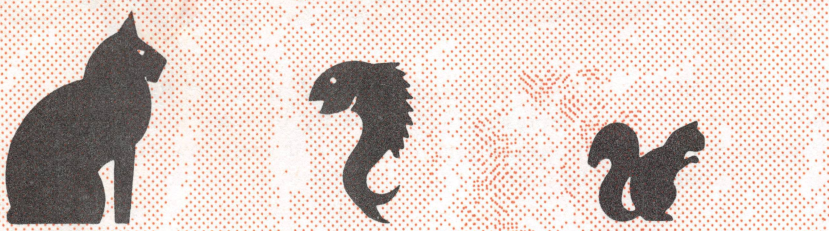


SPOT

BY PHYLLIS COLE, EDITOR

PET photo courtesy of Utter Chaos

Commodore's PET is a factory assembled personal computer based on a 6502 microprocessor. The unit includes a keyboard, cassette tape unit, CRT, some graphics, upper and lower case, and an extended 8K BASIC. The system with 4K of user memory costs $595; the 8K model costs $795. For details, see the last 2 issues of People's Computers.

Until there's some sort of formal PET owners' organization, *People's Computers* will provide space each issue as a forum for PET people. The name of the forum will be SPOT (*S*ociety of *P*et *O*wners and *T*rainers) unless something better comes along. Possible uses for the forum include swapping software and ideas (PET projects? and maybe Teachers' PET?) and complaints (obviously PET Peeves). Perhaps projects that involve hooking the PET to other devices should come under the heading ComPETible Stuff?

TEACHERS' PET

The 'we' of this article refers to a group of computer professionals – including your editor – who have purchased a PET as part of a project aimed at integrating computers into the daily routines at a local school. To date we've concentrated on preparing a wide variety of sample programs that we've shown to both kids and teachers. As of January we'll begin workshops with teachers and junior high age students to teach them how to program in a version of PILOT. The workshops will focus on the development of a body of programs to support the topics that will be studied in the latter half of the school year in the junior high. In addition, some programs for younger kids will be developed; the junior high school will serve as a primary resource for introducing the PET and PILOT to others in the school We plan to include even nursery-school age children in the project – already we have a demonstration program for youngsters that age.

TAPE TIPS

We've found that the most reliable tapes to use on the PET are Maxell, TDK, and Memorex. We've had some problems with the cheapest Radio Shack tapes. We definitely do not recommend Scotch tapes: we confirmed reports that when used as computer tapes they snag–although the problems of snagging may be reduced if very short tapes are used.

In its manual, Commodore recommends using 'Nortronics' Brand tape head cleaner. They recommend 'Nortronics', 'Hand-de-mag' and 'Robins' brands of head demagnetizers. Tape deck head cleaning and demagnetizing needs to be done every 50-100 hours of tape running time or when you have trouble reading tapes reliably.

SOFTWARE AVAILABILITY

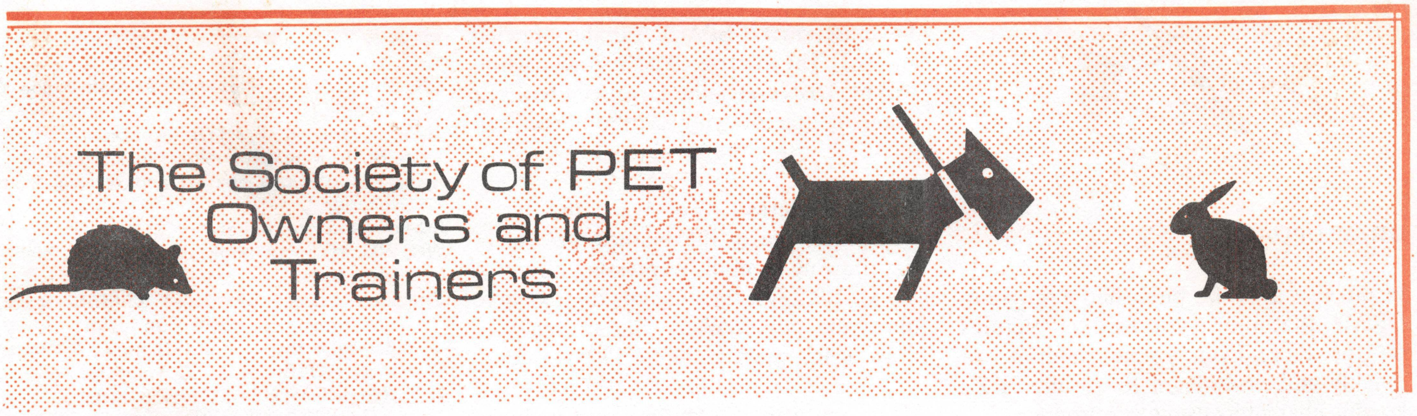
PET programs are beginning to be advertised–let us know which ones you've tried, and whether you recommend that others purchase them. Some of the programs we've been working on are now available. For a description of the programs, a price list, and a licensing agreement send a stamped, self-addressed envelope to Computer Project, Peninsula School, Peninsula Way, Menlo Park, CA 94025.

DIAL-A-PET

We've used a board from The Net Works (see announcement section) to connect our PET over the phone to other computers–and so we got the listings that accompany this article.

The PCNET (Personal Computer Network) Committee is involved in setting up protocols (standards) to enable different kinds of home computers to communicate with one another. We're implementing the protocols on the PET in BASIC. Using Commodore BASIC and The Net Works' board we can barely keep up a 30 character per second speed. However, we expect to be able to implement all the PCNET protocols in BASIC. So far, our PET can automatically dial the phone, and can receive and acknowledge packets of information. (See 'Computer Networks' by Larry Tesler, Volume 6, Number 2). Other folk are implementing them on other home computers, often in assembly language. You'll be hearing more of these efforts in later articles.

To receive the latest draft of the PCNET protocols, send $3.00 to cover replication and mailing costs to Dave Caulkins, 437 Mundel Way, Los Altos, CA 94022.



The Society of PET Owners and Trainers

NOTATION

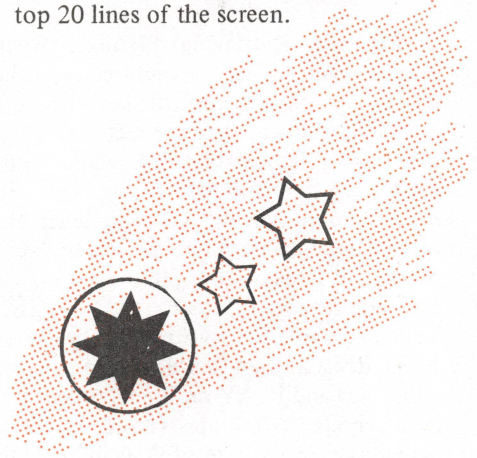
Most printers can't print PET graphics. Also, Commodore BASIC lets you put cursor control characters into PRINT statements, and inside the quote marks displays special *reverse* characters when you do so. And most printers can't cope with printing in reverse (i.e. white on black).

To help make PET listings more readable, we've decided to indicate special stuff inside square brackets, using commas to separate items. Sometimes a number precedes an item to tell how many times it should be repeated. We use HOME, CLR, RVS, OFF and INST as shown on PET keys. Sometimes SPACE is used to show where a space should be typed. A single character indicates that the *graphic* shown as the *shifted* character should be used. Examples are provided in Figure 1.

Finally, you may have noticed colons at the beginning of some program lines. Colons indent lines inside FOR loops to help show the limits of the loop.

STARS

Here's a familiar number-guessing game that we've adapted for the PET. The program randomly selects a number from 1 to 100 for you to guess. You try to guess the number, then the program prints out from 1 to 7 stars (asterisks) depending on how close your guess is. When you guess the number, the screen goes blank then 40 stars are randomly displayed in the top 20 lines of the screen.



Listing says	You type	The screen shows	What happens when program is run
PRINT "[CLR, Q]"	PRINT " CLR key shift-Q "	PRINT " black heart on white white dot on black "	Cursor goes home (upper left) and screen clears, white dot printed in home position.
PRINT"[HOME,3 DOWN,S]"	PRINT " HOME key cursor down 3 times shift-S "	PRINT " black S on white 3 black Qs on white white heart on black "	Cursor goes home, then down 3 times (i.e. to row 4), next prints a white heart.

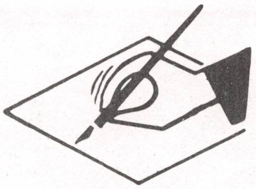
Examples of Notation for Special Characters
Figure 1

100 sets lower case mode.
110 clears the screen; moves the cursor to second row.
120-210 prints out the instructions.
220 selects X, the random number to be guessed.
230-240 gets the guess, G.
250 branches to line 40 if the correct number was guessed.
260 calculates D, the 'distance' that the guess G is from the correct answer, X.
270-290 loops to print from 1 to 7 stars, depending on the value of D.

D:	1	2	4	8	16	32	64
Q:	0	1	2	3	4	5	6
No.*:	7	6	5	4	3	2	1

300 gets next guess.
405 clears screen.
410-430 loops to print 40 stars in randomly chosen locations in the first 20 rows of the screen. In these 20 rows there are 40*20 or 800 locations to choose from. We randomly select Y, an integer from 1 to 800. The command POKE 32768+Y, 42 prints an asterisk (ascii code 42) in the Yth location on the screen.
440-490 moves the cursor home, then to the 22nd row (the second blank line), prints a message about how to restart the game.
500 waits until something is typed from the keyboard.
510-530 if RETURN was pressed, starts the game again; if anything else was typed, ends the game.

```
10 REM STARS
100 POKE 59468,14
110 PRINT "[CLR,DOWN]"
120 PRINT "WELCOME TO MY GALAXY. I'M IN CHARGE"
130 PRINT "OF THE STARS HERE. PLAY MY GAME OF"
140 PRINT "***STARS*** AND GET SOME STARS FOR"
150 PRINT "YOURSELF.": PRINT
160 PRINT "I WILL THINK OF A WHOLE NUMBER FROM"
170 PRINT "1 TO 100. YOU TRY TO GUESS IT. THE"
180 PRINT "MORE STARS I PRINT THE CLOSER YOU ARE."
190 PRINT "IF I PRINT 7 STARS ******* YOU ARE"
200 PRINT "VERY VERY CLOSE!!!"
205 PRINT: PRINT
210 PRINT "OK STARSEEKER, I'M THINKING OF A NUMBER."
220 X=INT(100*RND(1))+1: N=1
230 PRINT
240 INPUT "WHAT'S YOUR GUESS";G
250 IF G=X THEN 400
260 D=ABS(G-X)
270 FOR Q=LOG(D)/LOG(2) TO 6
280 :PRINT "*";
290 NEXT Q
300 N=N+1: GOTO 230
400 REM GOT IT
405 PRINT "[CLR]"
410 FOR Q=1 TO 40
420 :Y=INT(800*RND(1))+1: POKE 32768+Y,42
430 NEXT Q
440 PRINT "[HOME, 21 DOWN]";
450 PRINT "YOU GUESSED MY NUMBER IN";N;"TRIES!!!"
460 PRINT: PRINT "PRESS RETURN TO PLAY AGAIN."
470 PRINT "PRESS ANY OTHER KEY TO STOP."
480 GET A$: IF A$="" GOTO 510
490 IF ASC(A$)=13 GOTO 205
500 PRINT: PRINT "BYE FOR NOW!": PRINT
510 END
```



DRAW UPDATE

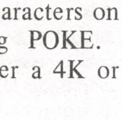
In our last issue we published a program, DRAW, that allows you to draw pictures on the PET. Even young children can quickly and easily learn to use the PET's graphic characters using this program.

Now we've added to the program. You can save on tape and retrieve the pictures that you draw. We've also modified the program to PRINT the characters on the screen, instead of using POKE. The program will run on either a 4K or 8K PET.

The Target. The program treats the screen as a grid of cells, 38 across and 24 down. When it starts, it clears (blanks) the screen and displays a large round dot (the 'drawing symbol') in a cell near the center. That cell is the initial 'target cell'. A white square blinks at you occasionally to let you know where the target cell is.

To draw a picture made of dots, use the program's target-motion keys. They are not the same as the PET's cursor keys. Instead, the digit keys 1 through 9 are used to make the target move one cell in any of eight directions.

7	8	9
4	5	6
1	2	3



Pretend the target cell is on the '5' key. To move it left, press '4'; to move it up and right, press '9'; and so on. Whenever the target moves, it will inscribe the drawing symbol in its new cell.

The Drawing Symbol. When you are tired of dots, press any graphic key. Shifting is only needed for the graphics on the 1-9 key. The graphic character on the key will become the new drawing symbol. It will be inscribed in the target cell.

Press RVS, and the color of the drawing symbol will be reversed. Subsequently chosen drawing symbols are not affected. To erase, make SPACE be the drawing symbol. The reverse SPACE draws white stripes. DEL erases the target cell without changing the drawing symbol. You can 'un-DEL' using the '5' key.

Other Features. To move the target without changing the picture, get rid of the drawing symbol by pressing either of the CRSR keys, then use the digit keys to move the target. When you are ready to draw again, press a graphics key or RVS.

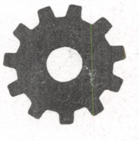
When you want to admire your drawings without the target cell blinking at you periodically, press RETURN. Then, to make it blink again, type any other key; '5' is a good choice.

To move the target to the center of the screen, press HOME. To start a new picture, press CLR (note that you'll have to shift). To stop drawing so you can do something else with your PET, first press RETURN and then press STOP.

Saving and Retrieving Pictures. When you're ready to save a picture, type the left-arrow cursor control key–i.e. the one that points to the tape recorder. First the cursor stops blinking while your picture is measured; during the 30 seconds or so that this takes, little streaks of light may dance about the screen. Next the first 3 lines of your picture are erased and you're asked 'SAVE FILE NAME?' Type in a name for your picture and press RETURN and then you'll be told to press RECORD and PLAY on the recorder. It'll take about a minute (more or less, depending on the size of your picture) to save your picture on tape. When saving is complete the program clears the screen and displays the initial drawing symbol, a dot, in the center of the screen.

Now rewind the tape. To retrieve your picture from tape and print it on the screen, type the up-arrow cursor control key–the one that points to the screen.

Notice the 'save' (cursor left) and 'retrieve' (cursor up) commands involve shifting–hopefully this will help avoid accidental typing of these commands.



Annotations. Here are brief descriptions of major elements of this program.

variables

LX, MX, LY, MY: least and maximum X and Y used.
X, Y, L: target X, Y, screen location.
GR$, R: drawing symbol graphic and reversal.
PG$, PR: previous graphic and reversal from location L.
SC=screen cursor address.
WH, BL, WT, FL: blink timers (white, black, wait, flash).
Character codes: SP=space; SS=shift space; CV=screen/ascii conversion factor; SH=shift bit; US=unshift mask; BY=byte mask; QT$=quote; CR=carriage return; RE=reverse; DE=delete; IR=initial reverse (OFF); XR=exchange reverse (XR–OFF=RVS).

line(s)

5 puts system into graphics mode (as opposed to lower case mode).
6-20 sets up constants, initializes variables; note that the first and last characters in strings H$ and V$ are cursor controls.
30 sets initial drawing symbol to a dot (shift-Q) and clears the screen.
40-60 prints dot in the center of the screen.
100 looks for keystroke.
150 converts keystroke to unshifted ascii.
200 checks that the target is blinked off.
250 RETURN causes a long blink.
300 resets short blink.
400-800 handles number key, graphic key, DEL, RVS and the cursor control keys.
850-875 the LEFT cursor causes a 'save' on tape; UP retrieves from tape.
1000 encodes symbol for display.
1200 resets drawing symbol.
1250 reverses drawing symbol.
1300-1500 displays symbol in target cell.
1700-2275 moves target in direction indicated by number key.
2300 locates new target.
2325 stores previous graphic.
2350 stores previous reverse information.
2400 checks for drawing symbol.
2500-2700 blinks the target if there's no symbol to draw.
3000-3020 blinks the target while you're not doing anything.
4000-4020 reverses the color of the target cell.
4500-4510 handles delete (DEL) key.
5000 initializes variables used in storing the picture on tape.
5010-5130 determines the size of the picture.
5500-5615 saves 3 lines of the picture in an array.
5618-5627 clears first 3 lines of the screen.
5630-5635 requests file name for the picture.
5640-5700 stores picture onto tape. Lines 5675-5677 turn the tape recorder motor on for 3 jiffies (3/60 second) and then off. This code is needed *only* for early PETs, which tend to have trouble reading data files. This technique should be used every 191 characters (1 record) or, as here, more often.
6000-6015 opens file, reads X1 and Y1 coordinates, checks status bit to see if end of file reached.
6020-6040 positions picture in center of screen.
6050-6100 reads and prints picture, closes file.

```
1 REM PET DRAWING PROGRAM
2 REM (c) 1977 PENINSULA SCHOOL
3 REM PERMISSION TO USE, BUT NOT SELL

5 POKE 59468.12
6 SC=32768: SP=32: SS=160: CV=191
7 BY=255: SH=128: US=127: QT$=CHR$(34)
8 LX=1: MX=38: LY=0: MY=24: DIM E$(2)
9 H$="[LEFT]"+CHR$(0)+"[RIGHT]"
10 V$="[UP]"+CHR$(0)+"[DOWN]"
15 N1$="1": N9$="9": CR=13: RE=18: DE=20
20 WH=50: BI=53: WT=5: XR=164: IR=146
30 GR$="[Q]": PRINT "[CLR]":
40 Y=INT((MY+LY)/2): X=INT((MX+LX)/2)
45 R=IR
50 PRINT "[HOME]":LEFT$("[12 DOWN]",Y);SPC(X);
60 GOTO 2300

100 GET C$: IF C$="" GOTO 3000
150 C=ASC(C$) AND US
200 IF FL>=WH THEN GOSUB 4000
250 IF C=CR THEN FL=-1E8: GOTO 100
300 FL=WH-WT
400 IF C$>=N1$ AND C$<=N9$ THEN 1700
450 IF C>=SP GOTO 1000
500 IF C=DE GOTO 4500
600 IF C=RE GOTO 1200
700 IF C$="[DOWN]" OR C$="[RIGHT]" THEN R=IR: GR$=""
750 IF C$="[HOME]" THEN GR$="": GOTO 40
800 IF C$="[CLR]" GOTO 30
850 IF C$="[LEFT]" GOTO 5000
875 IF C$="[UP]" GOTO 6000
900 GOTO 100

1000 GR$=CHR$(C+SH): R=IR
1100 GOTO 1300

1200 IF GR$="" THEN GR$=PG$: R=PR
1250 R=XR-R: PRINT CHR$(R);
1300 PRINT GR$:"[LEFT]";
1500 PG$=GR$: PR=R
1600 GOTO 100

1700 PX=X: PY=Y
1750 X=X+C+1-3*INT((C+2)/3)
1800 IF X<LX THEN X=LX
1900 IF X>MX THEN X=MX
2000 Y=Y+1-INT((C-49)/3)
2100 IF Y<LY THEN Y=LY
2200 IF Y>MY MY
2250 PRINT MID$(H$.X-PX+2.1);
2275 PRINT MID$(V$.Y-PY+2.1);
2300 L=SC+40*Y+X
2325 PG$=CHR$(PEEK(L) OR SH)
2350 PR=IR-(PEEK(L) AND SH)
2400 IF GR$<>"" GOTO 1300
2500 GOSUB 4000
2600 FOR DL=1 TO WT: NEXT DL
2700 GOSUB 4000
2800 GOTO 100

3000 FL=FL+1
3010 IF FL=WH THEN GOSUB 4000
3020 IF FL=BI THEN FL=0: GOSUB 4000
3030 GOTO 100

4000 PR=XR-PR: PRINT CHR$(PR):
4010 PRINT PG$: "[LEFT]":
4020 RETURN

4500 PG$=" ": PR=IR
4510 PRINT "[OFF, SPACE, LEFT]":CHR$(R):
4520 GOTO 100

5000 X0=MX: X1=LX: Y0=MY: Y1=LY: K=SC
5010 FOR Y=LY TO MY
5020 :FOR X=0 TO 39
5030 ::C=PEEK(K): K=K+1
5050 ::IF C=SP GOTO 5100
5060 ::IF X<X0 THEN X0=X
5070 ::IF X>X1 THEN X1=X
5080 ::IF Y<Y0 THEN Y0=Y
5030 ::IF Y>Y1 THEN Y1=Y
5100 :NEXT X
5130 NEXT Y

5500 SY=0
5510 FOR Y=Y0 TO Y1
5512 :K=SC+40*Y+X0
5515 :RV=0: L$=""
5520 :FOR X=X0 TO X1
5530 ::C=(PEEK(K)-SP AND CV)+SS
5535 ::V=C>BY: IF V=RV GOTO 5545
5540 ::RV=V: L$=L$+MID$("[RVS. OFF]".V+2.1)
5545 ::K=K+1: L$=L$+CHR$(SH*RV+C)
5600 :NEXT X
5615 :E$(SY)=L$: SY=SY+1
5618 :IF SY<3 AND Y<Y1 GOTO 5685
5620 :IF Y>Y0+2 GOTO 5660
5625 :PRINT "[HOME. OFF]";
5626 :FOR I=0 TO 2: FOR J=0 TO MX
5627 ::PRINT " ";: NEXT J: PRINT: NEXT I
5630 :INPUT "[HOME]SAVE FILE NAME"; NM$
5635 :PRINT "[HOME]";
5640 :OPEN 1,1,1,NM$
5650 :PRINT#1, Y1+1-Y0
5655 :PRINT#1, X1+1-X0
5660 :FOR I=0 TO SY-1
5665 ::PRINT#1, QT$; E$(I); QT$
5670 ::E$(I)=""
5675 ::POKE 59411,53: T=TI
5676 ::IF TI-T<3 GOTO 5676
5677 ::POKE 59411,61
5678 :NEXT I
5680 :SY=0
5685 NEXT Y
5690 CLOSE 1
5700 GOTO 30

6000 OPEN 1
6005 INPUT#1, Y1
6010 INPUT#1, X1
6015 IF ST GOTO 6100
6020 Y0=INT((MY+LY+1-Y1)/2)+1
6030 X0=INT((MX+LX+1-X1)/2)+1
6040 PRINT MID$("[CLR, 12 DOWN]".1,Y0);
6050 FOR Y=1 TO Y1
6060 :INPUT#1.L$
6070 :PRINT SPC(X0);"[LEFT]";L$;
6080 :IF Y<=MY THEN PRINT
6090 NEXT Y
6100 CLOSE 1
6110 GR$=""
6120 GOTO 40
```

OTHER STUFF

We've made our first use of the 8-bit user port: we can flash a 40-watt lamp on and off under program control. The circuit we used is from *The First Book of Kim,* which is reviewed in this issue. The 115 volt AC circuit must be isolated from the 5 volt digital circuit. We used a $1.50 opto-isolator to do this, and a $1.40 Triac (electronic switch) to operate the lamp.

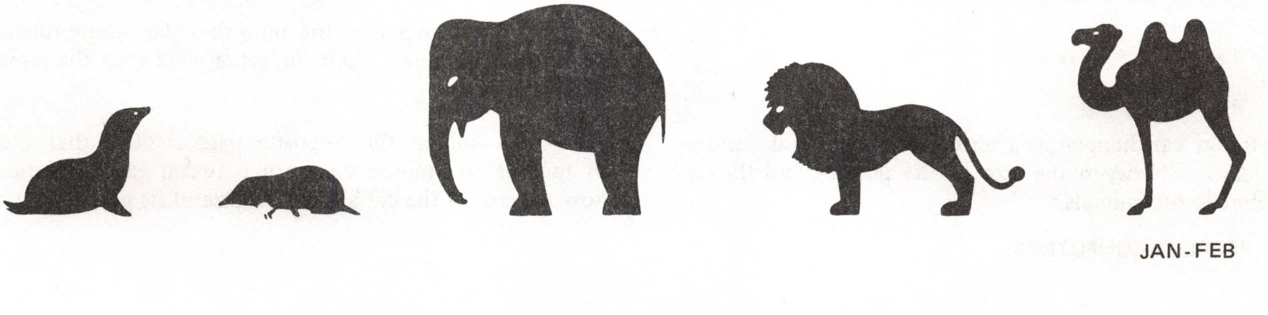
We've also used the PET to assist in analyzing data from a questionnaire. The program is designed so that inexperienced non-programmers can easily enter data, using one DATA statement per questionnaire. For easy reference, the DATA statement line number is the same as the sequential identification number assigned to each questionnaire. Checking the data after it was entered into the computer was facilitated by a program that displayed data for any specified questionnaire in an easy-to-read form.

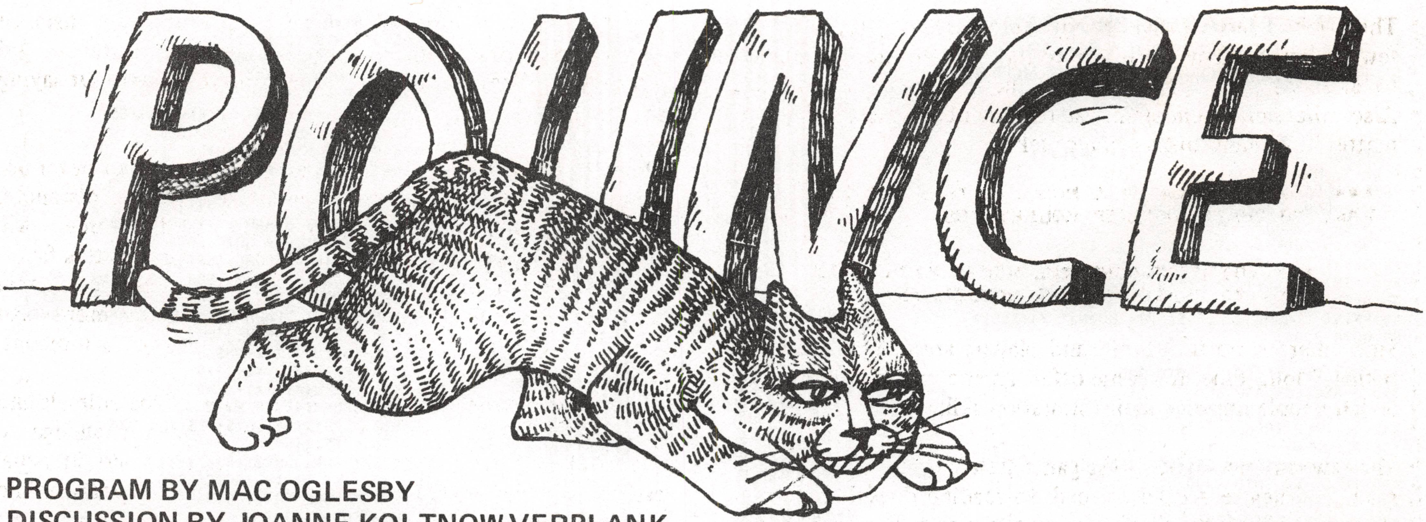
The PET and its associated questionnaire analysis programs have attended various meetings where survey results were discussed. To the delight of all concerned, questions such as 'What happens if we change factor X by amount Y?' and 'What's the median of Z?' were readily answered, and, when appropriate, the results were displayed as a graph. When the questions involved complex calculations whose results were not obvious, the computerized approach saved many hours of analysis and most likely forestalled many hours of discussion, since in a brief time period it was possible to explore many possible alternatives in depth.

A nice feature we've only recently discovered: you can list programs at a readable speed (about 2 lines per second) by holding down the RVS key while doing a listing. For more tips and another PET program, see the letters section of this issue.

We've seen draft versions of an introductory PET manual and a tutorial tape which introduces BASIC. Both are written for the novice. The tape's 10 lessons go into more detail than does the manual. I learned useful new information even with a brief glimpse at the tape—you don't need semi-colons to separate items in a PRINT statement—PRINT "HI, "N$ works fine. The PET User's Manual now being written promises to be the more complete work that many of us await. Commodore isn't even guessing when documentation will be available.

Last but not least, we hear that as of the first week of December Commodore was producing 100 4K PETs a week, and an unknown number of 8K systems.





PROGRAM BY MAC OGLESBY
DISCUSSION BY JOANNE KOLTNOW VERPLANK,
Community Computer Center

In this article we bring together two long-time supporters of People's Computers. *Mac Oglesby is a teacher in Vermont; from his time-sharing terminal in a log cabin he creates games for classroom use. We've published many of Mac's kid-tested games over the last few years.*

Joanne Verplank, the Director of Community Computer Center, has introduced thousands of kids to computers over the last few years. Here she shares with our readers her experiences with one of her favorite games, Mac's POUNCE.

We have to estimate and compare, visually, all the time – while driving, walking, or giving directions, in order to buy raw materials or decide which supermarket line is the shortest. However, visual skill building, which includes estimation and comparison, is often ignored in schools. POUNCE, an amusing chase game by Mac Oglesby, offers a chance for us to practice estimating and comparing short distances. POUNCE can be used by players at several levels of experience. Beginners easily learn to play, yet the game has enough variety to challenge the more advanced.

WHAT HAPPENS?

Players are shown representations of the cat and the mouse.

```
        ## = CAT
        :: = MOUSE
```

When the game begins, the mouse is at the left margin and the cat is some distance away. The computer prompts POUNCE!!, waiting for a number to be typed.

```
          ::                ##

          POUNCE!!
```

The cat pounces toward the mouse, moving a distance corresponding to that number, in this case 4.

```
                      4

          ::                ##

          POUNCE!!
```

Several things can happen as a result of a pounce, depending on the relation between the size of the pounce and the distance between the animals.

1. If the size of the pounce matches the distance: The cat lands on the mouse and catches it, and the game ends.

```
::##

POUNCE!!
        1
*** YOU'VE CAUGHT THE MOUSE WITH 5 POUNCES!!
WANT TO CHASE ANOTHER MOUSE? YES
```

2. If the size of the pounce is smaller than the distance: The cat approaches the mouse, but doesn't get it.

```
:                                          #

POUNCE!!
         15

:                    #
```

3. If the size of the pounce is greater than the distance: The cat jumps over the mouse, landing on the other side. (In order for the play to remain on the paper, the whole frame of reference is shifted to the right.)

```
::::                                         ####

POUNCE!!
         17

####                                 ::::
```

(Since the cat always pounces toward the mouse, the player doesn't have to worry about direction when making pounce decisions.)

4. Sometimes, when the size of the pounce is *almost* the distance to the mouse, the mouse runs away.

```
POUNCE!!
         4

::::####

OH! OH! THE MOUSE SEES THE CAT!
  RUN, MOUSE, RUN!

####                                 ::::
```

A small percentage of the time that the mouse runs, it will run into its hole. Then the game ends with the mouse the winner.

Two runs are on the opposite page. Notice that the sizes of the cat and mouse can vary between games. Notice, too, how the size of the cat affects the size of its pounce.