



Add a point-and-click interface, complete with windows and pull-down menus, to your BASIC programs

CoCo Does Windows and a Whole Lot More

By Tony Zamora

Recently commercial programs such as *CoCo Max*, *Lyra* and *Multi-View* have introduced the CoCo Community to the power of pull-down menus, pop-up windows and a point-and-click user interface. However, we have had no easy way to

interface, complete with windows and pull-down menus, to your own BASIC programs.

Just Point and Click

A point-and-click interface allows users to work the way they think. Instead of remembering long commands or cryptic control codes, users use a mouse (or joystick) to move a cursor around the screen. When they want to do something, they point the cursor at an object on the screen and press the mouse button.

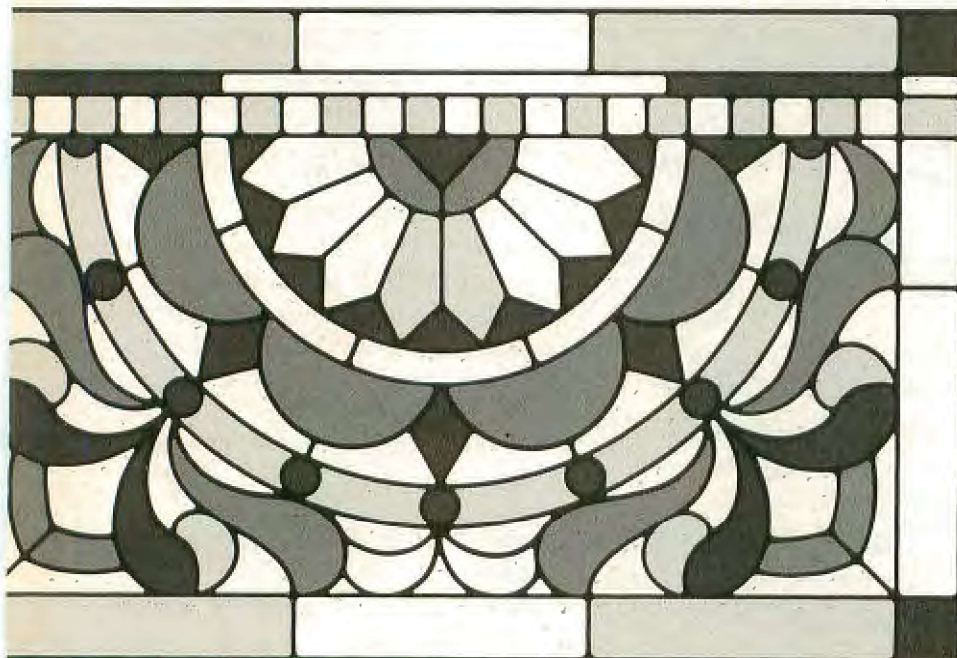
When *CoCo Desktop* is run, a list of menus appears at the top of a high-resolution graphics screen. You will see a small, arrow-shaped cursor on the screen. With the mouse plugged into the right-joystick port, move this cursor onto one of the menu titles and press the mouse button. A list of menu choices will pop up, and you will be able to select one of the items by moving the arrow up and down the menu. When you select an item, it is highlighted in reverse video. To choose the highlighted option, release the button. If you release the button when no option is highlighted, no selection is made.

Menus Made Easy

It is easy to create your own menus. All you need to do is put the names of the menus and the list of options you want to appear in the menu in DATA statements near the beginning of your program. Put the menu title first, followed by the choices in the order you want them to appear. The list should end with a special choice called END (all uppercase). For example, if you wanted

incorporate these features into a BASIC program. *CoCo Desktop* is a set of routines for the CoCo 3 allowing you to add the power of a point-and-click

Tony Zamora is currently a computer science major at Rose-Hulman Institute of Technology. When he's not programming, Tony enjoys reading, music and distance running.



to create an Option menu containing the choices Send, Receive, Clear Buffer and Baud Rate, you would write a line like the following:

```
100 DATA Options, Send, Receive,
Clear Buffer, Baud Rate, END
```

After the last menu has been put in a DATA statement, you need to denote the end of the menus with another DATA statement containing only END.

You can use two special options when creating menus. The first separates menu choices with a dotted line. To do this, enter a hyphen as one of your menu options, so it appears between the two choices you want separated. This feature is useful when you want to group a set of choices together or isolate an option. If you want the options Send and Receive grouped together in the menu, the DATA statement appears as follows:

```
100 DATA Options, Send, Receive, -,
Clear Buffer, Baud Rate, END
```

When the Options menu is chosen, a

dotted line appears between the options Receive and Clear Buffer. The second feature defines a special menu title that looks like a rainbow. When you use it, place it in the first menu in your DATA statements so it appears at the top left of the menu bar. This menu contains a list of small subroutines, called desk accessories, which perform actions that may be unrelated to the main program. For example, a pop-up calculator or a mini-text editor for making notes would be a useful desk accessory. In order to get the rainbow menu to appear instead of a menu title, put two @ signs in the DATA statements, as is done in the following example:

```
100 DATA @@, Calculator, Note Pad,
Get Info, END
```

Adding Windows and Dialog Boxes

CoCo Desktop has some subroutines that let you use windows anywhere on the screen. The first of these starts at Line 12000. This routine draws a window on the screen. It automatically saves what was on the screen, so you don't have to worry about losing any-

thing. Variables WX and WY tell *CoCo Desktop* where to draw the window. The x coordinate for the top left corner of the window goes in WX(1), and the y coordinate for the top left corner goes in WY(1). The coordinates for the bottom right-hand corner go in variables WX(2) and WY(2). After setting these variables, use a GOSUB 12000 statement to draw the window on the screen. To erase a window and replace the part of the screen it covered, call the subroutine at Line 13000.

A dialog box is similar to a window, but it contains an area in which you can type a response. Usually this kind of window is used to get input from the user. Because there are several different kinds of input (strings, numbers, etc.), *CoCo Desktop* provides a general structure, which can be modified to handle any kind of input. The dialog routine in the program draws a window and an area in which the user can type. It allows a user to enter a string consisting of digits. The part of the program calling this routine converts the string to a number and checks to see if the input is in the desired range. By modifying the



ARK ROYAL GAMES is drastically cutting prices and reducing our inventory on most of our CoCo products. Prices have been slashed on even our new programs. Send a SASE for complete price listing or \$1.00 for catalog (refunded with first order).

Better hurry. When item is depleted it will not be restocked.

EXAMPLES

ACES (64K Disk) WWI Flight/Combat simulator	\$15
DOUGHBOY (64K Disk) WWI Real Time Combat	\$14
COMPANY COMMANDER (32K) Tactical War Game	\$15
ALL MODULES FOR COMPANY COMMANDER	\$10
COMPANY COMMANDER SCENARIO CREATOR (32K)	\$12
OKINAWA (64K Disk) WWI Marine Invasion	\$12
LUFTLOTTE (32K) Battle of Britain	\$14
FIRE ONE! (CoCo 3 Disk) Sub Warfare in WWII	\$15
PRO FOOTBALL (CoCo 3) 1 or 2 players	\$12
BATAAN (64K Disk) Two games in one	\$10
TUNIS (32K) Battle in North Africa	\$ 8
GUADALCANAL (32K) America Strikes Back	\$ 7
BOMBER COMMAND (32K)	\$ 6

And more! Almost all prices have been cut. Call or write for price list.

ARK ROYAL GAMES
Post Office Box 14806 • Jacksonville, FL 32238
(904) 221-5712

Include 50 cents per program shipping and handling.
Florida residents add 6% sales tax.



CINSOFT
PROGRAMS • PERIPHERALS • SUPPLIES • SERVICE

Fast Delivery...
Friendly Service

Now in our 7th year!

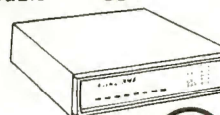


Avatex Modems

* NEW LOW PRICES *

Avatex 1200e	\$ 79	Avatex 2400	\$169
with Coco Cable	89	with Coco Cable	179
		(Coco 3 only)	
		with RS-232 Cable*	205

Reviewed in
April, 1988
Rainbow!



Avatex 1200e, Cable
AUTOTERM... \$119



Avatex 2400, Cable
AUTOTERM... \$209

*Coco 1, 2 requires Deluxe RS-232 Pak

• Call •

513-396-SOFT

• Shop by Modem •

513-396-SHOP



2235 Losantiville, Cincinnati, OH 45237

SHIPPING will be charged at our ACTUAL COST
Ohio residents add 5.5% Sales Tax COD add 3.00

subroutine to accept characters instead of digits, dialog boxes can process string input.

The dialog routine contains some statements that draw a blinking cursor in the area where the user types. There is a provision for checking if the user

clicked on buttons marked "OK" and "Cancel." Often people choose a menu option that pops up a dialog box and then decide they did not want to change anything. A Cancel button lets this person abort without any ill effects. The OK button is an alternative to pressing

ENTER when finished typing. It can also be used to accept a default value displayed in the dialog box.

To create a dialog box, draw a window where you want the box to appear. Then you draw the box where the user will type. Do this with the HLINE command with the ,B option. Once everything has been drawn in the window, set variables P, PY and L. P is the horizontal screen coordinate where the user will type; PY is the vertical coordinate; and L is the maximum length of the string the user can enter. Note: P and PY must be between 0 and 79 because they are used in an HPRINT command. Study lines 1000 through 1240 to see an example of the use of the dialog routines.

Moving Around

The workhorse of *CoCo Desktop* is a routine at Line 10000, which checks the mouse and draws the arrow-shaped cursor on the screen. This routine works like BASIC's INKEY\$ command. The difference is that instead of getting a key press, the subroutine at Line 10000 gets a mouse event. A mouse event is a signal that the mouse was moved or the user clicked. If the user moved the mouse, the routine moves the cursor to its new position and returns. If the button was pressed, the routine responds by setting variables CX and CY. CX and CY are used to return to the place on the screen where the button was pressed. For example, if the user clicked at screen position (300,45), CX is 300 and CY is 45. If the user moved the mouse but did not click, CX and CY both have the value negative one (-1).

When you click in the menu bar at the top of the screen, a menu pops up, and you can select one of the choices. If you select any of the options, a variable called SELECT is assigned a number based on the order of the choices. For instance, if you choose the first option of the first menu, SELECT will be assigned the number one. If you choose the second option, SELECT will be two. If the first menu contains four options, and you choose the first option in the second menu, SELECT will have the value five. The numbers are assigned in the order in which they appear in the DATA statements. If there is a dotted line in the menu, it does not count as a choice. If the user does not select anything, SELECT will be zero.

When you call Line 10000 in your program and want to respond to the choice made, use an ON SELECT GOTO or ON SELECT GOSUB statement. If SELECT is zero, none of the line numbers in the ON

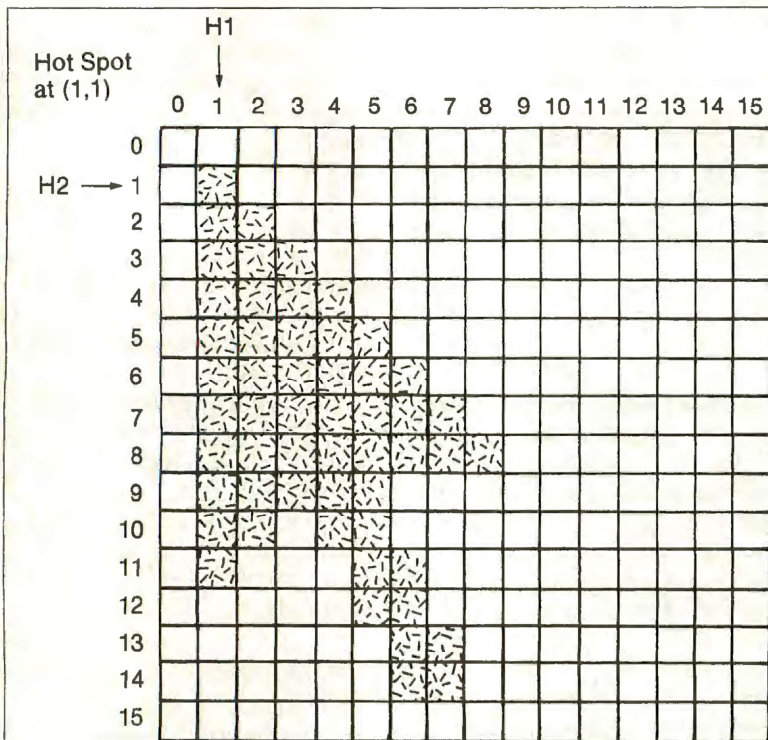


Figure 1

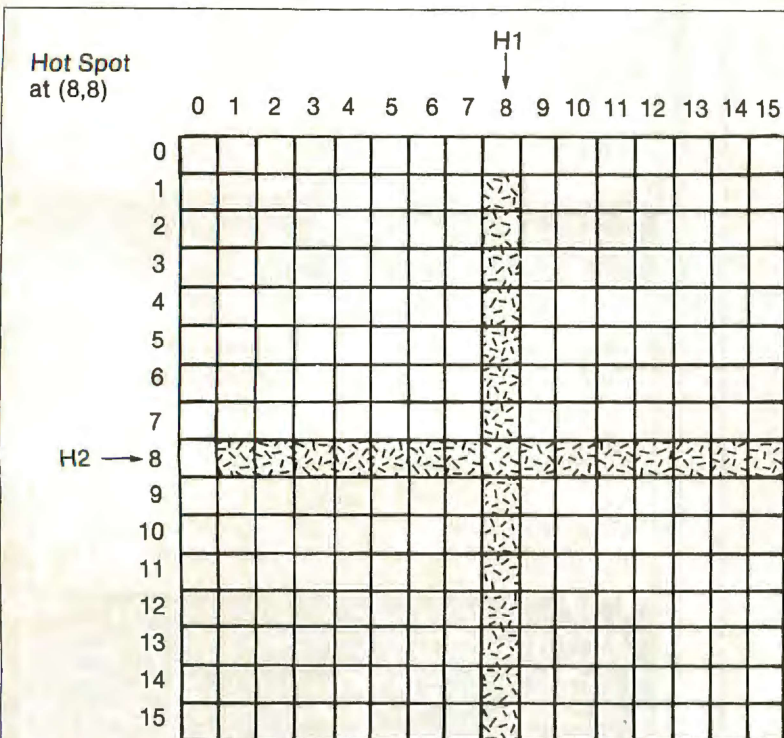


Figure 2

GOTO statement will be executed, and the program will drop through to the next statement.

Other Features

In some applications, you will need to clear the screen. So you don't have to worry about redrawing the menu titles, Line 14000 contains a routine that clears the screen and redraws the menu bar. Using this routine instead of the CLS command will keep you from having to worry about maintaining the menus.

Once you have created a window, use it for any purpose. Nothing drawn in a window affects any object outside or under the window. When you close the window, everything in the window will be erased, and the screen will appear as it did before the window was opened. However, if you draw outside the window, those changes remain intact even when you close the window. If part of an object is not in the window when the window is closed, only the part of the object in the window will be erased. This also applies to text drawn with the HPRINT command.

Problems can arise when you draw objects on the screen. When a window is opened over the cursor or a shape is

drawn on top of the cursor, part of the object or window will be erased when the cursor is moved. This happens because of the way the cursor is displayed. Whenever the cursor is drawn, it saves part of the screen. If you draw on the screen without erasing the cursor, the cursor replaces the portion of the screen it saved, wiping out whatever you just drew. Therefore, erase the cursor before you draw anything, and replace it when you are done. This will prevent the cursor from destroying any of your work. A subroutine at Line 15000 will clear the cursor and replace the area it covered. The routine at Line 16000 redraws the cursor. Using these routines guarantees the screen is redrawn correctly.

The subroutines that draw the cursor use Variable CURSOR\$. This variable contains a string used by the DRAW statement to draw the cursor. By creating different strings and assigning them to CURSOR\$, you can have several different cursors. A cursor must not be larger than 16-by-16 pixels, and you must define the cursor's *hot spot* (the point of the cursor aligned with the mouse). For example, the hot spot for the arrow cursor is (1,1) near the upper-left corner.

If you define a cursor shaped like a cross-hair, the hot spot would probably be (8,8), close to the center of the grid (see Figures 1 and 2). Store the horizontal component of the hot spot in Variable H1 and the vertical component in H2. When creating your own cursors, define the DRAW string so it will start at the hot spot. If you switch between several cursors in the same program, erase the old cursor with the routine at Line 15000 before calling Line 16000 to draw the new cursor.

Special Notes

You need to be careful about using some aspects of *CoCo Desktop*. The program uses the high-speed poke, so if your program does disk I/O, make sure you slow down the CoCo with POKE 65496,0 before each disk access. You can speed it up again with POKE 65497,0 when you are done.

When windows are drawn or menus are popped up, the area of the screen covered is saved in an HGET buffer. Because of this, it is possible to crash the program if menu options are too long or you try to create too-large windows.

If the windows get too large, the buffer will not be big enough to save the



The COCO hardware store

Super Controller I \$99.95

A Superb Controller. Along with the included C-DOS, plug-in three more software selectable DOSes or 2764 or 27128 EPROMs burned to your liking. The Internal M.E.B. lets you add Disto incredible Super Add-ons.

Multi-Board Adapter \$59.95

This Multi-Board is an adapter that plugs in any Disto Super Controller, Ramdisk or MEB Adapter. It includes a new and improved Printer Port (Centronics compatible), a faster Real Time Clock (works at 2MHz.) and a true RS-232 Serial Port (external 12 volt AC adapter required). DB25 cable included. It fits neatly inside the metal case and is still within Tandy's power limits. It also works with or without a Multi-Pak.

Mini Controller \$74.95

- Radio Shack/Tandy controller compatible
- Works on all COCOs with or without Multi-Pak
- 2 DOS switcher
- Accepts 24 or 28 pin EPROMs
- Low power draw and Gold plated edge connectors

RS-232 SuperPack \$49.95

- A Stand-Alone (Multi-Pak required) adapter that gives the user a true RS-232 Serial Port.
- Completely compatible with OS9's ACIA software.
- Compatible with software that requires the Tandy Deluxe RS-232 Pack.
- DB-25 cable included.

\$130.

Fantastic Super Controller II

- Under OS-9:
- Buffered Read/Write sector achieved without halting the CPU.
- Continual use of keyboard even while reading or writing to disk.
- System's clock no longer loses time during Read & Write.
- NMI is blocked and transferred to IRQ in software for low CPU overhead.
- Completely Interrupt driven for fast & smooth Multi-Tasking operations.
- Drivers written by KEVIN DARLING
- Now Available at your Local Radio Shack store PN 90-2009

SUPER ADD-ONS

- Real Time Clock & Printer Interface \$34.95
- Mini EPROM Programmer \$54.95
- Hard Disk Interface \$49.95
- Hard Disk Interface with RS-232 \$79.95
- Super RAM 3 ZeroK Board \$24.95
- MEB Adapter \$24.95



Master Card and Visa Accepted

1-514-383-5293

We accept phone orders • Call for Canadian Prices
Include S&H of \$4 or \$8 if order exceeds \$75
Sorry: No personal cheques

10802 Lajeunesse, Montreal, Quebec, Canada H3L 2E8

entire area, causing an ?FC (Function Call) Error. One possible solution to this is to reserve larger buffers with the HBUFF command. The commands that allocate the buffers are in Line 50. Buffer 1 is used to store the area under the cursor. Buffers 2, 3 and 4 are used for menus, and Buffer 5 keeps the part of the screen under the windows. The values for these buffers allow moderately large windows and menus. If you use small menus, you may be able to decrease the buffer for menus and increase the buffer for windows.

If you use long menus, you may need to increase the DIM statements in Line 150. Variable MENU\$ holds the menu titles. It currently holds 10 menu titles. If you need more than 10, increase the dimension of the array. CHOICE\$ keeps the names of the menu options. NU stores the number of the menu, which is returned in SELECT. NU should have the same dimensions as CHOICE\$. HEIGHT must be dimensioned to the maximum number of menu choices, and WI and RANGE must have the same dimension as MENU\$.

I chose HSCREEN 3 as the graphics mode because it provides high resolution without using a lot of memory. *CoCo Desktop* can be modified to use other graphics modes with more colors. If you make this modification, be careful: The other graphics modes use more memory to provide the extra colors, so you will not have as much memory for windows and menus. If you are using medium-sized windows, this should not be a problem. However, big windows might need more memory, so make sure your windows don't get too big.

If you are using DATA statements in your program, be careful that your data values do not conflict with the values containing the menu options. As long as your DATA statements come after the DATA statements containing the menu choices, there should be no problems. There is only one exception: When you use a RESTORE statement to reread your data, the statement will restore not only your data but also data for the menu items. You will have to use some dummy READ statements to skip the menu choices, so you can read your own data.

Using Variables

In order to minimize conflict between your variables and the variables used by the program, the variables *CoCo Desktop* uses all start with the letters CD or the letter C, followed by a digit. As long as your program does not use any variables with these names, everything

C1, C2, C3, C4	Loop Variables
C5	Unused
C6	Unused
C7	Current x-position of cursor
C8	Current y-position of cursor
C9	Previous x-position of cursor
C0	Previous y-position of cursor
CD(0)	Counts the number of menu items
CD(1)	Checks if the routine has been entered
CD(2)	Unused
CD(3)	Unused
CD(4)	Loop Variable
CD(5)	Length of the menu option
CD(6)	Left margin of menu
CD(7)	Number of menus
CD(8)	Vertical position of dotted line
CD(9)	Unused
CD(10)	Unused
CD(11)	Scaling factor for menus
CD(12)	Unused
CD(13)	Right margin of menu
CD(14)	The menu that was chosen
CD(15)	Scaling factor for menus
CD(16)	Scaling factor for menus
CD(17)	Scaling factor for menus
CD(18)	Top margin of menu
CD(19)	Number of characters in the menu bar
CD(20)	Loop Variable
CD(21)	Loop Variable
CD(22)	Bottom margin of menu

Table 1: Program Variables

should work fine. However, there are some variables the program uses to communicate with your program that you must avoid as well. These are SELECT, CX, CY, WX, WY, CURSOR\$, RAINBOW\$, ARROW\$, MENU\$, CHOICE\$, NU, HEIGHT, WI and RANGE. (See Table 1 for an explanation of the variables.) When selecting variable names, remember that only the first two letters of a variable are significant in BASIC.

Conclusion

The listing includes all the *Desktop* routines and a demonstration program using these routines to create a point-and-click Tic-tac-toe game. The game is for two players and does not recognize wins and losses because the demo program was designed to demonstrate the use of windows and pull-down menus. By studying the program, you will be able to get an idea of how the routines work and how to better use them in your own programs. Experiment with the program and feel free to modify it to suit your particular needs.

The routines in *CoCo Desktop* should let you use menus and windows in your programs with a minimum of problems. There are many ways the program can provide a friendly user interface. These range from painting and drawing programs, to point-and-click spreadsheets, to mouse-driven word processors. If you have a collection of short programs, you can draw icons for each one and have them execute when the user clicks on the icon. This provides a nice alternative to the traditional text menus. Alternate high-resolution fonts for the HPRINT command can add individuality and style to your program. The possibilities are limited only by your imagination. Be creative and have fun.

(Questions or comments about the program may be directed to the author at 5500 Wabash Avenue, Box 568, Terre Haute, IN 47803. Please enclose an SASE when requesting a reply.) □

✓ 170189	1210124
34097	10120123
55056	1024086
690119	10390197
8401	10490125
10005	12080221
1080253	END134

The Listing: DESKTOP

```


Ø ' COPYRIGHT 1989  FALSOFT, INC
1Ø POKE&HFØ15,&H21 'MAKE SOLID H
PRINT CHARACTERS
2Ø POKE65497,Ø
3Ø PALETTE Ø,63:PALETTE1,Ø
4Ø PCLEAR1
5Ø HBUFF 1,512:HBUFF 2,2256:HBUF
F3,512:HBUFF4,512:HBUFF5,4Ø96
6Ø HCOLOR1
7Ø HSCREEN3
8Ø POKE&HFF9A,63 'SET BORDER TO
WHITE
9Ø CLEAR 2Ø96
1ØØ DATA @@,About The Desktop,Ge
t Info,Key Caps,Music,END
11Ø DATA File,New Game,-,Quit,EN
D
12Ø DATA Edit,Undo,-,Cut,Copy,Pa
ste,END
13Ø DATA Options,Foreground,Back
ground,END
14Ø DATA END
15Ø DIM CD(22),MENU$(1Ø),CHOICES$
(1Ø,2Ø),NU(1Ø,2Ø),HEIGHT(2Ø),WI(
1Ø),RANGE(1Ø)
16Ø ARROW$="CØD13M+3,-3M+1,+1DM+
1,+1D2R3U2M-1,-1U2M-1,+1U2R4M-9,
-9DC1D1ØM+1,-1U8M+1,+1D6RDRD2RD2
RULU2LU2LU5M+1,+1D3RU2M+1,+1DR"
17Ø RAINBOW$="D3R1U3M+1,-1URDUM+
1,-1R3M+1,-1R7M+1,+1R3M+1,+1DRUD
M+1,+1RD3LU3D3BL3U2LD2U2M-1,-1LU
RLM-1,-1L7M-1,+1LDRLM-1,+1LD2RU2
D2BR3U2RD2U2M+1,-1R3M+1,+1D2RU2"
18Ø CURSOR$=ARROW$:H1=1:H2=1
19Ø CD(1)=1
2ØØ CD(18)=16
21Ø CD(7)=1:CD(Ø)=1
22Ø READ MENU$(CD(7))
23Ø IF MENU$(CD(7))="END" GOTO 3
5Ø
24Ø HEIGHT(CD(7))=1
25Ø WI(CD(7))=Ø
26Ø READ CHOICES$(CD(7),HEIGHT(CD
(7)))
27Ø CD(5)=LEN(CHOICES$(CD(7),HEIG
HT(CD(7))))
28Ø IF CHOICES$(CD(7),HEIGHT(CD(7
)))="END" GOTO 33Ø
29Ø IF CD(5)>WI(CD(7)) THEN WI(C

```

```

D(7))=CD(5)
3ØØ IF CHOICES$(CD(7),HEIGHT(CD(7
)))<>"-" THEN NU(CD(7),HEIGHT(CD
(7)))=CD(Ø):CD(Ø)=CD(Ø)+1
31Ø HEIGHT(CD(7))=HEIGHT(CD(7))+
1
32Ø GOTO 26Ø
33Ø CD(7)=CD(7)+1
34Ø GOTO 22Ø
35Ø CD(7)=CD(7)-1:RANGE(Ø)=8
36Ø FOR C1=1 TO CD(7)
37Ø HEIGHT(C1)=HEIGHT(C1)-1
38Ø CD(19)=CD(19)+LEN(MENU$(C1))
+2
39Ø RANGE(C1)=RANGE(C1-1)+(LEN(M
ENU$(C1))+2)*8
4ØØ NEXT
41Ø RANGE(C1)=64Ø
42Ø IF CD(19)>8Ø THEN PRINT"Menu
bar Options Too Long":END
43Ø FOR C1=1 TO CD(7)
44Ø FOR C2=1 TO HEIGHT(C1)
45Ø IF CHOICES$(C1,C2)<>"-" THEN
CHOICES$(C1,C2)=" "+CHOICES$(C1,C2
)+STRING$(WI(C1)-LEN(CHOICES$(C1
,C2)))+1,32)
46Ø NEXT C2
47Ø NEXT C1
48Ø GOSUB 17ØØØ 'DRAW MENUBAR
49Ø ''' END MENU STEUP
5ØØ ''' YOUR PROGRAM STARTS HERE
51Ø DIM GR(2,2)
52Ø TURN$="X":FOR T=Ø TO 2:FOR T
2=Ø TO 2:GR(T,T2)=Ø:NEXT T2,T
53Ø HLINE(2ØØ,75)-(44Ø,75),PSET:
HLINE(2ØØ,115)-(44Ø,115),PSET
54Ø HLINE(28Ø,35)-(28Ø,155),PSET
:HLINE(36Ø,35)-(36Ø,155),PSET
55Ø GOSUB 1ØØØØ
56Ø ON SELECT GOTO 93Ø,7ØØ,79Ø,7
7Ø,9ØØ,99Ø,91Ø,66Ø,66Ø,66Ø,1ØØØ,
1ØØØ
57Ø IF CX<2ØØ OR CX>44Ø OR CY<35
OR CY>155 GOTO 55Ø
58Ø IF CX=28Ø OR CX=36Ø OR CY=75
OR CY=115 GOTO 55Ø
59Ø XI=INT((CX-2ØØ)/8Ø):YI=INT((
CY-35)/4Ø)

```



SPECIAL OFFER

for owners of VIP Writer, Telewriter, Textpro and Word Power. You can step up to the dazzling MAX-10 for only \$49.95. Send proof of purchase (original disk or first page of original manual) with your order.

Max-10: the only super fast word processor with "What You See Is What You Get" and Graphics. See around page 19 for full details and ordering info. VIP Writer, Telewriter, Textpro and Word Power are trademarks of SD Enterprises, Cognitec, Cer-Comp, and Microcom Software, respectively.


```

600 IF GR(XI,YI) THEN SOUND 55,1
:GOTO 550 ELSE GR(XI,YI)=1
610 A=XI*80+200:B=YI*40+35:MOVE=
1:LX=XI:LY=YI
620 GOSUB 15000
630 IF TURN$="X" THEN HLINE(A+10
,B+5)-(A+70,B+35),PSET:HLIN(A+1
0,B+35)-(A+70,B+5),PSET:TURN$="O
" ELSE HCIRCLE(A+40,B+20),30:TUR
N$="X"
640 GOSUB 16000
650 GOTO 550
660 WX(1)=208:WY(1)=70:WX(2)=430
:WY(2)=120:GOSUB 12000
670 GOSUB 15000:SOUND 55,1:HPRIN
T(29,11),"Sorry, not implemented
":HPRINT(31,12),"Click to contin
ue":GOSUB 16000
680 GOSUB 10000:IF CX=-1 AND CY=
-1 AND SELECT=0 GOTO 680
690 GOSUB 13000:GOTO 560
700 WX(1)=170:WY(1)=70:WX(2)=470
:WY(2)=130:GOSUB 12000:GOSUB 150
00
710 HPRINT(23,10),"The CoCo Desk
top - A programming"
720 HPRINT(23,11),"environment w
hich supports Pop-Up"
730 HPRINT(23,12),"Windows and P
ull-Down Menus.":HPRINT(23,14),"
Available Memory ":HPRINT(42,14
),MEM:HPRINT(49,14),"Bytes"
740 GOSUB 16000
750 GOSUB 10000:IF CX=-1 AND CY=
-1 AND SELECT=0 GOTO 750
760 GOSUB 13000:GOTO 560
770 POKE65496,0:PLAY"T3L2FL8GB-A
GL404CCL8CDO3AB-L4GGL8GB-AGFO4CO
3GAL4F":POKE65497,0
780 GOTO 550
790 WX(1)=70:WY(1)=60:WX(2)=310:
WY(2)=112:GOSUB 12000:GOSUB 15000
800 FOR C=65 TO 90:HPRINT(11+C-6
5,9),CHR$(C):NEXT
810 FOR C=97 TO 122:HPRINT(11+C-
97,10),CHR$(C):NEXT
820 FOR C=33 TO 58:HPRINT(11+C-3
3,11),CHR$(C):NEXT
830 FOR C=59 TO 64:HPRINT(11+C-5
9,12),CHR$(C):NEXT
840 FOR C=91 TO 96:HPRINT(17+C-9
1,12),CHR$(C):NEXT
850 FOR C=123 TO 126:HPRINT(23+C
-123,12),CHR$(C):NEXT
860 GOSUB 16000
870 GOSUB 10000:IF CX=-1 AND CY=
-1 AND SELECT=0 GOTO 870
880 GOSUB 13000:GOTO 560
890 GOTO 550
900 GOSUB 14000:GOTO 520 'CLEAR

```

```

SCREEN AND REDRAW MENUBAR
910 IF MOVE=1 THEN GR(LX,LY)=0:M
OVE=0:IF TURN$="O" THEN TURN$="X
":HLINE(A+10,B+5)-(A+70,B+35),PR
ESET:HLIN(A+10,B+35)-(A+70,B+5)
,PRESET ELSE TURN$="O":HCIRCLE(A
+40,B+20),30,0
920 GOTO 550
930 WX(1)=228:WY(1)=30:WX(2)=410
:WY(2)=90:GOSUB 12000
940 GOSUB 15000 'CLEAR CURSOR
950 HPRINT(32,5),"The CoCo Deskt
op":HPRINT(33,7),"By Tony Zamora
":HPRINT(33,9),"Copyright 1988"
960 GOSUB 16000 'DRAW CURSOR
970 GOSUB 10000:IF CX=-1 AND CY=
-1 AND SELECT=0 GOTO 970
980 GOSUB 13000:GOTO 560
990 POKE 65496,0:CLS:RGB:END
1000 TEMP=SELECT:WX(1)=144:WY(1)
=16:WX(2)=496:WY(2)=64:GOSUB 120
00:GOSUB 15000
1010 IF TEMP=11 THEN HPRINT(20,3
),"Enter the new foreground colo
r." ELSE HPRINT(20,3),"Enter the
new background color."
1020 HPRINT(53,3),"Cancel":HPRIN
T(55,5),"OK"
1030 HCIRCLE(421,25),8,,1,.5,.75
:HLIN(421,21)-(471,21),PSET:HLI
NE(421,34)-(471,34),PSET:HCIRCLE
(421,30),8,,1,.25,.5:HCIRCLE(471
,25),8,,1,.75,0:HCIRCLE(471,30),
8,,1,0,.25:HLIN(413,26)-(413,29
),PSET:HLIN(479,26)-(479,29),PS
ET
1040 HCIRCLE(421,41),8,,1,.5,.75
:HLIN(421,37)-(471,37),PSET:HLI
NE(421,50)-(471,50),PSET:HCIRCLE
(421,46),8,,1,.25,.5:HCIRCLE(471
,41),8,,1,.75,0:HCIRCLE(471,46),
8,,1,0,.25:HLIN(413,42)-(413,45
),PSET:HLIN(479,42)-(479,45),PS
ET
1050 HLINE(157,37)-(352,50),PSET
,B:GOSUB 16000
1060 P=20:PY=5:L=23:GOSUB 1110 '
INPUT THE COLOR
1070 IF ST$<>STRING$(L," ") THEN
ST=VAL(ST$) ELSE 1100
1080 IF ST>63 THEN SOUND 55,1:GO
TO 1060
1090 IF TEMP=11 THEN PALETTE 1,S
T ELSE PALETTE 0,ST:POKE&HFF9A,S
T
1100 GOSUB 13000:IF SELECT=0 THE
N GOTO 550 ELSE GOTO 560
1110 R=P:ST$=STRING$(L," "):GOSU
B15000:HPRINT(R,PY),ST$:GOSUB160
00
1120 GOSUB 10000:K$=INKEY$:TR=TI

```



```

MER:IF TR-INT(TR/3)*3=0 THEN HLI
NE(R*8,PY*8-1)-(R*8+1,PY*8+8),PS
ET,B
1130 IF SELECT<>0 THEN ST$=STRIN
G$(L," "):RETURN
1140 !CHECK TO SEE IF THEY CHOSE
"CANCEL" OR "OK"
1150 IF CX>=421 AND CX<=471 AND
CY>=21 AND CY<=30 THEN ST$=STRIN
G$(L," "):RETURN
1160 IF CX>=421 AND CX<=471 AND
CY>=37 AND CY<=46 THEN RETURN
1170 HLINE(R*8,PY*8-1)-(R*8+1,PY
*8+8),PSET,B
1180 IF K$="" GOTO 1120
1190 IF K$<>CHR$(8) GOTO 1220
1200 R=R-1:IF R<P THEN R=P
1210 MID$(ST$,R-P+1,1)="":GOSUB
15000:HPRINT(R,PY),"":GOSUB1600
0:GOTO 1120
1220 IF K$=CHR$(13) THEN RETURN
1230 IF K$>="0" AND K$<="9" THEN
IF R<L+P THEN MID$(ST$,R-P+1,1)
=K$:GOSUB15000:HPRINT(R,PY),K$:G
OSUB16000:R=R+1:IF R>L+P THEN R=
L+P
1240 GOTO 1120
9999 'GET A MOUSE EVENT
10000 SELECT=0
10010 C7=INT(JOYSTK(0)*9.9+H1):C
8=INT(JOYSTK(1)*2.783+H2)
10020 IF BUTTON(0)<>0 AND C8<10
AND CD(1)<>1 THEN HPUT(C9-H1,C0-
H2)-(C9-H1+16,C0-H2+16),1:GOSUB
10120:CX=-1:CY=-1:RETURN
10030 IF C7=C9 AND C8=C0 GOTO 10
100
10040 IF CD(1)<>1 THEN HPUT(C9-H
1,C0-H2)-(C9-H1+16,C0-H2+16),1
10050 IF C7-H1<0 THEN C7=H1
10060 IF C8-H2<0 THEN C8=H2
10070 HGET(C7-H1,C8-H2)-(C7-H1+1
6,C8-H2+16),1
10080 HDRAW "BM"+STR$(C7)+"", "+ST

```

```

R$(C8)+CURSOR$
10090 C9=C7:C0=C8:CD(1)=0
10100 IF BUTTON(0)<>0 AND C8>=10
THEN CX=C7:CY=C8:RETURN
10110 CX=-1:CY=-1:RETURN
10120 CD(14)=1:SELECT=0:C7=INT(J
OYSTK(0)*9.9+H1)
10130 IF C7>RANGE(CD(14)) THEN C
D(14)=CD(14)+1:GOTO 10130
10140 IF CD(14)>CD(7) GOTO 10530
10150 CD(6)=RANGE(CD(14)-1):CD(1
3)=(WI(CD(14))+2)*8+CD(6)-1:CD(2
2)=HEIGHT(CD(14))*8+15
10160 HGET(CD(6)-1,11)-(CD(13)+1
,CD(22)+1),2
10170 HLINE(CD(6)-1,10)-(CD(13)+
1,CD(22)+1),PSET,B:HGET(CD(6),0)
-(RANGE(CD(14))-1,9),3:HPUT(CD(6
),0)-(RANGE(CD(14))-1,9),3,PSET
10180 HLINE(CD(6)-1,0)-(CD(6)-1,
9),PSET
10190 HCOLOR0:HLINE(CD(6),11)-(C
D(13),15),PSET,BF:HCOLOR1
10200 HLINE(CD(13)+2,12)-(CD(13)
+2,CD(22)+1),PSET
10210 CD(4)=CD(6)/8
10220 FOR C3=1 TO HEIGHT(CD(14))
10230 IF CHOICE$(CD(14),C3)="-"
THEN HPRINT(CD(4),C3+1),STRING$(
WI(CD(14))+2," "):CD(8)=(C3+1)*8
+3:FOR C4=CD(6) TO CD(13) STEP 2
:HSET(C4,CD(8)):NEXT:GOTO10250
10240 HPRINT(CD(4),C3+1),CHOICE$
(CD(14),C3)
10250 NEXT
10260 GOTO 10430
10270 IF BUTTON(0)=0 GOTO 10480
10280 C7=INT(JOYSTK(0)*9.9+H1):C
8=INT(JOYSTK(1)*2.783+H2)
10290 IF C7=C9 AND C8=C0 GOTO 10
270
10300 IF C8<9 AND (C7<RANGE(CD(1
4)-1) OR C7>RANGE(CD(14))) AND C

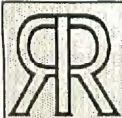
```

Join the MIDI revolution -- Turn your music synthesizer into a professional recording studio!

With a little imagination and artistry, you can control a synthesizer with your trusty CoCo and produce professional quality music. Hooked up to an inexpensive synthesizer (such as the Yamaha PSS-480), your CoCo will produce incredibly rich 8+ part music. We carry the most complete line of music products for the Color Computer. Our programs are not toys! They are used world-wide by professional musicians. All come backed by a guarantee of satisfaction. We have been in the CoCo music business for over 5 years and are committed to providing you with the best.

CoCo MIDI 3: A powerful MIDI recorder/sequencer; includes hardware MIDI interface and cables.. Multipak or Y cable required. \$149.95. **FB01EDIT** Edit and creat new FB-01 voices on your CoCo 3. Requires CoCo MIDI hardware pack. \$29.95. **FB-01 Calc** creates custom configurations for your FB-01 and Lyra. \$19.95. **Music Library** is a collection of 4-part music to play on your CoCo (no extras needed!). Each disk \$5.00. **Musica 2**, the companion music composition program, is only \$24.95. **Lyra Lybrary** is an extensive collection of 6-8 part music (over 13 disks) to play on your MIDI synthesizer. Each disk \$14.95. **Lyra:** a powerful yet easy-to-use MIDI music composition program. 8 voices plus much more. Comes with MIDI cable. Only \$59.95. **Yamaha PSS-480** 12 note multitimbral synthesizer: call for price and availability.

Ordering information: send check or money order. Sorry, no credit cards. COD is ok.



Rulaford Research

P.O. Box 143
Imperial Beach, CA 92032
(619) 690-3648 (evenings 6-10 PT)


```

7>8 GOTO 10480
10310 CD(15)=FIX((C8-8)/8)
10320 CD(17)=FIX((C0-8)/8)
10330 CD(16)=CD(15)*8+8
10340 CD(11)=CD(17)*8+8
10350 HPUT(C9-H1,C0-H2)-(C9-H1+1
6,C0-H2+16),1
10360 IF C7<CD(6) OR C7>CD(13) O
R C8<CD(18) OR C8>CD(22) GOTO 10
420 'IF C7 IS OUTSIDE, GOTO
10370 IF C9<CD(6) OR C9>CD(13) O
R C0<CD(18) OR C0>CD(22) THEN IF
CHOICES$(CD(14),CD(15))<>"-" THE
N HGET(CD(6),CD(16))-(CD(13),CD(
16)+7),4:HPUT(CD(6),CD(16))-(CD(
13),CD(16)+7),4,PRESET:GOTO 1043
0 ELSE GOTO 10430
10380 IF CD(16)=CD(11) GOTO 1043
0
10390 IF CHOICES$(CD(14),CD(15))=
"-" THEN HGET(CD(6),CD(11))-(CD(
13),CD(11)+7),4:HPUT(CD(6),CD(11
))-(CD(13),CD(11)+7),4,PRESET:GO
TO 10430
10400 IF CHOICES$(CD(14),CD(17))=
"-" THEN HGET(CD(6),CD(16))-(CD(
13),CD(16)+7),4:HPUT(CD(6),CD(16
))-(CD(13),CD(16)+7),4,PRESET:GO
TO 10430
10410 HGET(CD(6),CD(11))-(CD(13)
,CD(11)+7),4:HPUT(CD(6),CD(11))-(
CD(13),CD(11)+7),4,PRESET:HGET(
CD(6),CD(16))-(CD(13),CD(16)+7),
4:HPUT(CD(6),CD(16))-(CD(13),CD(
16)+7),4,PRESET:GOTO 10430
10420 IF C9>=CD(6) AND C9<=CD(13
) AND C0>=CD(18) AND C0<=CD(22)
THEN IF CHOICES$(CD(14),CD(17))<>
"-" THEN HGET(CD(6),CD(11))-(CD(
13),CD(11)+7),4:HPUT(CD(6),CD(11
))-(CD(13),CD(11)+7),4,PRESET
10430 IF C7-H1<0 THEN C7=H1
10440 IF C8-H2<0 THEN C8=H2
10450 HGET(C7-H1,C8-H2)-(C7-H1+1
6,C8-H2+16),1
10460 HDRAW "BM"+STR$(C7)+", "+ST
R$(C8)+CURSOR$
10470 C9=C7:C0=C8:GOTO 10270
10480 HPUT(C9-H1,C0-H2)-(C9-H1+1
6,C0-H2+16),1
10490 HPUT(CD(6)-1,11)-(CD(13)+1
,CD(22)+1),2:HGET(CD(6),0)-(RANG
E(CD(14))-1,9),3:HPUT(CD(6),0)-
(RANGE(CD(14))-1,9),3,PRESET
10500 HLINE(CD(6)-1,0)-(CD(6)-1,
9),PRESET
10510 IF C7-H1<0 THEN C7=H1
10520 IF C8-H2<0 THEN C8=H2

```

```

10530 HGET(C7-H1,C8-H2)-(C7-H1+1
6,C8-H2+16),1
10540 HDRAW "BM"+STR$(C7)+", "+ST
R$(C8)+CURSOR$
10550 C9=C7:C0=C8
10560 C7=INT(JOYSTK(0)*9.9+H1):C
8=INT(JOYSTK(1)*2.783+H2)
10570 IF C7>=CD(6) AND C7<=CD(13
) AND C8>=CD(18) AND C8<=CD(22)
THEN SELECT=NU(CD(14),CD(15))
10580 RETURN
11999 'POP UP A WINDOW
12000 GOSUB 15000
12010 HGET(WX(1),WY(1))-(WX(2),W
Y(2)),5
12020 HLINE(WX(1),WY(1))-(WX(2),
WY(2)),PSET,B
12030 HCOLOR0
12040 HLINE(WX(1)+1,WY(1)+1)-(WX
(2)-1,WY(2)-1),PSET,BF
12050 HCOLOR1
12060 HLINE(WX(1)+4,WY(1)+2)-(WX
(2)-4,WY(2)-2),PSET,B
12070 HLINE(WX(1)+5,WY(1)+2)-(WX
(1)+5,WY(2)-2),PSET
12080 HLINE(WX(2)-5,WY(1)+2)-(WX
(2)-5,WY(2)-2),PSET
12090 GOSUB 16000
12100 RETURN
12999 'ERASE THE CURRENT WINDOW
13000 GOSUB 15000:HPUT(WX(1),WY(1
))-(WX(2),WY(2)),5:GOSUB 16000:RE
TURN
13999 'CLEAR THE SCREEN AND REDR
AW THE MENUBAR
14000 HCLS:GOSUB 17000:CD(1)=1:C9
=-1:C0=-1:RETURN
14999 'ERASE THE CURSOR
15000 HPUT(C9-H1,C0-H2)-(C9-H1+1
6,C0-H2+16),1:RETURN
15999 'REDRAW THE CURSOR
16000 HGET(C9-H1,C0-H2)-(C9-H1+1
6,C0-H2+16),1
16010 HDRAW "BM"+STR$(C9)+", "+STR
$(C0)+CURSOR$
16020 RETURN
16999 'DRAW MENUBAR
17000 HLINE(0,10)-(639,10),PSET
17010 IF MENU$(1)="@" THEN HDRA
W "BM12,4"+RAINBOW$ ELSE HPRINT(2
,0),MENU$(1)
17020 C2=2:C1=2
17030 C1=C1+LEN(MENU$(C2-1))+2
17040 IF C2>CD(7) THEN RETURN
17050 HPRINT(C1,0),MENU$(C2)
17060 C2=C2+1
17070 GOTO 17030

```