VIC 1212

PROGRAMMER'S AID CARTRIDGE™

Cartouche d'instructions pour le programmeur™

commodore

The information in this manual has been reviewed and is believed to be entirely reliable. No responsibility, however, is assumed for inaccuracies. The material in this manual is for information purposes only, and is subject to change without notice.

Retyped and redrawn in 2006 for your enjoyment and for future generations by François Léveillé. '-)

PROGRAMMER'S AID USER MANUAL

TABLE OF CONTENTS

TABLE OF FIGURES

SECTION ONE

INTRODUCTION TO VIC 20 PROGRAMMER'S AID CARTRIDGE

1.1 Introduction

The VIC 20 PROGRAMMER'S AID CARTRIDGE has been designed to help both new and experienced BASIC programmers to write, edit and debug programs quickly and easily. This is achieved by the AID commands which are automatically incorporated into the VIC operating system when the cartridge is inserted. The cartridge also assigns some of the AID commands and some BASIC keywords to the function keys thus giving the programmer his own shorthand notation during program writing. This, together with the facility for the programmer to assign his own functions to these keys, make PROGRAMMER'S AID an invaluable programming accessory.

This manual does not attempt to teach BASIC programming on the VIC. If you have no knowledge of BASIC programming, please refer to one of the following:

VIC 20 User Manual (supplied with your computer).

An Introduction to BASIC Parts 1 and 2 by Andrew Colin (part of the VIC learning series).

1.2 The PROGRAMMER'S AID Manual

This manual is divided into three parts as outlined below:

SECTION ONE – INTRODUCTION TO PROGRAMMER'S AID

This section outlines PROGRAMMER'S AID in broad terms. It also explains how to load and start PROGRAMMER'S AID. The conventions used by this manual in describing the format of each PROGRAMMER'S AID commands are also included.

SECTION TWO – THE COMMANDS OF PROGRAMMER'S AID

In this section, the format of each PROGRAMMER'S AID is stated, its purpose is given, its uses explained and an example shown. The commands are listed in the logical order in which they might be used when writing a BASIC program such as the one included in Section Three of this manual.

SECTION THREE – USING PROGRAMMER'S AID AS A TOOL

This section illustrates the speed and efficiency of using PROGRAMMER'S AID when programming in VIC BASIC. It demonstrates how some of the cartridge's attributes were used to write, edit and debug a simple program.

1.3 Starting PROGRAMMER'S AID

The PROGRAMMER'S AID CARTRIDGE must always be inserted or removed from the VIC with the power off. The cartridge is inserted into the expansion port on your VIC with the label facing up. (See your VIC 20 User Guide.)

If a VIC 1010 Memory Expansion Board is in use, this must also be turned off. The VIC 20 PROGRAMMER'S AID CARTRIDGE may be used with VIC 1212 VIC Machine Code Monitor and/or VIC 1211A Super Expander Cartridges. It may also be used with expansion RAM in the VIC 1010 Memory Expansion Board. Please note, however, that some operations may conflict if changing from one cartridge to another. Therefore it is recommended that the VIC be turned off to effect a good "switch".

To begin using PROGRAMMER'S AID type:

SYS 28681 or SYS 7 * 4096 + 9 and press the RETURN key.

Your VIC will display: –

-- PROGRAMMER'S AID --

READY

All the PROGRAMMER'S AID commands are now included in the operating system of your VIC and may be used at any time like any other BASIC command.

When PROGRAMMER'S AID is started, the VIC is automatically in PROGRAM mode which means that the four function keys on the right of the VIC keyboard have been assigned programming commands such as GOTO, CHR$(and GOSUB. (See Section 2.3.1) The

alternative to PROGRAM mode is EDIT mode which assigns edit commands to these same keys. (See Section 2.3.2.)

There are twelve AID commands available in each mode. Each of the function keys may be pressed on its own, with the SHIFT key held down or with the CTRL key held down. When the key is pressed, one of 12 preassigned functions is displayed on the screen. Sections 2.3.1 and 2 describe the function keys and the two modes of PROGRAMMER'S AID in more detail. If he wishes, the programmer may assign his own values to these keys, i.e. different from the values generated by the PROGRAMMER'S AID CARTRIDGE. (See section 2.4.8.)

1.4 PROGRAMMER'S AID Commands

The following is a list of AID commands which are added to your VIC operating system by the PROGRAMMER'S AID CARTRIDGE:

Commands used when debugging and executing programs:
HELP, TRACE, STEP, OFF, DUMP

Commands for assigning the function keys:
KEY, EDIT, PROG, KILL

These commands can only be entered into the VIC in DIRECT mode, i.e. they cannot be incorporated into the body of the program.

PROGRAMMER'S AID also gives you six special program editing functions which can be used by holding the CTRL key down and pressing a letter on the keyboard. CTRL and U for example will erase all the characters on the line where the cursor is located when the command was executed. The six special program editing functions are detailed in Section 2.5

1.5 Entering Commands

All the commands in PROGRAMMER'S AID are executed by first typing the command and then pressing the RETURN key. Some BASIC commands assigned to the function keys have a built in RETURN and the command is executed simply by pressing the key. These commands are RUN, EDIT, RETURN, TRACE, STEP and PROG. (See Sections 2.3.1 and 2.3.2.)

1.6 Indication of Errors

If an AID command is not spelt correctly, the message "SYNTAX ERROR" will be displayed to the screen. You must retype the command using the correct spelling.

SECTION TWO

THE COMMANDS OF PROGRAMMER'S AID

2.1 Introduction

In Section Two, the two modes of PROGRAMMER'S AID are described. The AID commands are listed in the order in which they might be used when writing a BASIC program such as the one in Section Three. The use of the function keys is detailed and instructions are given to enable the programmer to assign his own values to these keys.

2.2 Function Keys

PROGRAMMER'S AID makes extensive use of the VIC's function keys. There are only four function keys on the keyboard but there are many more functions than keys.

Pressing the keys normally, you obtain functions F1, F3, F5 and F7. Holding down the SHIFT key and pressing these same keys, you get functions F2, F4, F6 and F8. By holding down the CTRL key and pressing the keys, you obtain functions F9, F10, F11 and F12. (Note that this last group of numbers is not printed on the function keys.)

2.3 PROGRAM and EDIT Modes

PROGRAMMER'S AID has two modes of operation each of which assigns different commands to the function keys on the right of the VIC's keyboard. When you initialize the PROGRAMMER'S AID CARTRIDGE, the VIC is automatically in PROGRAM mode.

2.3.1 PROGRAM MODE

In this mode, PROGRAMMER'S AID assigns the following BASIC keywords to the function keys:

NOTE: The (RETURN) indicated below will appear on the screen as a left-arrow in reverse-field following the last character inside the quotation marks.

KEY 1, "LIST"
KEY 2, "MID$("
KEY 3, "RUN (RETURN)"
KEY 4, "LEFT$("
KEY 5, "GOTO"
KEY 6, "RIGHT$("
KEY 7, "INPUT"
KEY 8, "CHR$("
KEY 9, "EDIT (RETURN)"
KEY 10, "GOSUB"
KEY 11, "RETURN (RETURN)"
KEY 12, "STR$("

By pressing the CTRL and F1 keys together or typing EDIT and pressing RETURN, the mode changes to EDIT mode.

2.3.2 EDIT MODE

In this mode, the following program editing commands are assigned to the function keys:

KEY 1, "LIST"
KEY 2, "AUTO"
KEY 3, "RUN (RETURN)"
KEY 4, "DELETE"
KEY 5, "FIND"
KEY 6, "CHANGE"
KEY 7, "TRACE (RETURN)"
KEY 8, "STEP (RETURN)"
KEY 9, "PROG (RETURN)"
KEY 10, "RENUMBER"
KEY 11, "MERGE"
KEY 12, "OFF (RETURN)"

NOTE: Commands in one mode can be typed in full whilst you are in the other mode, i.e. the PROGRAMMER'S AID modes do not limit the commands available. They simply enable you to use the shorthand form of one set or the other.

Here is a summary of the points to remember about PROGRAM and EDIT modes:

1. The primary importance of both PROGRAM and EDIT modes is to assign special commands to the VIC's function keys in order that you can write, edit and debug your programs quickly and easily.

2. You can type in any command in either mode at any time.

3. After initialization of the PROGRAMMER'S AID CARTRIDGE, the VIC is automatically in PROGRAM mode.

4. To switch from one mode to the other, simply press CTRL and F1 keys together. Alternatively you may type PROG or EDIT and press RETURN.

2.4 AID Commands

2.4.1 CONVENTIONS

The format of each AID command in this manual is presented using the following notation:

1. Items written in capital letters should be typed exactly as shown. There is no need to use the SHIFT key to obtain capitals.

2. Those items enclosed in brackets indicate a user-specified entry such as a range of program lines or character string. The brackets are printed for illustrative purposes only and should not be typed.

3. Other symbols, such as quotation marks and commas, must be typed exactly as shown.

4. The return key is indicated by (RETURN).

5. The start line number of a command is indicated by (sln).

6. The finish line number of a command is indicated by (fln).

2.4.2 AUTO

FORMAT : AUTO (sln), (interval between lines)
or AUTO

PURPOSE : To increment and display program line numbers automatically.

After initializing the PROGRAMMER'S AID CARTRIDGE, the use of the command AUTO without specifying either a start line number or an interval between lines, will automatically display line number 100. A line number 10 larger than the preceding one will be displayed each time you type in a line of BASIC code and press RETURN. Alternatively, you may use AUTO to specify your own start line number and increment.

AUTO will display the line number following your last program entry regardless of whether or not you deleted any program lines from memory during the editing of your program. You may if you wish use AUTO (specifying the next sequential line number of your program as the start line number) to bridge the gap in line numbers thus created. You may also use the RENUMBER command (see Section 2.4.3) to create a wholely sequential program listing.

If RETURN is pressed immediately following the display of a line number, the AUTO command will be canceled.

NOTE: If you use NEW to erase an existing program from memory, you must specify the start line number again with the AUTO command.

EXAMPLE : When you first start up your VIC with PROGRAMMER'S AID inserted, to display program line numbers automatically beginning at line 100 in intervals of 10 lines:

COMMAND : AUTO (RETURN)

DISPLAY : 100

TYPE : PRINT (RETURN)

DISPLAY : 100 PRINT
110

RESULT : Each time you enter a line of code and press RETURN, a line number 10 larger than the previous line number will be displayed to the screen.

EXAMPLE : To display program lines numbers automatically beginning at line 50 in intervals of 5 lines:

COMMAND : AUTO 50,5 (RETURN)

DISPLAY : 50

TYPE : PRINT (RETURN)

RESULT: 50 PRINT
55

2.4.3 RENUMBER

FORMAT : RENUMBER (sln), (interval between lines)

PURPOSE : To automatically renumber all program lines and all GOTO and GOSUB entries.

RENUMBER will automatically change all of the line numbers in your program so that the program listing then begins at the start-line you specify and all subsequent line numbers will be incremented by the interval that you have defined. RENUMBER will also change the destination line numbers of all GOTO and GOSUB entries so that they match the newly-numbered program lines. This command is especially valuable if you need space in the program to insert more code.

EXAMPLE : To renumber all the program lines and the GOTO entry of the following program:

```
100 REM RENUMBER COMMAND
110 PRINT "HELLO I'M VIC"
120 FOR L = 1 TO 1000
```

```
121 NEXT
130 PRINT "(SHIFT/CLR/HOME)":GOTO 110
```

COMMAND : RENUMBER 200,3 (RETURN)

TYPE : LIST (RETURN)

DISPLAY :
```
200 REM RENUMBER
      COMMAND
203 PRINT "HELLO I'M VIC"
206 FOR L = 1 TO 1000
209 NEXT
212 PRINT"(SHIFT/CLR/HOME)":
      GOTO 203
```

2.4.4 DELETE

FORMAT : DELETE (sln) – (fln)

PURPOSE : To delete program lines from VIC's memory.

DELETE operates on a line range in the same way as the BASIC command LIST. The various formats of the DELETE command are listed below:

DELETE (ln)	Deletes line (ln) only
DELETE (sln) – (fln)	Deletes all lines between (sln) and (fln)
DELETE – (fln)	Deletes all lines from the start of the program to line (fln)
DELETE (sln) –	Deletes all lines from line (sln) to the end of the program.

EXAMPLE : To delete the first two lines from the following program:

```
10 REM DELETE COMMAND
20 REM ONE OF THE MANY
30 REM USEFUL VIC 20
40 REM PROGRAMMER'S AID
50 REM COMMANDS
```

COMMAND : DELETE – 20 (RETURN)

TYPE : LIST (RETURN)

DISPLAY :
```
30 REM USEFUL VIC 20
40 REM PROGRAMMER'S AID
50 REM COMMANDS
```

RESULT : Lines 10 and 20 have been removed from the program

EXAMPLE : To delete lines 30 and 40 of the same program:

COMMAND : DELETE 30 – 40 (RETURN)

TYPE : LIST (RETURN)

DISPLAY : 50 REM COMMANDS

RESULT : Lines 30 to 40 have been removed from the program.

2.4.5 FIND

FORMAT : FIND (BASIC code), (sln) – (fln)

or : FIND "(character string)",(sln) – (fln)

or : FIND (character), (sln) – (fln)

PURPOSE : To search the program for a given BASIC code or character string and display the program line(s) where it appears.

FIND operates on a range of lines in the same way as a LIST command. Its various formats are listed below:

FIND (char), – (fln)	FINDs from the beginning of the program to line (fln).
FIND (char),(sln) –	FINDs from line (sln) to the end of the program.
FIND (char),(sln) – (fln)	FINDs from line (sln) to line (fln).

where (char) indicates a BASIC code, character or character string to be located.

FIND searches for a character or a BASIC code in a program and displays all lines that contain that character or BASIC code except those lines where the character code is enclosed in quotation marks. This includes all occurrences in REM statements. If you do not wish to include REM statements, enter REM with a set of quotation marks immediately following, e.g. REM "REMARK". In this way, items in REM statements will only be displayed to the screen if a search is made for a character enclosed in quotation marks, i.e. a character string.

If a character string search is made, the FIND command will display all the lines where the string is contained within quotation marks.

Holding down the CTRL key will slow down the rate at which program lines are displayed to the screen. The STOP key will abort the FIND command.

EXAMPLE : To find the character in the following program:

```
10 REM FIND COMMAND
20 PRINT "ABCDEFGH VERTICALLY"
30 A$ = "ABCDEFGH"
40 FOR C = 1 TO 8
50 PRINT MID$(A$,C,1)
60 PRINT "– – – – – – – – –"
70 NEXT C
80 REM "CONTINUE
```

COMMAND : FIND C (RETURN)

DISPLAY :
```
10 REM FIND COMMAND
40 FOR C = 1 TO 8
50 PRINT MID$(A$,C,1)
70 NEXT C
```

RESULT : Every program line containing a C will be displayed to the screen.

EXAMPLE : To FIND the character string "ABCD" in the same program:

COMMAND : FIND “ABCD” (RETURN)

DISPLAY : 20 PRINT “ABCDEFGH
VERTICALLY”
30 A$ = “ABCDEFGH”

RESULT : Every line containing the character string “ABCD” will be displayed to the screen.

2.4.6 CHANGE

FORMAT : CHANGE (old code),(new code), (sln) – (fln)

or : CHANGE (old string),(new string), (sln) – (fln)

PURPOSE : To search for an existing BASIC code or character string and replace it with a new BASIC code or character string.

If the character string to be modified is enclosed in quotation marks, CHANGE will only replace the matching character strings in the program which themselves are bounded by quotation marks. Note that in REM statements, BASIC codes which are not enclosed in quotation marks , for example 10 REM PRINT, will not be changed by the command CHANGE PRINT,PRINT #,1000 – 2000.

CHANGE operates on a line range in the same way as does the BASIC command list and its various formats are given below:

CHANGE (oc),(nc), – (fln)
Search from the beginning of the program to line (fln) replacing (oc) with (nc).

CHANGE (oc),(nc),(sln) –
Search from line (sln) to the end of the program replacing (oc) with (nc).

CHANGE (oc),(nc),(sln) – (fln)
Search from line (sln) to line (fln) replacing (oc) with (nc).

where oc refers to the old BASIC code or character string and nc refers to the new BASIC code or character string.

EXAMPLE : To change the character string “ABCDEFGH” to “12345678” in the preceding program:

COMMAND : CHANGE “ABCDEFGH”, ”12345678” (RETURN)

DISPLAY : 20 PRINT “12345678
VERTICALLY”
30 A$ = “12345678”

RESULT : All character strings “ABCDEFGH” will be changed to “12345678” and each line where the CHANGE occurs will be displayed to the screen.

2.4.7 EDIT

FORMAT : EDIT

PURPOSE : To change from PROGRAM mode to EDIT mode.

The EDIT command gives easy access to twelve program editing commands which are automatically assigned to the function keys when the cartridge is inserted into your VIC. (See section 2.3.2.)

You may type the word EDIT or press the CTRL and F1 keys to enter the EDIT mode. The function keys are then assigned the following commands: *NOTE: (RETURN) indicates a built-in carriage RETURN following the command. This is displayed on the screen as a reverse-field left-arrow. (See section 2.4.9 to add a built-in carriage RETURN.)*

KEY 1, “LIST”
KEY 2, “AUTO“
KEY 3, “RUN (RETURN)”
KEY 4, “DELETE“
KEY 5, “FIND”
KEY 6, “CHANGE“
KEY 7, “TRACE (RETURN)”
KEY 8, “STEP (RETURN)“
KEY 9, “PROG (RETURN)”
KEY 10,“RENUMBER”
KEY 11,“MERGE”
KEY 12,“OFF (RETURN)“

EXAMPLE : To enter EDIT mode from PROGRAM mode:

COMMAND : EDIT (RETURN)

DISPLAY : EDIT

RESULT : The function keys are now assigned program editing commands.

2.4.8 KEY

FORMAT : KEY

or : KEY number, “code”

PURPOSE : To list the commands assigned to the function keys which will then allow you to change their assignments.

KEY allows you to display the information assigned to the function keys and then change them if you wish. The function keys can represent anything which facilitates program writing e.g. a BASIC keyword, a graphic symbol, a number, a character string, etc., or a combination of these. The only restriction is that the information assigned to any one key must be 10 characters long or less. By using the abbreviated form of BASIC codes, e.g. ? for PRINT, you can increase the amount of information you assign to each key.

After the command KEY is entered and the twelve function key commands are displayed, the function keys can be redefined by typing the letters CLR followed by (RETURN) (to enter the function key change mode), moving the cursor to the key you wish to change, overwriting the existing command and pressing (RETURN). PROGRAMMER'S AID will automatically enter the last set of quotation marks following the last character of your entry. Thereafter, each time that key is pressed, the new command will automatically be displayed on the screen.

NOTE: If your new entry is shorter than that which was previously assigned to the function KEY, press the SPACE BAR to delete the surplus characters and press RETURN.

To change another key, clear the screen, use the KEY command again and repeat the above procedure.

Alternatively you can simply type the word KEY, the number of the key you wish to change, a comma and the new command in quotation marks. Press RETURN to enter the change.

IF, after redefining a KEY command, the message "ILLEGAL QUANTITY ERROR" is displayed, you have exceeded the 10 character maximum command length. If you cannot reduce the size of the command by abbreviation, assign part of the command to another function key.

The reverse-field left arrow after some KEY commands indicates a carriage return which is executed automatically after the command is entered.

WARNING: The new assignments of the function keys will not be retained if you change to the alternative mode. If you assign commands to the function keys in PROGRAM mode and then change to EDIT mode, when you return to PROGRAM mode, the function keys will retain their initial values.

EXAMPLE : To assign the BASIC code PRINT to the F1 function key:

COMMAND : KEY 1,"PRINT" (RETURN)

RESULT : You can now display the BASIC code PRINT on the VIC's screen by simply pressing the function key labeled F1.

2.4.9 ADDING CARRIAGE RETURNS

To eliminate the need to press RETURN after a function key command, you may add a built in RETURN to the command.

Use either method for changing KEY assignments described above. Either add the code

+ CHR$13

after the second set of quotation marks and then press RETURN or carry out the following instructions prior to the second set of quotation marks:

(a) Hold down the CTRL key and press the RVS ON key. (This turns on the reverse mode.)

(b) Type the left-arrow key (on the top left of the VIC keyboard).

(c) Type the end quote marks (") and press RETURN.

When you press this function key, you will automatically generate a RETURN after the command.

2.4.10 HELP

FORMAT : HELP

PURPOSE : To display the line on which an error occurred during program execution and highlight the position of the error in reverse field characters.

HELP will only work if the command is given immediately after an error has been detected by the BASIC interpreter and whilst the error message is displayed on the screen. If the STOP key is pressed whilst a program is running, HELP will only indicate the last program line which was executed prior to the STOP command.

NOTE: Because of the way in which the BASIC interpreter works, the exact error may not always be displayed. It will, however, be very close to the reverse field area.

EXAMPLE : To find the error in the following program:

```
10 FOR CO = 1 TO 10
20 PRINT CO + 2 * 3.142
30 NEXT C
```

TYPE : RUN (RETURN)

DISPLAY :
```
7.284
?NEXT WITHOUT FOR ERROR
IN LINE 30
READY
```

COMMAND : HELP (RETURN)

DISPLAY : 30 NEXT C (the letter "C" appears in reverse field display).

2.4.11 DUMP

FORMAT : DUMP

PURPOSE : To display the values of all variables except those in arrays.

The variables will be listed in the order in which they were defined in the program and will be displayed in the format:

variable name = value

The value of a variable can be changed by moving the cursor over the value, typing the new value and re-executing the program from the point after the line containing the original variable definition. If there are a lot of variables, holding down either SHIFT or CTRL key will control the rate of variable display to the screen. Pressing the STOP key will halt the DUMP completely.

EXAMPLE : To display the value of all variables in the following program:

```
10 A$ = "RANDOM NUMBERS"
20 PRINT A$
30 X = INT(RND(8) * 15) + 1
40 Y = INT(RND(8) * 7) + 1
50 R = X * 16 + 8 + Y
60 POKE 36879,R
70 FOR CO = 1 TO 1000:NEXT CO
80 GOTO 20
```

TYPE : RUN (RETURN)

ACTION : After a few seconds, press the STOP key

DISPLAY : READY

COMMAND : DUMP (RETURN)

DISPLAY : A$ = "RANDOM NUMBERS"
X = 5
Y = 6
R = 94
CO = 995

2.4.12 TRACE

FORMAT : TRACE

PURPOSE : To display the number of the program line as it is being executed.

The TRACE command is entered before executing a program. A "window" will appear in the top right corner of the VIC's screen. The window will display program line numbers as they are executed. A maximum of six lines can be displayed in the window at any one time in the format: # (line number). The lines in the window will scroll up so that the last program line number executed will appear at the bottom of the list inside the window.

CAUTION: The window will overwrite anything that would have been displayed in its position on the screen. Therefore if you are using the BASIC code "INPUT", the place for the user to respond must be below the window containing the line numbers currently being executed.

The use of TRACE will slow down program execution but the display of line numbers may still appear too quickly to be able to follow. Hold the CTRL or SHIFT keys down to control the rate at which program lines are displayed on the VIC's screen. The lines will be displayed at a speed of approximately 2 lines per second. To cancel the TRACE command, see Section 2.4.14.

EXAMPLE : To display the program lines while a program is executing:

COMMAND : TRACE (RETURN)

TYPE : RUN (RETURN) and hold down the SHIFT key

DISPLAY : See figure 2-1.

Figure 2-1. A TRACE Display

2.4.13 STEP

FORMAT : STEP

PURPOSE : To halt the program after each program instruction and display the first line number of the next instruction.

If the STEP command is executed before running the program, each program instruction will be executed individually.

A "window" on the screen will contain the program lines associated with that instruction (to a maximum of six lines) and the first line of the next instruction. Pressing the SHIFT or CTRL key will cause the next instruction to be executed and the line number of the following instruction to be displayed. Holding either of these keys down will cause the program to execute continuously.

Pressing the STOP key will halt program execution. To terminate the STEP command see Section 2.4.14.

EXAMPLE : To execute a program by single stepping:

COMMAND : STEP (RETURN)

TYPE : RUN (RETURN)

RESULT : A “window” appears at the top right of the screen displaying the first line number of the first instruction preceded by #. Note that this line has not been executed. (See figure 2-2.)

ACTION : Press the SHIFT key.

RESULT : The first instruction of your program is executed and the first line number of the next instruction displayed in the window.

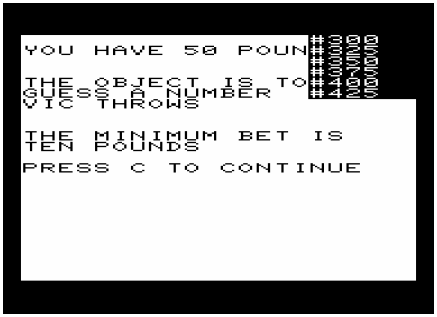

Figure 2-2. The Screen During Single-step

2.4.14 OFF

FORMAT : OFF

PURPOSE : To cancel the TRACE and STEP modes.

OFF cancels the commands TRACE and STEP. The “window” will disappear from the screen and the program will execute at normal speed.

EXAMPLE : To return to normal program execution following a STEP command as in the previous example:

COMMAND : OFF (RETURN)

TYPE : RUN (RETURN)

RESULT : The window has disappeared and the program executes normally.

2.4.15 PROG

FORMAT : PROG

PURPOSE : To change from EDIT mode to PROGRAM mode.

The PROG command gives access to twelve commands which are automatically assigned to the function keys when the cartridge is inserted into your VIC. (See section 2.2.2.)

When the PROGRAMMER’S AID cartridge is started up, you are in PROGRAM mode. If you have entered EDIT mode (see Section 2.3), to return to this mode type the word PROG or press the CTRL and F1 keys. The function keys are assigned with the following commands:

KEY 1, “LIST”
KEY 2, “MID$“
KEY 3, “RUN (RETURN)”
KEY 4, “LEFT$(“
KEY 5, “GOTO”
KEY 6, “RIGHT$(“
KEY 7, “INPUT”
KEY 8, “CHR$(“
KEY 9, “EDIT (RETURN)”
KEY 10, “GOSUB”
KEY 11, “RETURN (RETURN)”
KEY 12, “STR$(“

NOTE: (RETURN) indicates a built in carriage RETURN following the command. This appears on the screen as a reverse-field left-arrow. (See Section 2.4.9 to add a built in carriage RETURN.)

EXAMPLE : To enter PROGRAM mode from EDIT mode:

COMMAND : PROG (RETURN)

DISPLAY : PROG
READY

2.4.16 MERGE

FORMAT : MERGE “(Program name)”,(Device number)

PURPOSE : To load a previously saved program or subroutine and incorporate it into the program currently stored in the VIC’s memory.

The device number refers to the number of the peripheral on which the program or subroutine to be MERGEd is stored. The device number is 1 for a cassette unit and 8 for a disk unit. If not specified, the device number defaults to 1, i.e. cassette. The program can be specified in the same way as for the BASIC command LOAD. If a program name is not specified, the next program on the cassette or the first program on the diskette is MERGEd.

CAUTION: If line numbers are duplicated in both programs, the lines with the same number will follow each other, i.e. the program will be interspersed rather than appended. If this is likely to occur, use the RENUMBER command

(see Section 2.4.3) to renumber the program currently in memory so that you can effect a good MERGE.

EXAMPLE : To MERGE the program named "VIC 1" on the cassette unit with the program currently in memory:

COMMAND : MERGE "VIC 1",1 (RETURN)

DISPLAY : PRESS PLAY ON TAPE

ACTION : Press the PLAY button on the cassette deck.

DISPLAY : LOADING VIC 1
READY

RESULT : The two programs are now merged.

2.4.17 KILL

FORMAT : KILL

PURPOSE : To cancel the function of the PROGRAMMER'S AID CARTRIDGE.

When the command KILL is executed, the function of the PROGRAMMER'S AID CARTRIDGE will be cancelled, i.e. the VIC will behave as if the cartridge were not in place. PROGRAMMER'S AID slows down program execution speed due to such things as memorizing the execution position for the HELP command. It is recommended that you use the KILL command when the program is completed and free of errors.

NOTE: KILL leaves the assignement of the function keys unaltered, i.e. you may still press the function keys to display their assigned values.

To re-enter PROGRAMMER'S AID, carry out instructions outlined in Section 1.3 of this manual.

EXAMPLE : To cancel PROGRAMMER'S AID:

COMMAND : KILL (RETURN)

RESULT : PROGRAMMER'S AID is no longer in effect.

2.5 Special Editing Functions

PROGRAMMER'S AID gives you six special program editing functions which can be used by holding the CTRL key down and pressing a letter on the keyboard. They are as follows:

CTRL A Scrolls up a program list.
CTRL E Cancels quotes in insert mode.
CTRL L Blanks the characters to the right of the cursor on the screen line.
CTRL N Erases from the screen all characters in the program after the cursor.
CTRL Q Scrolls down a program list.
CTRL U Blanks the line of the screen on which the cursor is positioned.

They are all self-explanatory, except perhaps for CTRL E. This command is handy if you are inserting information between quotation marks on a program line. Use of CTRL E will permit you to move the cursor without displaying the cursor character.

NOTE: If you use one of the special editing functions to erase characters, it will only remove them from the screen. It does not change the program line(s) in the VIC's memory. In order to change the contents of memory, you must type in new characters and press RETURN or the lines must be deleted.

EXAMPLE : To erase all the characters on the line:

```
100 REM THIS IS SUPERFLUOUS
```

when the cursor is positioned on that line:

COMMAND : CTRL U

DISPLAY : A blank line

RESULT : Line 100 is removed from the screen, but the code is still in memory.

SECTION THREE

USING PROGRAMMER'S AID AS A TOOL

3.1 Introduction

This section will show you the stages of writing a complete BASIC program using PROGRAMMER'S AID. A sample program will be used together with instructions and illustrations designed to demonstrate the advantages PROGRAMMER'S AID gives in program writing, editing and debugging.

3.2 The BASIC Program

The BASIC program illustrated here is a simple random number "dice game" giving the player an initial stake and asking him to match a number between 1 and 6. The color and sound attributes of the VIC are used in only limited ways in order not to confuse the less experienced VIC BASIC programmer.

3.3 The Procedure

Insert the PROGRAMMER'S AID cartridge into the expansion port of your VIC with the power OFF. Switch the power on and type

SYS 28681 (RETURN)

Your VIC is now in PROGRAM mode. Carry out the instructions listed below to list and redefine the commands that have been automatically assigned to the function keys. Type:

KEY (RETURN)

Now type the letters:

CLR (RETURN)

Redefine the function keys to appear as below by moving to the proper location, typing the new characters on that line and pressing the RETURN key. To change the next command, clear the screen and repeat the process. (See Section 2.4.8.)

KEY 1, "LIST"
KEY 2, "FOR"
KEY 3, "RUN"
KEY 4, "NEXT"
KEY 5, "GOTO"
KEY 6, "G– A$:IFA$="
KEY 7, "INPUT"
KEY 8, "PRINT"
KEY 9, "IFA$="
KEY 10,"GOSUB"
KEY 11,"RETURN"
KEY 12,"POKE"

NOTE: G– *(obtained by typing G, holding the SHIFT key down and typing E) is the abbreviated form of the BASIC keyword "GET". For a comprehensive list of the abbreviated form of all BASIC keywords, see Appendix D of your VIC 20 Computer Guide.*

WARNING: Do not attempt to enter EDIT mode after redefining the function keys. This will replace the newly defined functions with the original information assigned by the PROGRAMMER'S AID CARTRIDGE. All EDIT commands that are required can be typed in full.

3.4 Entering the Program

Begin numbering your program lines at line 50 in increments of 25 lines by typing the following:

AUTO 50,25 (RETURN)

The VIC will display the line number 50 and await your input. (See Section 2.2.2.) As you press RETURN at the end of each line of code, the next line number will be displayed to the screen.

Type in the program below using the function keys. They are shown enclosed in square brackets, e.g. [F4]. When you have finished entering the program, press RETURN immediately following the next automatically displayed line number to cancel the AUTO command.

```
50   [F12] 36879,30
75   N=50
100  [F8] "(SHIFT/CLR HOME)":[F8] TAB (5)
     "VIC DICE GAME"
125  [F8] "PRESS P TO PROCEED"
150  [F6] "" THEN 150
175  [F9] "P" THEN 250
200  [F5] 150
225  REM **GAME RULES
250  [F8] "(SHIFT/CLR HOME) YOU HAVE
     "N" POUNDS":[F8]:[F8]
275  [F8] "THE OBJECT IS TO"
300  [F8] "GUESS A NUMBER"
325  [F8] "VIC THROWS"
350  [F8] "(CRSR DOWN/CRSR DOWN) THE
     MINIMUM BET IS"
375  [F8] "TEN POUNDS"
400  [F8]:[F8] "PRESS C TO CONTINUE"
425  [F6] "" THEN 425
450  [F9] "C"THEN 525
475  [F5] 425
500  REM **RANDOM NUMBER
525  [F8] "(SHIFT/CLR HOME)":A%=RND(1)
     *6+1
550  [F7] "WHAT IS YOUR BET";B
575  IF B<10ORB>N THEN [F10] 925:[F5]525
600  [F8] "CRSR DOWN/CRSR DOWN)
     PLEASE SELECT NUMBER"
625  [F8] "1-6"
650  [F6] "" THE 650
675  IF VAL(A$)<1 OR VAL(A$)>5 THEN[F8]
     "(SHIFT/CRSR DOWN/SHIFT CRSR
     DOWN/ SHIFT/CRSR DOWN/SHIFT/
     CRSR DOWN/SHIFT/CRSR DOWN/
     SHIFT/CRSR DOWN)":[F5] 600
700  X=VAL(A$):[F8]"(CRSR DOWN/CRSR
     DOWN)YOUR NUMBER WAS"X
725  [F2]Q=1 TO 1000:[F4]
750  [F8]"(CRSR DOWN/CRSR DOWN)
     VIC'S NUMBER WAS"A%
775  IF X=A%THEN N=N+B*2:[F8]
     "(CRSR DOWN/CRSR DOWN)YOU WIN":
     [F10] 1250:[F5] 825
800  [F8]"(CRSR DOWN/CRSR DOWN)VIC
     WINS": [F2]Q=1 TO 2500:[F4]
825  N=N-B
850  IF N<10 THEN 1050
875  IF N>100 THEN 1200
900  [F5] 1050
925  [F8]"(SHIFT/CLR HOME)"
950  [F8]"(CLR/HOME)NUMBER OUT OF
     RANGE"
975  [F2] Z=1 TO 1000: [F4]Z:[F5]525
1000 REM**ERROR ROUTINE
1025 [F11]
1050 REM**GAME OVER
1075 REM**ROUTINE
1100 [F8]"(SHIFT/CLR HOME)"
1125 [F2]S=1 TO 5:[F8]"(CLR HOME/CRSR
     DOWN/CRSR DOWN)YOU'RE BROKE!!":
     [F2]Q=1 TO 500: [F4]Q
1150 [F8]"(CLR HOME/CRSR DOWN/CRSR
     DOWN/CTRL RVS ON)YOU'RE
     BROKE!!":[F2]Q=1 TO 150:[F4]Q,S:RUN
1175 [F8]"(SHIFT/CLR HOME)"
1200 [F2]S=1 TO 10:[F8]"(CLR HOME/
     CRSR DOWN/CRSR DOWN)VIC'S
     BROKE!!":[F2]Q=1 TO 2500:[F4]Q:
     RUN
1225 REM**COLOR AND SOUND
1250 REM**SOUND ROUTINES
1275 [F12]36878,15:[F2]L=148 TO 220 STEP
     .7:[F12]36876,L:[F4]L
1300 [F2]W=1 TO 5:[F12]36879,26:[F2]FF=1
     TO 250:[F4]FF
1325 [F12]36879,30:[F2]FF=1 TO 250:[F4]FF,W
1350 [F2]L=200 TO 128 STEP-1:[F12]36876,L:
     [F4]L
1375 [F12]36878,0:[F12]36876,0
1400 [F11]
```

To run the program, type the following:

[F3] (RETURN)

If you have typed the program as it is indicated above, the screen will appear as shown in Figure 3-1.

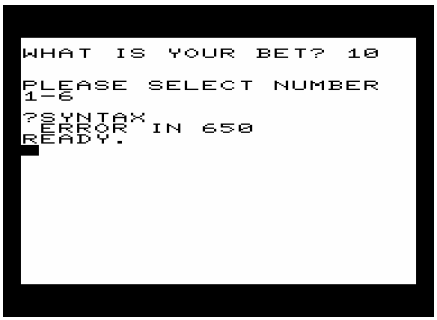

Figure 3-1. The First Error

3.5 Locating the Error

Something has happened to prevent the program executing successfully. To discover where the error occurred type:

HELP (RETURN)

The screen will appear as shown in Figure 3-2.

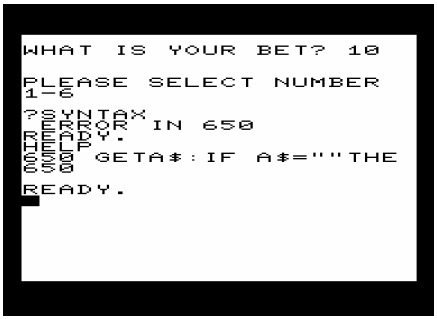

Figure 3-2. The First Error Highlighted

You will notice that the quote mark is highlighted in reverse-field in line 650. This indicates the approximate position of the error. You will no doubt have noticed that "THE" should have been the BASIC code THEN. Correct this error and run the program again.

Now another problem is obvious. The player appears to be out of money when he shouldn't be. To discover how much money he does have, press the STOP key after the BROKE message has flashed a few times, and type:

DUMP (RETURN)

The screen will appear as shown in Figure 3-3.

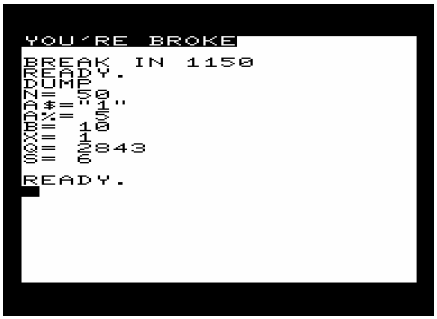

Figure 3-3. Dump of Variables

The program is obviously not functioning correctly. You can see what has gone wrong by using the STEP command (see Section 2.4.12) to execute the program instruction by instruction. Because the BASIC code INPUT has been used in this program near the top of the screen, you must change line 550 to read as follows:

550 [F8]" (SHIFT/CLR HOME/CRSR DOWN/ CRSR DOWN/CRSR DOWN/CRSR DOWN/CRSR DOWN/CRSR DOWN/ CRSR DOWN/CRSR DOWN)":[F7] "WHAT IS YOUR BET";B

The program is still not running correctly because the user runs out of money even when betting within his limits. To discover the reason why this is happening, type the following:

STEP (RETURN)
RUN (RETURN)

When you reach the instruction which displays whether VIC wins or you win, check the number of the line that appears at the bottom of the window. During the first pass through the program, providing the bet did not exceed 40 pounds, the next line should be the point at which the game restarts, i.e. line 250. In fact, line 1050 is displayed in the window. The fault lies in line 900. It reads GOTO 1050 when it should read GOTO 250. Change the code and the program should execute normally.

The STEP and TRACE commands slow down program execution (see Sections 2.4.12 and 2.4.13), so if you use either of these commands initially, type OFF (see Section 2.4.14) to enable the program to run at the correct speed. The command KILL will cancel PROGRAMMER'S AID enabling you to run the program at normal speed. (See Section 2.4.17.)

3.6 Summary

Here is a summary of the steps to follow to write programs with PROGRAMMER'S AID:

1. Initialize the PROGRAMMER'S AID CARTRIDGE.
2. Decide whether the commands automatically assigned to the function keys will be helpful to you. If not, use the command KEY to display and reassign your own commands to these keys.
3. Use the AUTO commands to automatically display program line numbers.
4. Enter your program.
5. If you wish to use a subroutine or another program which was stored previously, use the MERGE command to combine it with your program.
6. Use the DELETE command to remove lines from memory if required.
7. Use the RENUMBER command if, after deleting lines, you wish to create a wholely sequential program listing.
8. RUN the program.
9. If an error occurs, use the HELP command to display the line where the error occurred.
10. Correct the error and RUN the program again.
11. Use the DUMP command if you wish to view the values of all non-array variables.
12. If you wish to display the line(s) where a particular character, character string or BASIC code occurred in your program, use the command FIND.
13. Use the CHANGE command if you wish to alter any code(s) in your program.
14. If an error occurs because of an incorrect GOTO or GOSUB entry, use the STEP or TRACE command to view the program lines as they are being executed.
15. Correct any errors and use the command OFF to cancel the STEP or TRACE mode.
16. When everything works O.K., use the KILL command to cancel the function of PROGRAMMER'S AID.
17. Remember to save your corrected program.

INDEX

SUMMARY OF PROGRAMMER'S AID COMMANDS

The commands are executed when the RETURN key is pressed.

(sln) indicates the start line number.
(fln) indicates the finish line number.
(char) indicates a character, BASIC code, or character string.

FUNCTION KEY ASSIGNMENTS

	PROGRAM MODE	*EDIT MODE*
KEY 1,	"LIST"	"LIST"
KEY 2,	"MID$("	"AUTO"
KEY 3,	"RUN (RETURN)"	"RUN (RETURN)"
KEY 4,	"LEFT$("	"DELETE"
KEY 5,	"GOTO"	"FIND"
KEY 6,	"RIGHT$("	"CHANGE"
KEY 7,	"INPUT"	"TRACE (RETURN)
KEY 8,	"CHR$("	"STEP (RETURN)
KEY 9,	"EDIT (RETURN)"	"PROG (RETURN)"
KEY 10,	"GOSUB"	"RENUMBER"
KEY 11,	"RETURN (RETURN)"	"MERGE"
KEY 12,	"STR$("	"OFF (RETURN)"

SPECIAL EDITING FUNCTIONS

CTRL A	Scrolls up a program list
CTRL E	Cancels quotes in insert mode
CTRL L	Erases all characters after the cursor on the same line
CTRL N	Erases all characters in the program after the cursor
CTRL Q	Scrolls down a program list
CTRL U	Erases all the characters on the line containing the cursor

To start PROGRAMMER'S AID, type SYS 28681 or SYS 7*4096+9 and press the RETURN key.

commodore