CERN-Data Handling Division
DD/79/8
October 1979
J.M. Howie

CERN LIBRARIES, GENEVA

CM-P00059820

GD3 - A SYSTEM FOR TWO DIMENSIONAL
COMPUTER GRAPHICS

(Submitted to Computers and Graphics)

DD-jt

GD3 - A SYSTEM FOR TWO DIMENSIONAL
COMPUTER GRAPHICS

J.M. Howie
CERN, Geneva, Switzerland

ABSTRACT

GD3 is a system for graphic input and output, which has been developed at CERN over the past several years primarily for the laboratory's scientific community. An important feature of the system is the production of an intermediate data file called the "display file", which allows a user's program to be truly device independent. The system is written largely in FORTRAN and although the main implementation has been on CDC 7000/6000 series computers at CERN, it has also been implemented on other machines both within CERN and at laboratories outside CERN.

The main facilities available to the user are described, including the system's interactive capabilities. Use has also been made of GD3 to provide the graphics part of other special user oriented packages developed at CERN.

1. Introduction

The GD3 system (Graphical Display System) has been developed at CERN (European Organisation for Nuclear Research) over the past several years to provide the CERN computer community with a tool for producing graphic output. The system provides the user with an extensive set of FORTRAN callable routines which the user can call in a selective way to produce his desired picture or set of pictures. Because of the nature of the work done at CERN, the package tends to be more oriented towards the needs of physicists and engineers rather than to other fields of work.

To produce an output for a specific device always involves a two-step process. First of all the GD3 routines generate an output file (called the display file) which represents the data in a device independent form and then this file is treated by another program (called the interpreter) to produce an output appropriate for the particular device. This general philosophy is shown in Fig. 1. It means, of course, that a separate interpreter is needed for each particular device which is going to be supported, but in practice this is not a major drawback, since the greater part of the code in an interpreter is still device independent and therefore common to all interpreters. The big advantage of having the intermediate file is that the user does not have to make any change in his program in going from one device to another, so he can very well use a graphics terminal or the line printer to obtain a quick look at this output whilst debugging a program and then when fully satisfied, direct the display file to a plotter or microfilm device to obtain his final output.

The output devices which are currently supported at CERN are line printer, a Calcomp plotter, a Benson plotter and Tektronix 4006, 4012 and 4014 graphic display terminals. In the past, a Ferranti refresh display and microfilm output device were also supported, but these latter two devices are no longer in operation at CERN.

The GD3 system also allows users to write interactive programs for a graphics display terminal and it can accept input from an alphanumeric keyboard, a function keyboard, a light pen or a tracking cross, depending on what input devices are available on the display terminal.

Most of the package is written in FORTRAN and the main implementation at CERN has been on the CDC 7600 and 6000 series computers. However it is not a difficult job to transfer the system to other computers and this has been done both within CERN and in laboratories outside CERN. The most recent implementation has been on an IBM 370/168 in the computer centre at CERN.

The documentation for the program consists of a set of internal CERN reports (1, 2, 3).

2. Coordinate Systems

All pictures to be drawn by GD3 routines are considered to be two dimensional. Two basic coordinate systems are used in the program, referred to as the "DISPLAY" coordinates and the "USER" coordinates. The "DISPLAY" coordinates (in both x and y directions) are constrained to lie with the range $0 \rightarrow 1023$, and the maximum "DISPLAY" area PQRS (see Figure 2) is the area which is linearly mapped by a GD3 interpreter onto the plot area of the output device. The "USER" coordinate system is a coordinate system defined by the user himself and he must specify maximum and minimum values in both x and y directions. Thus, in Fig. 2 the user defines the area ABCD by specifying coordinates (XMIN, YMIN) for the point A and coordinates (XMAX, YMAX) for the point C.

A mapping is then defined (under user control) which will map the "USER" area ABCD onto a sub-region WXYZ of the total "DISPLAY" area. At any time within his program the user can choose to work in either the "USER" coordinate system or the "DISPLAY" system. He can also choose to redefine the "USER" coordinate system and the sub-region in the "DISPLAY" area onto which it will be mapped. In that way he can construct separate independent pictures within the "DISPLAY" area. The resolution which is kept within the "USER" coordinates when they are recorded onto the display file is about 1 in 2^{17} .

3. The GD3 routines

a) General comments

All GD3 routines and common blocks within the routines have names beginning with the two letters TV. This convention is very useful for avoiding duplicate names and also for track back information when GD3 detects an error.

An important concept in GD3 is that routines can be called with a variable number of arguments. This, in general, simplifies the calling sequences and makes the routines easier to use. However, it is non-standard FORTRAN and it means that most GD3 routines have to first call a particular routine (called NOARG), which returns an argument whose value is the number of arguments that the GD3 routine was called with. Therefore to implement GD3 on a particular computer it is clearly desirable that the machine code assembly of FORTRAN routines is done in such a way that the argument list is easily accessible.

b) Some basic routines

A list of the main routines in the GD3 package is given in Appendix I along with a brief description of the function of each routine. Within the package there is a wide choice of both high level and low level routines. At the high level end the user can plot graphs (TVGRPH), draw histograms (TVHIST) and make contour plots (TVCONT) by giving the minimum amount of information. The routines will automatically compute a sensible scaling for the given function, draw axes with the necessary labelling and text annotation, and plot the function. All parameters needed at each stage of a plot have carefully chosen default values built into the package, which the user does not have to concern himself with. However it is very easy for the user to request fairly standard alternatives, such as smoothing of the data before it is plotted, or to introduce logarithmic or polar coordinates. For more detailed variations such as changing the position or length of the tick marks on an axis, or plotting a curve with a dotted line instead of a continuous one, the user normally has to change the value of a parameter in a common block, or else give additional arguments in the calling sequence of the routine.

For the more sophisticated user who prefers to build up his own pictures, the basic primitives for plotting points (TVPLOT), drawing lines (TVDRAW) or outputting text (TVTEXT) exist, as well as facilities for creating sub-pictures (symbols) which can be inserted in different parts of the main picture.

c) Hardware and software characters

Another important aspect of the package is that it provides two forms of text characters. The first form, referred to as hardware characters, makes use of the fact that most graphical output devices have some form of hardware character generator which produces a limited set of characters - usually the FORTRAN character set plus a few special characters. In GD3 there are assumed to be four possible sizes for these characters, and two orientations (horizontal and vertical). However the actual availability of these facilities varies from device to device and, for each device, this information has to be included in the device dependent part of the appropriate interpreter.

A much more flexible form is the software character set which is generated at the interpreter stage. This is completely free in its sizes and angles of orientation, and also has a much greater number of available characters (four alphabets of 63 characters each and upper and lower case for each alphabet). It also has the advantage that it is independent of the device to which the pictures are output and a size of 1.0 will give exactly the same number of characters per line (85) and the same number of lines per page (64) on every device.

d) Interactive facilities

For the user with an appropriate graphical terminal, GD3 can be used in an on-line mode. In this mode the display file becomes simply a buffer, which when full is passed directly to the interpreter, which then converts the graphic data to the required form for the display. To perform graphic input, GD3 uses basically two routines. First the routine TVINA, which writes out a message onto the screen to indicate to the user what action is required of him, and which also enables the input devices that the user will be permitted to use (the other devices are software disabled). The second routine, TVIN, is called to collect the information input by the user, e.g, a text string from the alphanumeric keyboard or coordinates of points digitised with the tracking cross.

Other routines are available which provide the following facilities:

- i) to name part of a picture
- ii) to delete a named item from the screen
- iii) to make a copy on a separate file of the current contents of the screen (for hard copy)
- iv) to decode free format input from the keyboard.

e) Some examples

Figures 3, 4 and 5 show some examples of pictures produced with GD3.

4. The Display File

The structure of the display file is very straightforward. First there is a header record containing the text "DISPLAY FILE", which is used to identify the file and then a set of data records for each picture (a frame in GD3 terminology). Records are of fixed length except for the last record of each frame, and are built up as a series of characters. On the CDC machines the characters are 6 bits long, but on the ILM/370 a character is defined to have 8 bits. Special characters are used at the beginning of a record to define whether it is the first record, last record or some other record within the frame, but apart from these characters the rest of the characters in each record are regarded as forming a continuous sequence without any breaks, even if this means splitting a data item consisting of more than one character over two records. The conventions defining the various records are illustrated in Fig. 6.

The sequence of characters representing the data consists of a series of control characters, where each control character is followed by a certain number of characters of information. For example the line from (x_1, y_1) to (x_2, y_2) would be output as a base move (control character 1) followed by six characters of coordinate information, three for x_1 , and three for y_1 ; then the vector draw (control character 3) is followed by the six characters (x_2, y_2) . In most cases the number of characters following a control character is fixed and only in the case of text output (control character 16) is it variable. In this case a special end of text marker is used to indicate when the next control character will occur. The end of data for a frame is signalled by the control character 0.

5. Further Developments

With a package like GD3, which is being used regularly by many people, there will always be cases where some user finds it limited or would like some additional facilities easily available. This usually implies the addition of other high level routines to the system. Over the last few years some additions have been made, but often it is not clear whether the facility in question should be added to the basic system or whether it should be used as a separate program which makes use of GD3. At CERN there are certainly two other important program packages which make use of GD3 for their basic graphic output.

The first of these is the SIGMA system (4) which is a programming language that has been developed specially for interactive scientific computing, and the other package is HPLOT (5), which is an aid to physicists for formatting their histogram and scatter plot outputs into forms suitable for talks and publications.

Routines for three dimensional graphics, which include calls to the GD3 package have been developed and hence some 3-D graphics facilities are available to the GD3 user.

6. Acknowledgements

The original concept of the GD3 system was due to P. Blackall and A. Yule in 1967, with A. Yule carrying out most of the initial implementation. In more recent years though, R. Miller has been responsible for all additions and improvements which have been made to the system.

REFERENCES

1. R. Miller, GD3 Long Write-up, CERN Computer Centre, Program Library number J510 (1976).
2. C.S. Curran and J.M. Howie, GD3 User Guide (CDC), CERN-Data Handling Division internal report DD/US/8 (1979).
3. C.S. Curran and J.M. Howie, GD3 User Guide (IBM), CERN-Data Handling Division internal report DD/US/27 (1979).
4. J. Reinfelds and C.E. Vandoni, SIGMA 76, in Information Processing 77 (editor B. Gilchrist), p. 963.
5. H. Watkins, HPLOT Users Guide, CERN-Data Handling Division DD/76/19(1976)

GD3 routines in
a library

Figure 1: General representation of the GD3 system

Figure 2: GD3 Coordinate systems

Figure 3: A plot using GD3; example 1

GD3

TEST PROGRAM

Figure 4: A plot using GD3; example 2

Figure 5: A plot using GD3; example 3

Figure 6: The display file

APPENDIX I

List of the main routines in the GD3 package

An * in front of the routine name means that it is coded in machine code.
All other routines are written in FORTRAN.

1. Administrative routines

TVBGN	Initialize the GD3 system
*TVCHAR }	Routines for packing and unpacking characters -
*TVCKCH }	mainly for operations on the display file
TVEND	Terminate the display file
TVHELP	Output error messages
TVNEXT	Terminate a picture and start the next one
*TVTRCE	To provide error traceback for TVHELP

2. Basic routines

TVARC	To draw circles and arcs
TVAX	To draw an axis with tick marks, labels and a title
TVCLIP	To clip lines that extend beyond the drawing area
TVCONT	To make a contour plot of a function
TVCSET	To change pens
TVDRAW	Vector drawing
TVDDRW	Vector drawing with dotted lines
TVEBAR	Plot points with error bars
TVFUNC	Plot graph of a given function
TVGRID	To draw a grid on a plot
TVGRPH	To draw a graph
TVHIST	To plot a histogram
TVLBL	Put numeric labels on an axis
TVMAP	Choose suitable range for "USER" coordinates
TVMODE	Used to set various modes such as which coordinate system is in operation or whether logarithmic scaling is in force, etc.
TMSS	Output of software characters
TVNUMB	Output a number as a text string
TVPLOT	Plot points or characters
TVRNG	Set range for "USER" or "DISPLAY" coordinates
TVSET	Set certain types of output such as size of hardware characters, intensity level, etc.
TVSHAD	Shade a polygonal area
TVTEXT	Output hardware characters.

3. Sub-picture routines

TVSCCL	Delete sub-picture from list
TVSDEF	Start definition of a sub-picture
TVSEND	End definition of a sub-picture
TVSUB	Draw a sub-picture.

4. Interactive routines

TVCOPY	Switch on hard copy mode
TVDELT	Delete an item from display
TVFREE	Decode free format input
TVIN	Get interactive input
*TVINA	Enable interactive input
TVNAME	Name an item to be displayed.