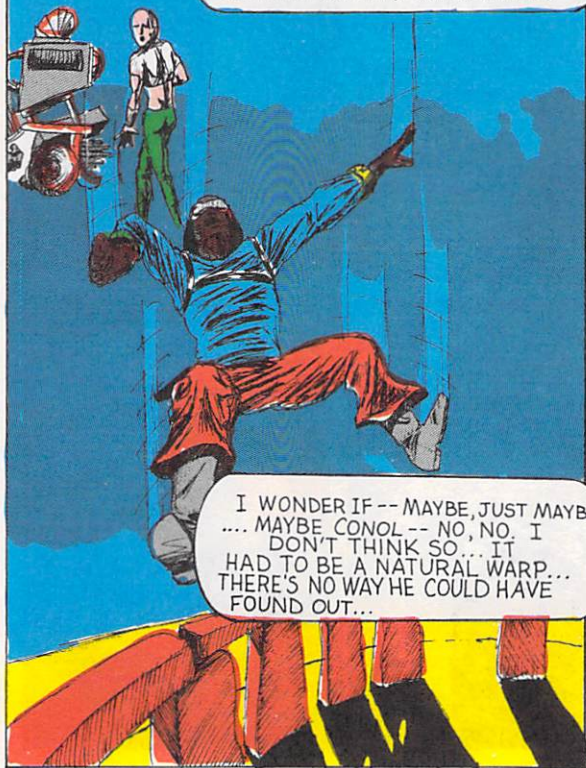




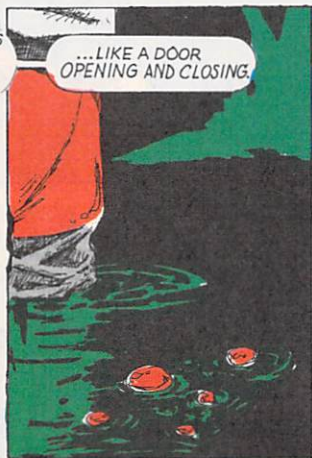




ANYWAY, SUFFICE IT TO SAY THAT AT A CRUCIAL MOMENT I SUDDENLY FOUND MYSELF SWEEPED UP BY ONE OF THOSE TIME WARPS AND DEPOSITED AT STONEHENGE IN THE TWENTIETH CENTURY SAY, WAITAMINNIT!!



I WONDER IF -- MAYBE, JUST MAYBE MAYBE CONOL -- NO, NO. I DON'T THINK SO... IT HAD TO BE A NATURAL WARP... THERE'S NO WAY HE COULD HAVE FOUND OUT...





WITH A SICKENING GURGLE,

THE GROANING SWAMP SHUDDERS,...

... HEAVING FROM THE MUCK
ITS FIERCEST OF VENGEFUL
SPIRITS! THE SWAMP DOES
NOT LIKE TRESPASSERS...

... AND THE
WILL OF THE
SWAMP IS THE
LAW OF
THE LAND!



THE SLIME-DWELLERS
LURCH MENACINGLY CLOSER...



THESE THINGS JUST
KEEP COMING!



I'M HOLDING MY OWN,
BUT IT'S JUST A MATTER
OF TIME BEFORE--



AAACCHH!



THAT'S IT!
WARP OR NO
WARP, WE'VE
GOTTA GET
OUTTA HERE!!






Play the game here.







... BUT I THINK
I KNOW A WAY TA
FIND OUT.

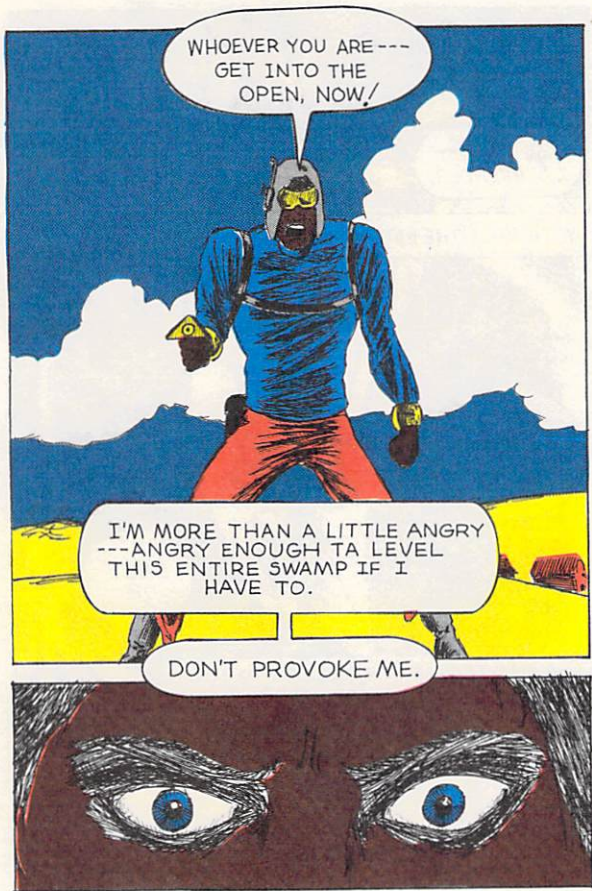


"... SHOULD JUMBLE UP
OUR FRIENDS' THOUGHTS
ENOUGH TO DRIVE
HIM INTO THE OPEN!"



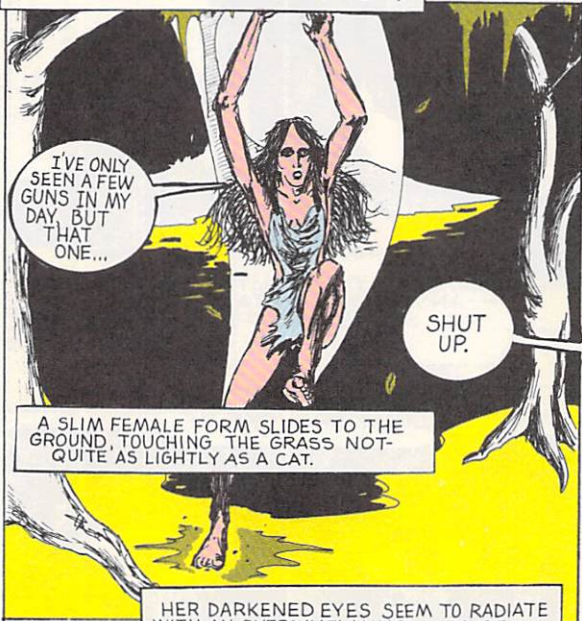
THIS QUASI-TELEPATHIC
THOUGHT TRANSLATOR,
SET ON HIGH....

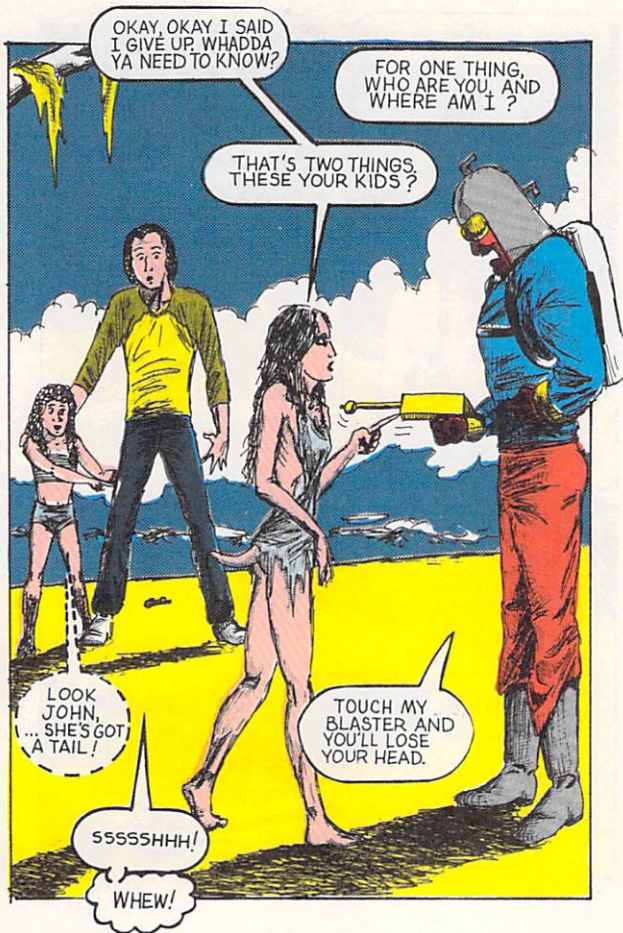
WHA---OH!



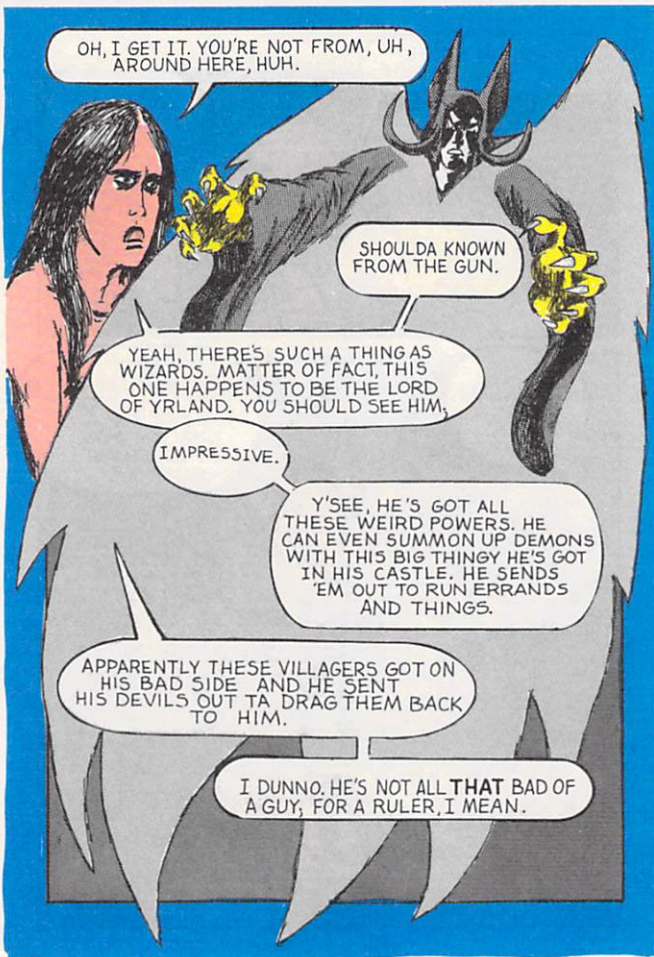


WITH THE LITHE EASE OF AN ACROBAT,









OH, I GET IT. YOU'RE NOT FROM, UH, AROUND HERE, HUH.

SHOULDA KNOWN FROM THE GUN.

YEAH, THERE'S SUCH A THING AS WIZARDS. MATTER OF FACT, THIS ONE HAPPENS TO BE THE LORD OF YRLAND. YOU SHOULD SEE HIM,

IMPRESSIVE.

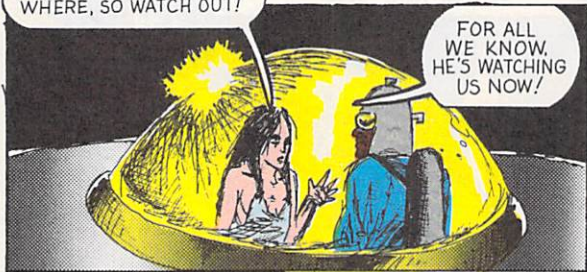
Y'SEE, HE'S GOT ALL THESE WEIRD POWERS. HE CAN EVEN SUMMON UP DEMONS WITH THIS BIG THINGY HE'S GOT IN HIS CASTLE. HE SENDS 'EM OUT TO RUN ERRANDS AND THINGS.

APPARENTLY THESE VILLAGERS GOT ON HIS BAD SIDE AND HE SENT HIS DEVILS OUT TA DRAG THEM BACK TO HIM.

I DUNNO. HE'S NOT ALL **THAT** BAD OF A GUY, FOR A RULER, I MEAN.

HE'S GOT EYES EVERYWHERE, SO WATCH OUT!

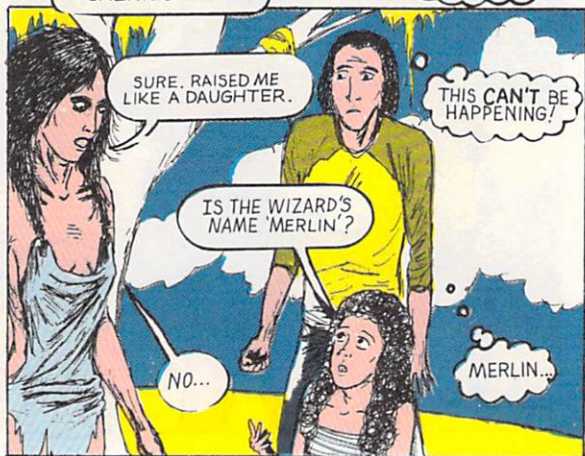
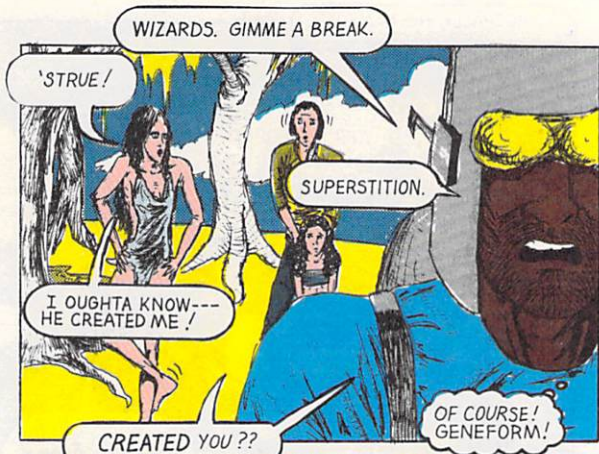
FOR ALL WE KNOW, HE'S WATCHING US NOW!

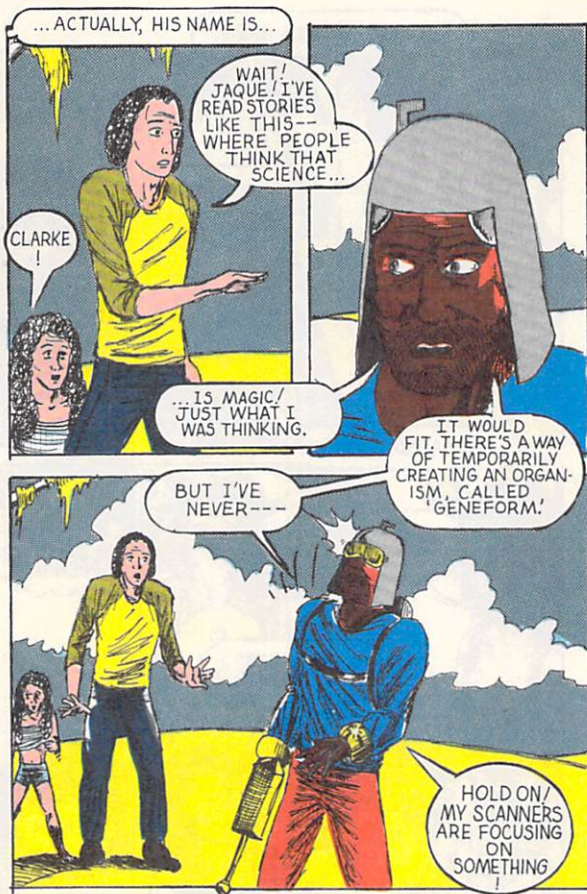


SO, D'REEN, YOU'VE FOUND MORE VISITORS!

I'LL HAVE TO SEE ABOUT BRINGING THEM HERE... AND PERHAPS THIS TIME I MIGHT EVEN BAG YOU, MY REBELLIOUS DAUGHTER!









WITHOUT WARNING
THE VERY GROUND
CAVES IN ON ITSELF,
OPENING A PIT
WITHOUT DEPTH...

... UNLEASHING A
SAVAGE SWARM
OF TUNNELLING
TERRORS BENT
ON ONE PUROSE:

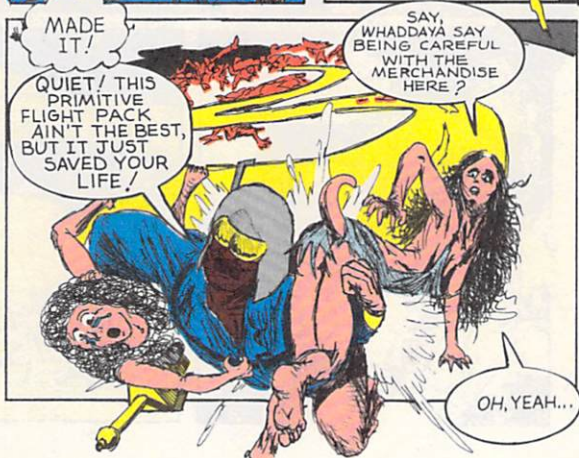
**GET
BACK!**



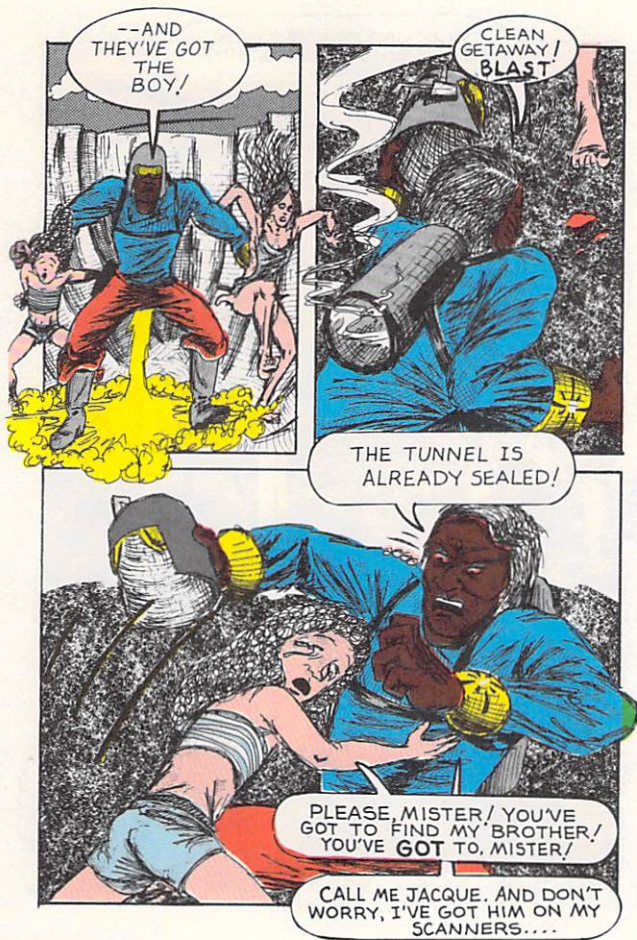
... THE
ABDUCTION
OF THE
UNLIKELY
QUARTET!













Play the game here.



...AND HE'S GONNA LEAD
ME TO WHOEVER'S BEHIND
ALL THIS...

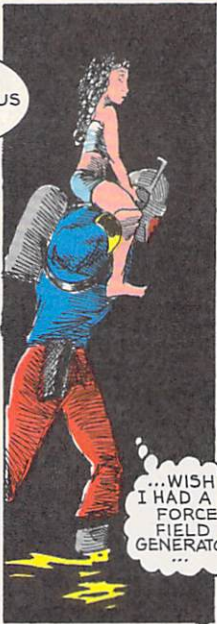
MINUTES LATER...



...UH, DON'T YOU
THINK IT'S A
LITTLE DANGEROUS
TO COME BACK
IN HERE?

...STUPID
LEECHES...

DON'T WORRY. MY POSITRON
BLASTER HASN'T FAILED ME YET...



...WISH
I HAD A
FORCE
FIELD
GENERATOR
...







--THERE!!

THE ANCIENT
MOUND JUTS UP
DEFIANTLY
FROM THE MUCK...

...ENSHRINING
A GRIM DOOR
THAT COULD
WELL LEAD TO
THE DEPTHS
OF HELL
ITSELF!

IT'S THE OLD CAVERN
ENTRANCE!
THERE ABIDE MONSTERS!

WELL, THAT'S WHERE THE SIGNAL STOPPED, SO THAT'S WHERE I'M GOING!



ERIN, YOU STAY HERE WITH D'REEN...

ER-I CAN'T WATCH HER! I CAN BARELY LOOK AFTER MYSELF IN THIS PLACE!

BESIDES, I WANNA STAY WITH YOU, JACQUE! I WANNA HELP FIND JOHN!



IT'LL BE DANGEROUS--- BUT OKAY. BUT YOU'LL HAFTA KEEP QUIET, AND YOU'LL HAFTA DO WHAT I SAY, OR ELSE I'LL LEAVE YA THERE TA DIE AND FIND MY OWN WAY HOME! GOT THAT?

Y-YESSIR...



...STAND BACK...

SCANNERS SHOW
THE HALL
IS CLEAN.



READY,
GIRL?

YEAH.

COMING, D'REEN?



NOT ME!

I MAY BE DUMB,
BUT I'M NOT
SUICIDAL!



... NOSSIR, NOT ME



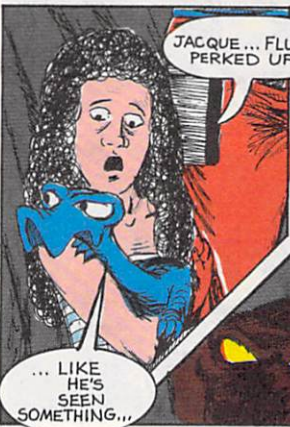
HOPE I DIDN'T MAKE A MISTAKE...



... WISH MY SCANNERS WOULD STOP FRITZING...



JACQUE ... FLUX'S PERKED UP...

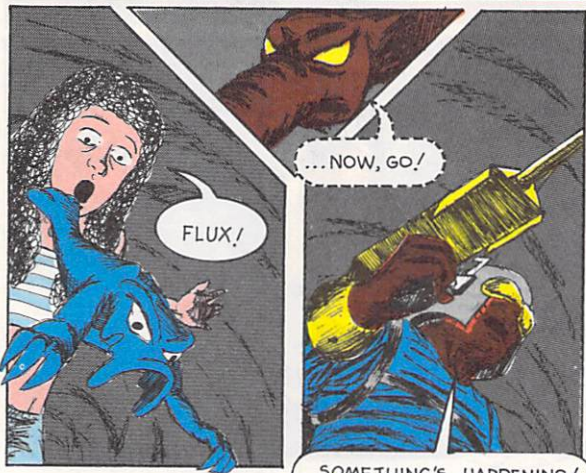
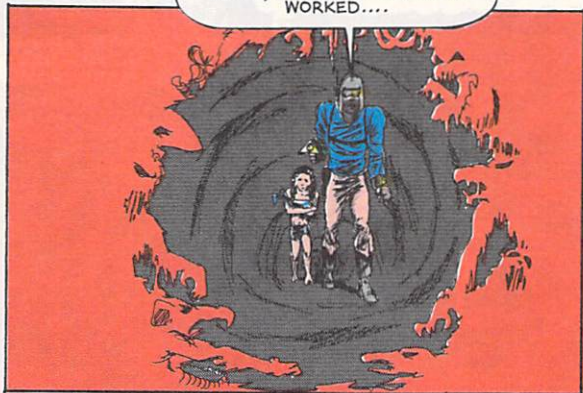


... LIKE HE'S SEEN SOMETHING...



YEAH, I FEEL IT, TOO...

BLAST, I WISH MY SCANNERS
WORKED....



FLUX!

... NOW, GO!

SOMETHING'S HAPPENING!

WITH A BESTIAL HOWL, THE VERY WALLS SUDDENLY TRANSFORM THEMSELVES INTO A CLAWING MASS OF FANGS AND TALONS!

HERE ABIDE MONSTERS!

NO!!

HOLY—!

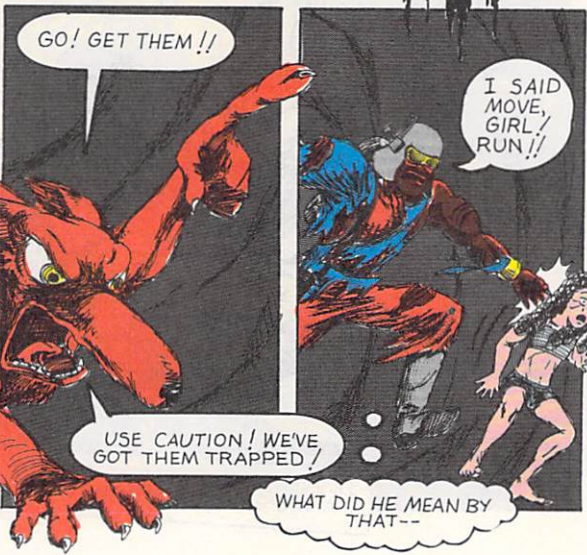


... AND THE FRAY IS ENTERED!
THE CAVERNOUS HALL IS FILLED
WITH THE INSANE HOWLS OF
THE ONCOMING GHOULS, THE
THUNDEROUS BOOMS OF
THE POSITRON BLASTER, AND
THE ECHOING SCREAMS OF
A HORRIFIED CHILD!!





WE'VE BROKEN THROUGH!
C'MON, ERIN, MOVE!



GO! GET THEM!!

I SAID
MOVE,
GIRL!
RUN!!

USE CAUTION! WE'VE
GOT THEM TRAPPED!

WHAT DID HE MEAN BY
THAT--



Play the game here.

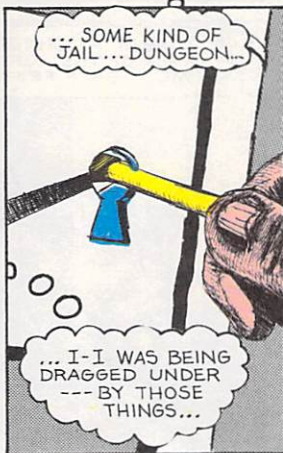
MEANWHILE, IN A PLACE NOT FAR AWAY, A BATTERED JOHN STRUGGLES FROM A RESTLESS SLUMBER...



TO FACE A HARSHER REALITY...



OOOOH, ...
..MY BACK...
WH-WHERE
..AM I...
?

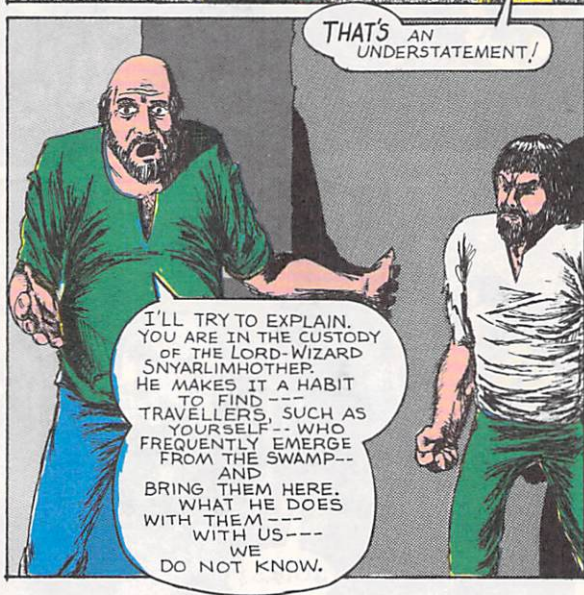


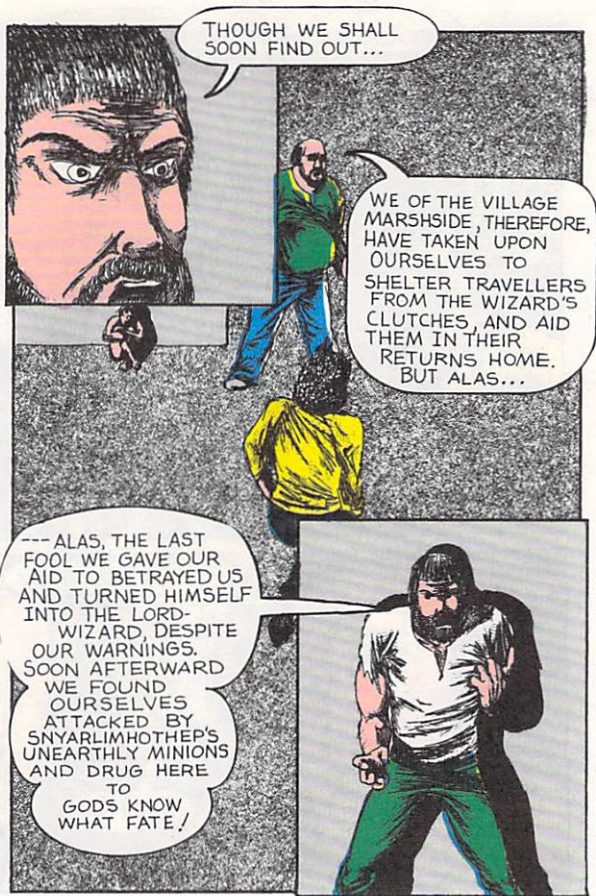
... SOME KIND OF
JAIL ... DUNGEON...

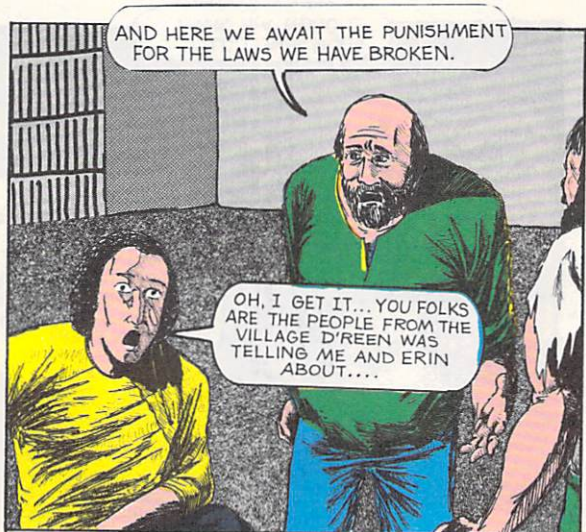
... I-I WAS BEING
DRAGGED UNDER
--- BY THOSE
THINGS...



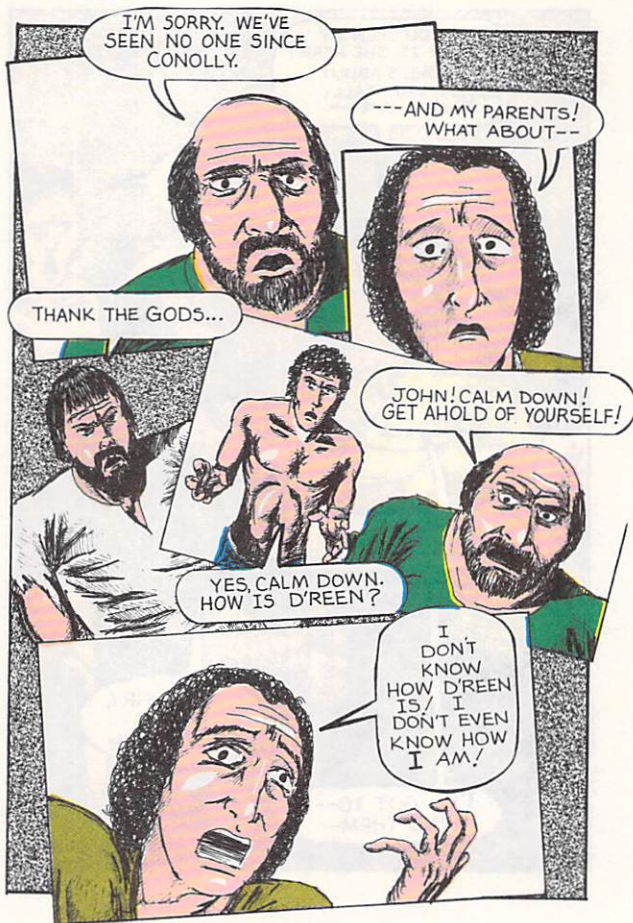




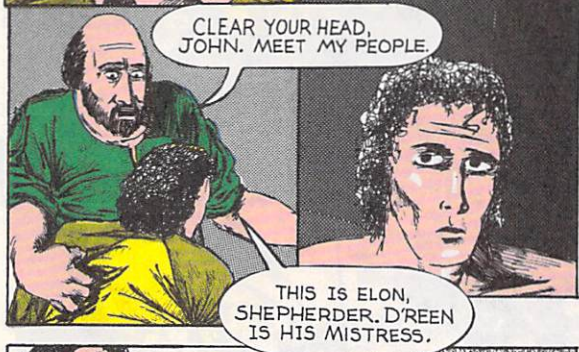


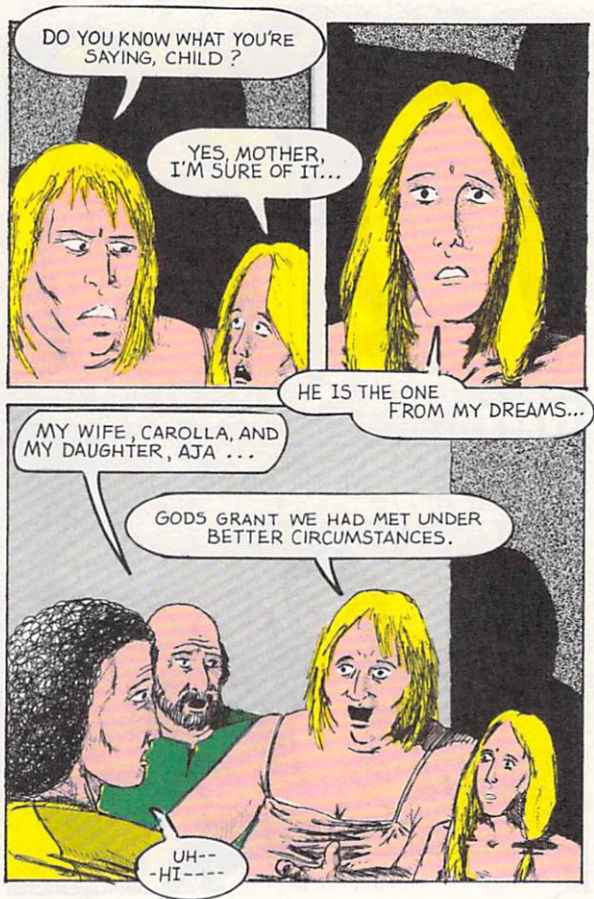












THE GIRL-- AJA-- LOOKING AT ME---



-- LIKE SHE KNOWS ME-----

I'M-- I'M STILL CONFUSED... WHAT
WOULD SOME POWERFUL WIZARD
WANT WITH ME ?



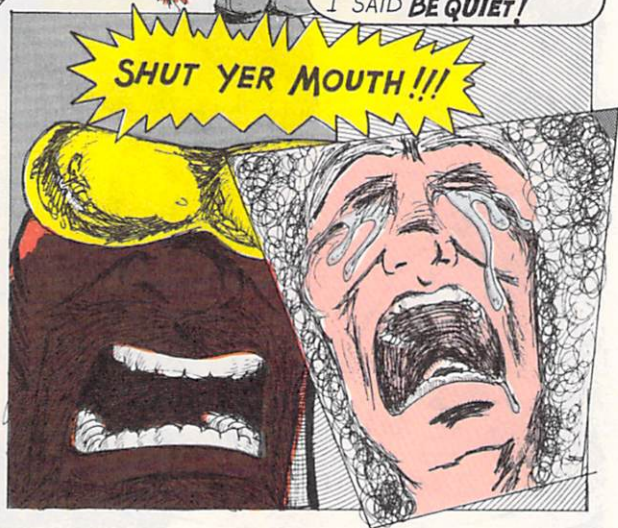
JOHN!

WHY I WANT YOU, MY BOY ?

... A FAR MORE THREATENING
DRAMA IS BEING PLAYED OUT
BENEATH THE SURFACE OF
THE SWAMP, NOT MANY
MILES AWAY!

FOR BLOOD'S SAKE, ERIN,
STOP SCREAMING! MY SCANNERS
ARE JAMMED ENOUGH AS IT IS!



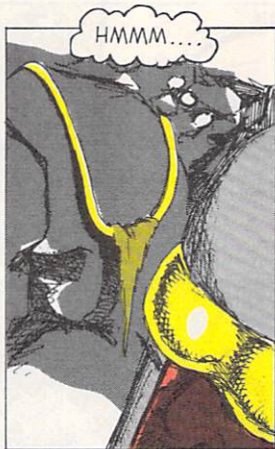


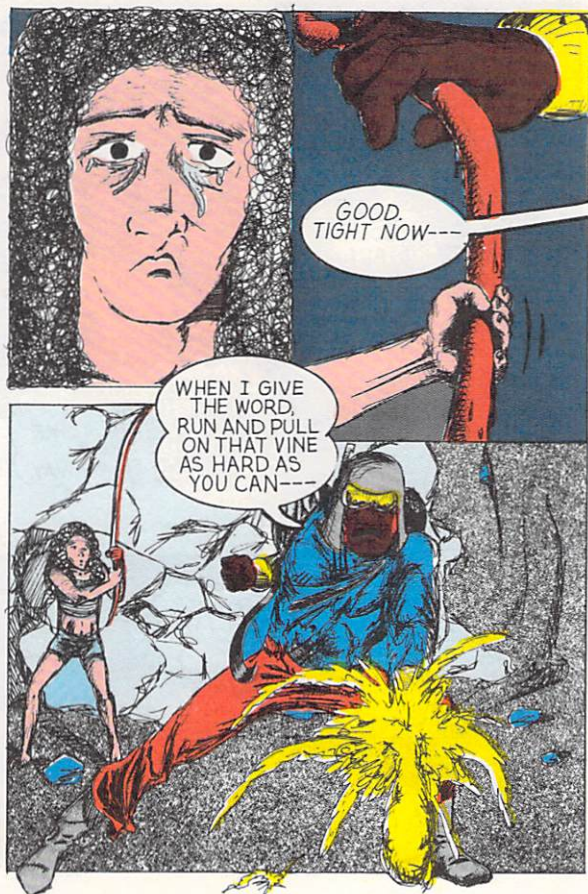




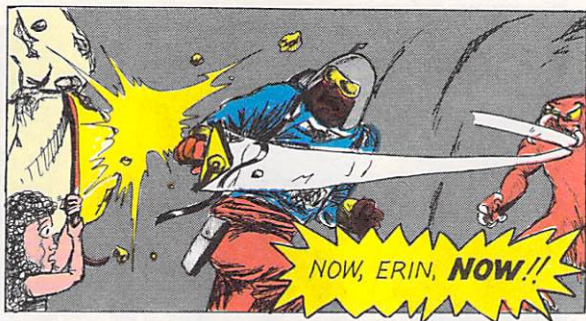
-- J-JACQUE--?









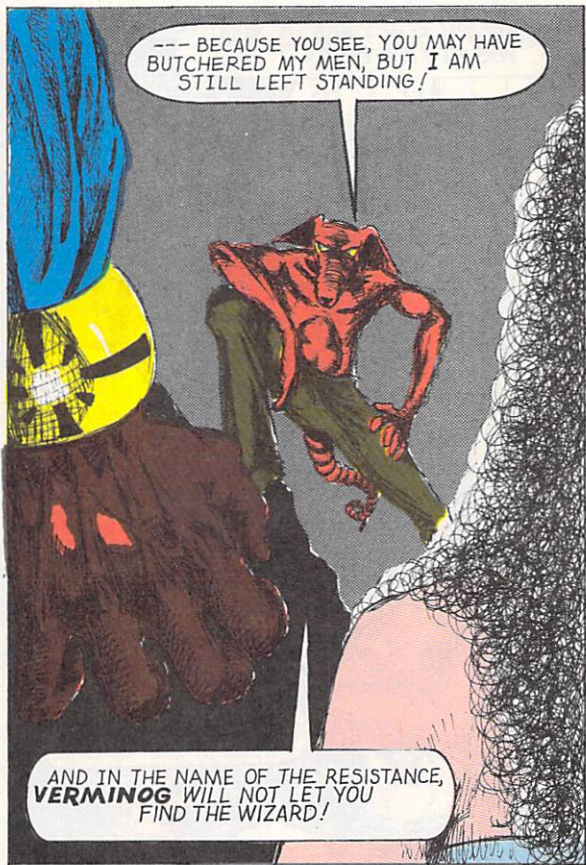



THE CAVE WALLS BUCKLE UNDER THE
FORCE OF THE BLOW---











Y'KNOW, I'VE BEEN RUNNING ALL DAY,
AND I THINK I'M GLAD I'VE FINALLY
MET SOMEONE IN CHARGE!

STAND BACK, ERIN...
I THINK I'M
GOING TO BLOW HIS
STINKIN' HEAD OFF!

THE
CHALLENGE
IS
ACCEPTED...

... WHILE THE LIVES OF TWO INNOCENT,
UNWILLING CHILDREN LAY HELPLESSLY
ON THE LINE!

... AND THOUGH
THEY MAY
BE
DEFIANT...

... AND THOUGH
THEY MAY
BE
SPIRITED....

.... THEY ARE STILL CHILDREN,
WHO HAVE UNEXPECTEDLY
BECOME ----

TIMELOST!

Battle at Stonehenge

How long can you defy the terrible Warbirds of the Wizard? Test your skill with a positron beam as you fight off the dreaded Warbirds in the Battle of Stonehenge. But beware: if a Warbird reaches the ground, the game is over.

Enter the following program and run it. You control the Mystery Man at the bottom of the screen. Press the "S" key to move left and the "D" key to move right. Use the space bar to fire his Positron Blaster when he is directly under the Warbird! To get more points, wait until the Warbird is as close as possible to the Mystery Man.

Game Program Listing

```
8 POKE788,52
9 LT=0
10 POKE52,48:POKE56,48:CLR
20 CS=256*PEEK(52)+PEEK(51)
22 POKE56334,PEEK(56334)AND254
24 POKE1,PEEK(1)AND251
30 FORI=CSTOCS+511:POKEI,PEEK(I+53248-CS)
   ):NEXT
32 POKE1,PEEK(1)OR4
34 POKE56334,PEEK(56334)OR1
35 POKE53272,(PEEK(53272)AND240)+12
36 FORK=1TO12:READX:POKE13000+K,X:NEXT
38 DATA169,127,141,0,220,173,1,220,141,2
14,50,96
```

```
40 FORI=0TO7:READJ:POKECS+I,J:NEXT
50 DATA60,153,90,60,24,24,60,24
60 FORI=0TO7:READJ:POKECS+28*8+I,J:NEXT
70 DATA60,24,24,60,90,153,60,24
80 FORI=0TO7:READJ:POKECS+37*8+I,J:NEXT
90 DATA24,152,255,61,60,60,228,4
100 FORI=0TO7:READJ:POKECS+33*8+I,J:NEXT
110 DATA24,25,255,188,60,60,39,32
120 FORI=0TO7:READJ:POKECS+27*8+I,J:NEXT
130 DATA255,255,255,255,255,255,255,255
140 FORI=0TO7:READJ:POKECS+29*8+I,J:NEXT
150 DATA3,3,3,3,3,3,3,3
152 FORI=0TO7:READJ:POKECS+31*8+I,J:NEXT
153 DATA192,192,192,192,192,192,192,192
155 FORI=0TO7:READJ:POKECS+38*8+I,J:NEXT
160 DATA0,0,0,0,255,255,255,255
170 PRINT"[SC]"
172 PRINT"BATTLE AT STONEHENGE":PRINT
173 PRINT:PRINT:PRINT"PRESS SPACE BAR TO
FIRE":PRINT
174 PRINT"PRESS 1 TO MOVE LEFT":PRINT:PR
INT"PRESS 2 TO MOVE RIGHT"
175 PRINT:PRINT:PRINT
176 PRINT"NAME? (19 LETTERS MAX)":INPUTN
$:PRINT"[HM]":PRINT"[SC]"
177 POKE53281,7
180 S=1024:CL=55296:DI=54272
190 FORI=1277TO1279:POKEI,38:POKEI+DI,0:
NEXT
195 FORI=1288TO1290:POKEI,38:POKEI+DI,0:
NEXT
200 FORI=1TO6:READJ:POKEJ,29:POKEJ+DI,0:
NEXT
205 DATA1317,1357,1397,1328,1368,1408
207 FORI=1TO6:READJ:POKEJ,31:POKEJ+DI,0:
NEXT
```

```
208 DATA1319,1359,1399,1330,1370,1410
210 FORI=1754TO1758:POKEI,27:POKEI+DI,0:
NEXT
215 FORI=1769TO1773:POKEI,27:POKEI+DI,0:
NEXT
220 FORI=1TO16:READJ:POKEJ,27:POKEJ+DI,0
:NEXT
225 DATA1795,1835,1875,1797,1837,1877,18
10,1850,1890
226 DATA1812,1852,1892,1915,1917,1930,19
32
230 X$="SCORE="
240 FORI=1024TO1143:POKEI,27:POKEI+DI,0:
NEXT
245 PRINTSPC(5)X$
250 SC=0:LL=22:RR=24:SM=54272:GOSUB760
255 R=INT(8*RND(X))+16:SH=INT(5*RND(X))+
9
260 FORC=SHTO22:IFC=22THEN680
270 POKES+40*LL+RR,37:POKECL+40*LL+RR,1
280 POKES+40*C+R,0:POKECL+40*C+R,2
290 SYS13001
300 IFPEEK(13014)=254THEN410
310 IFPEEK(13014)=247THEN350
340 IFPEEK(13014)=239THEN530
345 GOTO490
350 POKES+40*LL+RR,32
360 RR=RR+1
370 IFRR>23THEN420
375 GOTO480
380 POKES+40*LL+RR,32
390 LL=LL+1
400 IFLL>22THEN450
405 GOTO480
410 POKES+40*LL+RR,32
```

```
420 RR=RR-1
430 IFRR<16THEN360
435 GOTO480
440 POKES+40*LL+RR,32
450 LL=LL-1
460 IFLL<17THEN390
480 POKES+40*LL+RR,33:POKECL+40*LL+RR,1
490 POKES+40*C+R,32
500 POKES+40*(C+1)+R,28:POKECL+40*(C+1)+
R,2
510 FORL=SMTOSM+24:POKESM,0:NEXT
512 POKESM+5,8:POKESM+6,143:POKESM+24,15
514 POKESM+1,200:POKESM,125:POKESM+4,33
516 FORT=1TO10:NEXT:POKESM+24,0
520 NEXTC
530 LK=LL-1:FORML=LKTO5STEP-1
550 POKES+40*ML+RR,30:POKECL+40*ML+RR,6:
IFML=CTHEN570
565 GOTO580
570 IFRR=RTHEN620
580 POKES+40*ML+RR,32
590 IFC>MLTHEN490
600 NEXTML:GOTO490
620 POKES+40*ML+RR,32
630 FORL=SMTOSM+24:POKESM,0:NEXT
635 POKESM+5,8:POKESM+6,143:POKESM+24,15
640 POKESM+1,7:POKESM,1:POKESM+4,129
645 FORT=1TO1000:NEXT:POKESM+24,0
650 SC=SC+C:GOSUB740:GOSUB760:GOTO255
680 GOSUB810
690 GETA$
700 IFA$="Y"THEN850
710 IFA$="N"THEN730
720 GOTO690
730 END
```



```
740 POKES+40*LL+RR,32:POKES+40*C+R,32:FORI=1076TO1081:POKEI,32:NEXT
750 FORI=1427TO1627:POKEI,32:NEXT:RETURN
760 Y$=STR$(SC):FORX=1TOLEN(Y$)
780 POKE1075+X,ASC(MID$(Y$,X,1)):POKE55347+X,1:NEXT:RETURN
810 SN=1
811 IFSC>HSTHENHS=SC:HS$=N$:SN=0
820 Y$="HIGH SCORE="+STR$(HS)
825 FORX=1TOLEN(Y$):GOSUB900:POKE1443+X-INT(LEN(Y$)/2),A1
827 POKE1443+DI+X-INT(LEN(Y$)/2),1:NEXT
830 Y$="BY "+HS$:IFSN=1GOTO834
831 IFLT=0GOTO834
832 FORZ=1TOLEN(Y$):IFPEEK(1025+Z)=27THENLT=Z+2:Z=LEN(Y$)
833 NEXT
834 IFLT=OTHERLT=LEN(Y$)
835 FORX=1TOLT:GOSUB900:POKE1523+X-INT(LT/2),A1
836 POKE1523+DI+X-INT(LT/2),1:NEXT
837 Y$="ANOTHER GAME?"
840 FORX=1TOLEN(Y$):GOSUB900:POKE1596+X,A1:POKE1596+DI+X,1:NEXT:RETURN
850 Y$="NAME? (19 LETTERS MAX)":FORX=1TOLEN(Y$):GOSUB900:POKE1592+X,A1
851 POKE1592+X+DI,1:POKE1025+X,27:POKE55297+X,0:NEXT
852 INPUT"[CD][CD][CD][CD][CD][CD][CD][CD][CD][CD][CD][CD]";N$:PRINT"[HM]":GOSUB740:GOTO250
900 A1=ASC(MID$(Y$,X,1)):IFA1>63ANDA1<96THENA1=A1-64
910 RETURN
```

Variables in this Program Include:

- CL = Start of color code memory table
 CS = Start address of new character set
 DI = CL-S or number to add to S to get color code table
 S = Screen start address
 X\$ = String constant "SCORE="
- A1 = Screen code number
 HS = Highest score
 HS\$ = String of player's name having highest score
- LL = Y coordinate of Jacque
 ML = Y coordinate of Missile
 N\$ = Present player's name
 R = X coordinate of Warbird
 RR = X coordinate of Jacque
 SC = Score
 SH = Y coordinate of Warbird
 SM = Start address of sound registers
 LT = Length of high score name
 SN = Indicator for length of HS\$ when SC<HS

Polling Keyboard

Below is the short machine-language program to enable polling in BASIC.

```

A97F      LDA  #$7F      LOAD A REGISTER
                        WITH $7F
8D00DC    STA  $DC00     STORE 127 DECIMAL
                        AT ADDRESS $DC00
  
```

```
AD01DC  LDA  $DC01  LOAD A REGISTER
                               WITH CONTENTS OF
                               ADDRESS $DC01
8DD632  STA  $32D6  STORE CONTENTS
                               OF A REGISTER IN
                               RAM @ ADDRESS
                               $32D6
60      RTS      RETURN FROM
                               SUBROUTINE TO
                               BASIC
```

LDA, STA, etc., are mnemonics or instructions. Directly to the right are parameters associated with those instructions. The \$ means the parameter is a hex number. If a # sign precedes the \$ sign, for example, LDA #\$7F, it means to load a register with the hex number (7F) or decimal 127. Without the # sign, the instruction would be to load a register with the contents of memory address \$7F.

The hex numbers in the left column are the numbers to be poked into a protected memory area as the machine-language program. These hex numbers must be in decimal to be poked into consecutive memory locations. (See line 38 of "Stonehenge" listing.)

Number 127 is loaded into the A register, then stored at address \$32D6. This number selects a specified row from a keyboard matrix; in this case a row containing the numbers 1 and 2 and the space bar.

Depending on which key is held down, a number indicating that key will be present at address

\$DC01. This number is then stored in address \$32D6 or decimal 13014 in RAM. BASIC can now PEEK 13014 to find the number. The comments in the right column explain the flow.

Note: The 6502 microprocessor requires addresses to be specified low byte first, then high byte. This explains why the address in the left column is reversed—i.e., DODC is actually DC00 or 56320 decimal.

How the Program Works

8 Disables the BASIC RUN/STOP routine, allowing keyboard polling

10 Protects 512 bytes for new character set

20 Determines start address of new character set

30-35 Transfer first 65 characters of standard upper case from ROM to new character address

36-38 Load machine-language routine for polling keyboard

40-160 Change selected screen characters to characters needed for the game. Note: Do not replace screen characters (usually alphabet) needed in the game.

40-70 Create two Warbird figures to simulate movement, and 80-100 do the same for a man

170-175 Instructions for playing game

176 Inputs player's name for keeping track of highest score. Caution: More than 19 letters in a name can ruin the display. Use only up to 19 letters.

180 Puts constants into variable table. S is screen start address, CL is start of color code memory table, DI is the difference (CL-S).

190-256 Use the custom characters to draw the screen

230-245 Draw border around "SCORE="

250 Initializes variables. SC is score, LL is Y coordinate of man, RR is X coordinate of man. Subroutine 760 prints score on screen.

255 Creates random coordinates for the Warbird's flight start. SH is Y coordinate, and R is X coordinate.

260 Determines length of Warbird's flight

270-280 Print man and Warbird original start positions on screen

290-340 Keyboard polling. SYS is BASIC's entry to a machine-language program.

300 PEEKs for key "1", 310 for "2", and 340 for space bar. (See description provided.)

345 If no key is pressed, program goes to move Warbird.

350-460 Determine movements of man. Old figure is first erased, and coordinate S is determined for new position.

480 Draws man in new position with arms and legs reversed to simulate running

490 Erases Warbird

500 Draws Warbird in new position with wing position changed to simulate flight

510-516 Produce a sound tone for movement of Warbird

520 Starts movement over again

530-620 Routine to move Positron Blast. LL is Y coordinate of Warbird, ML is missile Y coordinate, and C is end of missile path.

550-570 Determine if Warbird has been hit, and if so, go to produce sound blast at 630 and 645, using white noise

650 Determines new score, goes to subroutine to erase old score, goes to subroutine to print new score, and goes back and starts Warbird from top of screen again

680 Goes to routine to ask if another game should be started

690-730 Give player choice of new game or not

740-755 Erase Warbird, man, score, and printing that shows previous best score

760-780 Subroutine to print score. 760 converts score to a string, and 780 pokes the score in.

810-910 Subroutine updating old scores and inquiring for another game. 811 uses the variable HS to keep track of highest score, and HS\$ is the string containing the name of player with highest score.

820-827 Put high score and name on screen and center the display

830-836 Do the same for "By Name"

837-840 Print "ANOTHER GAME ?"

850-852 Let new player input name

900-910 Convert CHR\$ code numbers to screen code numbers

How You Can Change the Program:

1. Make the Warbird move in a zigzag pattern down the screen, instead of straight down.
2. Penalize the player for missed shots. For example, give 10 points for every Warbird the man hits and take away 5 points for every one missed.

3. Allow multiple Warbirds on the screen, instead of just one at a time.
4. Allow the Warbird to descend in random amounts. For example, have the Warbird descend one to three columns at a time. You may also allow this speed as an option. A player may select beginner or expert game play, setting the speed accordingly.
5. In this and many of the other games, allow a fixed number of shots and creatures to appear. You win if you get them all. For example, change this game to allow 100 shots and 100 Warbirds. Can you play a perfect game?

Attack of the Slime Creatures

Hold off the Slime Creatures! How long can you blast the terrors as they come closer and closer?

The Slime Creatures appear and disappear as they swim underwater toward Jacque. Then suddenly they appear for a moment as they surface to attack Jacque.

Shoot at them with Jacque's gun by pressing the space bar, and score more points the closer they are. You'll hear a sound if they're hit. Otherwise, the Positron Blast will have missed because they have dived into the swamp before being hit. But watch out. If you miss, the Swamp Creatures may randomly move up to twice as far compared to when you don't fire. Your only clue as to how long they'll appear is a series of beeps before they surface as their air bubbles rise to the surface. The number of beeps is proportional to how long they'll be visible. Also, as they come closer, the pitch and intensity of the sounds increase. So, if you listen carefully, you will have important clues as to where the Swamp Creatures appear and for how long.

Game Program Listing

```
2 PRINT"[SC]":PRINT"  ATTACK OF
THE":PRINT:PRINT"  SLIME CREATURES"
3 PRINT:PRINT:PRINT"  PRESS SPACE BAR TO
FIRE ":PRINT:PRINT:PRINT
5 POKE788,52
10 POKE52,48:POKE56,48:CLR
20 CS=12288
21 POKE56334,PEEK(56334)AND254
23 POKE1,PEEK(1)AND251
30 FORI=0TO511:POKEI+12288,PEEK(I+53248)
:NEXT
31 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
)OR1
32 POKE53272,(PEEK(53272)AND240)+12
34 FORK=1TO12:READX:POKE13000+K,X:NEXT
35 DATA169,127,141,0,220,173,1,220,141,2
14,50,96
40 FORI=0TO7:READJ:POKECS+37*8+I,J:NEXT
50 DATA24,152,255,61,60,60,228,4
52 FORI=0TO7:READJ:POKECS+27*8+I,J:NEXT
53 DATA255,247,247,119,182,213,251,7
60 FORI=0TO7:READJ:POKECS+30*8+I,J:NEXT
70 DATA227,201,145,1,113,241,161,129
72 FORI=0TO7:READJ:POKECS+29*8+I,J:NEXT
73 DATA240,239,213,182,119,247,247,255
80 FORI=0TO7:READJ:POKECS+31*8+I,J:NEXT
90 DATA193,225,241,240,240,240,240,248
100 FORI=0TO7:READJ:POKECS+28*8+I,J:NEXT
110 DATA255,255,0,0,255,255,255,255
114 FORI=0TO7:READJ:POKECS+I,J:NEXT
116 DATA255,255,255,255,255,255,255,255
120 POKE53280,0:POKE53281,0:PRINT"[SC]"
121 PRINT"NAME?(19 LETTERS MAX)":PRINT:
INPUTN$:PRINT"[HM]":PRINT"[SC]"
```

```
122 FORI=1024TO1103:POKEI,0:POKEI+54272,
5:NEXT
124 H$="[CD] [CL] [CL] [CL] [CL] [CL] [CL] [CL]
[CL] [CL] [CL] [CL] [CL] [CL]HIGH SCORE":H$=H
$+STR$(HS)+" BY ":HS$=H$+HS$
125 X$="[WH] SCORE=" " :PRINTSPC
(5)X$;HS$:PRINT"[HM] "
130 I=3:S=1024:CL=55296:DI=54272:SID=542
72
140 SC=0:X=0
145 IFTA=2THEN440
290 FORJJ=22TO11STEP-1
300 FORII=ITO35
310 POKES+40*JJ+II,0:POKECL+40*JJ+II,2
320 NEXTII
330 X=X+1
340 IFX>3THEN350
345 GOTO370
350 I=I+1
360 X=0
370 NEXTJJ
430 REM **INITIAL POSITIONS
440 POKES+40*12+4,37:POKECL+40*12+4,6
450 M=INT(9*RND(X))+10
460 M=M-INT(3*RND(X))-1
462 FORJO=1TO100*M
463 NEXTJO
465 IFM<6THEN730
470 GOSUB900
472 TT=RND(X)*M
473 FORJJ=1TOTT
474 FORL=SIDTOSID+24:POKEL,0:NEXT
475 FORTM=1TOM-6:POKESID+14,5:POKESID+18
,16:
476 POKESID+3,1:POKESID+24, TM+128:POKESI
D+6,240
```

```
477 POKESID+4,65:FR=5389
478 FORT=1TO20
481 POKESID+0,107.5:POKESID+1,21
482 NEXTT
483 POKESID+24,0
484 NEXTTM
490 GOSUB920
510 FORJ=1TOTTT*10
530 SYS13001
540 IFPEEK(13014)=239THEN590
550 NEXTJ
560 GOSUB900
570 GOTO460
590 FORRR=5TOM
600 TT=TT-1
610 IFTT<=0THEN810
620 POKES+40*12+RR,28
630 POKES+40*12+RR,0
640 IFRR=MTHENGOSUB930:GOSUB900:GOTO660
650 NEXTRR
660 POKESID+3,1:POKESID+24,143:POKESID+6
,240
662 POKESID+4,129:FR=5389
664 :FORT=1TO100
666 POKESID+0,87.5:POKESID+1,15
668 NEXTT
670 POKESID+24,0
690 SC=SC+(30-M):Y$=STR$(SC)
700 FORX=1TOLEN(Y$)
710 POKES+54+X,ASC(MID$(Y$,X,1)):POKECL+
54,1
720 NEXT
725 GOTO450
730 M=5:GOSUB920:IFTA<>2THENA=2:HS=SC:HS
$=N$:GOTO765
```

```
732 IFN$=""THEN765
735 IFSC>HSTHENHS=SC:HS$=N$
765 Y$="ANOTHER GAME ?":FORX=1TOLEN(Y$):
GOSUB950:POKE1156+X,A1:POKE1156+DI+X,1:N
EXT
770 GETA$
775 IFA$="Y"THEN788
780 IFA$="N"THEN870
785 GOTO770
788 PRINT"[SC]":GOTO122
800 FORX=1092TO1243:POKEX,32:NEXT:GOTO14
0
810 POKES+40*12+RR,0:GOSUB900
820 FORJ=RRTO21
830 POKES+40*12+J,28
840 POKES+40*12+J,0
850 NEXTJ
860 GOTO460
870 END
900 POKES+40*12+M,0:POKES+40*13+M,0:RETU
RN
920 POKES+40*13+M,30:POKES+40*12+M,30:PO
KES+40*13+M,31:RETURN
930 POKES+40*12+M,27:POKES+40*13+M,29:FO
RX=0TO100:NEXT:RETURN
950 A1=ASC(MID$(Y$,X,1))
955 IFA1>63ANDA1<96THENA1=A1-64
960 RETURN
```

Variables in this Program Include:

CL = Start of color code memory table

CS = Start address of new character set

DI = CL-S or number to add to S to get color
code table

- S = Screen start address
- X\$ = String constant "SCORE ="
- A1 = Screen code number
- HS = Highest score
- HS\$ = String of player's name having highest score
- M = X coordinate of Slime Creature
- N\$ = Present player's name
- TT = Time allowed for player to fire

How the Program Works

- 2-3 Clear screen and print instructions
- 5 Disables run/stop routine, allowing keyboard polling by machine language
- 10 Protects 512 bytes below the screen start address for new character set
- 20-23 Determine start address of new character set
- 30-31 Transfer first 65 characters of standard upper case from ROM to new character address
- 34-35 Load machine-language routine for polling keyboard
- 40-116 Change selected screen characters to characters needed for the game
- 120 Initializes custom character set
- 121 Lets player input name for keeping track of highest score
- 122 Prints green border for score
- 124 HS is used to store high score, and HS\$ stores player's name
- 125 Prints "SCORE=" and then prints "(HOME)" to move cursor back to start

130-140 Put constants into variable table and initialize variables. I is left screen, S is screen start address, CL is start of color code memory table, and DI is difference (CL-S).

145 TA=2 is key to bypass drawing screen after first game

290-370 Draw swamp

440 Prints Jacque on screen

450 Randomly selects Slime Creature's start position

460 Randomly determines how far Slime Creature moves each time

462-463 Timing loop that randomly determines how long the Slime Creature is under water

465 Determines if a Slime Creature got Jacque; if so, then start another game

470 To subroutine to erase Slime Creature

472-484 Create a random-length series of tone bursts, in which the length is proportional to how long the Slime Creature is visible

490 To subroutine to print a Slime Creature

500-550 TT gives random amount of time for player to fire. SYS at 530 is BASIC's entry command to a machine-language program. 540 peeks to see if space bar is pressed. NOTE: to make game more difficult, line 510 can be changed by decreasing TT's multiplier (10) to a smaller number.

560 To subroutine to erase the Slime Creature

570 Goes back and moves the Slime Creature again

590-650 Move Positron Blast, and if time runs out (TT=0), then the Slime Creature disappears and

Blast continues to end of swamp. If not, then subroutine 930 prints an explosion, 900 erases it, and 660-670 is the sound of the explosion.

690-720 Print score

725 Goes back and repeats routine

730 TA=2 is key to bypass drawing screen after first games. The Slime Creature is shown grabbing Jacque.

765 Asks about another game

770-785 Input player's decision

800-810 Erase printing and return for another game

820-850 Routine to move blast to end of swamp

860 Back to move a Slime Creature again

900 Erases Slime Creature subroutine

920 Get Slime Creature subroutine

930 Explosion subroutine

950-960 Convert CHR\$ code numbers to screen code numbers

How You Can Change the Program

1. Allow multiple Slime Creatures to appear and disappear. You would allow Jacque to move on the screen to position him in the row of the creature you want to shoot.
2. Limit the number of positron blasts and Slime Creatures per game. For example, allow 20 Slime Creatures and 20 Positron Blasts to see if Jacque can hit them all. You should also display the remaining Positron Blasts.

Peril of the Pitdemons

Now, you must fight off a fierce attack by underground creatures of the Wizard! In this game, you control Jacque's blaster as he fights against the swarming demons from the dark pit.

You move Jacque, using the "1" control to go left and the "2" control to move right. Use the space bar to fire the Positron Blaster. Your score is shown on the left of the screen. You can play until a Pitdemon reaches the top row, and then the game ends. You get more points the closer you let the Pitdemons approach Jacque.

Game Program Listing

```
5 POKE788,52
10 POKE52,48:POKE56,48:CLR
12 DIMAA(7,2)
13 FORJ=1TO7:FORI=1TO2:READAA(J,I)
14 NEXTI:NEXTJ:J=0:IFTA=2THEN590
15 DATA21,13,21,10,21,9,21,12,21,8,21,7,
21,11
20 CS=12288
21 PRINTCHR$(142)
22 POKE56334,PEEK(56334)AND254
23 POKE1,PEEK(1)AND251
```



```
30 FORI=0TO511:POKEI+12288,PEEK(I+53248)
: NEXT
31 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)
)OR1
32 FORK=1TO12:READA:POKE13000+K,A: NEXT
34 DATA169,127,141,0,220,173,1,220,141,2
14,50,96
35 POKE53272,(PEEK(53272)AND240)+12
40 FORI=0TO7:READJ:POKECS+27*8+I,J: NEXT
50 DATA24,152,255,61,60,60,228,4
60 FORI=0TO7:READJ:POKECS+28*8+I,J: NEXT
70 DATA24,61,128,188,60,60,39,32
80 FORI=0TO7:READJ:POKECS+29*8+I,J: NEXT
90 DATA24,61,128,188,60,61,66,128
100 FORI=0TO7:READJ:POKECS+30*8+I,J: NEXT
110 DATA24,188,126,61,60,188,66,1
120 FORI=0TO7:READJ:POKECS+31*8+I,J: NEXT
130 DATA24,24,24,24,24,24,24,24
140 FORI=0TO7:READJ:POKECS+37*8+I,J: NEXT
150 DATA255,255,255,255,255,255,255,255
160 POKE53280,0 :POKE53281,0
:PRINT
161 POKE53280,14:Y$="SCORE="
162 PRINT"[CD][CD][22 SPACES][YL] PERIL
OF THE[CD][CD][12
SPACES][CL]PITDEMONS"
163 PRINT"[CD][CD][22 SPACES][WH] 1=LEFT
"
164 PRINT"[CD][22 SPACES] 2=RIGHT      "
165 PRINT"[CD][22 SPACES] SPACEBAR= FIRE"
167 S=1024:CL=55296:DI=54272:SID=54272:
168 FORX=1064TO1085:POKEX,37:POKEX+DI,7:
NEXT
170 FORX=1TOLEN(Y$):A1=ASC(MID$(Y$,X,1))
:IFA1>63ANDA1<96THENA1=A1-64
```

```
180 POKE1049+X,A1:POKE1049+X+DI,5:NEXT
200 LL=10:IFTA=2THEN500
350 READP:IFP=255THEN500
380 READU,C1:FORX=UTOC1:POKES+40*P+X,37:
POKECL+40*P+X,7:NEXT
390 GOTO350:
400 DATA5,0,4,5,17,21,6,0,4,6,17,21,7,0,
3,7,18,21
410 DATA8,0,3,8,18,21,9,0,2,9,19,21,10,0
,1,10,20,21,11,0,1,11,20,21
420 DATA12,0,0,12,21,21,13,0,0,13,21,21,
14,0,1,14,20,21
430 DATA15,0,1,15,20,21,16,0,2,16,19,21,
17,0,2,17,19,21,18,0,3,18,18,21
440 DATA19
,0,4,19,17,21,20,0,4,20,17,21,21,0,5,21,
16,21,22,0,5,22,16,21,255
500 POKES+40*3+LL,27:POKECL+40*3+LL,5
520 IFTA=2THENRESTORE:GOTO12
590 IFXX<>0THEN620
600 GOSUB810
620 POKES+40*3+LL,28:POKECL+40*3+LL,5
625 SYS13001
630 IFPEEK(13014)=239THEN1050
640 IFPEEK(13014)=254THEN690
650 IFPEEK(13014)=247THEN740
660 GOSUB810:GOTO620
690 POKES+40*3+LL,27:POKECL+40*3+LL,4:PO
KES+40*3+LL,32
700 LL=LL-1:IFLL<7THEN750
720 POKES+40*3+LL,27:POKECL+40*3+LL,5
730 GOTO780
740 POKES+40*3+LL,32
750 LL=LL+1:IFLL>13THEN700
770 POKES+40*3+LL,27:POKECL+40*3+LL,5
```

```
780 GOSUB810:GOTO620
810 IFXX=0THEN870
820 M=INT(RND(X)*7)+1:A=AA(M,1):B=AA(M,2)
)
821 Z=B
850 POKES+40*A+B,32
860 IFXX=1THEN880
870 FORM=1TO7
875 IFM>7THENM=7
880 AC=AA(M,1):AB=INT(RND(X)*5)+1:AC=AC-
AB:B=AA(M,2)
881 IFB<4THENB=B+4
882 IFB>15THENB=B-3
920 IFAC>5THEN970
930 POKES+40*3+B,29:POKECL+40*3+B,4:AA(M
,1)=4:GOTO1310
960 IFAA(M,1)<6THEN1310
970 POKES+40*AC+B,29:POKECL+40*AC+B,4
980 FORL=SIDTOSID+24:POKEL,0:NEXT
981 POKESID+5,9:POKESID+6,0
982 POKESID+24,15-AC/2:DR=50
983 POKESID+1,25:POKESID,177-AC*8:FORT=1
TODR:NEXT
990 AA(M,1)=AC
1000 IFXX=1THEN1030
1010 NEXTM:XX=1
1030 RETURN
1050 FORJ=1TO7:IFLL=AA(J,2)THEN1100
1070 NEXTJ
1100 FORLK=5TO19
1110 POKES+40*LK+LL,31:POKECL+40*LK+LL,1
:POKES+40*LK+LL,32
1130 IFLK=AA(J,1)THEN1170
1140 NEXTLK
1150 GOSUB810:GOTO620
```

```
1170 AA(J,1)=21:GOSUB1210:GOTO620
1210 POKES+40*LK+LL,32
1220 POKEL,0
1222 FORX=15TO1STEP-1:
1223 POKESID+24,X:POKESID+4,129:POKESID+
1,40:POKESID,200:NEXT:
1230 POKESID+4,0:POKESID+5,0
1250 SC=SC+32-LK:Y$=STR$(SC)
1260 FORX=1TOLEN(Y$)
1270 POKE1057+X,ASC(MID$(Y$,X,1)):POKE10
57+X+DI,5
1280 NEXT
1290 RETURN
1310 TA=2
1350 Y$="ANOTHER GAME?"
1355 FORX=1TOLEN(Y$):GOSUB1550:POKE1904+
X,A1:POKE1904+X+DI,5:NEXT
1360 GETA$
1370 IFA$="Y"THEN1415
1380 IFA$<>"N"THEN1360
1390 END
1415 FORJ=1TO7
1420 FORJ=1TO7
1430 RE=AA(J,1)
1440 CE=AA(J,2)
1450 POKES+40*RE+CE,32
1455 NEXTJ:XX=0
1460 FORX=1058TO1062:POKEX,32:NEXT:SC=0
1465 FORX=1119TO1127:POKEX,37:POKEX+DI,7
:NEXT
1500 CLR:PRINT"[HM]":RUN
1550 A1=ASC(MID$(Y$,X,1)):IFA1>63ANDA1<9
6THENA1=A1-64
1560 RETURN
```


Variables in this Program Include:

- CL = Start of color code memory table
- CS = Start address of new character set
- DI = CL-S or number to add to S to get color code table
- S = Screen start address
- Y\$ = String constant "SCORE ="
- A = Present Y coordinate of Pitdemon
- AA = Array of X & Y coordinates of Pitdemons
- AC = Next Y coordinate of Pitdemon
- A1 = Screen code number
- B = X coordinate of Pitdemon
- LK = Y coordinate of Blast
- LL = X coordinate of Jacque and of the Blast

How the Program Works

- 5 Disables the basic run/stop routine, allowing keyboard polling
- 10 Protects 512 bytes below the screen address for new character set
- 12-15 Create an array with the coordinates for the seven Pitdemons
- 20-23 Determine start address of new character set
- 30-31 Transfer first 65 characters of standard upper case from ROM to new character address
- 32-35 Load machine-language routine for polling keyboard
- 40-150 Change selected screen characters to characters needed for the game. 40-70 create two

Jacque figures used alternately to simulate movement. 80-110 do the same for the Pitdemons, but only one figure is used in the game to speed up their movement.

160 Initializes custom character set

165 Clears screen to black background and creates string for score

167 Puts constants into variable table. S is screen start address, CL is start of color code memory table, DI is the difference (CL-S).

168 Top of pit in yellow

170-180 Draw "SCORE =". A1 is required to convert CHR\$ numbers for alphabet to screen code numbers.

200 LL= Jacque's initial X coordinate. TA is key to keep from redrawing screen for other than the first game.

350-440 Draws pit

500 Places Jacque at start position

520 TA=2 sends routine back to restore original data for Pitdemon's positions for next game.

590-600 XX-0 is key to move all Pitdemons, randomly, once, before Jacque is allowed to fire. This is true only for the first time through the routine at game start.

620 Shows Jacque's arms and legs move

625-650 Keyboard polling: SYS is BASIC's entry to a machine-language program. 630 peeks for the space bar, 640 for letter "C" and 650 for letter "B."

660 Starts process over to move Pitdemons again

690-770 Determine movements of Jacque. Old

figure is erased, and coordinates are determined for next position.

780 Starts process over and moves Pitdemons again

820-850 Randomly determine which Pitdemon will move next and erases that Pitdemon

860 Bypasses movement of all seven Pitdemons to move only one going once through routine

870-880 Determine the amount of next move by Pitdemon. Changing the #5 in $5 * \text{RND}(X)$ to a lesser # will make game easier.

920 If Pitdemon not already out of pit, routine goes on to move Pitdemon.

930 Places Pitdemon out of pit and goes to another game routine

970 Places Pitdemon on screen at next position

980-983 Short burst of tone with frequency and volume determined by distance from top of pit

990 Stores last position of Pitdemon

1010 Next M is loop to 870 to set all seven Pitdemons on first pass through routine, and $XX=1$ is where key is set to place only one Pitdemon at a time.

1030 The return for the several GOSUB 810 routines

1050-1070 Check Jacque's position in respect to Pitdemons

1100-1140 Move Positron Blast to Pitdemon. If Pitdemon is not there or out of range, the Pitdemon goes nearly to end of screen. LK is Y coordinate of blast, and LL is Y coordinate.

1150 Back to move Pitdemon again and then back to 620 to permit player to move or fire

1170 Sets Pitdemons back to pit bottom, goes to 1210 for sound blast, then back to 620 for moving or firing

1210 Erases missile

1220-1230 Sound blast using white noise

1250-1290 Determine and print score

1310 Sets key so screen need not be redrawn for new games

1350-1355 Print "ANOTHER GAME?"

1360-1390 Let player stop or play again

1415-1500 Erase players, and score; printing ready for another game

1550-1560 Convert CHR\$ code numbers to screen code numbers

How You Can Change the Program

1. Let a Warbird attack from above while the Pitdemons are climbing up from below.
2. Add some different types of creatures climbing up and give different points for them.
3. Let some of the Pitdemons become invisible and give extra points for hitting them.

In the Caverns of Carnage

Jacque and Erin are in the fearsome and eerie Caverns of Carnage, attacked by the savage Crusher beasts. How long can they survive the Chase that no one has ever gotten through? How far can YOU get them through the caverns?

In this game, Jacque is on the right and Erin on the left. Jacque's Positron Blaster will destroy the loathsome Crushers and score points for each one disintegrated. You get more points the closer they are when you blast them. Each new cavern is narrower and more treacherous than the previous one, but you score more points. Watch out—if a Crusher gets Jacque or Erin from the front or if they crash into a wall or Crusher, the game is over. Use the "S" key to move them left, the "D" key to move them right and hold the space bar to fire.

Game Program Listing

```
10 PRINTCHR$(142)CHR$(8)CHR$(147)" [BK] [C  
D] [CD] [CD] [CD] [CD]          CAVERNS OF CAR  
NAGE"  
20 PRINT"[CD] [CD] PRESS 'S' TO MOVE LEFT  
[17 SPACES]PRESS 'D' TO MOVE RIGHT
```

```

30 PRINT"          PRESS SPACEBAR TO FIRE
40 PRINT"[CD] [CD]          ONE MOMENT PLEA
SE"
50 POKE52,48:POKE56,48:CLR
60 POKE53280,7:POKE53281,07:C$="[BK] [WH]
[RD] [PR] [GR] [BL] [CY]":LV=1:S1%=12282
70 C=53248:C0=40960:C1=54272
80 POKE56334,PEEK(56334)AND254:POKE1,PEE
K(1)AND251
90 FORT=0TO511:POKET+12288,PEEK(T+C):NEX
TT
100 POKE1,PEEK(1)OR4:POKE56334,PEEK(5633
4)OR1:POKE53272,(PEEK(53272)AND240)+12
110 FORX=1TO5:READA:FORG=A*8TOA*8+7:READ
P:POKE12288+G,P:NEXTG,X
120 DATA27,24,25,255,61,60,60,39,112,28,
24,255,182,60,60,66,130,130
130 DATA29,24,61,126,18,60,205,66,8,29,2
4,24,24,24,24,24,24,24
140 DATA30,255,255,255,255,255,255,255,2
55
150 FORSN=54272TO54296:POKESN,0:NEXT:SO=
54272:POKESO+5,17:POKESO+6,9
160 PRINT"[HM] [CD] [CD] [CD] [CD] [CD] [C
D] [CD] [CD] [CD] [CD] [CD] [CD] [CD] [C
D] [CD] [CD] [CD] [CD] [CD] [CD] [CD]";
170 DL=15:S%=1024:D%=1024+40*DL:POKE5328
0,7:POKE53281,7:S1%=12288
180 Z%=20:L=40:SW%=20
190 LK=1:POKED%+Z%-40,32:POKED%+Z%-39,32
:POKESO+24,15
200 IFPEEK(1024)<>32THENLV=LV+1:PRINT"[H
M] [HM] [BK] [CD] [CD] [CD] [CD] [CD]    LEVEL"L
V:SW%=SW%-2:
210 IFSW%<=0THENSW%=0:DL=DL+2

```

```
220 A$=MID$(C$,LV,1)
230 N0=INT(RND(1)*16)+5:N1=INT(RND(1)*16)
    )+5
240 POKESO,N0:POKESO+1,N1
250 POKESO+4,33
260 POKED%+Z%,27:POKEC1+Z%,2
270 POKED%+Z%+1,28:POKEC1+Z%+1,6
280 RN=RND(1):POKESO+4,32
290 IFRN>.3THEN320
300 L=L+1:IFLV>7THENLV=1
310 GOTO340
320 IFRN<.7THEN340
330 L=L-1
340 IFL<=15THEN360
350 L=15
360 IFL>=1THEN380
370 L=1
380 PRINT"[SC][CD][CD][CD][CD][CD][CD][C
D][CD][CD][CD][CD][CD][CD][CD][C
D][CD][CD][CD][CD][CD][CD][CD]";
390 FORX=1TOL:PRINTA$"^";:NEXT
400 J=INT((SW%-1)*RND(1))
410 PRINTSPC(J);"[GN][ENGLISH POUND]";SP
C(SW%-J-1);
420 FORX=1TO40-SW%-L:PRINTA$"^";:NEXT
430 POKED%+Z%,27:POKED%+C1+Z%,2
440 POKED%+Z%+1,28:POKED%+C1+Z%+1,6
450 POKED%+Z%-40,32:POKED%+Z%-39,32
460 N=PEEK(D%+40+Z%)
470 IFN<>32THEN890
480 N=PEEK(D%+41+Z%)
490 IFN<>32THEN890
500 N=PEEK(197)
510 IFN=60THEN740
520 IFN=13THEN570
```

```
530 IFN=18THEN660
540 POKED%+Z%,32:POKED%+C1+Z%,7
550 POKED%+Z%+1,32:POKED%+C1+Z%+1,7
560 GOTO190
570 N=PEEK (D%+Z%-1)
580 IFN<>32THEN890
590 POKED%+Z%+1,32:POKED%+C1+Z%+1,7
600 POKED%+Z%,32:POKED%+C1+Z%,7
610 Z%=Z%-1:POKED%+Z%-38,32
620 IFZ%<1THEN690
630 POKED%+Z%,27:POKED%+C1+Z%,2
640 POKED%+Z%+1,28:POKED%+C1+Z%+1,6
650 GOTO190
660 N=PEEK (D%+Z%+2)
670 IFN<>32ANDN<>27THEN890
680 POKED%+Z%,32
690 Z%=Z%+1:POKED%+Z%-41,32
700 IFZ%>40THEN610
710 POKED%+Z%,27:POKED%+C1+Z%,2
720 POKED%+Z%+1,28:POKED%+C1+Z%+1,6
730 GOTO190
740 POKED%+Z%-40,32:POKED%+Z%-39,32
750 FORLK=INT ( (D%-1024) /40) +1TO25:TS=LK
760 N=PEEK (S%+TS*40+Z%+1)
770 IFN=30THENLK=26:GOTO190
780 IFN=28THENLK=26:GOTO830
790 POKES%+TS*40+Z%+1,29:POKES%+C1+TS*40
+Z%+1,0
800 POKES%+TS*40+Z%+1,32:POKES%+C1+TS*40
+Z%+1,7
810 NEXT LK
820 GOTO190
830 POKES%+TS*40+Z%+1,32
840 FORT=15TOOSTEP-.4
850 POKESO+24,T:POKESO+4,129:POKESO+5,12
```


- J = Variable used to randomly select length of the left edge of the cavern
- L = Location of left end of the cavern
- LK = Row location currently PEEKed in POSITRON Loop
- LV = Current cavern level
- N = Temporary variable used to store value PEEKed from a variety of locations
- N1 = Random sound - high value
- P = Temporary variable used to POKE 8 new bytes for each custom character
- RN = Random value determining location of monsters
- SO = Base address of sound registers
- T = Temporary counter used to POKE first 64 characters from ROM into RAM
- TS = Stores value of variable LK, used to properly exit FOR-NEXT loop when LK is less than 26
- X = Temporary counter used to POKE values for the 5 custom characters into RAM
- D% = Row where characters appear (Integer)
- S% = Start of screen memory (Integer)
- S1% = Start of newly created character set (Integer)
- SW% = Maximum width of the cavern...Decreases with increasing level of difficulty (Integer)
- Z% = Horizontal position of characters (Integer)

A\$ = Current cavern color, selected from C\$
(String)

C\$ = Contains Color control-characters for
cavern walls (String)

How the Program Works

10-40 Select and "lock" upper-case characters,
clear screen, and print instructions

50 Partitions memory for the custom character
set and clears it

60 Sets screen and border color, and defines
variables used later

70 Defines address of ROM character set, offset
between old and new sets, and color nybble

80 Turns off keyscan-interrupt while creating
new characters in RAM and switches VIC-II chip to
view ROM character set

90 Pokes first 64 characters from ROM into new
RAM locations

100 Returns VIC-II chip to normal status, and
returns keyscan interrupt to normal status

110 Reads data statements for the 5 custom
characters used, and POKES values into RAM

120-140 Data statements replacing characters
with screen codes 27 [, 28 £, 29], and 30 †

150 Clears all registers of the sound
synthesizer, and sets volume to maximum

160 Clears screen and positions cursor on
bottom line

170 Sets values for difficulty level, and character
positions on the screen

180 More values used in setting characters' positions

190-200 Clear current position of hero and girl, and check to see if cavern has reached the top of the screen. If it has, clear the screen and change cavern color and difficulty level.

210 If cavern width is less than 0, maintain it at 0.

220-250 Get random notes and POKE them into sound synthesizer

260 Places girl on the screen and sets her color to red

270 Places hero on the screen and sets his color to blue

280 Gets a random value for RN, and turns sound off

290 Tests random value for placement of cavern wall (L)

300 Adds one to the value of L...Checks difficulty level

310-370 More calculations of the cavern-wall placement

380 Positions cursor for next cavern segment added

390-420 Print left wall in color A\$, the monster in green, and the right wall in color A\$

430-450 Reposition girl and hero on the new line, clearing old positions

460-490 Is a monster or a cavern wall in front of either the girl or the hero? If so, the program will "GOTO 780", ending the game.

500-530 Which key (if any) has been pressed?
Move girl and hero or fire positron accordingly.

540-550 Erase girl and hero from current screen positions

560 Returns to line 190; continue progress through cavern if no key was pressed

570-580 Look at position below the girl...IF the space is not empty, GOTO 890.

590-600 Clear characters from their last position

610-650 Move both characters down and right IF space is empty, place girl and hero in new positions, and return to line 190

660-730 Look at the position below the hero...If the space is not empty, GOTO 890. If it is empty, place the characters in their new positions, and GOTO 190.

740 Clears last positions of girl and hero

750-760 POSITRON BLASTER LOOP

BEGINS...Check spaces where blaster fires, below the hero

770 If space is empty, or if it is a cavern wall, ignore it and go to line 190.

780 If there is a monster there, exit the loop and begin sound effects at line 830.

790-800 Place a "bullet" in the space and clear it again

810 If nothing was found, go through the loop again examining one line lower on the screen.

820 POSITRON BLASTER LOOP ENDS...GOTO 190.

830-850 A MONSTER WAS

DESTROYED...Blank the position of the monster, and turn on the sound effects, gradually decreasing the volume to simulate an echo.

860 Adds points to score, based upon the distance from hero to monster, and level of difficulty

870 Prints the new score on the screen

880 Goes back to line 190 and continues the game

890 A MONSTER GOT THEM...Erases the position behind the hero and the girl

900 Moves the cursor back to the bottom of the screen, and prints the score

910 Prints "ANOTHER GAME" message

920 Waits for a moment

930 Has a key been pressed? If not, waits until one is pressed

940 If only the "Y" key was pressed, begins the game again, with score of 0.

950 If the key pressed was NOT "N" then goes back to 810.

960 GAME ENDS...

How You Can Change the Program

1. Change Jacque and Erin to sprite-graphics, so that they move more freely.
2. Make the cavern walls invisible to Jacque and Erin.
3. Add additional sound effects with the unused sound channels.
4. Deduct points from the score for each Positron Blast that misses.

5. **Modify the program to use joysticks instead of the keyboard.**
6. **Allow Positron Blasts to create an opening in the cavern for Jacque and Erin to escape.**
7. **To help Jacque and Erin, add other creatures that can also destroy cavern monsters.**

Escape

You are in the horrible prison of the Warlord and are attempting to escape with some of your fellow prisoners. But to escape, you must leave through the River of Death by passing through the Gates of Destruction.

In this game, you can help 100 prisoners to escape by guiding them through the gates. No prisoner will get through without your help. No one has ever got all 100 safely through. You guide each prisoner's boat down the river and through the gates, using the "S" key for left and the "D" key for right. Because the current is so swift, there's no way you can stop the boats. You must maneuver them just right to pass through the gates—if possible.

Watch out for the rocks at the edge of the river. They will also make you lose a prisoner if one touches them. You'll also notice that the shoreline changes as the game progresses. This represents the tide exposing and covering some of the rocks at the edge. Good luck.

Game Program Listing

5 P=5

10 PRINTCHR\$(147)

20 POKE56,48:POKE52,48:POKE53280,6:POKE5
3281,3:POKE56334,PEEK(56334)AND254:POKE1
,PEEK(1)AND251

```
30 FORT=0TO511:POKE12288+T,PEEK(53248+T)
:NEXTT:S=1024:H=40:D=55296:R=24
40 FORT=1TO2:READA:FORN=0TO7:READB:POKE1
2288+A*8+N,B:NEXTN,T
50 DATA0,255,255,255,255,255,255,255,255
60 DATA27,24,60,36,36,36,36,60,24
70 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334
)OR1:POKE53272,(PEEK(53272)AND240)+12
80 DIMA(41)
90 FORT=1TO40STEP2
100 A(T)=12+INT(3*SIN(2*T*[PI]/20)+.5)
110 A(T+1)=28-INT(3*SIN(2*T*[PI]/20)+.5)
120 NEXT
130 FORT=1TO40STEP2
140 GOSUB1000
200 NEXTT:ES=0
210 REM INITIAL POSITIONS
220 C=20:W=5
230 POKES+R*H+C,27:POKED+R*H+C,6
240 M=100
250 FORT=1TO24STEP2
253 CN=CN+1
254 A(T)=12+INT(3*SIN((T+CN)/P+2))
255 A(T+1)=39-A(T)
256 POKES+R*H+C,32:POKED+R*H+C,3
257 A(T+1)=39-A(T)
258 GOSUB1000
260 FORN=A(T)+1TOA(T+1)-1
270 POKES+R*H+C,32:POKED+R*H+C,3
280 B=PEEK(197)
290 IFB=18THEN330
300 IFB=13THEN380
310 GOTO430
320 REM BOAT GOES RIGHT
330 C=C+1:IFC>38THEN380
```

```
350 IFPEEK(S+(R-1)*H+C)=0ORPEEK(S+R*H+C-1)=0THENC=C-1:GOTO740
360 GOTO410
370 REM BOAT GOES LEFT
380 C=C-1:IFC<2THEN330
400 IFPEEK(S+(R-1)*H+C)=0ORPEEK(S+R*H+C+1)=0THENC=C+1:GOTO740
410 REM WALL AND HERO ADVANCE
430 IFPEEK(S+R*H+C)=0ORPEEK(S+(R-1)*H+C)=0THENGOTO740
440 R=R-1
450 POKES+R*H+C,27:POKED+R*H+C,6
456 FORQ=1TO100:NEXT
458 POKE54296,31:POKE54272,10:POKE54273,N-6:POKE54276,33:POKE54277,36
459 POKE54278,128:FORQ=1TO100:NEXT:POKE54276,34
460 POKES+(T+9)*20+N,0:POKED+(T+9)*20+N,0
465 IFPEEK(S+R*H+C)=0THEN740
470 IFR<5THENPOKES+R*H+C,32:POKED+R*H+C,3:GOSUB880:GOTO510
480 NEXT:POKE54296,15:POKE54272,1:POKE54273,8:POKE54276,129:POKE54277,128
481 FORQ=1TO90:NEXT:POKE54276,128
490 GOSUB880
500 NEXT:GOTO250
510 POKE54296,15:POKE54272,50:POKE54273,9:POKE54276,33:POKE54277,64
511 FORG=1TO100:NEXT:POKE54272,100:POKE54273,15
512 FORG=1TO100:NEXT:POKE54272,200:POKE54273,60
520 FORG=1TO500:NEXT:POKE54276,32:POKE54296,0
```

```
530 PRINTCHR$(19);CHR$(144);"FREEDOM!!!  
"  
540 R=23:C=20  
550 ES=ES+1  
560 PRINT"ESCAPED=";ES  
570 M=M-1  
580 PRINT"# LEFT=";STR$(M);" "  
590 IFM=0THEN620  
600 IFT>=24THEN250  
610 NEXTT:IFM>0THEN250  
620 PRINT"ANOTHER GAME?(Y=YES,N=NO)"  
710 B=PEEK(197):IFB=64THEN710  
720 IFB=25THENPRINTCHR$(147);:GOTO130  
730 END  
740 POKE54296,31  
745 POKE54272,100:POKE54273,6:POKE54276,  
33:POKE54277,128:FORG=1TO300:NEXT  
746 POKE54272,10:POKE54273,4:POKE54276,3  
3:POKE54277,128:FORG=1TO300:NEXT  
760 POKE54296,0:POKE54276,32  
770 PRINTCHR$(19);CHR$(144);"ANOTHER  
LOST!!!":FORG=1TO500:NEXT  
780 PRINT"ESCAPED=";ES  
790 M=M-1  
800 PRINT"# LEFT=";STR$(M);" "  
810 IFM=0THEN620  
820 IFPEEK(D+R*H+C)=0THENR=R-1  
825 F=PEEK(S+R*H+C):F1=PEEK(D+R*H+C)  
830 POKES+R*H+C,27:FORB=1TO4:POKED+R*H+C  
,2:FORA=1TO100:NEXTA  
840 POKED+R*H+C,6  
850 FORA=1TO100:NEXTA,B:POKES+R*H+C,F:PO  
KED+R*H+C,F1:GOSUB880  
860 IFT>=24THEN250  
870 R=24:C=20:NEXTT:IFM>0THEN250
```



```
875 GOTO250
880 FORN=A(T)+1TOA(T+1)-1:POKES+(T+9)*20
+N,32:POKED+(T+9)*20+N,3:NEXT
890 RETURN
1000 FORN=0TOA(T)
1010 POKES+(T+9)/2*H+N,0:POKED+(T+9)/2*H
+N,5
1020 NEXTN
1030 FORN=A(T+1)TO39
1040 POKES+(T+9)/2*H+N,0:POKED+(T+9)/2*H
+N,5
1050 NEXTN
1060 RETURN
```

Variables in this Program Include:

P = A constant used to control the shape of the banks. P is proportional to the period of the sine wave that makes up the banks.

T = A variable that stores the current row of the gate

S = The first position on the screen

H = The number of characters/positions per row

D = The first position in color memory

A = Variable used for time delay when boat flashes, changing colors after collision. Increasing A will make the boat flash longer.

N = Current position of gate going across the river

B = Used to test the keyboard

- R = Row of the boat. Initially, R=22 since the boat is on bottom row.
- C = Number of columns from the left of boat. Initially, C=11.
- M = Number of prisoners
- CN = Variable that controls change in river shape
- Q = Variable used for timing delay
- G = Used in timing delay for sound
- F = Stores code of character at old position of boat
- F1 = Stores color of character at old position of boat

How the Program Works

5 This is a constant that is used to set the curviness of the river boat. P is proportional to the period of the size wave that makes up the river bank. Try different values of P, such P=1 and P=10 to see different bank shapes. See what is your best score for different Ps.

10 Clears the screen

20 Protects the part of the memory that we will use later to create the monsters and the other special graphic characters

30 Copies the Commodore 64's character memory to a place where we can change it later. The last part of this line sets up some of the variables that will be used later in the program.

40 Reads the information contained in the data statements that reconfigure the character set

50 Contains the information needed to change

60 the character set

70 Tells the Commodore 64 to use our character set

80 Sets aside enough memory to accommodate our A array

90 Sets up the way the river will look

100 Defines the left-hand side of the bank

110 Defines the right-hand side of the bank

120 Says to go on to the next line

130 Starts the loop that will determine the banks. This line should not be changed.

140 Jumps to line 1000 and returns when finished

200 Takes the next value or the next row of bank

210 The next section is the beginning of the main loop.

220 Initializes a few variables

230 Puts the hero's ship on the bottom of the screen

240 How many ships we will use to try to get through the river

250 This is the main loop that controls the row of the black barrier that crosses the river.

253 Increases a constant that is used to determine where the bank is drawn. (Change the constant if you like.)

254-255 Recalculate the position of the bank

256 Blanks the hero's ship from the screen

257 Does the same as line 255

258 Does the same as line 140

260 Controls the gate itself instead of its row

270 Makes the hero's ship disappear from the screen to get him ready to advance to his next position

280 This is the keyboard test. If a key is pressed, then its code appears in memory location 197.

290 Tests to see if the key pressed was a "D."

300 Tests to see if the key pressed was an "S."

310 If no key was pressed, then advance both hero and wall.

320 This section is for going right.

330 Adds one to "C" and then tests to see if it is too close to twenty. If it is, then goes left.

350 Tests to see if the hero is going to run into something; if he is, then goes to 7000.

360 Advances both hero and wall

370 This section is for going left.

380 Subtracts one from "C" and then tests to see if it is too close to two; if it is, then goes right.

400 Tests to see if the hero is going to run

410 Advances both the hero and the wall

430 Subtracts one from the hero's current row

440 Tests the hero's current position to make sure that he hasn't run into anything

450 Puts the hero on the screen in his new position

456 This is a delay to slow down the game. You can delete this line if you want the game to go much faster. If you think the game is too fast, then increase this value.

458-459 This is the sound routine that sounds as the wall advances (should not be changed).

460 Puts the new section of wall on the screen

465-470 Test to see if the hero has reached the top of the screen; if he has, then the wall will disappear and music will play.

480 Does the same with the next section of wall

490 Clears out the old wall

500 Does the same with a new wall

510 Sound! The first part turns on the volume.

The second part plays a tone on the highest generator. The next part is a delay to hear the tone. The tone is now higher and then another delay. The tone is then changed again.

520 This is another delay, but this one is going to last longer. But, again, the tone generator is turned off after the tone.

530 Prints "FREEDOM!!!" at the top of the screen, signifying that the hero has escaped

540 Resets the row and column where the hero will start again

550 Adds one to the number that have escaped

560 Prints "ESCAPED," then the number

570 Subtracts one from the number of attempts left

580 Prints "# Left," then the number left

590 Tests to see if you have no more attempts left. If not, then GOTO the ending message.

600 If the "T" loop is already finished, then restarts that loop.

610 Takes the next row (for the wall)

620 Asks if you'd like to play again

710 This section tests the keyboard again. If no key is pressed, then the program will wait until one is pressed.

720 If the "Y" key was pressed, then the program will shift the bank by one position. This will also reset the score.

730 Ends the program

740 Turns on the volume to begin another sound routine

745-746 The first part of this line starts the middle tone generator of the SID to a medium pitch. The second part is a time delay to hear the note. The third part makes the tone lower in pitch. Then there is another time delay.

760 Turns off the middle tone generator and the volume

770 Prints the captured message on the screen, and then waits a few seconds

780 Prints "ESCAPED" and the current score

790 Subtracts one from the number of attempts left

800 Prints "# LEFT" and the number of attempts left

810 Tests to see if the number of attempts is zero

820 Tests to see if the hero was run aground. If he was, then one will be subtracted from his row. If it is, then the ending message is printed with the score.

825 Remembers what was at the position where the boat crashed. After the boat has finished flashing, whatever was there will reappear.

830 Flashes the boat on the screen. First, it flashes red and waits.

840 Changes the hero's color back to blue

850 Waits a few seconds before going back to red. After the flashing, the characters that were remembered are replaced on the screen.

860 If the wall has already reached the last row, then restarts the loop

870-875 Reset the hero's row and column, then goes back and either takes the next 'T' or restarts the 'T' loop. Resets the 'T' loop

880 Sets up a loop to clear away the previous wall. (Used many times in the program and shouldn't be changed)

890 Returns from wherever the program was before it came

1000 Starts the loop for the banks to be placed on the screen in a different place.

1010 Places the left side according to the loop above

1020 Takes the next piece

1030 Now the right side

1040 Places the right side on the screen

1050 Takes the next piece

1060 Goes back to where you came from

How You Can Change the Program

1. Allow multiple gates to make it harder for the prisoners to get through.
2. Allow the prisoners' boats to stop temporarily. For example, let them stop on islands. Give extra points if the prisoners reach the islands.
3. Make rocks randomly appear in the water to make it harder for the boats to get through.

Rockfall!

Things are bad, bad, bad. In fact, they're worse. The Wizard's creatures are attacking, and rocks are crashing down as Erin pulls the vine. How long can Jacque and Erin survive?

In this game, Jacque is on the right and Erin on the left. The Wizard's creatures attack from the sides. The following controls are used:

S moves Jacque and Erin left

D moves Jacque and Erin right

< fires the Positron Blaster left

> fires the blaster right

Space bar fires the blaster up

If a rock hits Jacque or Erin, the game is over. Likewise, if a creature gets to either of them, the game ends.

The rocks initially start falling from near the middle of the screen. When the rocks reach the row just above the people, the rocks will disappear and a new row will appear if the game isn't over.

This new row will start one row closer to the heroes. Now you'll get one more point per rock disintegrated. The rocks will keep starting off one row closer until row 15, and you'll get 15 points per rock or creature destroyed at this time.

Creatures can appear, starting from two spaces to the right or left of Jacque. As soon as

you disintegrate a creature, another will appear at a different location and will start moving toward Jacque and Erin. You score more points the closer they get before you blast them. Your score will be continually updated on the screen.

Game Program Listing

```

5 DIMA(40)
10 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
20 PRINT"[SC]";
30 W=920:S=1024:V=40:C=55296:NR=30:SC=0:G=39:SR=12:R=12:F=1:Z=20
31 POKE52,48:POKE56,48
40 FORT=0TO511:POKE12288+T,PEEK(53248+T):NEXT:FORX=1TO12:FORT=1TO9
41 READB(T):NEXTT:FORN=2TO9
50 POKE12288+B(1)*8+N-2,B(N):NEXTN,X
51 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1:POKE53272,(PEEK(53272)AND240)+12
60 PRINT"[BK][CS]";:POKE53281,7:POKE53280,7
70 DATA27,24,25,255,61,60,60,39,112
80 DATA28,24,25,255,188,60,60,66,128
90 DATA29,24,25,255,188,60,60,39,32
100 DATA30,24,152,255,61,60,60,39,112
110 DATA31,24,56,124,117,115,117,36,66
120 DATA34,24,56,125,115,112,124,36,24
130 DATA35,24,28,55,87,199,55,36,82
140 DATA36,24,28,190,206,30,62,36,24
150 DATA37,0,0,0,255,255,0,0,0
160 DATA38,255,255,126,126,60,60,60,24
170 DATA39,24,24,24,24,24,24,24,24

```

```
180 DATA40,255,255,255,255,255,255,255,2
55
190 FORJ=1TO10:FORT=0TO21:POKES+J*V+T,40
:POKEC+J*V+T,0:NEXTTT,J:X$="SCORE="
195 PRINT" [BK] [HM] ";X$
200 FORI=1TONR:A(I)=38:NEXT:FORI=1TO8:A(
INT(NR*RND(1)+.5))=32:NEXT:R=SR
201 POKES+W+Z,27:POKEC+W+Z,6
210 POKES+W+Z-1,29:POKEC+W+Z-1,2
220 N=PEEK(197)
230 IFN=60THEN730
240 IFN=13THEN290
250 IFN=18THEN360
260 IFN=47THEN940
270 IFN=44THEN1050
280 GOTO430
290 POKES+W+Z,32:POKEC+W+Z,7
300 POKES+W+Z-1,32:POKEC+W+Z-1,7
310 Z=Z-1
320 IFZ<6THEN380
330 POKES+W+Z,28:POKEC+W+Z-1,2
340 POKES+W+Z-1,30:POKEC+W+Z,6
350 GOTO430
360 POKES+W+Z,32:POKEC+W+Z,7
370 POKES+W+Z-1,32:POKEC+W+Z-1,7
380 Z=Z+1
390 IFZ>34THEN310
400 POKES+W+Z,28:POKEC+W+Z,6
410 POKES+W+Z-1,30:POKEC+W+Z-1,2
420 I=0:GOTO430
430 I=0
440 T=0
450 R=R+1
460 IFR<23THEN550
470 N=PEEK(S+22*V+Z)
```



```
480 IFN<>32THEN880
490 N=PEEK(S+22*V+Z-1)
500 IFN<>32THEN880
510 IFR<19ORR=19THEN530
520 SR=SR+1:SR=SR+1:IFSR>19ORSR=19THENSR
=12
530 FORT=1TONR:POKES+R*V+T-V+4,32:NEXT
540 GOTO200
550 FORY=1TO20:NEXT
560 POKE54296,15:POKE54276,65:POKE54277,
9:POKE54273,R+5:POKE54272,R*5
561 POKE54275,8
570 I=0:FORT=1TO30
580 I=I+1
590 POKES+R*V+4+I,A(T):POKEC+R*V+4+I,A(T
)
600 POKES+R*V+4+I-V,32
610 NEXTT:POKE54296,0:POKE54276,64
620 FF=1
630 POKES+W+F,32:POKEC+W+F,7
640 R1=INT(RND(1)*8):F=F+FF+R1
650 IFF>=Z-2THEN850
660 POKES+W+F,31:POKEC+W+F,0
670 GG=1
680 POKES+W+G,32:POKEC+W+G,7
690 R2=INT(RND(1)*8):G=G-GG-R2
700 IFG<=ZTHEN870
710 POKES+W+G,36:POKEC+W+G,5
720 GOTO220
730 FORLK=22TORSTEP-1
740 POKES+LK*V+Z,39:POKEC+LK*V+Z,2
750 POKES+LK*V+Z,32:POKEC+LK*V+Z,7
760 NEXTLK
770 IFA(Z-4)=32THENGOTO430
780 GOSUB1150:A(Z-4)=32
```

```
790 SC=SC+R:PRINT" [HM] [CR] [CR] [CR] [CR] [C
R] [CR] "SC
800 GOTO430
820 SC=SC+Q:PRINT" [HM] [CR] [CR] [CR] [CR] [C
R] [CR] "SC
830 PRINT" [BK] [HM] ";X$;SC
840 RETURN
850 POKES+W+Z-1,34:POKEC+W+Z-1,5
860 GOTO880
870 POKES+W+Z,31:POKEC+W+Z,6
880 IFSC>HSTHENHS=SC
890 X$="TRY AGAIN
?" :PRINT" [BK] [HM] ";X$;" [HM] [CD] [CD] [CD] [
CD] [CD] [CD] [CD] [CD] [CD] [CD] [CD] [CD] [
CD] SCORE: ";SC:PRINT"HIGH: ";HS
891 RESTORE:FORN=1TO500:NEXT
900 N=PEEK(197):IFN=64THEN900
910 IFN=25THEN10
920 IFN<>39THEN900
930 END
940 SH=Z-2
950 FORTH=SHTO2STEP-1
960 POKES+W+TH,37:POKEC+W+TH,2
970 POKES+W+TH,32:POKEC+W+TH,7
980 IFTH=FTHENQ=ABS(19-(Z-F)):GOSUB1150:
GOTO1010
990 NEXTTH
1000 GOTO430
1010 F=1
1020 POKES+W+OH,32:POKEC+W+OH,7
1030 GOSUB820
1040 GOTO430
1050 FOROH=Z+1TO39
1060 POKES+W+OH,37:POKEC+W+OH,0
1070 POKES+W+OH,32:POKEC+W+OH,7
```

```
1080 TH=OH
1090 IFOH=GTHENQ=ABS(19-(G-Z)):GOSUB1150
:GOTO1120
1100 NEXTOH
1110 GOTO430
1120 G=39
1130 POKES+W+B,32:POKEC+W+B,7
1140 GOTO1030
1150 POKE54296,15:POKE54276,129:POKE5427
7,128:POKE54278,128:POKE54272,128
1151 POKE54273,8
1152 FORT=1TO150:NEXT
1153 FORT=15TO0STEP-.08:POKE54296,INT(T)
:NEXT
1160 POKE54296,0:POKE54276,128
1170 RETURN
```

Variables Used in the Program Include:

- W = The row where the hero and the girl are printed
- S = The first position on the screen
- V = The number of characters/positions per row
- C = The first position in color memory
- NR = The initial number of rocks
- SC = The score, initially set to 0
- G = The variable position of the right monster
- SR = The position of the hero initially
- R = The current position of the rocks
- F = The position of the left monster
- Z = The current position of the hero
- T = A variable defined for convenience

- X = Another variable defined for convenience
- N = A variable used to test the keyboard
- J = A variable defined for convenience
- x\$ = A message that is printed on the screen ("Try Again")
- I = A variable defined for convenience
- Y = A variable used for storing a delay
- FF = The step value of the left monster.
Contains the number of columns the left monster moves
- GG = The step value of the right monster.
Contains the number of columns the right monster moves

How the Program Works

- 10 Tells the Commodore 64 to use our own character set
- 20 Clears the screen and any variables that may be in memory
- 30 Sets up some of the variables that may be used later or may remain constant
- 31 Reserves memory for new character set
- 40 Copies the Commodore 64's character set from its permanent memory to a place where we can change it. The last part of this line starts to read our characters from the data statements.
- 50-51 Continue reading the data
- 60 Clears the screen and makes the screen yellow
- 70-180 Define what the characters will look like. The first number in each line tells which character it

is replacing. (Change this to change the way the hero looks.)

190-195 Put the ceiling of the cave on the screen, and then prints the word "SCORE" in the upper left-hand corner of the screen

200-201 Set up the rocks that are going to fall, as if there were no blank spaces available. The last part of this line will print the hero on the screen.

210 Prints the girl on the screen next to the hero

220-270 Test the keyboard to see if the player is pressing a key. If a key is being pressed, then the program will go to a certain section.

280 Tells the program to go to the section that advances the monsters and the rocks, then goes to the section that tests the keyboard

290 Clears the hero from the screen and gets the hero ready to move left

300 This is the same as line 290 except it affects the girl instead

310 Subtracts one from the hero's position

320 Tests the hero's position to make sure that he isn't going too far to the left

330 Places the hero back onto the screen in a new position and a new stance

340 This is the same as line 330, but affects the girl.

350 Does the same as line 280

360 Does the same as line 290 except that the hero is going to move right

370 Does the same as line 360, but for the girl rather than the hero

380 Adds one to the hero's position

390 Makes sure that the hero isn't too far to the right

400 Does the same as line 330

410 Does the same as line 400, but affects the girl

420 Resets the variable I and then does the same as line 280

430 Resets the variable I

440 Resets the variable T

450 Adds one to the current row where the rocks are located

460 Tests to see that the rocks are still away from the hero and the girl

470 If the rocks are on the row just above the hero, this tests to see what is above the hero.

480 Tests what is above the hero to see if it is a rock or not

490 Does the same as line 470, but affects the girl

500 Does the same as line 480, but affects the girl

510 If the variable "SR" has not yet reached 9, then continue; if it has, then skip the next line.

520 Adds one to "SR"

530 Clears the old rocks

540 Goes back to line 200, which is where the rocks begin to fall

550 This is a timing loop that will slow the game down a little. If you want to speed up the game, you can decrease the number that is in the "Y" loop.

560 This is the beginning of a sound routine. The first part turns the volume up. The second part

turns the highest tone generator to a level that will get higher as the rocks get lower.

570 Starts a loop from 1 to 10

580 Adds one to the variable I

590 Puts a new rock onto the screen

600 Clears the previous rock from the screen

610 Takes the next value of the loop from 570

620 Resets the variable "FF" to one. This variable is how fast the creature on the left is allowed to approach the hero and the girl. Increase this value to make the game harder.

630 Clears the previous position of the left creature

640 Adds the variable "FF" to the left creature's current position "F"

650 Tests to see if the left creature is close enough to the hero or the girl to end the game

660 Places the left creature on the screen in his new position

670 Resets the variable "GG," which is how fast the right creature is allowed to approach the hero

680 Clears the previous position of the right creature

690 Subtracts the variable "GG" from the right creature's current position "G"

700 Tests to see if the right creature has approached the hero

710 Places the creature on the screen in his new position

720 Returns the program from wherever it came from

730 Where the hero's missile is controlled. The

loop counts from just above the hero to the row in which the rocks are located.

740 Places the missile on the screen

750 Clears the previous missile from the screen

760 Tells to advance the missile

770 Tests to see if the location above the hero contained a rock. If it was just a space, then go back and let the rocks and the creatures advance.

780 If there was a rock, then it will now become a space. The "GOSUB" goes to the section that makes an explosion and then comes back.

790 Adds the current row of the rocks to the current score. It is then printed on the screen just after the word "SCORE."

800 Just line line 280

820 Accessed when a creature is shot. This line adds one to the score and prints it just after the word "SCORE," just as before.

830 This will reprint the "SCORE" message if it was messed up along the way.

840 Returns the program to wherever it came from

850 Places the left creature at its last position before it attacks the hero

860 Tells the program to skip the next line

870 Places the right creature at its last position before it attacks the hero

880 Tests to see if the player's current score is higher than the current high score. If it is, then make the current score the new high score.

890-891 Print the "Try Again?" message. Then prints both the current score and the high score. The

program will then wait about two seconds.

900 Tests the keyboard to see if a key is being pressed. If not, then wait until one is.

910 If "Y" was pressed, then GOTO 2 to start a new game without clearing the high score. Remember that the high score will be cleared when the program is started over by running.

920 Tests to see if "N" was being pressed. If not, then go back to line 90. If "N" was pressed, then END.

930 Ends the program

940 Fires to the left. The variable "SH" is left of the hero. The hero will shoot to the left, and the missile will continue until it hits a creature.

950 Loop beginning to the left of the hero

960 Places the missile on the screen at the new position

970 Clears the previous missile position

980 Tests to see if the missile has reached the left creature. If it has, then GOTO the explosion section, then to 101, which will restart the left creature.

990 Takes the next missile position

1000 Does the same as line 280

1010 Resets the left monster's initial position

1020 Clears the old position of the monster

1030 Goes to the section that prints the score

1040 Repeats actions of line 280

1050 Fires a missile to the right. Sets up the loop that will fire at the right creature

1060 Places the missile at a new position

1070 Clears the missile from the screen

1080 Sets the variable "TH" equal to "OH"

1090 Tests to see if the missile has reached the creature. If it has, then GOTO explosion section, reset the creatures position, and GOTO line 112.

1100 If it hasn't, then take the next position.

1110 Does the same as line 280

1120 Resets the right creature's position

1130 Clears the old right creature from the screen

1140 Prints score and then repeats actions of line 280

1150-1152 The explosion section. The first part sets the pitch of the noise generator. This can be changed. The second part turns up the volume. The next part is a time delay of less than a second.

1153-1160 Start a counter that counts backwards. The volume is then set according to this counter. As the last value is reached, the noise generator is turned off.

1170 Returns the program to wherever it came from

How You Can Change the Program

1. Make the creatures appear and disappear so that it is harder to hit them.
2. Allow some colored rocks or Warbirds to appear, and give more points for shooting them.

SPECIAL OFFER— COLLECTOR'S ITEMS

Now you can order large, deluxe reprints of some of the most exciting scenes in *Timelost*[™]. Each print is 10 x 12 inches and reproduced on high-quality paper. Every copy is individually signed by the artist and author, Kris Andrews. Each is numbered to further guarantee its authenticity. As the popularity of *Timelost* increases, your prints will grow in value. Only a limited number of prints will be made. Order now before the supply runs out. These prints are real collector's items.

"Attack of the Warbirds" and "The Wizard Watches" are available for only \$11.95. For \$19.95, you will also receive "Caverns of Carnage" and "Jacque." Collect the complete set!

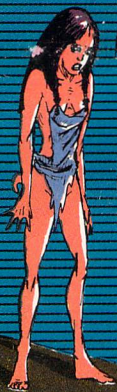
Mail your order today to Computext, Inc., P.O. Box 50942, Indianapolis, IN 46250. Allow 6-8 weeks for delivery.

NOTE: Please do not send requests to Que Corporation.

Timelost is the first book in a new unique series of integrated comic adventures and computer action games.

A teenage computer whiz and his young sister are swept through Time Holes and meet attacking Warbirds of the past, a mysterious man from the future, and a sinister Time Wizard.

Each lost-in-time adventure is combined with an arcade-type computer action game. Explanations help new users to learn programming and suggested variations challenge young users to create more game fun.



0

0-88022-074-0