BASIC program can be created to solve this numbering problem, but BASIC may be too slow. Solving this new problem leads to further complications which would not be necessary with a good operating system editor.

There are certain philosophical advantages to a small operating system. OS65D is small enough that its entire operation can be understood at once. This means hackers can modify and alter the system, not just by POKES and patches, but fundamentally, to suit their own needs. In my experience, most hobby OSI computer owners aspire to or already fall in this hacking category. The smallness of the system puts the user in direct contact with the most fundamental operating system commands and operations. Even though it is slightly more involved, this gives the user the very maximum of control over the system.

This article was written using disassemblies of OS65D V3.2 (NMHZ) Release November 1979. Future articles will cover: (1) the I/O routines; (2) the Disk routines; (3) the ROM, and (4) miscellaneous bits and pieces. The disassemblies I have made are fully annotated (by hand) and are available for those who would like to use them. Send a stamped, self-addressed postcard to me to determine availability.

Tom Berger
10670 Hollywood Blvd.
Coon Rapids, MN 55433

A Six-Gun Shootout Game For The OSI C1P

Charles L. Stanford

The Six Gun Shootout game is a very pleasant and fun activity, particularly for the six to twelve or so age group. But this article concerns more than just the mechanics of writing another BASIC game for the C1P. When I originally wrote the program almost two years ago, we were reasonably satisfied with it. Sure, it was slow. Every time a player moved his gunfighter up or down the screen, the graphics POKEs took a lot longer than desired. And remembering that the "1" key was UP and the "2" key was down took a lot away. Those of you who have seen my articles on Fast Graphics (COMPUTE II Issue 3) and on interfacing the Atari Joystick to the C1P (COMPUTE Issue 7) can grasp what happened. Making that program work like it should has taught us more about the workings of the machine, over the past year, than any dozen manuals or articles.

This article, then, is a summing up of the methods we used to speed up both the software and the hardware to make BASIC games both more fun and much more saleable in the not inconsiderable Software marketplace.

BASIC Program Description

The game runs much as the early Arcade versions did. Each player has his gunfighter, who can shoot across the screen. Three Cacti obstruct some of the view, and move to a new location after each shot. Each player can move up or down, and shoot. Each gets 15 shots, and 5 hits wins.

The BASIC program shown in Listing 1 is fairly well annotated with REMs, but a few of the routines bear some discussion. The initialization starting at Line 5 sets the screen up as though no joysticks were available. This was deliberate, and makes the game more universally useful. It is a good idea to do this on all games, whether for paddles or for joysticks. The scoring from Line 200 is handled indirectly through the Fast Graphics Machine Language subroutine. Thus the POKEs of the ASCII characters are to that program rather than to the

OSI

SOFTWARE FOR OSI

OSI

A JOURNAL FOR OSI USERS!!

The Aardvark Journal is a bimonthly tutorial for OSI users. It features programs customized for OSI and has run articles like these:

1) Using String Variables.

2) High Speed Basic On An OSI.

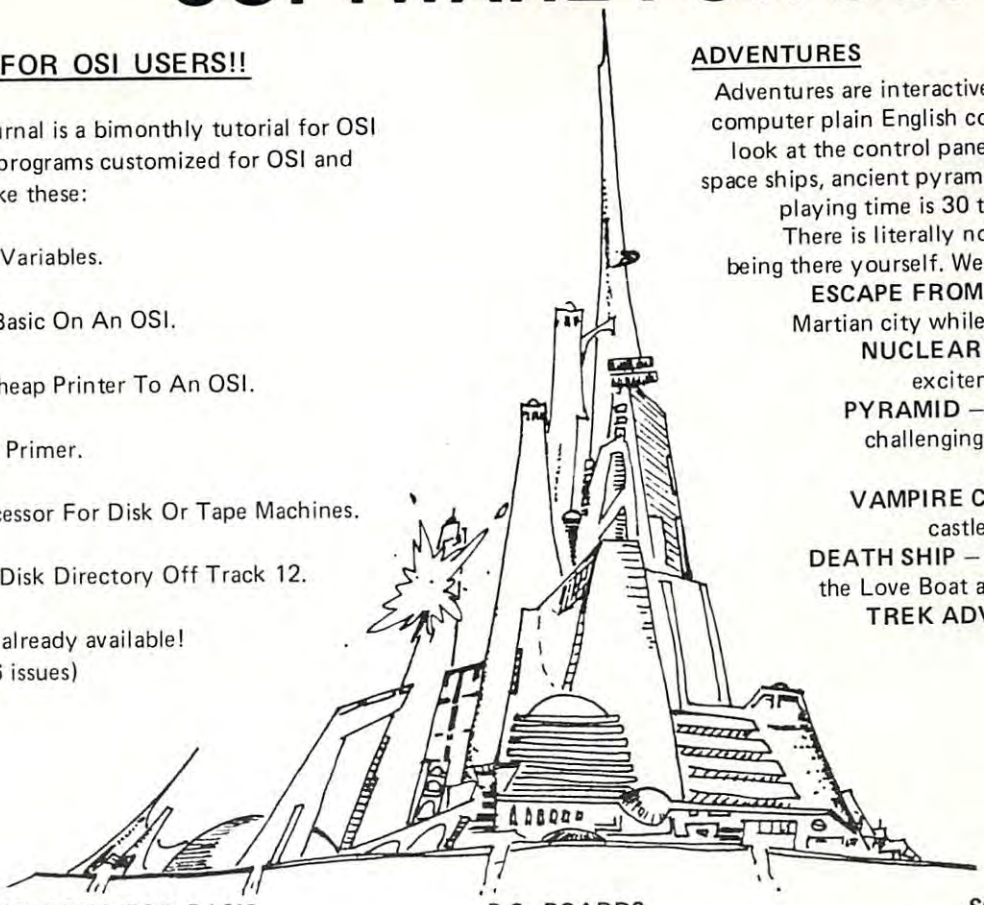
3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

Four back issues already available!
$9.00 per year (6 issues)



ADVENTURES

Adventures are interactive fantasies where you give the computer plain English commands (i.e. take the sword, look at the control panel.) as you explore alien cities, space ships, ancient pyramids and sunken subs. Average playing time is 30 to 40 hours in several sessions. There is literally nothing else like them – except being there yourself. We have six adventures available.

ESCAPE FROM MARS – Explore an ancient Martian city while you prepare for your escape.

NUCLEAR SUBMARINE – Fast moving excitement at the bottom of the sea.

PYRAMID – Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.

VAMPIRE CASTLE – A day in old Drac's castle. But it's getting dark outside.

DEATH SHIP – It's a cruise ship – but it ain't the Love Boat and survival is far from certain.

TREK ADVENTURE – Takes place on a familiar starship. Almost as good as being there.

$14.95 each

NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S – for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line.), Software selectable scroll windows, two instant screen clears (scroll window only and full screen.), software choice of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48, or 64 characters per line. All that and it sells for a measly $39.95.

C1E/C2E for C1/C2/C4/C8 Basic in ROM machines.
This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains an extended machine code monitor. It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Specify system $59.95

DISK UTILITIES

SUPER COPY – Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time – up to 12 tracks at a pass. It's almost as fast as dual disk copying. – $15.95

MAXIPROSS (WORD PROCESSOR) – 65D polled keyboard only - has global and line edit, right and left margin justification, imbedded margin commands, choice of single, double or triple spacing, file access capabilities and all the features of a major word processor – and it's only $39.95.

P.C. BOARDS

MEMORY BOARDS!! – for the C1P. – and they contain parallel ports!
Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel ports! It sells as a bare board for $29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

PROM BURNER FOR THE C1P – Burns single supply 2716's. Bare board – $24.95.

MOTHER BOARD – Expand your expansion connector from one to five connectors or use it to adapt our C1P boards to your C4/8P. - $14.95.

ARCADE AND VIDEO GAMES

ALIEN INVADERS with machine code moves – for fast action. This is our best invaders yet. The disk version is so fast that we had to add selectable speeds to make it playable.
Tape - $10.95 – Disk - $12.95

TIME TREK (8K) – real time Startrek action. See your torpedoes move across the screen! Real graphics – no more scrolling displays. $9.95

STARFIGHTER – a real time space war where you face cruisers, battleships and fighters using a variety of weapons. Your screen contains working instrumentation and a real time display of the alien ships. $6.95 in black and white - $7.95 in color and sound.

MINOS – A game with amazing 3D graphics. You see a maze from the top, the screen blanks, and then you are in the maze at ground level, finding your way through on foot. Realistic enough to cause claustrophobia. – $12.95

SCREEN EDITORS

These programs all allow the editing of basic lines. All assume that you are using the standard OSI video display and polled keyboard.

C1P CURSOR CONTROL – A program that uses no RAM normally available to the system. (We hid it in unused space on page 2). It provides real backspace, insert, delete and replace functions and an optional instant screen clear. $11.95

C2/4 CURSOR. This one uses 366 BYTES of RAM to provide a full screen editor. Edit and change lines on any part of the screen. (Basic in ROM systems only.)

FOR DISK SYSTEMS – (65D, polled keyboard and standard video only.)

SUPERDISK. Contains a basic text editor with functions similar to the above programs and also contains a renumberer, variable table maker, search and new BEXEC* programs. The BEXEC* provides a directory, create, delete, and change utilities on one track and is worth having by itself. – $24.95 on 5" disk - $26.95 on 8".

AARDVARK IS NOW AN OSI DEALER!

Now you can buy from people who can support your machine.

–THIS MONTH'S SPECIALS–

Superboard II	$279
C1P Model II	429
C4P	749

. . . and we'll include a free Text Editor Tape with each machine!

Video Modification Plans and P.C. Boards for C1P as low as $4.95

VISA

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

master charge

OSI

Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088
(313) 669-3110

OSI

screen. One routine, the "man dead" sequence, is still done in BASIC POKEs as a delay is called for here anyway.

The Joystick Interface

As described in the previous article mentioned above, the joysticks are interfaced to the keyboard such that any position can be directly related to the pressing of one or two keys. While the Atari Joysticks have eight positions around the center, only two of them are used in this game, and the others are "masked" out of existance. This is done with the routines beginning at Lines 700 and 750. Line 710's POKE K, 127 activates only Row 7 of the Keyboard. The first statement of Line 720 ORs away any columns except 5, 4, and 3, by forcing 1s into the others. Thus only keys 3, 4, and 5 are accepted as valid inputs. The next two statements of that line mask all but Column 5, so that a "shoot" command gets precedence over a "move" command. Finally, the other two keys are examined in Lines 725 and 730, and one of the move routines is addressed.

The routine at 750 works the same, except that Row 6 is activated, and the other player gets his chance. Each player is queried in turn, so one guy can't stand there and fill the other full of holes. The joystick works the same way as the keyboard, and is certainly a lot easier to use without a lot of practice, especially by the younger players.

Fast Graphics for the Six Gunners

The machine language graphics is done exactly the same way as the Choo Choo Collision demonstration program of the other previous article. A standard routine, shown in Listing 2, addresses a table of graphics symbols. These symbols are tailored for any game or other graphics screen display as shown in Figure 1. First, the Graphics Reference Manual is used to "draw" the characters desired, using a grid of sufficient size. Don't worry about screen location. The BASIC program takes care of that, by POKEing the table. Just lay out the characters, Determine the addresses of each of the elements of each character relative to the upper left corner of its grid, and couple that with the character code in making up the table. Each character should be ended with an #$FE (if there are more characters) or an #$FF (for the last character, or to end the routine).

Going Farther

You can just enter this program as-is, and have another nice game for your collection. Or you can dig a lot deeper, and quite possibly learn some techniques that will improve both your programming ability and some of those other games that run a bit slow, or get tiresome because the keyboard sequence is hard to use and remember. Anyone wishing to gain a deeper understanding of either the hardware or the software concepts described here should most certainly look to the other articles referenced.

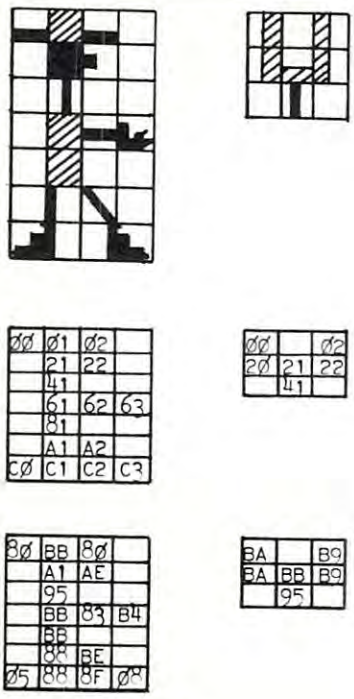


Figure 1. Graphics Development

Listing 1. Basic Program

```
5 FORS=ØTO9:PRINT:NEXT
7 REM-FOR ATARI JYSTKS
1Ø PRINT" SIX-GUN DUEL GAME
15 PRINT:PRINT:PRINT
2Ø PRINT"LEFT","RIGHT"
25 PRINT"----","-----
3Ø PRINT"  4      UP","    :
35 PRINT:PRINT"  3      DOWN","   Ø
4Ø PRINT:PRINT"  5      SHOOT","   -
45 PRINT:PRINT:PRINT"YOU HAVE 15 SHOTS EACH.
5Ø PRINT:PRINT:PRINT"FIVE HITS WINS!
55 PRINT:PRINT:PRINT"HIT SPACEBAR TO START
6Ø GOSUB1ØØ
65 GOSUB8ØØ
7Ø Z=1:L=Ø:R=Ø:SL=15:SR=15
75 POKE53Ø,1:K=57ØØ8:C=53445:D=54ØØ9
8Ø POKEK,253:IFPEEK(K)=239THEN9Ø
85 GOTO8Ø
9Ø X=USR(X):GOSUB2ØØ
95 GOTO7ØØ
1ØØ REM-MACH GRAPHICS
11Ø RESTORE
115 POKE11,34:POKE12,2:POKE254,96:POKE255,2
12Ø FORP=ØTO61:READM:POKE546+P,M:NEXT
13Ø DATA16Ø,Ø,169,32,153,Ø,211,153,Ø,21Ø,153,Ø,2Ø9,153,Ø
135 DATA2ØØ,2ØØ,2ØØ,241,234,234,234,16Ø,Ø,177,254,141,86,2,2ØØ
14Ø DATA177,254,141,87,2,2ØØ,177,254,17Ø,2ØØ,224,254,24Ø,236,224,255
145 DATA24Ø,8,177,254,2ØØ,157,68,2Ø9,2Ø8,236,96,234,234,234,234
148 REM-FIGURES NEXT
149 POKE133,255:POKE134,31:FORP=ØTO152:READM:POKE6Ø8+P,M:NEXTP
15Ø DATA197,2Ø8,Ø,128,1,187,2,128,33,161,34,174,65,149,97,187
155 DATA98,131,99,18Ø,128,187,161,136,162,19Ø,192,5,193,136,194,143
16Ø DATA195,8,254,17Ø,2Ø8,Ø,186,2,185,32,186,33,184,34,185,65
165 DATA149,254,174,2Ø9,Ø,186,2,185,32,186,33,184,34,185,65,149
17Ø DATA254,178,21Ø,Ø,186,2,185,32,186,33,184,34,185,65,149,254
175 DATA249,21Ø,1,128,2,187,3,128,33,173,34,161,66,149,96,181
18Ø DATA97,131,98,187,13Ø,187,161,189,162,143,192,5,193,136,194,143
185 DATA195,8,254,136,211,Ø,49,1,53,4,48,1Ø,48,13,49,14
19Ø DATA53,255,255,255,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
195 DATAØ,1,2,3,4,5,6,7,8,9,1Ø,11,12,13,14,15
199 RETURN
2ØØ REM-SCORE
21Ø POKE73Ø,48+L:POKE732,48+R
22Ø IFL=5ORR=5THEN29Ø
23Ø M=INT(SL/1Ø):N=SL-M*1Ø:IFM=ØTHENM=-16
24Ø U=INT(SR/1Ø):V=SR-U*1Ø:IFU=ØTHENU=-16
25Ø POKE726,48+M:POKE728,48+N
26Ø POKE734,48+U:POKE736,48+V
27Ø IFSL=ØANDSR=ØTHEN29Ø
28Ø RETURN
29Ø X=USR(X):GOTO6Ø
3ØØ REM-LEFT MAN DEAD
3Ø5 C=C+16Ø
31Ø POKEC+2,32:POKEC-31,32:POKEC-61,32:POKEC-62,32:POKEC-63,32
315 POKEC-95,32:POKEC-126,32:POKEC-127,32:POKEC-158,32:POKEC-159,32
32Ø POKEC-160,32
325 POKEC,143:POKEC+1,171:POKEC+3,14Ø
33Ø POKEC+32,187:POKEC+33,161:POKEC+34,148
335 POKEC+35,187:POKEC+36,187:POKEC+37,128
34Ø POKEC+38,176:POKEC+64,143
345 FORT=ØTO5ØØ:NEXT:C=C-16Ø:RETURN
35Ø REM-RIGHT MAN DEAD
355 D=D+15Ø
36Ø POKED+3,32:POKED+4,32:POKED-28,32:POKED-6Ø,32:POKED-61,32
365 POKED-62,32:POKED-92,32:POKED-124,32:POKED-125,32
```

```
370 POKED-155,32:POKED-156,32:POKED-157,32
375 POKED+3,139:POKED+5,171:POKED+6,136
380 POKED+32,178:POKED+33,128:POKED+34,187:POKED+35,187
385 POKED+36,148:POKED+37,161:POKED+38,187
390 POKED+70,136
395 FORT=ØTO5ØØ:NEXT:D=D-158:RETURN
400 IFSL=ØGOTO75Ø
405 Q=Ø
410 FORX=1TO3
415 IFC<E(X)-32ANDC>E(X)-128THENG=2:GOTO445
420 NEXT
425 IFC<D-96ORC>D+32THENG=23:GOTO445
430 G=23:GOSUB46Ø
435 GOSUB35Ø
440 L=L+1:GOTO45Ø
445 GOSUB46Ø
450 SL=SL-1
455 GOSUB8ØØ:GOSUB2ØØ:GOTO69Ø
460 FORX=C+1ØØTOC+1ØØ+G
465 POKEX,45
470 NEXT
475 FORT=ØTO2ØØ:NEXT
480 RETURN
500 IFSR=ØGOTO7ØØ
505 Q=1
510 FORX=1TO3
515 IFD<E(X)ANDD>E(X)-96THENG=2:GOTO545
520 NEXT
525 IFD<C-64ORD>C+64THENG=23:GOTO545
530 G=23:GOSUB56Ø
535 GOSUB3ØØ
540 R=R+1:GOTO55Ø
545 GOSUB56Ø
550 SR=SR-1
555 GOSUB8ØØ:GOSUB2ØØ:GOTO64Ø
560 FORX=D+94TOD+94-GSTEP-1
565 POKEX,45
570 NEXT
575 FORT=ØTO2ØØ:NEXT
580 RETURN
600 REM-LEFTMOVE
610 Q=Ø:IFC<53443THEN75Ø
620 C=C-32:GOTO64Ø
630 Q=Ø:IFC>54Ø19THEN75Ø
635 C=C+32
640 H=C:GOSUB9ØØ:POKE6Ø9,J:POKE6Ø8,M:X=USR(X):IFQ=1THEN7ØØ
645 GOTO75Ø
650 REM-RIGHTMOVE
660 Q=1:IFD<53443THEN7ØØ
670 D=D-32:GOTO69Ø
680 Q=1:IFD>54Ø19THEN7ØØ
685 D=D+32
690 H=D:GOSUB9ØØ:POKE689,J:POKE688,M:X=USR(X)
695 IFQ=ØTHEN75Ø
700 REM-LEFT INPUT
710 POKEK,127
715 Y1=PEEK(K)
720 Y1=Y1OR199:Y2=Y1OR247:IFY2=247THEN4ØØ
725 IFY1=239THEN6ØØ
730 IFY1=223THEN63Ø
735 GOTO75Ø
750 REM-RIGHT INPUT
760 POKEK,191
765 Z1=PEEK(K)
770 Z1=Z1OR199:Z2=Z1OR247:IFZ2=247THEN5ØØ
775 IFZ1=239THEN65Ø
780 IFZ1=223THEN68Ø
785 GOTO7ØØ
800 REM-CACTUS LOCATOR
810 EE=53414
820 FORX=1TO3
830 E(X)=EE+32*INT(22*RND(1))
840 E(X)=E(X)+X*4
850 H=E(X):GOSUB9ØØ
860 ONXGOSUB87Ø,88Ø,89Ø
865 NEXTX:RETURN
870 POKE644,J:POKE643,M
875 RETURN
880 POKE659,J:POKE658,M
885 RETURN
890 POKE674,J:POKE673,M
895 RETURN
900 REM-CHAR LOC SUB
910 J=INT(H/256)
920 M=H-J*256
930 RETURN
990 END

OK
```

```
Ø222 AØ ØØ     LDY,I
Ø224 A9 2Ø     LDA,I
Ø226 99 ØØ D3  STA,A,Y
Ø229 99 ØØ D2  STA,A,Y
Ø22C 99 ØØ D1  STA,A,Y
Ø22F 99 ØØ DØ  STA,A,Y
Ø232 C8        INY
Ø233 DØ F1     BNE
Ø235 EA        NOP
Ø236 EA        NOP
Ø237 EA        NOP
Ø238 AØ ØØ     LDY,I
Ø23A B1 FE     LDA,(I),Y
Ø23C 8D 56 Ø2  STA,A
Ø23F C8        INY
Ø24Ø B1 FE     LDA,(I),Y
Ø242 8D 57 Ø2  STA,A
Ø245 C8        INY
Ø246 B1 FE     LDA,(I),Y
Ø248 AA        TAX
Ø249 C8        INY
Ø24A E4 FE     CPX,I
Ø24C FØ EC     BEQ
Ø24E E4 FF     CPX,I
Ø25Ø FØ Ø8     BEQ
Ø252 B1 FE     LDA,(I),Y
Ø254 C8        INY
Ø255 9D 8Ø D3  STA,A,X
Ø258 DØ EC     BNE
Ø25A 6Ø        RTS
```

Listing 2. Machine Language Subroutine

```
Ø26Ø C5 DØ     Ø2A1 32 D1
Ø262 ØØ 8Ø     Ø2A3 ØØ BA
Ø264 Ø1 BB     Ø2A5 Ø2 B9
Ø266 Ø2 B9     Ø2A7 2Ø BA
Ø268 21 A1     Ø2A9 21 B8
Ø26A 22 AE     Ø2AB 22 B9
Ø26C 41 95     Ø2AD 41 95
Ø26E 61 BB     Ø2AF FE
Ø27Ø 62 B3     Ø2BØ F9 D2
Ø272 63 B4     Ø2B2 Ø1 8Ø
Ø274 81 B5     Ø2B4 Ø2 BB
Ø276 A1 B6     Ø2B6 Ø3 8Ø
Ø278 A2 BE     Ø2B8 21 AD
Ø27A CØ Ø5     Ø2BA 22 A1
Ø27C C1 88     Ø2BC 42 95
Ø27E C2 8F     Ø2BE 6Ø B5
Ø28Ø C3 Ø8     Ø2CØ 61 83
Ø282 FE        Ø2C2 62 BB
Ø283 9A D1     Ø2C4 82 BB
Ø285 ØØ BA     Ø2C6 A1 BD
Ø287 Ø2 B9     Ø2C8 A2 8F
Ø289 2Ø BA     Ø2CA CØ Ø5
Ø28B 21 B8     Ø2CC C1 88
Ø28D 22 B9     Ø2CE C2 8F
Ø28F 41 95     Ø2DØ C3 Ø8
Ø291 FE        Ø2D2 FE
Ø292 2E D2     Ø2D3 88 D3
Ø294 ØØ BA     Ø2D5 ØØ 31
Ø296 Ø2 B9     Ø2D7 Ø1 35
Ø298 2Ø BA     Ø2D9 Ø4 3Ø
Ø29A 21 B8     Ø2DB ØA 3Ø
Ø29C 22 B9     Ø2DD ØD 51
Ø29E 41 95     Ø2DF ØE 35
Ø2AØ FE        Ø2E1 FF
```

Listing 3. Machine Language Graphics Table ©