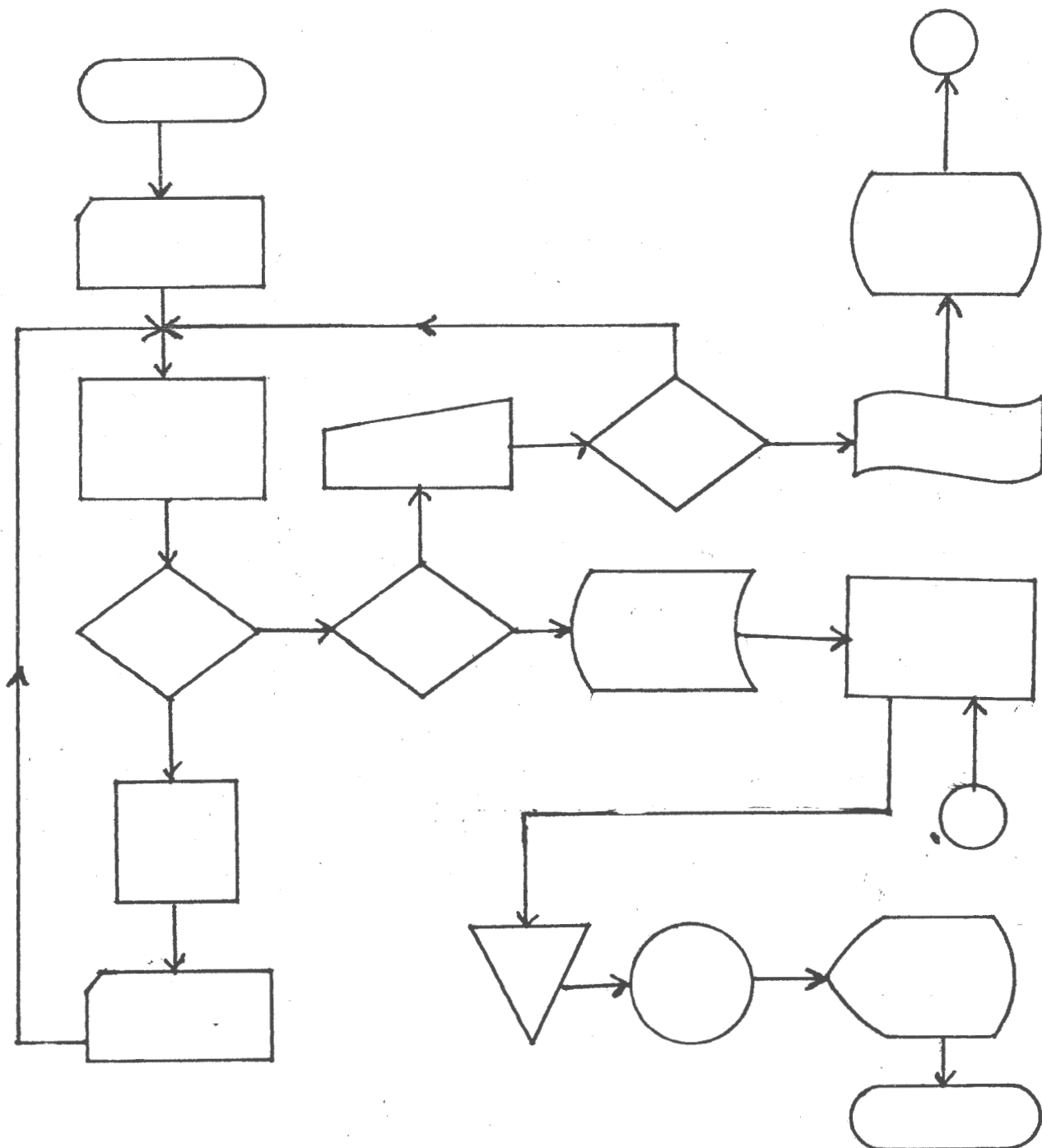


# OS ITEMS

Volume II No. 5

June 1980



AK

TABLE OF CONTENTS

Editor's letter	1
Vectors	2
Add an RS-232 Port to Your CIP	4
Dodge 'em	9
Source Update	10
Type Test	11
New Instructions for The 6502	12
Gopher	13
Useful Pokes/Peeks	14
Submarine	15
Dumb Terminal	17
Stupid Trek	18

Meets the first thursday  
of every month At  
ARISTO/POLKS  
314 Fifth Avenue  
New York, N.Y. 10001  
212/279-9034

Dave Gillett.....President  
Thomas Chens.....Secretary  
Hank Sacchi.....Treasurer

OSItems Editorial Staff

Thomas Chens Sol Lederman  
Mike Bassman Terry Terrance  
Larry Thaler

Editor this month...  
Larry Thaler

CLUB NEWS

Notes from May meeting:

Among other things, we discussed how  
to increase our membership. We  
decided to make up a mailing list, and  
to collect dues so we can send out the  
mail on a regular basis.  
Dues for next year are:  
\$10.00 for standard membership  
\$ 5.00 for student membership

PLEASE NOTE:

All the material in this  
magazine is the sole property  
of the writer. This includes  
all programs, articles and  
other material. They  
may not be copied, sold, or  
distributed without written  
consent from their owners.

## Editor's letter

This month's O.S.I. tems is loaded with great programs, useful subroutines, interesting mods, and funny articles, however, it is lacking one essential ingredient. When I started putting this issue together, I noticed that every article in it had to do with the Challenger 1P. There are no articles for any of the other systems, because NO ONE WROTE ANY !!!!! What the hell is the matter with you???? I keep seeing the same five or six people contributing to every issue. It's not because we like the ClP any better, it's just that the people who own other systems (C2,C4, etc.) never write anything. We ought to change the title to OSI ClP 'tems (well, that's getting a little carried away).

Seriously, why don't you guys write? C'mon, we write all the time (not that we're always right).

Let me tell you what the problem is. The problem is that the same four or five of us clowns are the only ones who really care. I mean, the three or four of us who really care enough to write, all own ClP's.

Now really, it's not that I'm so great or anything, I've only had my Cl for six months, and I can right nifty articles like this one..... Why can't you???? There are really only two or three of us who really do work on our systems and only one or two who have the time to actually write down what they're doing. So come on! what are you reading this stupid magazine for??? There's some fool right now who's waiting for you to submit your article on how you balanced your great grandmother's checkbook. Get it in!!! The none, or one of us who really care, need YOU.

## Vectors

By Mike Bassman

Vector is a word you may have heard before. It's another computer buzzword thrown around to confuse people. A vector is a place in memory where the computer stores an address to another section of memory. This address is usually the entry point of a common routine that Basic uses. Let me try to clarify this with an example. When you type SAVE, the computer looks at the contents of 220 and 221. (This is in hex, in decimal this would be 544 and 545). What it finds in these two locations are two more numbers, 96 and FF (150 and 255 decimal.). These numbers are the address of the Save routine, stored in Low byte, High byte format. The computer now knows that the Save routine is at FF96 (65430 decimal) and immediately begins execution there to perform your SAVE command. In general, the use for this is that if you want to change what a command does, or make it do something else entirely, you can change the vector so that it points to your own machine language program. Specifically, you can disable the SAVE command for program security (ref: Terry Terrance, OSI-tems, volume 2 number 4), add on to the input routine to give a true backspace (Aardvark C1 Cursor Control), or change the input routine so that it uses keyboard macros (Mike Cohen's Shorthand program). All these and much more can be done by changing vectors. There is a small problem, though. Even though almost every command has a vector, many are in ROM rather than in RAM. What this means is that those commands that have commands in ROM cannot be changed, and therefore cannot be used. What I have below is a list of those vectors that are in RAM, their addresses, and where Basic initially sets them to.

VECTOR.	ADDRESS		INITIAL VALUE	
	HEX	DECIMAL	HEX	DECIMAL
SAVE	220	544	FF96	65430
LOAD	21E	542	FF8B	65419
CTRL-C	21C	540	FF9B	65435
OUTPUT	21A	538	FF69	65385
INPUT	218	536	FFBA	65466
WARMSTART	1	1	A274	41588

Continued \_\_\_\_\_)

Now that we have the basic information, let's try some examples. Say, for example, you want Warmstart not to Warmstart, but to Coldstart. The address of Coldstart is BD11 (48401 decimal). The vector of Warmstart is at 1, so we must poke the low and high bytes of BD11 into 1 and 2 respectively. The low byte of BD11 is 11 (17 decimal) and the high byte is BD (189 decimal). These following commands will change the warmstart vector to point at coldstart.

```
POKE 1,17
POKE 2,189
```

Now try a Warmstart and see if it works.

Now let's try something more involved. I never did like the cursor that we get (the underline). I am going to write up a small machine language routine to get rid of it. To do this, though, we are going to have to change the Input vector. The Input routine is what Basic uses to get a character from the keyboard. What the program does is to store the tank character where the cursor is. The cursor location is stored in location 200 (512 decimal). The program then returns to where Basic normally continues. I will put the program in the free space in page 2, starting at 222 (546 decimal). Thus the Input vector must also be set to 222. Following is the program.

ADDRESS		MACHINE CODE	MNEMONICS	OPERANDS	COMMENTS
HEX	DECIMAL				
222	546	48	PHA		Store Accumulator.
223	547	8A	TXA		Put X in A
224	548	48	PHA		And store.
225	549	AE 00 02	LDX \$200		Get cursor position.
228	552	A9 FA	LDA #\$FA		Get tank character.
22A	554	9D 00 D3	STA \$D300,X		Store on screen indexed by X.
22D	557	68	PLA		Get former X from stack
22E	558	AA	TAX		and put in X.
22F	559	A9 22	LDA #\$22		Reset
231	561	8D 18 02	STA \$218		low byte of Input vector.
234	564	A9 02	LDA #\$2		Reset high byte
236	566	8D 19 02	STA \$219		of Input vector.
239	569	68	PLA		Retrieve former A from stack.
23A	570	4C BA FF	JMP \$FFBA		Go back to Basic.

Here we have a good example of the things that are possible if we work with vectors. I hope this article has opened new ways to modify Basic. Happy Computing !

## Add An RS-232 Port to Your ClP

by Larry Thaler

Since Peripherals are your computer's link to the outside world, they must play an important role in the functioning of any computer. In the Challenger 1P, we are blessed with many of these peripherals, or attachments. Among these, are the built in keyboard, the video output, and the cassette port. These 'arms' are what allows us to communicate with our computer, and what makes the computer able to communicate with us.

Although O.S.I. included with the Challenger 1P many of these peripherals, it sometimes becomes necessary to add others. If you want to add a printer so you can have hard copy of your programs, or if you'd like your computer to be able to communicate with other computers ( to exchange programs, act as an intellegent terminal, etc.), you'll need a special accessory port with signals known as RS-232. These RS-232 signals are universal; That is, the RS-232 signals coming out of a Challenger 1P are the same as those which come out of an IBM 370. This standardization allows us, the computer users, to link devices which were not specially made for our computers. Thus, a printer made for the TRS-80 can be used with the Challenger in the same way that a modem can be used.

Although O.S.I. went through all the trouble to include the printed circuit layout for an RS-232 port, it is unfortunate that they didn't think it was worth their time or money to put the components in. For this reason, we can take up where

O.S.I. left off, and add the necessary components without changing any of the internal wiring of the Challenger 1P.

This article will describe how to add these parts that O.S.I. left off, adding an RS-232 port to your computer. I originally did it so I could use my computer with a modem, and then be able to use large computer systems from my house. The modification is really simple, will cost about ten dollars in parts, and take two hours to complete. If you have never done anything like this before, get someone to help you, but if you know how to handle a soldering iron, then by all means, go right ahead. Warning- doing this modification voids your warranty, so if you're going to have trouble with your computer, don't do it.

Here's what you'll need:

quantity	description	Radio Shack #	price each	OSI part
1	7417 Integrated Cir Chip	*	*	U68
1	74LS14 I.C. Chip	*	*	U67
6	220 ohm resistors	271-015	2/19¢	R38-48 (even)
6	390 ohm resistors	271-018	2/19¢	R39-49 (odd)
3	10K ohm resistors	271-034	2/19¢	R's 63, 64, 66
1	4700 ohm resistor	271-030	2/19¢	R62
1	1000 ohm resistor	271-023	2/19¢	R72
1	470 ohm resistor	271-019	2/19¢	R65
1	1N914 diode (1N4048)	276-1122	10/99¢	D16
1	2N 4401 NPN Transistor	276-2009	79¢	Q2
1	2N 2907 PNP Transistor	276-2023	79¢	Q1
1	DPDT SWITCH on-none-on	275-614	\$2.19	SW1
2	16 Pin DIP sockets	276-1998	2/89¢	U67, 68

You'll also need:

1½ feet hook up wire	solder	soldering iron
diagonal cutters*	wire strippers	exacto knife
drill w/ ¼" bit.		

\* Radio Shack does not sell these components, they can be purchased from Hobby World 19511 Business Center Dr. Nothridge, Ca. 91324 (800) 423-5387 The cost is about 25¢ for U68 and 1.25 for U67. They can also be ordered through many electronic hobby places.

6

The first thing you have to do is find a place to install the switch (sw1). I put mine on the front lip of the computer, next to the plaque which says 'Challenger 1P'. This is a good place for it since it can be reached easily, and does not interfere with the installation of a '610' expansion board. First, I used a center punch to make an indentation where I wanted the hole. Next, I drilled out the hole using a 1/8" drill bit. Then, I widened the hole with a 1/4" bit. When you do this, be sure to remove the cover of the computer, and do the drilling far away from the 600 board, otherwise, little metal filings will land in your computer, and the next time you turn it on, you'll get a great surprise.

#### Electronic Instructions

1. Install the switch at W10. (see parts layout and schematic). Cut the jumper shorting what OSI calls W10. Connect switch according to schematic diagram. Keep the leads as short as possible. Note: OSI sheet 6 (in your owner's manual) lists 2 W10's, ours is the one on the right side of the page.
2. Insert sockets for U67 and U68 into their respective spaces on the board. Be sure pin 1 (the one with the cut, or hole in it) is facing in the right direction. Solder the leads from the bottom of the board. The plated-through holes will allow the solder to make good contact on the top as well.
3. Install resistors. All the resistors get inserted standing up. Note the way OSI installed their resistors. Solder all the resistors from the bottom of the board only. Be sure none of the leads are touching each other. Cut off the



excess leads from the bottom of the board using the diagonal cutters.

4. Install diode. D16 gets installed lying down, not like the resistors; see the way OSI inserted D1. Be sure you put the diode in the right direction. The part with the line is called the cathode, and goes in the same direction as the arrow points on the schematic.
5. Install transistors. Be careful when soldering, too much heat and they'll burn out. Be sure to install them in the right position, the base is the one in the center, the emitter has an arrow, and the collector doesn't have anything. See the spec. sheets that come with the transistors to find which pins correspond.
6. Insert the IC's into their sockets. Be sure you put the right IC's in their correct sockets. Be sure pin 1 is pointing in the right direction.
7. Check the bottom of the board for uncut resistor/diode/transistor leads. Check for any solder splashes (solder in the wrong places). Be sure there are no extra bits of wire lying around. Check the top of the board for all these too.
8. Install your modem or printer to J3 pin 4 (Rx Data) pin 2 (Tx Data) and pin 1 (ground). If your printer needs hand shaking, you'll have to hook up RTS and CTS also, and modify W3 (another article to come). If your device needs a negative voltage source, then cut the jumper at W10A (the lower one on sheet 6) and connect a negative voltage source to pin 7 (J3). A nine volt battery (+terminal to ground) should work just fine.

Now, you have RS-232 signals to use your modem or printer with. If the mod is not working, check your connections, and the polarity of the transistors and diode, not to mention the IC's! If your computer is not working at all, turn off the power FAST!!!!!! Go through step 7 again, very carefully. If there's still a problem, get help from a friend, sometimes you just miss something that you did.

Our frinds who decide thestandards for computer signals did us a real favor when they invented RS-232. Now, we like all the other computer hobbysts can enjoy the large list of peripherals available, and make the most out of our computers.

Editor's note: For a working dumb terminal program, see the listing by Danny Schwartz on page of this issue.

Dodge 'em

By Salomon Lederman

This program simulates a car race, with your car always remaining in the center of the screen. The keys 1 and 2 will move your car up and down. Your mileage is displayed on the screen. The thing which makes this program special is not that it's a great game, but it illustrates an extremely fast, machine language screen routine. The routine, installed in RAM in line 5000, and executed in line 50, moves cars from the left of the screen to the right. Each time it is called, the routine takes each character located within the race track, and moves it one place to the right. For this reason, it is necessary to Poke the car back onto the middle of the race track each time the subroutine is called.

```
1 REM***DODGE' EM, BY SALOMON LEDERMAN
2 REM***40 WATERSIDE PLAZA, #31C NY NY 10010
3 REM***212-685-9822
5 S=53545:G=.975
7 GOSUB5000
9 FORX=1TO30:PRINT:NEXTX
10 POKE11,34:POKE12,2
11 B=53762:R=9
12 B=B-32
15 K=57088:POKE530,1:POKEK,127
20 FORX=BTOB+30:POKEX,187:POKEX+32*R,187:NEXTX
30 Y=B+14+32*INT(R/2)
40 POKEY,0
41 SC=SC+10:GOSUB1000
42 IFPEEK(Y-1)>>32THEN500
45 IFRND(86)<GTHEN50
47 POKEB+32,1
50 POKEY,32:X=USR(X):POKEY,0
60 IFPEEK(K)=127THEN100
70 IFPEEK(K)=191THEN200
75 GOTO40
```

```

100 IFPEEK(Y-32)=32THENPOKEY,32:Y=Y-32:GOTO40
110 GOTO500
200 IFPEEK(Y+32)=32THENPOKEY,32:Y=Y+32:GOTO40
500 PRINT"YOU CRASHED"
510 FORX=1TO3000:NEXTX:RUN
1000 A$=STR$(SC)
1010 FORX=1TOLEN(A$):POKEX,X,ASC(MID$(A$,X,1))
1020 NEXTX:RETURN
5000 DATA162,255,189,0,210,157,1,210,202,203,247,96
5010 FORX=546TO557:READZ:POKEX,Z:NEXTX:RETURN

```

#### SOURCE UPDATE:

New listing of software and hardware sources somewhat revised

By A. Nonymous

#### HOHO SCIENTIFIC

No Address. No phone number.

Your original source of non working, mal loading software  
all at ridiculous prices and with no documentation

MOTTO: "Call your dealer"

#### ANTFARM TECHNICAL SERVICES

Surprising programs at incredible prices,  
all with 3 column printer listings.

MOTTO: "It pays to advertise"

#### GALACTIC CRUD ASSOCIATES

Discount software. Instead of buying 3 programs at  
\$6.00 each, ours come in 1 package for \$29.95

ADDRESS: Cell Block X, Sing Sing, Ossining NY

#### HANKY PANKY HARDWARE MODS

Astounding things done to your computer done by our staff  
of professional clowns and servicemen.

MOTTO: "It'll be ready tomorrow"

#### NO TICKEE NO SHIRTEE SOFTWARE

CO/ Lee's Laundry, Chinatown, NY

Real fancy software that costs more than your computer did.

MOTTO: "You BRK, you buy"

## Type Test

By Salomon Lederman

This program is a typing test, designed to estimate your speed and accuracy in typing. It uses a machine language subroutine which, when called, scans the keyboard to see if any keys are pushed, and returns the ASCII value of any character that is. If there is no key pushed, this routine jumps back to basic, and continues the program. This feature allows this program to keep track of the time while it checks your typing. Although this routine is very similar to the one in ROM, it has been changed at location 6260 (dec.) so that it can jump back to basic even though no key is pushed.

The program itself, will print a sentence on the screen, give a countdown, and tell you to start typing. If you make a mistake, the program will wait until you push the right key. When you're done typing the sentence, your score will be printed out. The program will also list any errors that you made. To change the sentence being used, change lines 10, 11, and 12 to put the sentence you want into A\$.

```
1 REM***TYPETEST***BY SALOMON LEDERMAN, 40 WATERSIDE PLZ #31-C NY NY
2 REM***ZIP:10010 TEL:212-685-9822
4 K=531:V=1/84:POKE15,255:DIME(255)
5 FORX=1TO30:PRINT:NEXTX:NT=3
10 A$="NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID"
11 A$=A$+" OF THEIR COUNTRY."
12 A$=A$+" THE QUICK SLY FOX JUMPED OVER THE LAZY BROWN DOG."
19 PRINTA$
20 FORX=64768TO64974:I=PEEK(X):POKEX-53624,I:NEXTX:POKE6260,65
25 FORX=1TO10:PRINT:NEXTX
30 FORJ=5TO0STEP-1:FORX=1TO500:NEXTX:PRINTJ;:NEXTJ
35 PRINT:PRINT"GO.....":PRINT:PRINT
40 PRINT:PRINT:POKE11,0:POKE12,24:FORJ=1TOLEN(A$)
```

```

55 FL=0
60 N=USR(X):IFPEEK(K)=2THENC=C+V:GOTO65
80 TE=ASC(MID$(A$,J,1)):IFPEEK(K)=TETHEN260
85 IFFL=1THENC=C+3*V:GOTO60
90 E=E+1:E(TE)=E(TE)+1:FL=1:GOTO60
200 PRINTCHR$(PEEK(K));:NEXTJ:PRINT:PRINT:PRINT:PRINT
210 PRINT"# OF SECONDS=";C:PRINT"# OF ERRORS=";E
225 CH=LEN(A$):W0=INT(.5+CH/5)
230 PRINT"# OF CHARACTERS=";CH:PRINT"# OF WORDS=";W0
250 PRINT:PRINT"THAT GIVES YOU A TYPING SPEED, ERRORS CONSIDERED, OF";
260 AV=60*((W0-E)/C)
270 PRINTAV;"W.P.M."
277 PRINT:PRINT:IFE=0THENPRINT:PRINT:END
280 PRINT"DO YOU WANT A LIST OF YOUR ERRORS";:INPUTB$
290 IFB$<>"Y"ANDB$<>"YES"THENPRINT:PRINT:END
295 PRINT:PRINT:PRINT"KEY ERRORS":PRINT"---"
297 IFE(32)>0THENPRINT"SPACE"TAB(7)E(32):E(32)=0
300 FORX=0TO255:IFE(X)>0THENPRINTCHR$(X)TAB(7)E(X)
310 NEXTX:PRINT:PRINT:END
350 END

```

# NEW INSTRUCTIONS FOR THE 6502

-----  
By Mike Bassman

AGB	Add garbage
BAH	Branch and hang
BLI	Branch and loop infinite
BPG	Branch on program bug
BPO	Branch on power off
CPB	Create program bug
CRN	Convert to roman numerals
DAO	Divide and overflow
ERM	Erase ROM
IAD	Illogical AND
IOR	Illogical OR
MDB	Move and drop bits
MWK	Multiply work
PAS	Print and smear
RBT	Read and break tape
RPM	Read programmer's mind
RSD	Read and scramble data
SVD	Scratch vital data
TPR	Tear paper
WID	Write invalid data
XIO	Execute invalid opcode
XPR	Execute programmer
TBS	Throw away 6502 and buy sixteen bit chip

# Gopher

By Salomon Lederman

This is a real time game to find a gopher who walks around in a 24 X 24 grid. The only clues you are given are provided by a trail which the gopher leaves as he walks around. Your character leaves a trail as well, which you can use to trap the gopher (he has a one in ten chance of breaking through your trail). When you know where the gopher is, shoot him. Save your zappers though, the fewer you use, the better your score.

- Controls:
1. turns your character counter clockwise
  2. turns your character clockwise
  3. moves you foward
  4. fire your zapper

```

1 REM***GOPHER, BY SALOMON LEDERMAN
2 REM***40 WATERSIDE PLAZA, #31C NY NY 10010
3 REM***212-685-9822
4 GOSUB1000
5 FORX=53200TO54300:POKEX,32:NEXTX
6 W=.6
10 FORX=53380TO53402:POKEX,131:POKEX+759,132:NEXTX
20 FORX=53411TO54115STEP32:POKEX,140:POKEX+24,139:NEXTX
30 POKE53379,221:POKE53403,222:POKE54147,220:POKE54171,223
40 G=54116+INT(RND(86)*20)
43 Y=53412+INT(RND(86)*20):POKEY,248:P=0
50 POKEG,187:T=-T:IFT=1THENGOSUB500
52 C=C+1
55 POKEY,248+P
60 R=INT(RND(86)*7)
70 IFPEEK(G-A(R))<>197ANDPEEK(G-A(R))<>32THEN200
80 G=G-A(R):GOTO50
200 IFPEEK(G-A(R))<>161THEN60
210 IFRND(86)>WTHEN60
220 GOTO80
500 IFPEEK(K)=127THEN600
510 IFPEEK(K)=191THEN650
520 IFPEEK(K)=223THEN700
525 IFPEEK(K)=239THEN750
530 RETURN
600 P=P-1:IFP<0THENP=7

```

```

610 GOTO530
650 P=P+1:IFP=8THENP=0
660 GOTO530
700 IFPEEK(Y-A(P))<>32ANDPEEK(Y-A(P))<>151THEN530
710 POKEY,161:Y=Y-A(P):GOTO530
750 IFPEEK(Y-A(P))=187THENPOKEY-A(P),32:GOTO770
760 GOTO530
770 IFPEEK(G)<>32THEN530
775 PRINT"YOU WIN IN";C;"TURNS"
780 RUN
1000 FORX=0TO7:READA(X):NEXTX:DATA32,31,-1,-33,-32,-31,1,33
1010 K=57088:POKE530,1:POKEK,127:T=1
1020 RETURN
9999 GOTO9999

```

### Useful Pokes

According to one Danny Schwartz....

Peek(57088) can be changed to anything near it.

ie: Peek(57078)

According to Larry Thaler...

Once your Aardvark cursor control program is installed, and running, you can change the character used for the cursor. Try Poke602,161 and you'll get a white box instead of the half tone.



Submarine

By Larry Thaler

Submarine is a game designed for one player. Your piece, located on the bottom of the ocean, must destroy the boats and submarines above you. The object is not only to sink as many ships as possible with the twenty torpedoes you are given, but to also get the boats and submarines which are the furthest away, and are moving the fastest. As you improve, the game gets harder.

```

1 REM*****
2 REM
3 REM   SUBMARINE
4 REM
5 REM   C1P 4K
6 REM
7 REM BY LARRY THALER
8 REM
9 REM*****
10 FORX=546T0571:I=PEEK(X+64490):POKE X,I:NEXT:POKE572,96
12 POKE11,34:POKE12,2
15 ST=32:PN=0:T=20:P=54095
25 X=USR(X)
30 PRINT:PRINT:PRINT:PRINT
35 PRINT"   SUBMARINE"
40 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
42 PRINT"WHAT IS YOUR ABILITY?"
43 INPUT"(1-HARD 3-EASY)";AB
45 X=USR(X)
50 PRINT"INSTRUCTIONS":PRINT:PRINT
55 PRINT"1 LEFT":PRINT"2 RIGHT":PRINT"3 FIRE":PRINT"7 QUIT"
60 PRINT:PRINT:PRINT:PRINT
65 FORW=1T03000:NEXT
70 X=USR(X)
110 FORA=53571T053595:POKEA,191:NEXT
115 FORA=53699T053723:POKEA,191:NEXT
120 CA=-1:CB=-1:CC=-1
125 POKE530,1:POKEP,215
130 POKE54084,161:POKE54106,161
135 POKE53512,48:POKE53513,48
190 FORQ=1T0100000
200 R=INT(RND(1)*20)
205 SPE=INT(RND(1)*AB+1)
210 IFR=10ANDLA=0THENLA=53563:SA=SPE:CA=0
220 IFR=12ANDLB=0THENLB=53667:SB=SPE:CB=0
230 IFR=14ANDLC=0THENLC=53819:SC=SPE:CC=0
300 IFLA<>0THENCA=CA+1
305 IFCA=SATHENCA=0:POKELA-1,181:POKELA,182:POKELA+1,32:LA=LA-1
310 IFLA<53539THENPOKELA,32:POKELA+1,32:LA=0
320 IFLB<>0THENCB=CB+1

```

```

325 IFCB=SBTHENCB=0:POKELB-1,32:POKELB,179:POKELB+1,180:LB=LB+1
330 IFLB>53691THENPOKELB,32:POKELB-1,32:LB=0
340 IFLC<>0THENCC=CC+1
345 IFCC=SCTHENCC=0:POKELC+1,32:POKELC,6:POKELC-1,5:LC=LC-1
350 IFLC<53795THENPOKELC,32:POKELC+1,32:LC=0
400 PK=57088:POKEPK,127
410 IFPEEK(PK)=127THENGOSUB500
420 IFPEEK(PK)=191THENGOSUB530
430 IFPEEK(PK)=223ANDLT=0THENGOSUB560
432 IFPEEK(PK)=253THEN9900
440 IFLT<>0THENGOSUB600
450 NEXTQ
499 STOP
500 IFPEEK(P-1)<>32THEN520
510 POKEP,32:POKEP-1,215:P=P-1
520 RETURN
530 IFPEEK(P+1)<>32THEN550
540 POKEP,32:POKEP+1,215:P=P+1
550 RETURN
560 T=T-1:IFT=-1THEN9900:REMEMD
570 LT=P-32:POKELT,149
575 POKE53527,T/10+48:POKE53528,(T/10-INT(T/10))*10+48.5
580 RETURN
600 POKELT,ST:ST=PEEK(LT-32):LT=LT-32
605 IFST=32ORST=191THEN700
610 FORA=1TO5:FORB=191TO194:POKELT-1,B:POKELT+1,B:POKELT,B
615 NEXTB:NEXTA
620 IFLT<53563THENVL=800-SA*100:LA=0
630 IFLT<53691ANDLT>53667THENVL=700-SB*100:LB=0
640 IFLT>53795THENVL=600-SC*100:LC=0
650 PN=PN+VL:POKELT-1,VL/100+48:POKELT,48:POKELT+1,48:FORC=1TO300
655 NEXTC
660 POKELT-1,32:POKELT,32:POKELT+1,32
670 POKE53510,INT(PN/1000)+48:POKE53511,(PN/1000-INT(PN/1000))*10+48.5
675 LT=0:ST=32
680 GOTO720
700 IFLT<53563THENLT=0:ST=32:GOTO720
710 POKELT,149
720 RETURN
9900 POKE530,0
9910 X=USR(X)
9920 PRINT:PRINT:PRINT"YOU GOT"PN"POINTS"
9922 PRINT:PRINT"AT LEVEL ";AB
9925 PRINT:PRINT:PRINT:PRINT
9930 PRINT:PRINT:PRINT"WOULD YOU LIKE"
9932 PRINT
9935 INPUT"TO TRY AGAIN";P$
9940 IFLEFT$(P$,1)<>"Y"THEN9980
9955 LA=0:LB=0:LC=0
9960 IFFPN>4500AND AB>1THENAB=AB-1:PRINT"TRY THIS NEXT ONE AT LEVEL";AB
9965 ST=32:PN=0:T=20:P=54095
9970 FORW=1TO1000:NEXT:GOTO70
9980 PRINT:PRINT"COME AGAIN SOON !"
9990 END

```

Dumb Terminal

By Danny Schwartz

This program allows you to turn your C1P into a dumb terminal for use with time-share systems. To use it, type in

POKE 122,4:POKE1024,0:NEW

before using it, or loading the program. The program works on an even parity system, and has been tested on such systems as CUNY (CALOS, WYLBUR), DEC 20, HP 2000, and PDP11.

```
0 REM BE SURE TO TYPE "POKE122,4:POKE1024,0:NEW" BEFORE LOADING PROGRAM
1 REM *****
2 REM
3 REM DUMB TERMINAL
4 REM
5 REM FOR C1P
6 REM
7 REM BY D. SCHWARTZ
8 REM
9 REM *****
10 DATA 173,0,240,74,144,12,173,1,240,41,127,201,127,240,3,32
20 DATA 45,191
21 FORX=754T0771:READI:POKEX,I:NEXT
30 FORX=772T0953:I=PEEK(64000+X):POKEX,I:NEXT
40 POKE853,156:POKE884,94
50 REM EVEN PARITY
51 DATA 160,9,162,0,24,42,144,1,232,136,208,249,133,17,138,41,1,24,106
60 DATA 106,5,17,32,177,252,76,242,2
70 FORX=954T0981:READI:POKEX,I:NEXT
100 PRINT "READY":POKE11,242:POKE12,2:X=USR(X)
190 END
```

OK

## Stupid Trek

By many

Do you remember the stupid game listed on page 13 of the Clp graphics manual? Well, here are two improved versions of it. The first one is great because it not only lets the enterprise shoot down the little guns, but the guns shoot back as well. You must get all the guns to win. The controls are: 1. moves you foward

2. moves you back

3. fires your lasers

The second listing is a slightly modified version of the first. In this one, the computer plays against itself, and the poor fool sitting at the keyboard does nothing.

These two might not be so great, but they're certainly better than the orignal.

```
1 REM *****
2 REM
3 REM STUPID TREK
4 REM
5 REM PLAY VERSION
6 REM
7 REM      C1P
8 REM
9 REM *****
10 IF F<>ATHEN60
21 FORX=0TO35:PRINT:NEXT
31 FORX=54051TO54071STEP2:POKEX,246:NEXTX
50 K=57088:A=53730:POKE530,1:POKER,127
60 REM
61 POKEA,11:POKEA+1,12:FORX=0TO50:NEXTX
70 IFPEEK(K)=127THEN100
75 IFPEEK(K)=191THEN200
80 IFPEEK(K)=223THEN300
90 IFPEEK(K)=253THEN400
95 .IF INT(RND(1)*5+1)=3 THEN330
100 POKEA,32:POKEA+1,32:A=A-1:GOTO60
200 POKEA,32:POKEA+1,32:A=A+1:GOTO60
300 FORR=A+31TO54071STEP31:POKER,189
310 FORX=0TO10:NEXTX:POKER,32:NEXTR
320 GOTO80
330 R=0
```

```

332 FORZ=54051T054071STEP2
335 IFPEEK(Z)=246THENR=R+1:IFR>INT(RND(1)*6)THEN345
340 NEXTZ:IFR>0THEN60
341 PRINT"YOU WON !!!":GOTO400
345 FORF=Z-31TOASTEP-31
350 POKEF,189:FORX=1TO10:NEXTX:IFPEEK(F-31)=32THENPOKEF,32:NEXTF:GOTO6
390 PRINT"YOU LOST !!!"
400 POKE530,0:INPUT"PLAY AGAIN":A$:IFLEFT$(A$,1)="Y"THEN RUN
410 PRINT"SEE YA ROUND":END

```

OK

```

1 REM*****
2 REM
3 REM STUPID TREK
4 REM
5 REM*****
6 IFF<>ATHEN60
21 FORX=0TO35:PRINT:NEXT
31 FORX=54051T054071STEP2:POKEX,246:NEXTX
50 A=53730
60 REM
61 POKEA,11:POKEA+1,12:FORX=0TO50:NEXTX
62 R=INT(RND(1)*5+1)
63 ONRGOTO300,100,100,330,330
70 GOTO62
80 REM
95 IF INT(RND(1)*5+1)=3 THEN330
200 POKEA,32:POKEA+1,32:A=A+1:GOTO60
300 FORR=A+31T054071STEP31:POKER,189
310 FORX=0TO10:NEXTX:POKER,32:NEXTR
320 GOTO80
330 R=0
332 FORZ=54051T054071STEP2
335 IFPEEK(Z)=246THENR=R+1:IFR>INT(RND(1)*6)THEN345
340 NEXTZ:IFR>0THEN60
341 GOTO400
345 FORF=Z-31TOASTEP-31
350 POKEF,189:FORX=1TO10:NEXTX:IFPEEK(F-31)=32THENPOKEF,32:NEXTF:GOTO6
400 FORT=1TO1200:NEXT:RUN

```

OK