3D PRECISION

BY ARNO HASNAES

DOCUMENTATION BY

Dr. HELMUT AIGNER & Dr. ANDY AIGNER



337

N T E N

1.

4 25.

```
INTRODUCTION
 1.1
           What Does 3D PRECISION Do for You?
 1.2
           Definitions
 1.3
           Making a Backup Copy - What Goes Where
 2.
        USING THE 3D OBJECT EDITOR
 2.1
           Screen Windows
 2.2
           Control Keys
 2.3
           An Example
 2.3.1
             DEFINE
 2.3.2
             INSERT
 2.3.3
             CURSOR
 2.3.4
            OBJECT
 2.3.5
            REMOVE
 2.4
          Further Menu Commands
 2.4.1
            FILE I/O
 2.4.2
            OBJECT CONTROL
 2.4.3
            PROGRAM CONTROL
        PROGRAMMING WITH THE BASIC TOOLKIT
Э.
3.1
          How to Go About It
3.2
          The New Keywords
4.
       PROGRAMMING WITH THE ASSEMBLER TOOLKIT
4.1
          How to Go About It
4.2
          Assembler Routines
4.3
          Assembler Definitions
4.3.1
            Camera definition
4.3.2
            Camera and object definition
4.3.3
            Object definition
4.3.4
            Vertex table
4.3.5
            Point table
4.3.6
           Line table
           Plot table
4.3.8
           Other parameters
Extra error codes
4.3.9
5.
       INDEX
```

+3:

.1

1. INTRODUCTION

What Does 3D Precision Do for You?

Thank you for buying 3D PRECISION.

3D PRECISION is a suite of programs which permits the creation and rapid manipulation (i.e. size changes as well as linear and rotary movement in space) of three-dimensional objects in any colour or combination of colours on the QL and Thor computers. It comprises:

3D Editor (treated in detail in Chapter 2) that will let you define, test, edit and save objects under menu and cursor control J if your computer has at least some extra memory;

- a SuperBASIC Toolkit (Chapter 3), which adds dozens of new commands to the SuperBASIC language, allowing you easy manipulation of

objects if you are at all proficient in that language;

- an Assembler Toolkit (Chapter 4), offering many subroutines for superfast manipulation of objects - all accessible to your programs written in assembler code (you supply the assembly-language know-how).

Your objects, together with an imaginary camera which views them, are placed in a three-dimensional 'world' with integer x, y co-ordinates ranging from -32768 to 32767. (In practice you may be slightly more restricted, as no object must be more than some 32000 'world units' away from the camera. Lines should also be shorter than 16000 units and object dimensions less than 15000 units, or you may get lines going beyond their intended endpoints or deformed or disappearing objects.)

There are a few further restrictions:

- Objects are of the wire-frame type, i.e. composed exclusively of lines and points. They are viewed as though anything but the lines and points were transparent; in other words, there is no hidden line removal (unless you provide for it in your own program). You may, however, define for each object a cut-off plane ('backplane') parallel to the screen, which will make everything on its off-camera side disappear.
- Curves on the surface of objects (e.g. on cylinders, cones, spheres) must be approximated by line segments. (The SuperBASIC CIRCLE command does nothing else; but in 3D PRECISION you have the option of specifying the number of corners to be used in the approximation.)

You may save individual objects or whole worlds on microdrive/disk and reload them singly or in combination.

You may also reconfigure nearly all of the default values that the program suite uses on start-up.

Please do not be put off by the technical appearance of this manual. Many parts of it are intended for use by BASIC and machine code programmers - if all you want to do is design and manipulate objects in 2D or 3D, ignore sections 3 and 4 of the manual.

while on the subject of the manual, Digital Precision is icknowledge the skill and expertise of Dr Helmut Aigner, who designed and wrote this treatise virtually single-handedly. 3D Precision is one of those rare programs whose production involved technical skills from nore than one country (Denmark, Austria & the UK). So did Professional istrologer, incidentally - Germany, India, Japan & the UK!

In all, your master medium(s) comprises the following files:

Part 1 DEFAULTS DEF3D

BOOT2

CLONE BIN

DEFTV8 DEF3D

DEFTV4_DEF3D

DEFBW_DEF3D

DEFCM DEF3D

GL PRT

GL BIN

PLOTTER BIN

3D_TOOLKIT_BIN RUNTIME_EXTS 3D_TOOLKITASM_BIN

HELPFILE

Part 2 3D_PRECISION_BIN

Part 3 BASICDEMO

BASICDEMOBOOT BASICEXT

BASICEXT_BIN
BASICEXTDEMO
BASICEXTDEMOBOOT
ASMDEMO

ASMDEMONDV_BIN ASMDEMOFLP_BIN ASMDEMONDVBOOT

ASMDEMOFLPBOOT

DRAWTYPE

DRAWTYPE_BIN
BASICDEMO ALL3D
BASICDEMO1 CAM3D
BASICDEMO2 OBJ3D
BASICDEMO3 OBJ3D
BASICDEMO4 OBJ3D
BASICDEMO5 OBJ3D
SPACE OBJ3D
LANDER_OBJ3D
STATION OBJ3D

Defaults for 3D PRECISION BIN preset to the values of DEFCM DEF3D. LRUN this to start Editor. This is LRUN by BOOT and will EXECUTE 3D_PRECISION_BIN. EXEC this to make a working copy (see Section 1.3). Copy this to DEFAULTS DEF3D if you use a TV in Mode 8 for display. Copy this to DEFAULTS_DEF3D if you use a TV in Mode 4 for display. Copy this to DEFAULTS DEF3D if you use a black-and-white monitor for display. Copy this to DEFAULTS DEF3D if you use a colour monitor for display. A plotter driver for plotters that use the GL plotter language.

(set plotter for ISO, A4 paper)
GL_PRT compiled by TURBO.
Same as PLOTTER BIN.
This is called by 3D_PRECISION when PLOT PIC. is used.
The BASIC Toolkit (21128 bytes).
Toolkit used by BOOT.
The Toolkit without the BASIC definitions (13472 bytes).
The Help file; you may add notes,

using an Editor, if you like.

The 3D Object Editor.

TA BASIC program that demonstrates the use of the BASIC Toolkit. LRUN this to start BASICDEMO. An Assembler file (METACOMCO) which demonstrates how to make further BASIC extensions. It also shows the use of DRAWTYPE. BASICEXT in machine code. 33: A BASIC prog to demo BASICEXT. LRUN this to see a demonstration. An Assembler file (METACOMCO) which shows how to make Assembler progs. ASMDEMO in machine code; uses MDV2_. ASMDEMO in machine code; uses FLP1_. LRUN this to see ASMDEMO (if on microdrive) run. LRUN this to see ASMDEMO (if on floppy disk) run. Draw routine for making points in different sizes. DRAWTYPE in machine code. Objects for demonstrations.

2 Definitions

all the following chapters you will come across the following erms:

A WORLD is a group of three-dimensional objects (or a single object if you insist) that can be resident in memory at the same time and saved and loaded as a whole. The definition of a world also includes the location of the three axes and of the camera.

An OBJECT is any combination of lines and points that you define as belonging together. You may create or place an object anywhere in a world, move it, enlarge or reduce it, rotate it, recolour it, and delete it.

The CAMERA is invisible but behaves like an object in many respects. It always exists in any given world, but it can be moved and rotated. This will produce an apparent motion or size change of the objects on the screen: moving or turning the camera in any direction will make the whole world (i.e. all objects in it) appear to move or turn in the opposite direction. This is of course as it should be but still takes some getting used to.

CO-ORDINATES are a set of three numbers (named x, y and z) used to describe locations in three-dimensional space. In the WORLD CO-ORDINATES used in 3D PRECISION, the greater the x co-ordinate of a point, the farther right it will be located on the screen. The greater the y co-ordinate of a point, the higher it will be located on the screen - at least if you forgo rotating the camera about a line parallel to the z axis. Increasing the z co-ordinate of a point will have no apparent effect on the screen (except for slight changes caused by perspective) until you rotate the object or the camera.

The GRID is an invisible pattern of gridpoints in the world which have the same mutual distance in all directions. This distance can be set by you, or you may accept the default; you may then move the cursor to the nearest gridpoint at any time.

A POINT is a dot, defined by a triplet of x, y and z co-ordinates (e.g. 5,-10,20) and drawn on the screen as part of an object.

A LINE is, strictly speaking, a line segment (as in SuperBASIC). It is defined by two triplets of co-ordinates for its endpoints and drawn on the screen as part of an object. The endpoints of a line are not considered to be POINTS as defined above.

An AXIS is one of three invisible lines defined in every world. The endpoints of the x axis have co-ordinates (-32768,0,0) and (32767,0,0), of the y axis have co-ordinates (0,-32768,0) and (0,32767,0) and of the z axis have co-ordinates (0,0,-32768) and (0,0,32767).

A VERTEX (plural: vertices) is an invisible point which may serve as a location for a visible point, for one of the two end points of a line, or for the intersection of several lines.

ample: A wire-frame cube consists of 12 lines, 8 vertices and no ints. (Remember: points must be isolated, not part of a line.)

The CENTRE of an object is a reference point from which all the vertices of the object are calculated. If you change the centre of an object, the whole object is recalculated. If you rotate an object, the axis of rotation (parallel to the x, y or z axis as specified by you) passes through its centre. All rotation angles must be expressed in degrees, not in radians as in SuperBASIC.

1.3 Making a Backup Copy - What Goes Where

Before you run any of the programs, you MUST make a backup copy by running the CLONE program. Murphy's Law is insistent on loss of files happening to YOU sooner or later if you fail to do this.

Follow the instructions given here and on the screen. Make sure that your destination medium is not write-protected and that it contains no files bearing any of the names listed in Section 1.1 (they would be overwritten) - if you are using cartridges, they should be entirely blank as there is no room on them for anything other than the 3D Precision files.

The clone program has be designed to be able to make a copy from (and/or to) disk or cartridge. In order to do this the copying has been split into 3 parts: each part can fit onto a cartridge that gives at least 212 sectors on formatting (or a 212 sector ramdisk). If you are using ramdisk as an intermediate stage for copying to floppy disk, it is most convenient to format just one ramdisk, but with at least 636 sectors so it can hold all three parts of the system. If you are using ramdisk as an intermediate stage for copying to cartridge, it is best to format 3 ramdisks (ram1, ram2 and ram3) with at least 212 sectors each.

If you have the Supertoolkit command WCOPY available you can bypass our backing up process and do a direct WCOPY: there is no messy copy protection system waiting to cause trouble for you!

First ensure that you have formatted the target medium(s).

The clone program is invoked by entering:

EXEC device CLONE BIN

where device is MDV1_, FLP1_ or whatever.

Press CTRL + C if on the QL so that the onscreen cursor flashes.

Now you are asked to enter the source (copy from), target (copy to) and the drive which is ultimately going to hold the copy (copy to run from) for part 1 of the copying.

Below are some examples of what you might enter in repay to the questions for part 1:

WHAT YOU HAVE AVAILABLE: 1 DRIVE 2 DRIVES 1 MDV 2 MDVS Copy from flpl flpl mdvl mdv1 Copy to raml flp2 raml mdv2 Copy to run from flpl flp1 mdv1

Note how ramdisk may be used as an 'intermediate' storage space if you do have only one disk drive or one microdrive (as opposed to one microcartridge!).

If you are making a copy to run from disk, repeat these answers for parts 2 and 3.

If you are making a copy to run from cartridge, the corresponding enswers for both parts 2 and 3 will be:

Copy from

mdv1 mdv1

Copy to

ram1_ mdv2_

Copy to run from

mdv2 mdv2

937

lote the change between part 1 and parts 2/3 for microdrives. It is ecessary because cartridge 1 is intended to be run from drive 1 while oth the others are intended to be run from drive 2.

ow you are asked to enter the name of the default file you want to se:

Filename For use with

DEFCM COLOUR MONITOR
DEFBW MONOCHROME MONITOR
DEFTV8 TV 8 COLOUR MODE

DEFTV4 TV 4 COLOUR MODE (MORE ROOM IN WINDOWS)

nsure the source medium (if the source is cartridge, this means artridge 1) is in the drive you specified as the source drive for art 1. Ensure that it is not write protected, as we will be adjusting he default file on it. Press a key, and the DEFAULT DEF3D file will ow be set to the one you have just selected.

nce you have answered the above questions the copying process will tart, pausing at the start of each part, where you will be asked to ress a key when ready. Before you answer each part you must ensure nat the source and target media appropriate for that part are present a the drives you specified.

ne copying may take a while - especially on cartridges. Pray be itient.

nen the backup is completed, put the master medium away in a safe lace and use only the backup medium (except for making a new backup something nasty happens to your working copy).

se the following instructions to run the programs.

run 3D PRECISION BIN enter:

LRUN FLP1 BOOT

if on microdrive, insert cartridge 1 in MDV1 and cartridge 2 in V2_, and enter:

LRUN MDV1 BOOT

To run the demonstrations enter:

LRUN FLP1 BASICDEMOBOOT

or

LRUN FLP1 BASICEXTDEMOBOOT

or

LRUN FLP1 ASMDEMOFLPBOOT

or

LRUN FLP1 BASICDEMOBOOT

or if on microdrive, insert cartridge 1 in MDV1 $_$ and cartridge 3 in MDV2 $_$, and enter:

_LRUN MDV2_BASICDEMOBOOT

or

LRUN MDV2 BASICEXTDEMOBOOT

or

LRUN MDV2_ASMDEMOMDVBOOT

or

LRUN MDV2_BASICDEMOBOOT

The computer should be reset before running a demonstration.

You have noticed that there is no copy protection on 3D PRECISION. Against the trend on top quality products, we have taken a gamble on your honesty. Please don't let us down. Only if you do distribute copies do we have to track you down and take — or get the authorities to take — whatever action is required under civil and criminal law. We still offer rewards for information which helps us to track down pirates. Please realise we would far prefer to be able to spend all our time more productively!

If you somehow find yourself in the possession of a pirated copy of 3D PRECISION, then please do the honest thing: order a legitimate copy. This will entitle you to software and documentation upgrades, and support from us if you run into problems.

Please note that we cannot undertake to answer technical queries on 3D PRECISION over the phone: please put your queries into writing. Often you will find that the very act of putting ink to paper (or fingers to keyboard) concentrates the mind wonderfully, and the "problem" vanishes. About 90% of telephone calls to us result from either not making a backup copy (if you screw up while making a backup, don't call us - just return the original medium to us with an SAE - we can't unscramble cartridges over the phone as yet!) or from not reading the manual (have you used the index or contents pages?). We are on your side - so please help us to help you.

If we have anything to add to this manual, you will find it on a file called UPDATES_DOC, which is intended to be read with Quill. This file will be present on cartridge 3 if you have ordered 3D PRECISION on cartridge. Please read it.

USING THE 3D OBJECT EDITOR

.1 Screen Windows

run the editor, insert cartridges 1 and 2 in the appropriate slots, the disk in slot 1, and type:

LRUN FLP1_BOOT or LRUN MDV1 BOOT

ou will then see the windows below. You must now select the program r pressing CTRL C on the QL; no action is required on the Thor.

19 ⁴	!			
HELP WINDO	W!	DRAW	WINDOW	
	:			
	-!			
	!		. CURSOR	
COMMAND	i		. CURSOR	
WINDOW	i			
	i			
	ĺ			
	1			
	!	•		
TNEO	DMATTCH	HINDOM		
INFO	RMATION	MINDOM		

e names of the windows, are for reference only; they do indicate eir main use, but part of the time the windows may be used for other ings.

HELP WINDOW:

Shows the current step sizes (in world units) for moving (MOVE) and rotating (TURN) objects.

INFORMATION WINDOW:

Shows the positions and rotation angles of the camera, cursor, points and lines and of any objects currently in memory, followed (if there is enough room left in the window) by the amount of memory used (the plot area is part of the work area); and the amount of space allocated can be changed with EDIT DEF - see below). For a detailed description of memory usage see 4.3.1.

2.2 Control Keys

a of Mai Africa

When the command window contains a menu of commands, the following keys will have special functions:

++ +↑F2**F**1 Changes cursor co-ordinates (x-+ y-+ z-+) CTRL $+ \leftarrow \rightarrow \downarrow \uparrow$ F2F1 Rotates cursor (if cursor is on an object or on the camera) ALT + ♣↑ Changes move step (-+) CTRL + ALT + Changes rotate step (-+) SHIFT + F4F5 If there is no room in the information window, this will scroll the list of objects. This also works when you choose a work object and the cursor. The keys are also used to control Box and Size. F4 F5 Selects command (-+) SPACE, ENTER Executes command ESC Usually takes you back to where you came F3 Updates draw window CTRL + F3 Changes screen mode Changes default cursor colour CTRL + F4 SHIFT + F1 Makes the draw window fill the whole screen SHIFT + F2 Makes the computer ask where to put the cursor SHIFT + F3 Shows camera views from all 6 possible directions. The direction number is shown before the work area. CTRL + 1 Moves the cursor to the closest vertex in the work object (if it is closer than the move step) CTRL + 2 Makes the cursor jump to the nearest point on the grid CTRL + CAPSLOCK Shows the Help pages in the draw window

You needn't learn all these keys now; you can always come back to this list when you need it.

The Editor cannot multitask, but provided that you exit through STOP PROGRAM (see later), it can be restarted by:

EXECUTE MDV2_3D_PRECISION_BIN or EXECUTE FLP1_3D_PRECISION_BIN

Note that from within the program an EXECUTE command is used to start the plotter driver.

3.3 An Example

.. 3.1 DEFINE

First let's define an object. Use the F4 and F5 keys and press SPACE (or ENTER) to select DEFINE. You will be asked for the object name: type 'A' and press ENTER. (The program will print everything in upper case.) Now you must decide how complex an object you are going to define:

OBJECT:
A
VTCS:10 (number of vertices)
PNTS:10 (number of points)
LINES10 (number of lines)

Remember that you must use one vertex for each isolated point and for each endpoint of a line; but if the same triplet of co-ordinates is to be used several times, a single vertex will do.

If you just press ENTER, the default (0) will be assumed. If you want to change any value, use the usual keys (). When you have answered the last question, (LINE), you will be returned to the main menu. Try to define a new object with the same name - it will show the definition of the object and the memory used by the definition. On the QL you can use about 100 to 200 lines before the program gets too slow, depending on the speed of your expansion hardware and, of course, on the length of the lines drawn on the screen.

.3.2 INSERT

So far our object has a name (A), and memory is reserved for it, but it is still empty. Now we will 'insert' something into it. The default is a point: select INSERT, using the F4 F5 keys, and press SPACE. You will now see a list of colours. Select the ink colour you want, using the F4 F5 keys, and press SPACE. You have now inserted a point into the object, which will become visible as you move the cursor away with the +>keys. As you will see presently, it is also possible to insert whole objects.

2.3.3 CURSOR

What about lines? Well, select CURSOR (F4 F5). You will see the following list:

CAMERA POINT LINE A

Yes - 'A' is the object you have defined before, but it will have to wait. Select LINE (F4 F5), then select Insert. When you now move the cursor (F2 F1), you will see a ('rubber') line going from the previous position to the current cursor position. When you select Insert again, you will be asked to choose the colour of the line. This question is asked twice, not because the program is deaf but because lines can have stipples, though only in standard drawing mode. As it happens, the Editor is loaded as a default in fast drawing mode, and you will have to perform some minor contortions to get into standard drawing mode for stipples. There is an error in the way MG ROM QLs handle points. Determine which QL version you have (PRINT VER\$ from SuperBASIC accomplishes this): if it is an MG you must enter GDRAWTYPE 1,0. If it is any other version (AH, JM or JS - if you have any other ROM, you are in real trouble!) then enter GDRAWTYPE 1,1 instead. The MG ROM point error can cause the 3D PRECISION editor to stop with an error message if a point is plotted. If you then choose two different colours, you must also select the stipple pattern to be used. If you want a solid colour - or if you stayed in fast drawing mode - just select the same colour again.

2.3.4 OBJECT

First define a new object called 'B'. When you then select OBJECT, you will see:

A B

Select 'B'. B is now the object you work on. Then change the cursor: select CURSOR and the 'A' object. You came now move the object with the cursor control keys. Try also to rotate it by holding the CTRL key down and then pressing one of the cursor control keys. Now select INSERT: You will be asked to select colours or to press ENTER (instead of SPACE). This can be used to make copies of objects in different colours and to build a large object from smaller ones. Just press ENTER for now and move the object away. Press F3 to update the world. If there are too few vertices, points or lines, an error message (or several) will be displayed on a red background in the information window. Select CAMERA under CURSOR. You can now move and rotate the camera.

2.3.5 REMOVE ONE

Select REMOVE, type 'A' and press ENTER. You will observe that the object 'A' is removed and that the 'B' object takes its place in the information window. Repeat this with 'B', and you have removed both objects you had defined. You could of course have done this by using the REMOVE MULtiple command in 2.4.2. If there are many and/or large objects after the last object to delete, the command will take a little time to be executed.

4 Further Menu Commands

ou may read what follows as needed. It is organised on the principle f indentation of sub-menus. Some of the headings are abbreviated on he screen.

.4.1 FILE I/O

Five different file types are used:

_OBJ3D an object.
_ALL3D a small file containing the number of cameras (always 1) and the number of objects saved as a world under this name by SAVE ALL.
_CAM3D the camera definition saved by SAVE ALL.
_DEF3D A file containing the defaults used by this program.
_PIC A common screen file.

Some of these are discussed below under DEF-I/O and PIC-I/O. Only the name, i.e. without the device name (e.g. flp1, mdv1_) and the above extensions, must be typed in answer to prompts generated by the following commands. You may want to change the device before trying this. (See DEFAULT EDIT)

LOAD OBJECT

First you will be asked for the file name of the object and then for its name when loaded. If the former cannot be found, the usual error message will be displayed.

SAVE OBJECT

This is the opposite of load." You are asked first for the name of the object and then for the file name. If a file with the same name (and extension) exists, you are asked whether to replace or not (type Y or N and press ENTER).

DIR OBJECT

This shows the filenames of all objects (_OBJ3D) on the default device. If you get tired of waiting, you can stop the listing by pressing ESC.

DELETE OBJECT

This will ask you for the name (remember, the name only) of the file you want to delete.

LOAD ALL-

This will load a whole world at once, provided it has been saved by SAVE ALL. If some objects have already been defined or loaded you are asked if you want to delete them (Y/N).

SAVE ALL

If you have just made 30 objects and want to save, you needn't wear out your fingers by 30 SAVE OBJECT commands. Just type the 'world' name, and all objects, as well as the camera position, will be saved under that name. You can then reload them all by LOAD ALL, or use load LOAD OBJECT to load objects one by one under the world name with the object's number at the end; e.g. 'BASICDEMO2'.

DIR ALL

Displays all the world names (_ALL) on the device. If you get tired of waiting, you can stop the listing by pressing ESC.

DELETE ALL

Use this to delete a world saved by SAVE ALL.

2.4.2 OBJECT CONTROL

This menu contains functions which work on the object selected by OBJECT.

aCHANGE VERTEX

This can be used to change a vertex's co-ordinates. You must first select the vertex (shown by a flashing dot) to change. The information window will show the number of the vertex and its x, y, z co-ordinates. After that you can move the vertex with the usual keys. If the vertex is shared by several lines, these lines will follow the vertex.

VERTEX CLEAR

When you have deleted some points or lines, you can delete their vertices. If you try to delete a vertex which is still in use, an IN USE error message will be displayed.

POINT INK

Select the point to change (F4 F5, SPACE). The point is 'highlighted' (!) by changing its colour to black. Then select the new ink colour (as for INSERTing points). You must select another point before you can see the change. In the information window you can see the colour of the point and the vertex it uses.

POINT CLEAR

Select the point to clear (F4 F5), press SPACE, and the point will be removed. In the information window you can see what used to be the colour of the point and the vertex it used.

LINE INK

Select the line to change (F4 F5, SPACE). The line will be included in select the ink colour (as for mINSERTing lines). You must select another line before you than see the change. In the information window you can see the colour of line and the vertices it uses.

LINE CLEAR

Select the line to clear (F4 F5), press SPACE, and the line is removed. In the information window you can see what used to be the colour of the line and the vertices it used.

33;

REMOVE MUL

It works as REMOVE ONE except, that it will remove multiple objects. You must type in the identifier for the first and the last object that you want to delete: see the list of objects in the information window. If there are many and/or large objects after the last object to delete, it will take some time.

BOX

First select the colour of the box. Then you can select the box size along the x axis by F4 F5. You can select other axes by SHIFT + F4 F5 (you can only select z when x > 0 and y > 0). You can change the position and rotation of a box with the usual keys: (CTRL + \leftarrow + F2F1).

CIRCLE

First you must decide whether you want the circle to consist of points or of lines (P/L). Then you must select its colour (as in INSERT point or line). Now, how many vertices (points or lines) should be used? The more vertices, the rounder the circle will become (there must be at least 5). You can now move, rotate and change size (SHIFT until you are satisfied, and then press SPACE.

SIZE

You can use this to change the size of an object. Pressing F4 F5 will change the integer part of the step factor, and SHIFT F4 F5 the fractional part. Now comes the fun: using the usual move keys, you can change the size along the different axes. If you don't want to wreck your object, press ESC; if it's all right, press SPACE.

CENTRE

When you have selected this function, any movement of the cursor keys will affect the reference point from which all vertices of the current object are calculated.

MIRROR "

This will make the negative x (or y, or z - it's your option) co-ordinates of all or selected vertices of an object positive, and positive ones negative. Remember, all co-ordinates are defined with respect to the centre of the object - so if you want to shift the mirroring plane, use CHANGE CENTRE first. You can choose whether to mirror from the current cursor position or from the work object (but mirroring will always affect the work object). You will only be prompted if necessary, i.e., if there is a cursor object and the cursor and work objects are not the same. Then you must decide about which axis to mirror and whether you want to 'mirror all' or choose the points or lines to mirror (F4/F5, followed by SPACE/ENTER). The old vertices disappear as their mirror images are formed; there is no way to have an object and its mirror image on the screen at the same time.

BACKPLANE

This is used to set a z co-ordinate behind which any lines and points will disappear. If you want to remove the backplane, press ESC.

2.4.3 PROGRAM CONTROL

This menu contains facilities to change the defaults in the DEFAULTS_DEF3D file and to save the screen.

The defaults are as follows:

ISL The crosses in the columns at left indicate whether these defaults are effective immediately, after START, or after LOAD DEFAULTS.

	LOAD DEFAULTS.	ately, after START, or after
ххх	STORAGE DEVICEFLP1_	
XXX	INSERT IN EDIT(1/0)1	DIR, DELETE. Mode used when editing
XXX	OVER MODE(-1/0)	Mode used when plotting
XXX	GRID10	objects (XOR or OVER). Distance between gridpoints. When you
vvv	DI OMMUD DELLE CO	press the 'TO GRID' key, the cursor will be moved to the nearest gridpoint.
XXX	PLOTTER DEVICESER1	The device the plotter is connected to or a filename
XX	MODE8	to direct output to file. Screen mode 4 or 8. If set to 0, the current mode is
х	NO. OF OBJECTS30	used.
	0202013	The maximum number of objects that can be
		loaded. The theoretical
		maximum is 1024, but there is not enough room in the
		dataspace of the Editor,
	÷	which is a Turbo-CHARGED
x	SIZE OF WORK AREA4226	SuperBASIC program.
	2 00 W 200	The work area used by the Editor.
X	SIZE OF PLOT AREA16	The plot area is part of
		the work area; it is used
х	LOAD ALL ON START	for the plot lists. The objects saved under
		this name are leaded.
XX X	CURSOR COLOUR7	Colour of the cursor.
X	CAMERA POS. X	Camera position.
X	Z500	
Х	CAMERA ANGLE X	Camera angle.
X	Y	
χ _α Χ	Z0	
ite ir X	SCALE YL192	Height of the screen window.
A	ZL288	Distance into the screen corresponding to YL.
Х	SCREEN FACTOR17128	Change this if a circle is not round.
XX	MOVE STEP50	Cursor move step.
XX XXX	ROTATE STEP10	Rotate step.
XXX	ERROR INK	Colour of error messages.

33:

EDIT DEFAULTS

When you select this, you will see a list of parameters in the information window. The changes will first be effective on restart or when you use LOAD DEFAULTS, except for the first four defaults: device, insert/overwrite, XOR mode and grid, which will be changed immediately). If you enter an illegal value, it will be changed to an acceptable value on exit (by ESC).

WINDOW

First you must select which window to change (F4 F5). Then if you, are in a mode 4, you can change the size of characters (and the window) by SHIFT . To move the window, use the arrows; to change the size of the window, ALT and the arrows; to change size in steps of one character, CTRL and the arrows. When finished, press ENTER; you will be prompted for changing the colour of the border, of the background and of the ink for the window. You can press ESC if you don't want any changes.

SCALE

This is used to change the height of the window (YL) and its equivalent into the screen (ZL). If a circle does not look round on you monitor or TV, you should change the x axis (SF for Screen Factor).

WARNING:

DO NOT EXPERIMENT WITH ANY EXTREME COMBINATIONS OF YL, ZL, SF. There is an ever so slight chance that you may crash the program.

KEYBOARD

You can change the function of keys by altering this list of ASCII values. The list will be displayed in the draw window, and editing will work as for EDIT DEFAULTS. Changes will first be effective after a restart or a LOAD DEFAULTS.

DEFAULT-I/O

This should be very easy to follow.

LOAD DEFAULTS

Type the name of a DEF3D file, e.g. DEFTV8, and press ENTER. You will now see the window change. Some of the above defaults will also be changed.

WARNING: The program cannot recover from a 'bad or changed medium' error in LOAD DEFAULTS.

SAVE DEFAULTS

Type the name of the file, e.g. DEFAULTS, and press ENTER. If a file of that name already exists, you will be asked whether you want to replace the file (Y/N). When the program is (re)started (or LOAD DEFAULTS selected), the windows (and some defaults) will be changed to the settings in force when they were saved by SAVE DEFAULTS (see above).

DIR DEFAULTS

Lists the _DEF3D files on the default device. If you get tired of waiting, you can stop the listing by pressing ESC.

DELETE DEFAULTS

If you enter the name of a _DEF3D file, it'll be deleted.

500 10 20

PICTURE-I/O

PLOT PICTURE

You are asked whether you want to plot all or only one object (type A/O). If you type 'O', you are asked to type in the identifier for the object you want to plot. The work object is suggested as default. Then you are asked which form the colour information must have. You can choose between Normal, B&W and Convert colours (N/B/C). The default is normal which does nothing. B&W will change all colours to black = O. Convert colours will convert whatever colours you want. You are asked to type the number of the colour to convert, and the number of the colour it must be converted to. You can repeat until all colours are specified to your satisfaction. Press enter when you are through. You may then type in a label for the drawing. When you press ENTER it will start plotting. You can stop by pressing ESC. You must have inserted a disk or tape in the memory device, with room for the PLOT3D file.

SAVE PICTURE

First you are asked for the filename; then whether you want 'this screen' (which means all the windows) or the draw window to fill all the screen. If you choose 'this screen', you must decide which of the 8 menu items you want. If the file name already exists, you will be asked whether you want to replace it (Y/N).

DIR PICTURES

Lists the _PIC files on the default device. It you get fed up, you can stop the listing by pressing ESC.

DELETE PICTURE

This deletes a _PIC file on the default device.

DIR

a de ir

Lists all files on the default device. If you get tired of waiting, you can stop the listing by pressing ESC.

DELETE

This will delete any file on the default device.

STOP PROGRAM

This will end the program, i.e. remove it from memory. You can restart it with:

EXECUTE MDV2_3D_PRECISION_BIN

EXECUTE FLP1 3D PRECISION BIN

PROGRAMMING WITH THE BASIC TOOLKIT

his chapter does not assume that you have read Chapter 2; thus some aformation is the same. You are, however, assumed to know aperBASIC. To use the Toolkit most effectively, you should also be repared to learn about some of its inner workings.

1 How to Go About It

fore you can use the BASIC Toolkit, you must load it - and at the me time reserve workspace - with a short SuperBASIC program along e following lines:

100 workarea = 40
110 a = RESPR(workarea * 1024)
120 LBYTES flp1_3d_toolkit_bin, a
130 CALL a,a+workarea*1024
140
: REMark set workarea = 40 K
: REMark reserve area
: REMark load BASIC toolkit
: REMark set memtop, link in
: REMark BASIC keywords

e only parameters that may change in this program depending on your quirements are the value of workarea and the device from which you ad the toolkit. Normally it is advisable to reserve a lot of rkspace. If space is at a premium, the following memory map may lp you to calculate exact memory requirements. All space shown on is map is by default placed after the toolkit, but can also be located in QDOS' common heap — see 4.1.

WORK AREA:			i sa ki
FILEPOS:	TOOLKIT	21128 bytes long	
OBJ.TABLE	CAM. DEF PLOT. AREA OBJ. DEF 1	(8 + no. of objects * 4) byte 256 bytes (see 4.3.1) For every object: 2 * (max. points * 8 + max. lines * 16 + 2) bytes 256 bytes (see 4.3.3)	es
	VERTEXTAB	max. vertices * 16 BYTES	(1) (2)
**	POINTTAB	max. points * 6 BYTES	
•	LINETAB	max. lines * 8 BYTES	937
	OBJ.DEF 2	same as OBJ.DEF 1	
	VETEXTAB		18
	POINTTAB		
AFTERFILE:	LINETAB	•	
THE THE	FREE MEMORY		
WORKTAB:	WORKTAB	(maximum no of vortices used	. h
MEMTOP:		(maximum no. of vertices used any of the loaded objects) * bytes	

n you have run the above program, you are ready to use the new erBASIC extensions in your own programs. (The demonstration gram BASICDEMO has its own boot program, BASICDEMOBOOT, to take of the initialisation.) Let's look at an even shorter sample gram that will show us the most important features:

```
200 GSTART 3,5
                                       : REMark start system:
202
                                       : REMark plot area = 3 K;
205
                                       : REMark max. 5 objects
210 GWINDOW 512,256,0,0,256,512,17128 : REMark open window over
212
                                       : REMark whole screen and
215
                                       : REMark set YL, ZL, SF
218
                                       : REMark (screen factor)
220 GCLS
                                       : REMark clear the screen
230 GPLACE 0,200,0
                                       : REMark place the camera
                                       : REMark (only 3 params.)
240 GLOADOBJ flp1 basicdemo2 obj3d
                                      : REMark load object
250 FOR T=1 TO 90
      GPLACE 1,0,0,7000-T*50
                                      : REMark place the object
     GROTATE 1,0,T*4,0
                                      : REMark rotate the object
280 GMAKE : GCLEAR : GPLOT
                                   . : REMark update the object
290 END FOR T
```

In order to understand the commands in the loop, it is important to know that for each vertex two sets of co-ordinates are stored (to prevent loss of precision): definitions and results; but only the latter are used directly for calculation and subsequent plotting. When an object is GROTATEd, new results are produced from the definitions, whereas GCHANGEOBJ duplicates the definitions into the results. Other commands, e.g. GSIZEOBJ and GCENTREOBJ, only change the definition, so that the changes they cause will not become evident until after the next GROTATE or GCHANGEOBJ.

In analogy to the two co-ordinate sets, the program also maintains two plotting lists for each object, a current list and a reserve list. GMAKE reads from the 'result' part of the vertex variables and places the values it has calculated in the current plot list. GPLOT plots from this list and swaps the two plot lists, so that the above values remain available in the reserve list while a subsequent GMAKE writes new values into the first list. This makes it possible to toggle between two states of an object without the need for recomputation.

In the above program, however, each set of computed values is used only once. GCLEAR makes the last image of the object disappear by replotting it in the background colour (from the reserve list, which is then used for next plot list), and GPLOT plots the next image from the current list. The first time through the loop, of course, the reserve list is still empty, so there is nothing for GCLEAR to act on.

There is a lot more to learn if you take the time to analyse the BASICDEMO program as a larger example of using the BASIC Toolkit.

The BASIC Toolkit is not re-entrant, i.e., only one program must use it at any time. Thus programs using it will not multitask, unless you can link the toolkit to a compiled program.

3.2 The New Keywords

Functions are marked as such. All other keywords refer to PROCedures.

Below are some names used frequently in the syntax descriptions. Any suitably-ranged numeric expression may be used in their place.

```
0:
         1 TO 1024
                          Object number
v:
         1 TO 32767
                          Vertex number
         1 TO 32767
p:
                          Point number
1:
         1 TO 32767
                          Line number
on/off:
            0/1
                          Turn something on or off
col:
         O TO 255
                          Colour
         -32768 TO 32513 If the clear flag is set (bit 15)
x, y, z: -32768 TO 32767
```

3:

GBACKPLANE o, on/off [, clipplane]

clipplane: -32768 TO 32767 Default for clipplane is zero

Turns backplane clipping of an object on or off and sets its clipping plane on the z axis.

GCAMADDR

A function that returns the address of the camera definitions (see 4.3.1).

GCAMCEN x, y, z

Sets the centre of rotation for the camera relative to its position.

GCAMCENX

A function that returns the x co-ordinate of the camera's centre of rotation relative to its position.

GCAMCENY

A function that returns the y co-ordinate of the camera's centre of rotation relative to its position.

GCAMCENZ

A function that returns the z co-ordinate of the camera's centre of rotation relative to its position.

GCAMX

A function that returns the x co-ordinate of the camera position.

GCAMXA

A function that returns the rotation angle of the camera about a line through its centre and parallel to the x axis.

GCAMY

1.1

A function that returns the y co-ordinate of the camera position.

GCAMYA .

A function that returns the rotation angle of the camera about a line through its centre and parallel to the y axis.

GCAMZ

A function that returns the z co-ordinate of the camera position.

GCAMZA

A function that returns the rotation angle of the camera about a line through its centre and parallel to the z axis.

GCENTREOBJ o, x, y, z

> Changes the centre of an object by adding x, y, z to the definitions. The change will first be effective after GROTATE or GCHANGEOBJ.

GCHANGEOBJ

Moves the definitions to the results (i.e., the object is not rotated). If you have just changed the definition of an object (e.g. by GCENTRE, GSIZEOBJ) and you must plot the object without rotation, your changes will first be effective after this command. If rotation is required, use GROTATE.

GCLEAR [o [TO o2]]

> Clears the object(s) by plotting them (from the reserve list, which is subsequently cleared) plotting background colour (or in their previous colour when in XOR mode; see GOVER). If two object numbers are given, the two objects and all objects with in-between numbers will be cleared. If no object is specified, all objects will be cleared.

GCLOSE

Closes the graphics window.

GCLS [col]

> Clears the screen. If no colour is given, the last GCLS or paper colour is used.

GDATAADDR

430

ar ir

A function that returns the address of certain parameters (see 4.3.8).

GDEFINEOBJ nv, np, nl

> nv: 0 to 32767 Number of vertices np: 0 to 32767 Number of points 33: Number of lines nl: 0 to 32767 Maximum size depends on available memory.

Defines a new object.

GDRAWTYPE point, line

> point: 0 to 1, (TO 32767) 0: QDOS line: 0 to 1, (TO 32767) 1: new fast algorithm, but no stipples 2: reserved for your

own algorithms

Sets the algorithm for drawing. If you choose 2 or larger, you must have linked in your own assembly plotting routine. You can use values larger than 2 to mean something special; you can make a whole set of different drawing modes (see the DRAWTYPE file).

GERROR

A function that returns the number of a trapped error (see GTRAPERROR).

The following error codes may occur in addition to those of QDOS:

- 1 Work area used up
- 2 Plot area used up
- 3 Too many objects; maximum set by GSTART

GFACTOR

1:

sf

sf = 8000 to 32767 screen factor (default 17128)

Changes the x/y ratio of the screen by changing the screen size (in world units) in the x direction.

GFINDLINE (o, 1, col, v1, v2, dummy)

dummy: Must always be set to -1

Line number from which to start the

search

col: Colour of the line (use -1 to search

for a line of any colour)

v1, v2: Numbers of the vertices the line

is using (in any order; use -1 once or twice if one or both vertex numbers are to be ignored)

in the search)

A function that returns the lowest number of a line conforming to the specifications given in the command, or, if no such line is found, zero.

GFINDMAX (o, %1, v2)

.:Vix

A function that returns the number of the vertex (ignoring vertex numbers lower than v1 or higher than v2) of an object which has the largest absolute value in any co-ordinate (x, y or z). This largest absolute value can then be found by GOBJMAX.

GFINDPOINT (o, p, col, v, dummy)

dummy: Must always be set to -1

p: Point number from which to start the

search

col: Colour of the point (use -1 to

search for a point of any colour)

Number of the vertex the point is using (use -1 if vertex numbers are to be ignored in the search)

A function that returns the lowest number of a point conforming to the specifications given in the command, or, if no such point is found, zero.

.1

GFINDVERTEX (o, v, x, y, z, range)

range:

х, у, в:

Search limits are +-range on x, y, z. If set to -32768, the whole

world will be searched.

V: Vertex number from which to start

the search Vertex co-ordinates

A function that returns the number of a vertex or, if no vertex has been found, zero. If a range other than -32768 is specified, the search will be made in the rotated results; else in the definitions.

GINK

12

o, col

col:

If col is -1, the old colours will return.

Sets the colour of an object.

GINSERTOBJ col, o1, o2

col:

If col is -1, the colours of ol will be used.

Inserts object ol in object o2. Duplicated points and lines are not inserted. If a positive error number results, it should be read as follows: bit 0 - too few vertices; bit 1 - too few points; bit 2 - too few lines.

GLINE col, xp, yp [, xp2, yp2]

It will plot a line using the algorithm set by GDRAWTYPE. Pixel co-ordinates are used when in fast line drawing mode. Graphic co-ordinates are used when in QDOS drawing mode. The values in the plotting lists are, of course, compatible with the requirements of this routine.

GLINE3D col, xp, yp, zp [, xp2, yp2, zp2]

It will plot a line in the 3D world. You specify its position just as an object (see GPLACE).

GLOADOBJ "filename"

filename: Name of 'object' file to be loaded

Loads an object and assigns it the number given by (number of objects previously loaded +1).

GMAKE [0 [TO 02]]

Calculates the object(s) from the results areas of its vertices and places the calculated values in the current plot list.

GMARKLINE o, 1, [12,] on/off

Marks or unmarks one or more lines. Marked lines will not be calculated during a GMAKE.

GMARKOBJ o [TO o2], on/off

Marks or unmarks one or more objects. GMAKE, GPLOT, GCLEAR and GSWAP will not take any action on marked objects.

GMARKPOINT o, p, [p2,] on/off

Marks or unmarks one or more points. Marked points will not be calculated during a GMAKE.

GMARKVERTEX o, v, [v2,] on/off

mno

Marks or unmarks one or more vertices. Marked vertices will not be calculated during a GMAKE.

GMODE

mno: 4/8 mode number

Sets the screen mode in the Toolkit. This command has to be preceded by the analogous SuperBASIC command, thus: MODE 4: GMODE 4

GMOVE [0,] forward/backward, right/left, up/down

Moves the object by the number of world units specified along each of the three axes. If no o is specified, the camera is moved as if it were an object.

GNOOBJ

A function that returns the number of objects in memory.

GNOOBJMAX

A function that returns the maximum number of objects as defined by GSTART.

GOBJADDR (o)

A function that returns the address of the object definitions (see 4.3.3).

GOBJMAX

A function that returns the maximum absolute value found by the most recently executed GFINDMAX.

GOBJX (o)

A function that returns the x co-ordinate of the centre of the object with respect to the camera position.

GOBJXA (o)

A function that returns the angle of rotation of the object about a line through its centre and parallel to the x axis.

GOBJY (o)

A function that returns the y co-ordinate of the centre of the object with respect to the camera position.

GOBJYA (o)

A function that returns the angle of rotation of the object about a line through its centre and parallel to the y axis.

25

GOBJZ (o)

A function that returns the z co-ordinate of the centre of the object with respect to the camera position.

GOBJZA (o)

A function that returns the angle of rotation of the object about a line through its centre and parallel to the z axis.

GOVER drawmode...

drawmode 0, -1 0 = normal -1 = XOR

Sets the drawing mode.

GPAPER [0,] col

Sets the colour of the background. If an object is specified, it will be cleared by being redrawn in that colour. The background colour - and hence the clearing action - is turned off if col = -1. This can be used to make patterns: as the object is not cleared, it will leave an image every time it is moved or rotated.

GPLACE [0,] xp, yp, zp

xp, yp, zp: -16384 TO 16383 x, y, z positions of object

Note: -32768 TO 32767 will be accepted, but the distance from the camera must not exceed 32767.

Sets the object at a given position. If no object is specified or o=0, the camera will be set at the given position.

GPLOT [0 [TO 02]]

Plots the object(s). After an object has been plotted, the plotting lists are swapped (see GSWAP).

GPOINT col, xp, yp

It will plot a point using the algorithm set by GDRAWTYPE. Pixel co-ordinates are used when in fast point drawing mode. Graphic co-ordinates are used when in QDOS drawing mode. The values in the plotting lists are compatible with the requirements of this routine.

. 3

GPOINT3D col, xp, yp, zp

It will plot a point in the 3D world. You specify its position just as an object (see GPLACE).

GREMOVELINE o, 1 [TO 12]

Removes the line(s) from the description.

GREMOVEOBJ o [,TO o2]

Removes one or more objects.

GREMOVEPOINT o, p [TO p2]

Removes the point(s) from the description.

33:

GREMOVEVERTEX o, v [TO v2]

Removes the vertex/vertices, unless used by a point or line, in which case an 'in use' error will be generated.

GROTATE [o,] xa, ya, za

xa, ya .za: -32768 TO 32767 xa, ya, za are the angles of rotation of the object about lines through its centre parallel to the x, y, z axis respectively.

Prepares rotation of an object by calculating new 'result' co-ordinates of its vertices from the vertex definitions and the command parameters. If no object is specified or o = 0, the camera will be rotated. GMAKE and GPLOT must follow if the rotation is to become visible.

GROTATEVERTEX no, v [TO v2] [,xa, ya, za]

xa, ya .za: -32768 TO 32767 xa, ya, za are the angles of rotation of the distribution of the distribution

Prepares rotation of some selected vertices in an object by calculating new 'result' co-ordinates from the vertex definitions and the command parameters. GMAKE and GPLOT must follow if the rotation is to become visible.

GROTATEVERTEXX no, v [TO v2] [,xa]

xa: -32768 TO 32767

Prepares rotation of some selected vertices in an object about a line through its centre and parallel to the x axis by calculating new 'result' co-ordinates from the current 'result' co-ordinates and the command parameters. GMAKE and GPLOT must follow if the rotation is to become visible.

GROTATEVERTEXY no, v [TO v2] [,ya]

ya: -32768 TO 32767

Prepares rotation of some selected vertices in an object about a line through its centre and parallel to the y axis by calculating new 'result' co-ordinates from the current 'result' co-ordinates and the command parameters. GMAKE and GPLOT must follow if the rotation is to become visible.

GROTATEVERTEXZ no, v [TO v] [,za]

za: -32768 TO 32767

Prepares rotation of some selected vertices in an object about a line through its centre and parallel to the z axis by calculating new 'result' co-ordinates from the current 'result' co-ordinates and the command parameters. GMAKE and GPLOT must follow if the rotation is to become visible.

GROTATEX x, y, z, xa

x, y, z: -32768 to 32767 Variables - NOT expressions whose current values (-32768 to
32767) represent the x, y, z
co-ordinates of a point before a
rotation of xa degrees about the
x axis, and are changed by the
command to the co-ordinates
after rotation

xa: -32768 to 32767 Angle of rotation

Rotates x, y, z about the x axis.

Note: If you use a compiler which does not support parameter passing by reference (say Supercharge) an alternative method of finding the changed co-ordinates will be found in the GRXVALUE, GRYVALUE and GRZVALUE functions.

GROTATEY x, y, z, ya

x, y, z: -32768 to 32767 Variables - NOT expressions whose current values (-32768 to
32767) represent the x, y, z
co-ordinates of a point before a
rotation of ya degrees about the
y axis, and are changed by the
command to the co-ordinates
after rotation

ya: -32768 to 32767 Angle of rotation

Rotates x, y, z about the y axis.

Note: If you use a compiler which does not support parameter passing by reference (say Supercharge) an alternative method of finding the changed co-ordinates will be found in the GRXVALUE, GRYVALUE and GRZVALUE functions.

GROTATEZ x, y, z, za

x, y, z: -32768 to 32767 Variables - NOT expressions whose current values (-32768 to
32767) represent the x; y, z
co-ordinates of a point before a
rotation of za degrees about the
z axis, and are changed by the
command to the co-ordinates
after rotation

za: -32768 to 32767 Angle of rotation

Rotates x, y, z about the z axis.

Note: If you use a compiler which does not support parameter passing by reference (say Supercharge) an alternative method of finding the changed co-ordinates can be found in the GRXVALUE, GRYVALUE and GRZVALUE functions.

GRXVALUE

A function that returns the x co-ordinate computed by the last GROTATEX, GROTATEY or GROTATEZ command.

GRYVALUE

A function that returns the y co-ordinate computed by the last GROTATEX, GROTATEY or GROTATEZ command.

GRZVALUE

A function that returns the z co-ordinate computed by the last GROTATEX, GROTATEY or GROTATEZ command.

GSAVEOBJ o, "filename"

Saves an object. If an object with that name already exists, it is overwritten.

GSCALE yl, zl

yl: 32-512 height of the screen window in world 3

co-ordinates

z1:/: 32-2048 distance into the screen equivalent

to yl in world co-ordinates

Sets the yl, zl parameters.

GSETLINE o, l, col, v1, v2, dummy

dummy: Must always be set to zero.

Defines a line. If 1 leaves a gap after the previously highest line number, intervening lines are marked.

GSETPOINT o, p, col, v, dummy

dummy: Must always be set to zero.

Defines a point. If p leaves a gap after the previously highest point number, intervening points are marked.

GSETVERTEX o, v, x, y, z

Defines a vertex. If v leaves a gap after the previously highest vertex number, intervening vertices are marked.

GSIZEOBJ o, intx, fracx, inty, fracy, intz, fracz

intxyz: Integer part of magnification factor fracxyz: Fractional part of magnification factor

(denominator 32768 implied, numerator can be 0 to

32767)

Changes the size of an object. As it changes the definition, it will first come into effect after GROTATE or GCHANGEOBJ. Magnification factors are given along each axis. Observe that the size is changed in a fixed world co-ordinate system. You can see that the object becomes 'deformed' when it is rotated:

GSTART pa, no

pa: plot area, i.e. area used to hold the plotting

lists

no: number of objects to be used

The first command in every program that uses the Tolkit. It sets the size of the plot area and the number of objects that can be loaded or defined. It can only be used once in a program.

GSTOP

GSWAP [o [TO o2]]

Swaps the plotting lists. Remember that the plotting lists are swapped by every GPLOT; so if you want to replot identically after a GPLOT, GSWAP has to be used first, in order to swap the lists back.

GTRAPERROR on/off Errors are trapped when this is set to 1.

Traps errors so that program execution continues. The error number can be read by GERROR.

GWINDOW xl, yl, xp, yp [, yl, zl, sf]

xl, yl Must be at least 16

Sets window position and size. The screen parameters can also be set (see GSCALE, GFACTOR).

PROGRAMMING WITH THE ASSEMBLER TOOLKIT

.1 How to Go About It

is chapter assumes that you have read the chapter on BASIC ogramming.

nen using the Assembler, you may load the BASIC Toolkit as described a section 3.1. You can, however, save some memory by using a TOOLKITASM_BIN, which does not contain any BASIC definitions. If you clear the link flag the BASIC definition will not be linked in, the basic toolkit.

u can change the position of the work area. Set FILEPOS to the art and MEMTOP to the end. The change must be made after a call to start of the toolkit and before the start of the Assembler program (START).

allow QDOS points and lines, some dummy variables must be defined. normal assembler, GBV.RIP must be set to point to a stack in the b and GBV.VVBAS to a reasonably large area. In BASIC extensions V.RIP should be set to BV.RIP and GBV.VVBAS to BV.VVBAS. This is ly necessary when using those commands that use the QDOS scale, dot d line traps (i.e. window, dot, line, clearobject and plotobject). LOADOBJECT and/or SAVEOBJECT are used, use BP.ABSIO to set dresses.

u can make your own drawing routines. D0-D4, A0, A3 need not be eserved. On entry A2 will point to the parameters.

r an example of drawing routines see DRAWTYPE and BASICEXT.

r an example of assembly see ASMDEMO.

r an example of BASIC extensions see BASICEXT, BASICEXTDEMO and SICEXTDEMOBOOT.

2 Assembler Routines

A5 (often) object number

A0 (often) camera address

A2 is always preserved or set to data address

A3 (often) object address

A4 (usually) undefined; use for call of routines

A5 , is always preserved or set to address of EXECLIST

START 4 \$00

This call is like GSTART.

INPUT

OUTPUT

331

D1.W plot area in K all undefined.

D5.W number of objects max 1024 every object uses 4 bytes

D6.W number of cameras (always 1)

Plots a point, using one of the algorithms set by DRAWTYPE.

INPUT		OUTPUT	
DO.W	colour	DO.L	undefined
D1.L	X position	D1.L	undefined
D2.L	Y position	D2.L	undefined
		T) 0 T	

D3.L D4.L undefined A3.L undefined

PLACE

\$02

a Sets the line start.

INPUT

INPUT

OUTPUT

D1.W	X1 position	D1.L	preserved
D2.W	Yl position	D2.L	preserved

LINE

\$03

Plots a line, using one of the algorithms set by DRAWTYPE.

INPUT		OUTPUT	
DO.W D1.W D2.W	colour X2 position Y2 position	DO.L D1.L D2.L D3.L D4.L	undefined undefined undefined undefined undefined
		A3.L	undefined

POINT3D \$05

> Plots a point in 3D, using one of the algorithms set by DRAWTYPE.

INPUT OUTPUT

DO.W	colour	all undefined
D1.W	x position	except: D7, A4
D2.W	y position	A0=camera address
D3.W	z position	

PLACE3D \$06

4 Sets the start of a line in 3D.

INPUT OUTPUT

D1.W	x position	D1.L	preserved
D2.W	y position	D2.L	preserved
D3.W	z position	D3.L	preserved

. 1

LINE3D

\$07

Plots a line in 3D, using one of the algorithms set by DRAWTYPE.

INPUT OUTPUT

DO.W colour all undefined,

D1.W x position except:

D2.W y position A0 camera address

z position D3.W

CONV16 \$09

44.

Calculates the x, y positions when z = 0.

INPUT OUTPUT

D1.W x position D1.W x pos. in 2D y position D2.W D2.W y pos. in 2D D3.W z position

D3.L

undefined

ROTATEINIT16 \$OA

Calculates rotate information.

INPUT OUTPUT

D1.W X angle(0 to 360) D1.W undefined D2.W Y angle(0 to 360) undefined D2.W Z angle(0 to 360) D3.W D3.W undefined addr. of calculated info A3.L preserved (9 words) A4.L undefined

ROTATE16 SOB :

> Rotates x, y, z about 0, 0, 0 using rotate information from ROTATEINIT16.

INPUT OUTPUT

D1.W x position D1.W rotated x D2.W y position D2.W rotated y 🔗 D3.W z position D3.W rotated z undefined D4.L D5.L undefined A3.L addr. of calculated info A3.L preserved

ROTATEX16 SOC

Works, like ROTATE16 but rotates about the x axis only.

ROTATEY16 \$OD

Works like ROTATE16 but rotates about the y axis only.

ROTATEZ16 \$OE

Works like ROTATE16 but rotates about the z axis only.

```
ADDRCAMOBJ
                $21
   Returns the addresses of the camera and object.
      INPUT
                                            OUTPUT
   D5.W
          object number
                                      D5.L
                                              preserved
   D6.W
          camera number (always 1)
                                      D6.L
                                              preserved
                                       D7.L
                                              undefined
                                      AO.L
                                              camera address
                                      A3.L
                                              object address
WINDOW
                $22
                                                            :)
   Changes the window size and position.
      INPUT
                                            OUTPUT
          x length min. 16
   D1.W
                                       all undefined
   D2.W
          y length min. 16
   D3.W
          x pos.
   D4.W
          y pos.
DRAWTYPE
               $23
   Sets the algorithm to be used in making points and lines.
      INPUT
                                           OUTPUT
          point type 0,1,2(to 32767) D1.L
                                             preserved
   D2.B
          line type 0,1,2 (to 32767) D2.L
                                             undefined
                                      AO.L
                                             undefined
                                      A1.L
                                             undefined
                                      A3.L
                                             undefined
                                      A4.L
                                             undefined
MODE
   Changes the internal screen-mode setting.
```

INPUT	OUTPUT	
Ð1.B mode 4/8	D1.L preserved 3: D2.L undefined A0.L undefined A1.L undefined A3.L undefined A4.L undefined	
CLS. \$25		
"Clears the window.		
INPUT	OUTPUT	
no input	D1.L undefined D3.L undefined A1.L undefined	

A4.L

undefined

.1

PAPER

\$26

Works like GPAPER, i.e., sets the screen colour. If D5 <> 0, it will be CLEARed by drawing the object in that colour.

*		•		•	_
	N	ш			т.
-	11		٠.	, ,	

OUTPUT

D1.B	colour	D1.L	preserved
		D3.L	undefined
D5.W	if 0 then screen	D5.L	preserved
		A1.L	undefined
A3.L	object address	A3.L	preserved
		A4.L	undefined

INK

\$27

Works like GINK, i.e., PLOTs the object by drawing it in that colour.

INPUT

OUTPUT

D1.B	colour	D1.L	preserved
		D3.L	undefined
		Al.L	undefined
A3.L	object address	A3.L	preserved

SCALE

\$28

Sets YL (the window height) and ZL (the depth corresponding to this), both in world co-ordinates.

Warnings: If low values of YL are used, precision suffers, resulting in strange deformation of objects as they move. Clipping at screen borders may not work at extreme combinations of YL, ZL, SF, so that system information may be overwritten, resulting in a crash (see GVARSF in 4.3.1).

INPUT

OUTPUT

D1.W YL 32 to 512 D2.W ZL 32 to 2048 all undefined

OVER

\$29

Sets the plotting mode.

INPUT

OUTPUT

· · · · · · · · · · · · · · · · · · ·		
D1.W mode 0 normal	D1.	L undefined
-1 XOR	D3.	L undefined
Att in	A1.	L undefined
25.	A4.	L undefined

CLOSE

\$2B

Close closes the window. The window can be reopened by WINDOW.

INPUT

OUTPUT

no input

A4.L undefined

STOP

12

\$2C

Stop resets the toolkit: if the window is open it will closed.

INPUT

OUTPUT

no input

A4.L undefined

ROTATECAMERA \$2D

Works like GROTATE with 3 parameters, i.e., rotates the camera about its centre (centre of window).

INPUT .. OUTPUT

D1.W XA(0 to 360) not -XA D1.L preserved D2.W YA(0 to 360) not -YA D2.L preserved D3.W ZA(0 to 360) not -ZA D3.L preserved

MOVECAMERA \$2E

Moves the camera.

INPUT

D1.W +-forward D1.L undefined D2.W +right/-left D2.L undefined D3.W +up/-down D3.L undefined D4.L undefined D5.L undefined AO.L camera address AO.L preserved

ADDRCAMERA \$30

Returns the address of the camera definition (see 4.3.1).

INPUT

D6.W camera number (always 1) D6.L preserved A0.L address of camera

ROTATEOBJECT \$31

937

Works like GROTATE with 4 parameters, i.e., rotates the object about its centre.

INPUT

XA(0 to 360) not -XA1 D1.W Di.L undefined D2.W YA(0 to 360) not -YA D2.L undefined W.Ed" ZA(0 to 360) not -ZA D3.L undefined A3.W object address A3.L preserved

MOVEOBJECT \$32

Moves an object,

INPUT		OUTPUT		
1.W	+-forward	D1.L	undef	
) W	right /_loft	DO T	3 - 6:	

D1 ined D2.W +right/-left undefined D2.L D3.W +up/-down D3.L undefined D4.L undefined D5.L undefined A3.L camera address A3.L preserved

.1

ADDROBJECT \$34

Returns the address of the object definition (see 4.3.3).

INPUT

OUTPUT

D5.W object number

D5.L preserved D7.W undefined

A3.L object address

MAKEOBJECT

y 1 7 \$35

Calculates an object and produces a plot list.

INPUT"

OUTPUT

to object number all preserved except: if 0 then one object. A0 camera address

D5.W object number (not if D1 = 0)

A3.L object address

(only if one object)

PLOTOBJECT

\$36

\$37

Draws an object from the plot list.

INPUT

OUTPUT

to object number all preserved except: if 0 then one object AO camera address

object number (not if D1 = 0) D5.W

A3.L object address (only if D1 <> 0)

CLEAROBJECT

INPUT OUTPUT

Draws an object in background colour from a plot list.

INPUT

OUTPUT

D1.W to object number if 0 then one object all preserved except: , 3: AO camera address

object number (not if D1 = 0)

A3.L object address

SWAPOBJECT \$38

Swaps the plot lists so an object can be plotted repeatedly.

INPUT

OUTPUT

to object number all preserved except: if 0 then one object A0 camera address D1.W

D5.W object number (not if D1 = 0)

A3.L object address

DEFINEOBJECT \$39

Defines a new object's size, i.e. the number of vertices, points and lines it can contain.

INPUT

OUTPUT

number of vertices

number of points number of lines D2.W D3.W

all undefined except: D0=error 1: WA 2: PA 3: TO

A3.L object address

REMOVEOBJECT \$3A

Deletes an object from memory.

INPUT

OUTPUT

to object number D1.W

all undefined

D5.W object number

LOADOBJECT \$3B

> Loads an object into memory. The device and filename in standard format must be pointed to by AO. (Errors: WA, PA, TO, FF)

INPUT

OUTPUT

AO.L pointer to name

all undefined except: D0 = error

SAVEOBJECT \$3C

Saves an object from memory. The device and filename in standard format must be pointed to by AO.

INPUT de

OUTPUT

D5.W object number

all undefined

AO.L points to the filename

INSERTOBJECT \$3F

Inserts an object into another object.

INPUT

OUTPUT

DO.W colour if -1 then copy colour

, D1.W to object number ™ D5.W

from object number

all undefined except: DO = error code

IF positive then

bit 0: too few vertices 1: too few points

53

2: too few lines

CENTREOBJECT \$40

Changes the centre of the object by adding the values to the definition. ROTATEOBJECT or CHANGEOBJECT must be used before the change will come into effect.

INPUT	OUTPUT
D1.W x change D2.W y change D3.W z change	D1.L preserved D2.L preserved D3.L preserved
A3.L object address	D4.L undefined A3.L preserved A4.L undefined

SIZEOBJECT \$41

TNDIIT

Changes the size of the object by multiplying the values into the definition. ROTATEOBJECT or CHANGEOBJECT must be used before the change will come into effect.

IN.	F01	(OUTPUT
D1.W D2.W	integer part 0 to 32767	D1.L D2.L	preserved preserved
D3.W	x: 1, y: 2, z: 3	D3.L D4.L D5.L	preserved undefined undefined
A3.L	object address	D6.L A3.L A4.L	undefined preserved undefined

MARKOBJECT \$42

INPUT

Marks one or more objects. If an object is marked, GMAKE, GPLOT, GCLEAR and GSWAP will have no effect on it.

INPUT	OUTPUT
D1.W to object number D2.W FLAG: 0 OR 1 D5.W object number	D1.L preserved if zero D2.L preserved D5.W preserved D7.W undefined A0.L undefined
	A3.L undefined

CHANGEOBJECT \$43

Moves the definition to the result.

INPUT		55	t c	DUTPUT
A3.L	object address		D1.L A1.L A3.L	undefined undefined preserved
WORKFREE	\$49			

Returns the number of free bytes in the work area.

OUTPUT

DO.W free bytes



PLOTFREE

S4A

Returns the number of free bytes in the plot area.

OUTPUT

DO.W free bytes

FINDMAX

\$4B

Finds the largest value of x, y, z vertex definitions in the specified range.

we IN	PUT	z, (OUTPUT	a.
i i i		DO.W	error or vtx	number
D1.W	vertex number	D1.L	undefined	114111202
		D5.W	max. value	
D6.W	to vertex number	D6.L	undefined	
		D7.L	undefined	
		Al.L	undefined	(A)
A3.L	object address	A3.L	preserved	
		A4.L	undefined	

ROTATEVERTEX \$4D

Rotates the vertex/vertices.

INF	TUT	(DUTPUT
DO.W	If 0, angle from object	DO.W	error or vtx number
D1.W	vertex number	D1.L	undefined
D2.W	XA	D2.L	undefined '
D3.W	YA	D3.L	undefined
D4.W	ZA	D4.L	undefined
	200	D6.L	undefined
	Aje.	A1.L	undefined
A3.L	object address	A3.L	preserved
		A4.L	undefined
ROTATEVER	TEXX \$4E		

\$4E

Rotates the vertex/vertices about a line through the centre of the object and parallel to the x axis. 93.

INE	PUT	OUTPUT	
DO.W	If O, angle from object	DO.W	error or vtx number
D1.W	vertex number	D1.L	undefined
-D2.W	XA	D2.L	undefined
M.EG	0	D3.L	undefined
D4.W	0	D4.L	undefined
ir ir		D6.L	undefined
		D7.W	undefined
		A1.L	undefined
A3.L	object address	A3.L	preserved
		A4.L	undefined

PLOTFREE

a

S4A

Returns the number of free bytes in the plot area.

OUTPUT

DO.W free bytes

FINDMAX

\$4B

Finds the largest value of x, y, z vertex definitions in the specified range.

u. IN	PUT	ę. (OUTPUT	2.3
a		DO.W	error or vtx	number
D1.W	vertex number	D1.L	undefined	
		D5.W	max. value	
D6.W	to vertex number	D6.L	undefined	92
		D7.L	undefined	
		A1.L	undefined	* (
A3.L	object address	A3.L	preserved	76
		A4.L	undefined	

ROTATEVERTEX \$4D

Rotates the vertex/vertices.

IN	PUT		OUTPUT
DO.W	If O, angle from object	DO.W	error or vtx number
D1.W	vertex number	D1.L	undefined
D2.W	XA	D2.L	undefined
D3.W	YA	D3.L	undefined
D4.W	ZA	D4.L	undefined
	j.	D6.L	undefined
	ije.	Al.L	undefined
A3.L	object address	A3.L	preserved
		A4.L	undefined
OFF WORKS PROPERTY.	DEED A 4 P		

ROTATEVERTEXX \$4E

Rotates the vertex/vertices about a line through the centre of the object and parallel to the x axis.

INI	PUT	C	OUTPUT	325
DO.W	If O, angle from object	DO.W	error or vtx	number
D1.W	vertex number	D1.L	undefined	15
.D2.W	XA	D2.L	undefined	25
M.EG	0	D3.L	undefined	
D4.W	0	D4.L	undefined	
At br	(6)	D6.L	undefined	
·-		D7.W	undefined	
		A1.L	undefined	
A3.L	object address	A3.L	preserved	
		A4.L	undefined	

REMOVELINE \$53

Removes a line.

INPUT		OUTPUT	
D1.W	vertex number	D1.L	undefined
D2.W	to vertex number	D2.L D7.L	undefined undefined
A3.L	object address	A1.L A3.L	undefined preserved
		A4.L	undefined

.3

337

3

4.3 Assembler Definitions

4.3.1 Camera definition

GVARRESTAB	EQU 18	Start of plot table
GVARCAMCENX		The production
	~	,
GVARCAMCENY	EQU 34	centre of the screen
GVARCAMCENZ	EQU 36	5
Size of screen	in world	co-ordinates:
GVARXL	EQU 48	Calculated from yl, zl
GVARYL	EQU 50	Default 256
GVARZL		Default 512
GVARXLH		Centre of screen
GVARYLH "	EQU 56	
GVARSF	EQU 58	
Size and posit:	ion of scr	een in pixel coordinates:
GVARX1	EQU 64	
GVARY1	EQU 66	
GVARX2	EQU 68	
GVARY2	EOU 70	

1.3.2 Object and camera definition

GVARX	EQU	0		
GVARX16	EQU	2	25	
GVARY	EQU	4	•	
GVARY16	EQU	6		
GVARZ	EQU	8		
GVARZ15	EQU	10		
GVARXA	EQU	12	Read	only
GVARYA	EQU	14	Read	
GVARZA	EQU	16	Read	

483

4.3.3 Object definition

4.3.4

4.3.5

Where a name is followed by '16' (e..g GVARX16), only this part of the 32 bits should be used in this version.

.s	snould be used	in this	s ve	rsion.
	GVARSIZE	EQU	22	The circ of an abiant Mana
		220	2.2	The size of an object. Used to clip objects. Object will not be
				clipped if set to zero.
	GVARHLIM	EQU	24	
	GVARFLAG	EQU	30	
		Pi Vi	er.	If flag 2 = 1, size calc. is on
				If flag $3 = 1$, backplane clip on
				High byte:
	No. i			If flag 0 = 1, object is on screen
	a GVARDISTX	EQU	32	If flag 1 = 1, object is marked Distance from camera to centre of
		-	807	object
	GVARDISTX16	EQU	34	Calculated if backplane clip or
	~~~			size <> 0
	GVARDISTY	EQU	36	
	GVARDISTY16 GVARDISTZ	EQU	38	
	GVARDISTZ16	EQU EQU	40 42	150
	GVARSX	EQU	44	Size of object (if floor set).
	GVARSY	EQU	48	Size of object (if flag set): Low word: integer
	GVARSZ	EQU	52	High word: fraction
	GVAROBJLEN	EQU	60	Length of the definition of the
		77		object when in memory (only
	CUADNUM	22.0	N-0 (000)	1 to GVARNV, P, L are saved)
	GVARNVM GVARNPM	EQU	64	Max. defined vertices
	GVARNLM	EQU EQU	66	Max. defined points
	GVARNV	EQU	68 80	Max. defined lines Number of vertices used
	GVARNP	EQU	82	Number of points used
	GVARNL	EQU	84	Number of lines used
	GVARVERTEXTAB	EQU	96°	Address of vertex table
	GVARPOINTTAB	EQU	100	Address of point table
	GVARLINETAB	EQU	104	
	GVARPLOTTAB2	EQU	128	
	GVARPLOTTAB1	FOU	100	plot lists
	CVIIII BOTTABI	EQU	132	GVARRESTAB+GVARPLOTTAB(1/2)
	v.			= the plot table (1/2) address
	Vertex table			
	VERTEXXD	EQU	0	Definition of object
	VERTEXYD	EQU	2	2011MICION OF ODJOCC
	VERTEXZD	EQU	4	
	VERTEXX	EQU	6	Rotated definition
3	VERTEXY	EQU	8	
	VERTEXZ VERTEXINFO	EQU	10.	PG 274: 4780 (1980) (1980) (1980) (1980)
	WENTERINEO.	EQU	12	Set to zero when created;
	VERTEXFLAG	EQU	15	some unused space here If bit 7 = 1 then not calculated
		EQU	16	ar bre , - 1 then not carturated
	Point table			
	COLOUR	EQU	0	If flag 15 = 1 then no plot
	VERTEX	EQU	2	Starts from vertex 0
	RESERVED	EQU	4	and some more
		EOU	6	

EQU

3.6 Line table

COLOUR	EQU	0	If flag 15 = 1 then no plot
VERTEX 1	EQU	2	Starts from vertex 0
VERTEX 2	EQU	4	Starts from vertex 0
RESERVED	EQU	6	and some more
	EQU	8	

3.7 Plot table

COMMAND		EQU	0	1 = point 3 = line	2 = place -1 = stop
COLOUR		EQU	2	0 11110	r - Scop
X	44.	EQU	4		
1	2 7 7	EQU	6		24
i.		EQU	8		

It is conceivable that the plot table format may change in future versions of 3D PRECISION.

3.8 Other parameters

ADDREXECLIST	EQU	0	
NOOBJN	EQU	4	
NOOBJMAX	EQU	6	
ERRORNO	EQU	8	
CHANNEL	EQU	10	Used for QDOS points/lines
CAMADDR	EQU	14	outer for Apon bother, lines
GBV.RIP	EQU	18	Set to stack address
GBV.VVBAS	EQU	22	bee to stack address
FLAGS	EQU	26	8: 1 = trap error
121100	LQU	20	9: 1 = after start
			10: 1 = after window
OBJMAX	BOIL		11: 1 = link BASIC
	EQU	28	, p_
FILEPOS	EQU;	30	Start of file (the default is the
			address after the toolkit (when
			loaded)). (For changing the address
			see 4.1).
MEMTOP	EQU	34	
AFTERFILE	EQU	38	Pointer to first free byte after
ATT THE RESIDENCE OF THE PARTY			last object
WORKTAB	EQU	42	First byte not free (used by make)
			(length = (max number of vertices in
			any object) * 16)
BASICDEF	EQU	46	Start of BASIC definitions (offset
	15 3 1		from start of 3D TOOLKIT BIN
POINTTYPEADDR	EQU	50	Address of linked draw routine for
· ·			point
LINETYPEADDR	EQU	54	Address of linked draw routine for
ie b			line
MODE -	EQU	62	Mode8: 0 mode4: 1
POINTTYPE	EQU	63	Type of point: 0, 1, 2 to 32767
LINETYPE	EQU	64	Type of line: 0, 1, 2 to 32767
OVER	EQU	66	Highest byte = 1: OVER mode will be
	~		used when drawing
DRAWSTACK	EQU	70	Address of co-ordinates put by the
	~ -		place routine for use in line draw
			routine
END	EQU	74	
		e de la composición dela composición de la composición dela composición de la compos	

.9 Extra error codes

ERR.WA	1	Work area used up
ERR.PA	2	Plot area used up
ERR. TO	3	Too few objects allocated in 'START'

5. INDEX

Entries referring to the new keywords in section 3.2 are arranged alphabetically within that section. Many other keywords for which a reference to 3.2 is given will be found in that section under the corresponding keyword with initial G; e.g., SCALE is discussed in section 3.2 under GSCALE.

	section 3.2 under	GSCI	TE				,	٠.	9 . /	00		1.5 Q	LDCUBBE	Q III
	Dood I Jin O. B and of	0007	*14:5											
	A D D D C A M C D A												587	0 012701
	ADDRCAMERA		•	•	•	•		•	•	•			. 4.	2.\$30
	ADDRCAMOBJ			•	•								. 4.	2.\$21
	ADDROBJECT ALT ANGLE ASMDEMOFLPBOOT ASMDEMOMPUBOOT ASMDEMOMDVBOOT ASMDEMOMDV BIN ASMDEMO ASSEMBLER DEFINIT ASSEMBLER ROUTINE					•							. 4.	2.\$34
	ALT	2 19	-					-					. 2.2	2.4.3
	ANGLE		12		142	(a)	81 - 82 81 - 82	2	757	12			1	2 3 2
	ASMDEMOFLPBOOT	2 (2)	12	27	020		16 16 16 18	50	200	100 100			· - 3	1 7
	ASMDEMOFILE BIN		•		•				•	•	•	•	•	. 1.1
	A SMDEMONDUROOT		•	•	•	•	•		•	•	•	•		- 1.1
	ASIDENOMOVBOOT		•	-	•	•			•					. 1.1
	ASUDENOUDA BIN	•	•	•	14			•	•	•		•		. 1.1
	ASMDEMO			¥8									. 1.	1 4.1
	ASSEMBLER DEFINIT	CMOI	4		4				9					. 4.3
	ASSEMBLER ROUTINE	S .			-	2 01		20	120	2 N		200 200	100 S	4.2
**************************************	ASSEMBLER TOOLKIT	V/	2	11011	100	- 18 - 10		76 940	959 80					7 4
	AXIS		-			•		•	i.		•			0.4.0
	BACKDIYME	•	*	•	-	•	•	•	3.	•	•		. 1.4	2.4.2
	DACKFLAME		•	•	•	. ,				•		2.4	.2 3.2	4.3.3
	BACKUP	KI 14		•	•			•					. 1.	1 1.3
	BASIC TOOLKIT .							•				1.1 2	2.3.3 3	. 4.1
	BASICDEMO		20		2							-	. 1.	1 3.1
	BASICDEMO ALL3D		-	0000		102 NE	20	1000 1100	70	20 00	63 55 63 55	2007		1 1
	BASICDEMOI CAMSD						-			B 8		1		
	BASTCDEMO2 OR 13D		•		•		-		•		•	•	•	. 1.1
	ASSEMBLER DEFINIT ASSEMBLER ROUTINE ASSEMBLER TOOLKIT AXIS BACKPLANE BACKUP BASIC TOOLKIT . BASICDEMO BASICDEMO ALL3D BASICDEMO1_CAM3D BASICDEMO2_OBJ3D BASICDEMO3_OBJ3D BASICDEMO4_OBJ3D BASICDEMO4_OBJ3D BASICDEMO5_OBJ3D		•	14	•			•	8		•	•		. 1.1
	DAGICDEMOS_OBJ3D		•	•	•			•	•		•	•		. 1.1
	BASICDEMO4_OBJ3D			*	¥		20	•	•			•		. 1.1
	BASICDEMO5_OBJ3D			4	¥									. 1.1
	BASICDEMOBOOT .								-		2 2	12	. 1.	1 3.1
	DACTOEVT													
	BASICEXT BIN .			10		70 5		350		B 3		1.5	• •••	1 1
	BASICEXTDEMO		(* •).	•	•	in .	•	•	*	•	Ť	•		
	BACICEVEDEMODOOF	· die	•	•	-		•	•	•		8	•	. <u>.</u>	1 4.1
	BASICEXT_BIN . BASICEXTDEMO . BASICEXTDEMOBOOT		٠	•	•		•	•	4			-	. 1.	1 4.1
	BOOT		•	*				•	•			•		. 1.1
	BOOT2		\$ * 8	•	¥3	9 9	•	-	•					. 1.1
	BOX		•		•		343	-	•				:	2.4.2
	BP.ABSIO			2										. 4.1
•)	BV RIP							-	20		2	1050	5 75 F	. 4.1
	BV.VVBAS		115	100			000	10				(E)	93	4 1
	CAMERA		•	1	1 1	· , ,	1 .		• •	· .	2 4	2 2	2 4 2	4 2 2
	CAMERA ADDRESS		•	. A	TI	. 4 4	. 1 2		3 4.	3.4	4.4	4 0	\$21 4.2	4.3.3
	CAMERA DEFINITION CENTRE	•	•	•		• •				•	3.1	3.2	4.3.1	4.3.2
	CENTRE	• •	₩ 48	•	1.2	2.4	. 2 3	3.2 4	4.2.	\$2D	4.2	.\$31	4.3.1	4.3.3
	CENTREOBJECT .	•		¥			•	=					4.3	2.\$40
	CHANGE :													2 4 2
	CHANGE SIZE CHANGE WINDOW				-			. 3	3.2	4.2.	\$40	4.2.	\$41 4.	2.\$43
	CHANGE SIZE			20	_	2000 20	120	9		2005 7630 220 12	B 100	70		3.2
	CHANGE WINDOW	- m		10				-		20.0			· A	2 622
	CIRCLE		•	•	•		•	•	•			3000	1 1 1	2.044
	CLEAR	•	•	•	•		283	•	•				1.1	2.4.4
	CLEAR	•	•	•	•		•	•	•			4.2.	\$26 4.7	2.\$42
	CLEAR FLAG		•	•	140		34	¥	•			12		. 3.2
	CLEAR SCREEN .	• •	•	•								4		. 3.2
	CLEAR WINDOW												4 1	2 425
	CLEAROBJECT CLIPPING				200		12	_		14 12		12	3.2 4.3	2.\$37
	CLIPPING		_					20	110011	3 2	4 3	1 4	2 \$28 4	4 3 3
	CLONE		-				•	•		~	3.0		1 . 7	1 1.3
	CLOSE		•	•		•	104	•	•		•	•		
-	CLOSE WINDOW	**	*		*		•	*	•			*	4.7	2.\$2B
. 1	CLOSE WINDOW		*					•	•				3.2 4.2	2.\$2B
	CLS	• •	•				•	•	•				. 4.7	2.\$25
	CLS	. 2.	3.3	2.	3.4	2.4	.2 2	.4.3	3.	13.	2 4	.3.5	4.3.6	1.3.7
	COMMAND	to a									82		4	4.3.7
	COMMAND WINDOW .		•		• 8						14			. 2.1
										-	-	-		

el,

OMPILED PRO	GRAM	5 4 6				920	9 <u>5</u>	727 SE				152			. 3.1
OMPILER .				200 100			5			10.00	4 1		•	100	. 3.2
ONFIGURING	•	2.40		•		•			•	•		85		7.8.0 ±	
	•	*				•			•	•	5 5	1.5			. 1.1
ONV16	•	•		. ,			4			658			*	4.3	2.\$09
O-ORDINATES		-	_								. 1.	2 2	.4.2	3.1	1 3.2
TRL		154			20	72	500								2.3.4
URRENT PLOT	LIST	г			5					880		1.5	. L	3.	
URSOR	LI,U.	Le	•	•	*	•	(E)				' _ :		4		
	¥0		¥	•					2.1	2.3		3.4			2.4.3
URSOR KEYS.	0.0		•						2.2	2.3	.1 2.	3.2	2.4	. 2 2	2.4.3
URVES										-	_	-	_		. 1.1
ATA ADDRESS	120	118 (S)	83 3		4		15 3 22	180 25	7.0	* *		e=	-	1000	4.2
EFAULT	120		100		•		•	•	•0					17	
	•	*	•	• •	2.0	•	•	• •	•6	, I	. 1 2.	3.1	2.4		4.3.1
EFAULT EDIT		•	•						8.5				3.5	. 2	2.4.1
EFAULTS_3D		***					•		33 • 8				•		. 1.1
EFAULTS DEF	3D =	. **			-	- 2	2			. 44		-		. 2	2.4.3
EFBW DEF3D	er	78 N					\$3 S	5 5 9	11.51	· ·		-	33.50	60 (00)	. 1.1
EFCM DEF3D	•	•	•	•	•		•		11.5	15	•		8.50		
	4		•	• •	•	•	•		•	•					. 1.1
EFINE	14		•				•		•		. 2.	3.1	3.2		2.\$39
EFINEOBJECT													3.2	4.2	2.\$39
EFINITION .		20 5		4 W	3.1	3.2	2 4.2	. \$40	4.2	\$41	4.2.	\$43	4.2	. \$41	3 4.3
EFTV4 DEF3D				65 00 1000					7070700			4 1 0	****		1.1
V <u>1</u>	•	•	•	•	•	•	•	• •	•	•	•	55	100		
EFTV8_DEF3D	•	•			•	•					10		•		. 1.1
ELETE	4												2.4	. 2 2	2.4.3
ELETE ALL .	*								4			2		. 2	2.4.1
ELETE OBJECT	Γ.	1923 S	2 1 2 1		1 10 10 10 10 10 10 10 10 10 10 10 10 10	- 10	1950 S	5 7597 S 7597	150	20 20	59 365 691 69		3050 S		2.4.1
ELETE PICTUI			-	ā (5)	i i	-		•		•	•		•		2.4.3
SVICE		31 4 3	•				•	•	•	•		•			
	•	*			•	•		•	•			*	•		2.4.3
[R	¥				•				•					. 2	2.4.3
IR ALL	40													. 2	2.4.1
IR DEFAULTS		040 0			- 4	20	12 1 12 1	5 7527 5 7527	10 8 <u>0</u>	W 8	5 E	89	121	. 2	2.4.3
[R OBJECT .	100	1000	20		1 10		(A)		15	5 8		100 100		3 183	2.4.1
RAWING ALGO	י אנדידי די		•	•	3.2		.\$01		400	: .	*		\$07	T	2.\$23
CHATHO WEGOT	TITLE				3 /				SHI	4	SUD	4 . /		4	4.823
ATTEND MODE				<u> </u>	~ . ~		501	7.2	. 400	T . AI .			901		
RAWING MODE		120			• •		. 501				,				3.2
RAWING ROUT	(NES			• •	•		•						. 4		
RAWING ROUT:			. 1 4		\$01		•		. \$05	4.2.	\$06	4.2.	. 4 \$07	. 1 4	3.2
RAWING ROUT:	(NES		.1 4				•			4.2.		4.2.	. 4	. 1 4	3.2 4.3.1 2.\$23
RAWING ROUT: RAWTYPE . RAWTYPE_BIN	(NES		.1 4				•			4.2.		4.2.	. 4	. 1 4	3.2 4.3.1 2.\$23
RAWING ROUT: RAWTYPE BIN RAW WINDOW	(NES		.1 4				•			4.2.		4.2.	. 4	4.2	3.2 4.3.1 2.\$23 1.1 2.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RAWY	(NES		.1 4				•			4.2		4.2.	. 4	. 1 4	3.2 4.3.1 2.\$23 1.1 2.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMMY	(NES		. 1 4				•			4.2.		4.2.	. 4	4.2	3.2 4.3.1 2.\$23 1.1 2.1
RAWING ROUT: RAWTYPE : RAWTYPE BIN RAW WINDOW RMMY : ROUTE DEFAULTS ROUTE :	(NES		.1 4				•			4.2.		4.2.	. 4	4.2	3.2 4.3.1 2.\$23 1.1 2.1 2.1 2.4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMMY	(NES		. 1 4				•			4.2.		4.2.	. 4	3.2 1 2	3.2 4.3.1 2.\$23 1.1 2.1 2.4.1 2.4.3
RAWING ROUT: RAWTYPE . RAWTYPE_BIN RAW WINDOW RAWY RAWTYPE_BIN RAW WINDOW RAWY RAWTYPE_BIN RAW	(NES		. 1 4				•			4.2.		4.2.	. 4	3.2 1 2 1 3 2	3.2 4.3.1 2.\$23 1.1 2.1 2.4.1 2.4.3 1.1 2
RAWING ROUT: RAWTYPE BIN RAW WINDOW MMY DIT DEFAULTS DITOR TER	(NES		. 1 4				•			4.2.		4.2.	. 4	3.2 1 2 1 3 2	3.2 4.3.1 2.\$23 1.1 2.1 2.4.1 2.4.3 1.1 2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMMY RAW DIT DEFAULTS RITER ROR CODE	(NES .1.		. 1 4				•			4.2.		4.2.	. 4	3.2	3.2 4.3.1 2.\$23 1.1 2.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RAW SIT DEFAULTS RITER RER CODE ROR MESSAGE	(NES .1.		. 1 4				•			4.2.		4.2. 3.4	. 4	3.2	3.2 4.3.1 2.\$23 1.1 2.1 2.4.3 1.1 2 2.3.5 2.4.1 3.2 2.551
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMMY RAW DIT DEFAULTS RITER ROR CODE	(NES .1.		. 1 4				•			4.2.		4.2. 3.4	. 4	3.2	3.2 4.3.1 2.\$23 1.1 2.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RAW SIT DEFAULTS RITER RER CODE ROR MESSAGE	(NES .1.		. 1 4				•			4.2.	\$06 1 2.:	4.2. 3.4 2.2.	. 4	3.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 33.2 2.\$51 3.2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RAWY RAWY RAWY RAWY RAWY RAWY RAWY RAW	(NES .1.		. 1 4				•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 2.3.5 2.3.5 2.4.1 3.2 2.4.3
RAWING ROUT: RAWTYPE BIN RAW WINDOW RAWY RAWY RAWY RAWY RAWY RAWY RAWY RAW	(NES .1.		. 1 4				•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3,2 1 2 1 3 2 2 4.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 3.2 2.551 3.2 2.4.3 4.2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY RIT DEFAULTS ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE	(NES .1.		. 1 4				•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3,2 1 2 1 3 2 1 3 2 4 2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 3.2 2.3.5 2.4.1 3.2 2.4.3 4.2 3.2 3.2 3.2 3.2 3.2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY	(NES .1.						•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3,2 1 2 1 3 2 2 4.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 3.2 2.4.1 3.2 2.4.3 4.2 3.3.2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY	INES .1.	1 4					•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3,2 1 2 1 3 2 1 3 2 4 2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 3.2 2.4.1 3.2 2.4.3 4.2 3.2 4.2 3.2 4.2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY	(NES .1.	1 4					•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3.2 1 2 1 3 2 1 3 2 4.2 2 3 3 3 1	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 2.3.5 2.4.1 2.3.5 2.4.1 3.2 2.3.5 2.4.1 4.2 4.3 4.2 4.3 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4 4.4
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY ROR CODE ROR MESSAGE ROR TRAPPIN C . ECLIST . ST MODE . LENAME . LEPOS	INES .1.	1 4					•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3.2 1 2 1 3 2 1 3 2 4.2 2 3 3 3 1	3.2 4.3.1 2.\$23 1.1 2.1 2.4.1 2.4.3 1.1 2 2.3.5 2.4.1 2.3.5 2.4.1 2.3.5 2.4.1 2.3.5 2.4.1 2.4.3 2.4.4 2.4.3 2.4.4 2.4.3 2.4.4 2.4 2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST . ST MODE . LENAME . LEPOS . LES ND	INES .1.	1 4					•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3.2 1 2 1 3 2 1 3 2 2 2 3 3 3 3 1	3.2 4.3.1 2.\$23 1.1 2.4.3 2.4.3 2.3.5 2.3.5 2.4.1 2.4.3 3.2 2.4.3 4.2 3.2 2.4.3 4.1 2.4.3
RAWING ROUTE RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN ROR CODE ROR CODE ROR MESSAGE ROR TRAPPIN ROR TRAPPIN ROTE CODE ROR MESSAGE ROR TRAPPIN ROTE CODE ROTE COD	INES .1.	1 4					•			4.2.	\$06 1 2.:	4.2. 3.4	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3.2 1 2 1 3 2 2 4.2 3.3 3.1 4.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 2.3.5 2.4.1 2.3.5 2.4.1 3.2 2.4.3 4.2 3.2 4.1 4.1 3.2 4.1 3.2 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN ROR CODE ROR CODE ROR MESSAGE ROR TRAPPIN ROR TRAPPIN ROT TRAPPIN RO	INES .1.	1 4					•			4.2.	\$06 1 2.:	4.2. 3.4	. 4	3.2 1.2 1.3 2.1 2.2 2.3 3.1 4.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.2.4.3 2.3.5 2.3.5 2.3.5 2.4.1 3.2 2.4.3 4.2 3.2 4.1 4.1 3.2 4.1 4.1 3.2 4.1
RAWING ROUTE RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN ROR CODE ROR CODE ROR MESSAGE ROR TRAPPIN ROR TRAPPIN ROTE CODE ROR MESSAGE ROR TRAPPIN ROTE CODE ROTE COD	INES .1.	1 4					•			4.2.	\$06 1 2.:	4.2. 3.4	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3.2 1.2 1.3 2.1 2.2 2.3 3.1 4.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 2.3.5 2.4.1 2.3.5 2.4.1 3.2 2.4.3 4.2 3.2 4.1 4.1 3.2 4.1 3.2 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN RAW WINDOW RAWTYPE BIN ROR CODE ROR CODE ROR MESSAGE ROR TRAPPIN ROR TRAPPIN ROT TRAPPIN RO	INES .1.	1 4					. \$03	4.2		4.2.	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3.2 1.2 1.3 2.1 2.2 2.3 3.1 4.2	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 1.1.2 2.3.5 2.3.5 2.4.3 2.4.3 3.2 4.2 3.2 4.1 4.1 3.2 4.1 4.1 4.1 3.2 4.1 4.1 3.2 4.1 4.1 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY RIT DEFAULTS ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE LENAME LEPOS LES ND NDMAX AG NCTIONS NCTION KEYS	INES .1.	1 4					•	4.2	. \$05	4.2. 	\$06 1 2.4 2 2.4	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 4.3.1 2.3.1 2.4.1 2.4.3 2.3.5 2.3.5 2.3.5 2.4.3 3.2 4.2 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY RIT DEFAULTS ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE . LENAME . LEPOS LES ND NDMAX AG NCTIONS NCTION KEYS ACKPLANE .	INES .1.	1 4					. \$03	4.2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 2.3.5 2.4.1 2.3.5 2.4.1 3.2 2.4.3 3.2 4.1 4.1 2.4.1 3.2 4.1 4.1 4.1 3.2 4.1 4.1 3.2 4.1 4.1 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY RIT DEFAULTS ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE LENAME LEPOS LES ND NDMAX NCTIONS NCTION KEYS ACKPLANE . V.RIP .	INES .1.	1 4					. \$03	4.2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 2.3.5 2.4.1 2.3.5 2.4.1 3.2 2.4.3 3.2 4.1 2.4.1 3.2 4.1 2.4.3 3.2 4.1 4.1 3.2 4.1 3.2 4.1 4.1 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY	INES .1.	1 4					. \$03	4.2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 4.3.1 2.3.1 2.4.1 2.4.1 2.3.5 2.4.1 2.3.5 2.4.1 3.2 3.2 4.1 3.2 4.1 3.2 4.1 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY	INES .1.	1 4					. \$03	4.2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 4.3.1 2.\$23 1.1 2.4.1 2.4.3 2.3.5 2.4.1 2.3.5 2.4.1 3.2 2.4.3 3.2 4.1 2.4.1 3.2 4.1 2.4.3 3.2 4.1 4.1 3.2 4.1 3.2 4.1 4.1 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY	INES .1.	1 4					. \$03	4.2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 4.3.1 2.1.1 2.1.1 2.4.1 2.3.5 2.4.1 2.3.5 2.4.1 2.4.3 2.4.3 2.4.3 2.4.1 2.4.1 2.4.3 2.4.1 2.4.3 2.4.1 2.4.3 2.4.1 2.4.3 2.4.3 2.4.1 2.4.3 2.4.3 2.4.1 2.4.1 2.4.3 2.4.1 2.4.1 2.4.3 2.4.1 2.4.3 2.4.1 2.4.1 2.4.3 2.4.1 2.4.3 2.4.1 2.4.3 2.4.1 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.1 2.5.4 2.5.
RAWING ROUT: RAWTYPE BIN RAW WINDOW MMY DIT DEFAULTS DITOR TER ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE LENAME LEPOS LES ND NDMAX AG NCTIONS NCTION KEYS ACKPLANE . V.RIP . V.VVBAS . AMADDR . AMCEN	INES .1.	1 4					. \$03	4.2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 3.1 3.1 3.2 4.1 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.1 2.4.3 2.4.1 2.4.3 2.4.1 2.4.3 2.4.1 2.4.3 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW MMY DIT DEFAULTS DITOR TER ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE LENAME LEPOS LES ND ND NCTIONS NCTION KEYS ACKPLANE . V.RIP V.VVBAS . AMADDR AMCEN AMCEN	INES .1.	1 4					. \$03	4.2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 3.1 3.2 3.1 2.3.1 2.4.1 2.4.3 2.3.5 2.4.1 2.3.5 2.4.1 2.4.1 2.4.3 2.4.1 2.4.1 2.4.3 2.4.1 2.4.3 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW MMY DIT DEFAULTS DITOR TER ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE LENAME LEPOS LES ND NDMAX AG NCTIONS . NCTION KEYS ACKPLANE . V.RIP V.VVBAS . AMADDR . AMCEN . AMCEN . AMCENY .	INES .1.	1 4					. \$03	2 2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 4.3.1 2.4.1 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.3 2.4.1 2.4.1 3.2 3.2 4.1 3.2 4.1 3.2 4.1 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2
RAWING ROUT: RAWTYPE BIN RAW WINDOW RMY RIT DEFAULTS ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST ST MODE LENAME LEPOS LES ND NDMAX AG NCTIONS NCTION KEYS ACKPLANE V.RIP V.VVBAS AMADDR AMCEN AMCENY AMCENZ	INES .1.	1 4		1.2.	\$01 	4.2	. \$03	2 2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 3.1 3.2 3.1 3.1 2.3.1 2.4.3 2.4.3 2.3.5 2.4.3 2.4.3 2.4.3 2.4.3 2.4.1 2.4.1 3.2 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1
RAWING ROUT: RAWTYPE BIN RAW WINDOW MMY DIT DEFAULTS DITOR TER ROR CODE ROR MESSAGE ROR TRAPPIN C ECLIST . ST MODE . LENAME . LEPOS LES ND NDMAX AG NCTION KEYS ACKPLANE . V.RIP . V.VVBAS . AMADDR . AMCEN . AMCEN . AMCEN . AMCENY .	INES .1.	1 4			\$01 		. \$03	2 2	. \$05	4.2. 	\$06 1 2. 2.4. 2 2. 3 4.	3.4 2.2.4.1	. 4 \$07 . 2 . 4 . 3 . 2 . 4	3,2 1,2 1,3 2,1 2,2 2,3 3,3 3,1 4,2 5,4	3.2 3.1 3.2 3.1 3.2 4.1 2.4.3 2.4.3 2.4.3 3.2 3.2 3.2 3.2 4.1 4.1 4.1 4.1 4.1 4.1 4.1 4.1

67534802	GCAMY			() =	20	(10)			12	2	74			-							3.2
	GCAMYA			124	2	320	10	9423	82	21	80	25 25	1000		Fig. 193	100	20	3250	100		3.2
1	GCAMZ			-	-		•	•		•	1.5	•		•	•	•	•	•		•	
100			•	•		•	123		•	•		•	•	•	•	•	•	•	•	•	3,2
	GCAMZA				-	150	2	•			14	-								*	3.2
	GCENTREO	BJ .	-		10	34	25	120	2	1023	12	23	1001		28	12	2	7527	8		3.2
	GCHANGEO	B.I								76	6	81	154		- 5		- 50	1970	· ~	4	3.2
	GCLEAR			•	•	-		•	•	•	7	•	•	•	•	107	*	•	3.	137.71	
						-	•		•	•	•	•	•	•	•			•	З.	1	3.2
	GCLOSE		11-11		20	14				7.4		40							· ·		3.2
	GCLS .		312																3.	4	3.2
	GDATAADD:	n .			10.00		•	•	•	•	•	•		•		•	•		٥.	ı	
			•		-	•					•	•		•				•		•	3.2
	GDEFINEO:	BJ .	- 2	¥			20				•										3.2
	GDRAWTYP	E .	52	107				* *										. 2	.з.	2	3.2
	GERROR			•	0.5		•	100	5	650		•	•	3			*	. 4		2	5 <u>0</u> 0500 <u>50</u> 40
		• •	•	-		2	•	•	•	•	•		•	**	•		•		•	•	3.2
	GFACTOR						•										*:			4	3.2
	GFINDLIN			-	154			-	20	323	0	-	12	2		_	20	74	_		3.2
	GFINDMAX	711 7					751	- 6	58	350	ē	7 . Z k	15 10		0.58		7.3			8.	3.2
	GFINDPOI	il .	•	•	•	•	•	•	•	•	•	•	•	•	85.8		*:	•	•	•	200 (A.A
				•	34			•	•	. •	•		•	•	•	•	•	•	•		3.2
	GFINDVER:	rex	*	48	•														្		3.2
	GINK .			25.45	14	20	190	12	5,20	32	63	1986	55	50	STATE OF THE STATE	50 50	10590	200			3.2
	GINK .		-					•	1	100	2	3 5 0		5	3.5	•	•	9 0	. .		(C)
			•	•	•	•	•	•	•	•	•	•		•	•	•	•	3.4	4.	2.	. \$27
	GINSERTO	ou .			4	•0	•	*		15	-	•		•	•	•	•			•	3.2
	GLINE								844	32	2	920		20			72	12	2	727	3.2
	GLINE3D .		01	202-1	131	55027	2000	86	0845	22	- -	1947) 1947	13	100	3.553) 331	5	1555	\$3 \$4	E)	SEE SEE	3.2
	GLOADOBJ	•		•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	
			•	•	•	*			•	•	•	•	•	•	•	•		•	•	•	3.1
	GLOADOBJE	ECT			2		16	2										12			3.2
	GMAKE .		-	5000	- 2	25	-		923	12	25	31-33 31 <u>2</u> -3	205	100000 100000	50		3.1	3 2	4	2	\$42
	GMARKLINE				-			•	•	•	86	•	⊕		•	•		3.2		4.	
			•	•	•	•	4				•	•		•	•	•	•	*	•	•	3.2
	GMARKOBJ		48	•		•				•	•										3.2
	GMARKPOIN	VT .	40				15	40			100	75				2		· ·	2	523	3.2
	GMARKVERT	CEX	20	-	101										127	- ES	\$550 5000	- A	20	2000 2000	3.2
	GMODE .		•		•		•	•	•	•			e.		•	*	•		•	•	
		•	•	(A	•	•	•	•	9		•	•	•	•	•	•	•		•	•	3.2
	GMOVE .	•	1 0	•	•	•	12	•		4	•			•	4		4	•	•		3.2
	GNOOBJ .				**	•	7.	21				_				25	200	9 8			3.2
	GNOOBJMAX	ς.	2020		20	1000											3.50	- EE		300	3.2
	GOBJADDR		1.0-1	y. =	-	A. • 12	•	•		•			•		*	•	0.00		•	•	
		•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•	•	3.2
	GOBJMAX .		0.00			•	-		9.155	9		36						E 7	-		3.2
	GOBJX .				alfr.	1.40				2			4	140	2		12	2 1		200	3.2
	GOBJXA .											25	2	10 <u>5</u> 2	- E	- 100 - 100 - 100	200		8		3.2
	GOBJY .			•	•	0.40	•	•					•		•	•	•		•		23649.5
		•		•	•	84	•	•	•	•		•	•	•	•	•	•	•	•	•	3.2
	GOBJYA .		•		₩1	949	*			2	•							5 7			3.2
	GOBJZ .		0.40	~	•	18	_	920		21	1000	22	27	10050 10061	200	48) 1883	080	50 H	253 253	CHAIN CHAIN	3.2
	GOBJZA .												86			•		5	•		3.2
1	COLLED			•	•	•							4	•	•	•	•		•		
		11.	•		•	14		•					•		•	•	•	93:			3.2
	GPAPER .	•	•		•	19	¥	350		2		12					•	3.2	4.	2.	\$26
	GPLACE .	-				22		0.22	10	21	9948		20	1050 1050							3.2
	GPLOT .	100-													5	٠,					
		*1	•		4		-		*				•	•	•		1.1	3.2	4.		
	GPOINT .		19		•			•			•	2	•	•			•	4	•	•	3.2
	GPOINT3D	•		*	0.40				4	-											3.2
	GREMOVELI	NE	90	300	2-8027	400															3.2
	GREMOVEOR		17	•		•	•		•	•	1.00							•	•	1300	58 59
				•	•					•		•	-	•	•	•	•	•	•		3.2
	GREMOVEPO						*	4	¥		**				•	•	•	•			3.2
	GREMOVEVE	RTEX	34							27	75601	2	20	52.0	_	20			20	-21	3.2
	GRID							•				į.		-		30	-	1	•		4.3
	GROTATE .	•		•		•								: -	• -	•					
	- 15 Mail 18 15 Mail 17 (17 (17 (17 (17 (17 (17 (17 (17 (17		3	*								¥		٦.1	. Э,	4	Ł.Z.	\$2D	4.	4.	10.0
	GROTATEVE			*					*		8		•	9						•	3.2
	GROTATEVE	RTEX	X		3440			100		850		-	25	526	12	60 20	50 500	2	84 25	2552 0353	3.2
	GROTATEVE						-								5	1		2	16		3.2
				•	•	•	*6	***										•			
	GROTATEVE	KIEX	6	*	•		•	•											•	٠	3.2
	GROTATEX	3 .		*	a•03		•	14											•		3.2
	GROTATEY				-		_	1200										15	10 40		3.2
	GROTATEZ	12.5%	1.5	*	6.76	-	*1											•	•		
			•	3.*3		•	•	•	•	•	•	•				•	•	₩ 8	•		3.2
)	GRXVALUE			•	(•)			•	*			2				-		€ 8	-		3.2
1	GRYVALUE								•				4						2		3.2
	GRZVALUE					-				•											3.2
	GUTANTOR				(1.50 pt)	-	3.7	3. 00	7 .			•	•				•				
		8 3 4																			2 2
	GSAVEOBJ	•	•	•								•1		•	•	•		* s	•		3.2
		•	•										•			•		•			3.2

SSETLINE .			2											3.2	•
SSETPOINT .						8 5 6 6				•		•		- 1 <u>- 1</u>	1
SETVERTEX .	2	12	10		100 A	141			•						
SSIZEOBJ .	25	100			50	1000	•		•	•		•			
START	98	-	29	- 10 E	₩.	•	•		•	• •			3.1	A STATE OF THE PARTY OF THE PAR	
SWAP	•		•			2.6				. 3.	1 3.2		L 4.2		
TRAPERROR .	•		8		76	•			1	¥ .		3.2	2 4.2	.\$42	
VARCAMCENX	•	•	•		% ₹3			*	19	2		•		3.2	
	•	•			•		6 94			¥		•	. 4	.3.1	
VARCAMCENY	•	*	•		1.00				14				. 4	.3.1	
VARCAMCENZ	•		•		•					g :		-	. 4	.3.1	
VARDISTX .	-	•	•		1							120	. 4	.3.3	
VARDISTX16			•											.3.3	
VARDISTY .		•	•		•						s 20	11.50 11.50		.3.3	-1
VARDISTY16		de.			. 8					828	2 46	1000		.3.3	
VARDISTZ .									1.5	70 - 07				.3.3	
VARDISTZ16	i.				4				•	5.4.5	• •	•		.3.3	
VARFLAG .	2		84						•		•			.3.3	
VARHLIM .	4			2 12	8 2			6 8 6 (8	•	(*)	• •	•		.3.3	
VARLINETAB	2	100			•	å - 16			•	•	• •				
VARNL		88953	- 15 10			S	•	•			•	•		.3.3	
VARNLM	•	•	*		• •	3 3	*		•1			•		.3.3	
VARNP	•	S C.	•			•	•11			•	•	4	1000	.3.3	
VARNPM	•	•	•	•			*0	•						.3.3	ę
VARNY	•	•	•	• •			•		*	* .		¥	. 4	.3.3	
200	*	•	•		•	8 .	S + 3		S. (3)	× .			. 4	.3.3	
VARNVM	•	-	•				9.68						. 4.	.3.3	
VAROBJLEN .	•	-	*				(c % 2)			8 9	E 15		. 4.	.3.3	
VARPLOTTAB1	•		•						8.				. 4.	3.3	
VARPLOTTAB2	•		•						120			2	. 4.	3.3	
VARPOINTTAB	•											20		3.3	
VARRESTAB .	(*)	*	20				5. - 00	e et						3.1	
VARSF							85.0		1. 5		٠,	2.\$		3.1	
VARSIZE .			180	60 61 12 23	5 S	100	- 12	2 310			- T.	. 2		3.3	3
/ARSX		-	100	of B	100	8.5). 		•	•	•	•			
/ARSY	12	- -	1900 1900			•	*	•	•		•	•		3.3	
/ARSZ	-	•				0.0				575					
		2.0					1.2		-	•	•	•		11000	
/ARVERTEXTAR	₩	•	-			188							. 4.	3.3	
/ARVERTEXTAB				· sp			* !gs						. 4.	3.3	
/ARX		•		• 4). • 4).	: :		• '13* •	· (*)	•				. 4. . 4.	3.3	
/ARX /ARX16		:	•	· •			A tax						. 4. . 4.	3.3 3.3 3.2 3.2	
/ARX /ARX16 /ARXA		:		- 412								•	. 4. . 4. . 4.	3.3	
/ARX /ARX16 /ARXA /ARY						•							. 4. . 4. . 4.	3.3 3.3 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY16													. 4. . 4. . 4. . 4.	3.3 3.3 3.2 3.2 3.2	d.
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA	•					**************************************							. 4. . 4. . 4. . 4.	3.3 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ		•							•				. 4. . 4. . 4. . 4. . 4.	3.3 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZ16			-		138 3350	(E)			•				. 4. . 4. . 4. . 4. . 4. . 4.	3.3 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZ16 /ARZA		•	-			•	• •	•	•				. 4. 4. 4. 4. 4. 4. 4. 4.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZ16		•	2			•			•				4. 4. 4. 4. 4. 4. 4. 4.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZ16 /ARZA						•	•		•				4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZ16 /ARZA /ARZA /INDOW						•			•				4. 4. 4. 4. 4. 4. 4. 4. 4.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZA /INDOW	INDO					•			•				4. 4. 4. 4. 4. 4. 3.1	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ /ARZ16 /ARZA /INDOW /LPFILE /LP WINDOW FORMATION WI	a.					•							4. 4. 4. 4. 4. 4. 4. 4. 4.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16	14 2000	W								2.4			4. 4. 4. 4. 4. 3.1	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ /ARZ16 /ARZA /INDOW /LPFILE /LP WINDOW FORMATION WINDOW K SERT	a Timb	w						2		2.4			4. 4. 4. 4. 4. 4. 3.1	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	. (
/ARX /ARX16	14 2000	w						2		2.4	.22.2.2.3.		4. 4. 4. 4. 4. 4. 3.1	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARYA /ARZ /ARZ /ARZA /INDOW /INDOW /LPFILE /LP WINDOW FORMATION WINDOW SERT SERTOBJECT USE	A BAG	W						2		2.4	.22.2.2.3.		4. 4. 4. 4. 4. 4. 3.1	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZA /INDOW /LPFILE /LP WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD	no.							2	. 3.2	2.4	.2.2.2.2.3.	4.342.4.2	4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY16 /ARYA /ARZ /ARZ16 /ARZA /INDOW /LPFILE /LP WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD YS	A BAG							2	. 3.2	2.4	.2.2.2.2.3.	4.342.	4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ /ARZ16 /ARZA /INDOW LPFILE LP WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD YS NDER_OBJ3D	a man	W						2	. 3.2	2.4	.2.2.2.2.3.	4.342.4.2	4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ /ARZ16 /ARZA /INDOW LPFILE LP WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD YS NDER_OBJ3D	months in			2.3.3	2.3.	4 3.	2 4.2	2		2.4	.2.2.	4.342.	4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ /ARZ16 /ARZA /INDOW .LPFILE /LP WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD YS NDER OBJ3D NE 1.2 2	months in	W				4 3.	2 4.2	2		2.4	.2.2.	4.342.	4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY16 /ARYA /ARZ /ARZ /ARZ16 /ARZA /INDOW CLPFILE /ILP WINDOW FORMATION WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD YS NDER_OBJ3D NE 1.2 NE3D	months in			2.3.3	2.3.	4 3.	2 4.22.\$39	2 . \$03		2.4 .3.2 .\$06	.2.2.2.3.		4. 4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2. 4.3 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZA /INDOW CLPFILE CLP WINDOW FORMATION WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD YS NDER_OBJ3D NE 1.2 NE3D NE CLEAR	i mo		.1	2.3.3	2.3.		2 4.22.\$39	2 . \$03		2.4 .3.2 .\$06	.2.2.		4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZA /INDOW /IND	in i		.1	2.3.3	2.3.		2 4.22.\$39	.\$03	.3.2	2.4	.2.2.2.3.2.3.3.53		4. 4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2. 4.2. 4.2. 4.2. 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZA /INDOW /INDOW /LPFILE /LP WINDOW FORMATION WINDOW FORM	in in it		.1	2.3.3	2.3.	4 3 .	2 4.22.\$39	2 . \$03	.3.2	2.4	.2.2.2.3.	4.342.4.2.34.2.	4. 4. 4. 4. 4. 4. 4. 3.1 4.2. 4.2. 4.2. 4.2. 4.2. 4.2. 4.2.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZA /INDOW /IND	in in it		.1	2.3.3	2.3.	4 3.	2 4.22.\$39	2 . \$03 4.2.			.2.2.2.3.2.3.3.553		4. 4. 4. 4. 4. 4. 4. 3. 1 4. 4. 3. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZA /INDOW /INDOW /LPFILE /LP WINDOW FORMATION WINDOW FORM	1.1		.1.	2.3.3	2.3.	4	2 4.22.\$39	2 . \$03 4.2.		2.4 .3.2 .\$06 4.2	.2.2.2.3.2.3.3.553		4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4	3.3 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2 3.2	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZ16 /ARZA /INDOW LPFILE LPFILE LPFILE SERT SERTOBJECT USE YBOARD YS NDER OBJ3D NE NDER OBJ3D NE NE3D NE CLEAR NE INK NE START VE TABLE			.1	2.3.3	2.3.	4.	2 4.22.\$39	2 . \$03 4.2.					4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4	3.3 3.2 3.2 3.2 3.2 3.3 3.2 2.2 2.2 2.1 2.1 2.1 2.1 2.2 2.2 2.3 3.3 2.2 2.2 2.3 2.3 2.3 2.3	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZ16 /ARZA /INDOW LPFILE LP WINDOW FORMATION WINDO			.1.	2.3.3	2.3.	43.	2 4.22.\$39	2 . \$03 4.2.		2.4 .3.2 .\$06 4.2			4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4	3.3 3.2 3.3 3.2 3.3 3.3 3.3 3.3 3.3 3.3	
/ARX /ARX16 /ARXA /ARY /ARY /ARY /ARY /ARZ /ARZ /ARZ16 /ARZA /INDOW LPFILE LP WINDOW FORMATION WINDOW FORMATION WINDOW SERT SERTOBJECT USE YBOARD YS NDER OBJ3D NE 1.2 NE3D NE LPAR NE INK NE START VE TABLE NK FLAG			.1.	2.3.3	2.3.	4.	2 4.22.\$39	2 . \$03 4.2.		2.4 .3.2 .\$06 4.2			4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4	3.3 3.2 3.2 3.2 3.2 3.3 3.2 2.2 2.2 2.1 2.1 2.1 2.1 2.2 2.2 2.3 3.3 2.2 2.2 2.3 2.3 2.3 2.3	

-	IOAD DICTUDE	7											
30	LOAD PICTURE	76		-									2.4.3
	LOADING	-				21				323	1	1 2 1	3.1 4.1
	LOADOBJECT .										4 1 2	2 4 1	4.2.\$3B
	MAKEOBJECT .	7.5		•			*	•		, Z	.4.1 J	.4 4.1	
		•		•		864				1.5			4.2.\$35
	MARKING	•		39		140	28	323	8	110		3 2	4.2.\$42
	MARKOBJECT .	0020						180	š (#)	10.	•		
	MAXIMUM	7,07	•		•	•		•		18	•	× •	4.2.\$42
	1100 100 10 10 10 10 10 10 10 10 10 10 1		8 7	4									. 3.2
	MEMORY	440		82	-	757						2 1 2	3.2 3.1
	MEMTOP			-			•	·				2.1 2.	J. 4 J. 1
	MENU				•		50		•				3.1 4.1
	198742110 800 1100 1100 1100 1100 1100 1100 1	15		12					•				2.2 2.4
	MIRROR MODE			10	22 94	12 6					A		2 4 2
	MODE			•	26			•		1 .	1 .		2.4.2
	MOUE				20		-		2.3.3	2.4	.3 3.2	4.3.1	4.2.\$24
	MOVE				1.0		2.1	2.3.4	4 3.2	4.2	. \$32 4	. 2. \$2D	4. 2 \$2E
	MOVE STEP .								TO 1070100 TON		. 532 4		
	MOVECAMERA			•			•		100	5. 3			2.4.3
			•	•	•								4.2.\$2E
	MOVEOBJECT .			41			161		20 <u>-</u>	_			4.2.\$32
	OBJECT ADDRE	SS.							70 0 5				4.2.\$34
	OBJECT CONTR	OT.		•	•				•		4	. 2 . \$21	4.2.534
	OBJECT CONTR	OP -		•									2.4.2 2 4.3.3
	OBJECT DEFIN		₹ .		14		- 2	21 10	60 12	2	3 1 3	2 4 3	2 4 3 3
	OBJECT TABLE OBJECT 1.1	_					Ø.	B) (6		**	J. 2 J	.2 1.5.	4 1.5.5
	OBIECT 1 1	1 2	2 2				. :	٠. ١	•				. 3.1
	OBJECT 1.1	1.2	2.3.	1 2	.3.2	2.3.	3 2.	3.42	2.3.5	2.4	.2.2.4	.3 3.2	4.2.\$31
)		4.	2.\$3	2 4	.2.\$3	35 4.	2 \$3	6 4 2	\$ \$37	4 2	638 4	2 627	4.2.\$3B
1.054				- 4	2 44	3D 4	0 40	7 4 2	1.007	4.4	. 220 -*	. Z. SSA	4.4.530
	OVER			4	. 4.53	3B 4.	2.530	3 4.2	2.\$3C	4.2	.\$3F 4		4.2.\$41
	OVER	0. €0						700				2 4 3	4.2.\$29
	PAPER							- 56 - 65					
	PAPER	3 .			•		•		•	10.5			2.4.3
				•		941 94	•			920		. 3.2	4.2.\$26
	PARAMETER PA	ssing				(42)	22	920	2	101			. 3.2
	PICTURES .					1920 12							
		ENER OTTO		•				٠ .					2.4.3
	PIXEL CO-ORD	LNAIE	δ.		*		20		•				4.3.1
	PLACE						20			565		4 2 \$0	2 4.3.7
	PLACE3D				-	•	-	120 0	•	•	10.51		
	PLOT	•		•	•		•		•				4.2.\$06
	PLOI			•	•		45		1. 2			. 3.2	4.2.\$27
	PLOT AREA .				20		0.27			. 2.	4 3 3	.1 3.2	4 2 6 A B
	PLOT LIST .			-	2 1	່ວ່າ	4 2 /		A .	. 4	. 4.3	4.2.\$3	1.2.34A
			* (*)			7 /	4 . 4 . 1	333 4	. Z . S 3	h 4	7 647	4 2 53	N 12 N
					V. 1	0.2						1.1.00	0 4.3.4
	PLOT PICTURE												
	PLOT PICTURE PLOT TABLE .		. ,.	٠									2.4.3
	PLOT TABLE .		· j·		•	in in						 . 4.3.	2.4.3 1 4.3.7
	PLOT TABLE . PLOTFREE .		· · · · ·		•							 . 4.3.	2.4.3
	PLOT TABLE . PLOTFREE . PLOTOBJECT .		A _{ic}	•	•	in in						4.3.	2.4.3 1 4.3.7 4.2.\$4A
	PLOT TABLE . PLOTFREE . PLOTOBJECT .		A _{ic}	•	•	in in						4.3.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE		i die		•	•	•	•	•		•	4.3.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE		i die		•	•	•	•	•		•	4.3.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36
· *	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE				•				•			. 4.3.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER POINT				•				•			4.3. 4.3. 4.3. 4.3. 4.3. 4.3.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE . PLOTTER . POINT . POINT .									1.2	2.3.2	4.3. 4.3. 4.3. 4.3. 4.3. 4.3.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05
)	PLOT TABLE . PLOTFREE PLOTOBJECT . PLOTTING MODE PLOTTER POINT POINTSD . POINT CLEAR									1.2	2.3.2	4.3. 4.3. 2.4. 2.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51
)	PLOT TABLE . PLOTFREE PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT STATE . POINT CLEAR POINT INK									1.2	2.3.2	2.4.2	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2
2	PLOT TABLE . PLOTFREE PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT STATE . POINT CLEAR POINT INK									1.2	2.3.2	2.4.2	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2
2	PLOT TABLE . PLOTFREE PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT STATE . POINT CLEAR POINT INK									1.2	2.3.2	2.4.2	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT . POINT CLEAR POINT INK . POINT TABLE .									1.2	2.3.2	2.4.3 2.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2
2	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT . POINT CLEAR POINT INK . POINT TABLE .									1.2	2.3.2	2.4.3 2.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT . POINT CLEAR POINT INK . POINT TABLE .									1.2	2.3.2	2.4.3 2.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$5	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$5	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
)	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
2	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$5	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$5	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$5	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE . PLOTFREE . PLOTOBJECT . PLOTTING MODE PLOTTER . POINT . POINT CLEAR POINT INK . POINT TABLE POINT . 1 . PROCEDURES .	2 2.	1 2.3	:::::::::::::::::::::::::::::::::::::::	2.3.	2 2.:	3.3 2	.3.4		1.2	2.3.2 . 3 \$01 4 4.2.\$5	2.4.3 2.3.3 2.3.3 1.4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.3.5 4.3.7 3 4.3.7
	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINT CLEAR POINT INK POINT TABLE POINT TABLE POINT TABLE POINT TABLE POINT TABLE PROGRAM CONTE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVE MUL REMOVE ONE REMOVE VERTEX RESERVE PLOT RESERVING WOR	2 2. OL EDITO	1 2.3 OR	· · · · · · · · · · · · · · · · · · ·	2.3.	2 2	3.3 2	3.4	3.2 4.2.	1.2 	2.3.2	2.4.3 2.3.3 1 4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3.7 2.4.3 3.5 3.4.3 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3
	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINT CLEAR POINT INK POINT TABLE POINT TABLE POINT TABLE POINT TABLE POINT TABLE PROGRAM CONTE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVE MUL REMOVE ONE REMOVE VERTEX RESERVE PLOT RESERVING WOR	2 2. OL EDITO	1 2.3 OR	· · · · · · · · · · · · · · · · · · ·	2.3.	2 2	3.3 2	3.4	3.2 4.2.	1.2 	2.3.2	2.4.3 2.3.3 1 4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3.7 2.4.3 3.5 3.4.3 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3
2	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINT CLEAR POINT INK POINT TABLE POINT TABLE POINT TABLE POINT TABLE POINT TABLE PROGRAM CONTE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVE MUL REMOVE ONE REMOVE VERTEX RESERVE PLOT RESERVING WOR	2 2. OL EDITO	1 2.3 OR	· · · · · · · · · · · · · · · · · · ·	2.3.	2 2	3.3 2	3.4	3.2 4.2.	1.2 	2.3.2	2.4.3 2.3.3 1 4.3.4 2.\$05 52 4.3.3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3.7 2.4.3 3.5 3.4.3 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3
	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINT CLEAR POINT INK POINT TABLE POINT TABLE POINT TABLE POINT TABLE POINT TABLE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVE MUL REMOVE ONE REMOVE VERTEX RESERVE PLOT RESERVING WOR RESET RESULT	2 2. OL EDITO	1 2.3 CE		2.3.	2 2	3.3 2	3.4	3.2 4.2.	4.2.	2.3.2	2.4.3 2.3.3 3 1.4.3.4 2.\$05 4.2.\$05 4.3.3 4.2.\$05	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$29 3 4.3.7 4.2.\$05 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3 3.4.3 3.4.3 3.4.1 2.4.3 3.4.3 3.4.3 3.4.1 2.4.3 3.4.2.\$52 4.2.\$53 4.2.\$53 4.2.\$52 4.2.\$52 4.2.\$52 4.3.5 5.2.\$52 4.3.5 5.3.6
	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINT CLEAR POINT INK POINT TABLE POINT TABLE POINT TABLE POINT TABLE POINT TABLE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVE MUL REMOVE ONE REMOVE VERTEX RESERVE PLOT RESERVING WOR RESET RESULT	2 2. OL EDITO	1 2.3 CE		2.3.	2 2.:	3.3 2		3.2 4.2.	1.2 	2.3.2 .3 \$01 4 4.2.\$!	2.4.3 2.3.3 3 2.\$05 2.3.3 2.\$05 2.3.3 2.\$05 4.2.\$05 2.3 2.3 2.3 2.3 2.3 3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$51 4.2.\$51 4.2.\$05 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3 3.4.3 3.4.3 3.4.3 3.2.4.3 3.4.3 3.2.\$53 4.2.\$53
	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINTSD POINT CLEAR POINT INK POINT TABLE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVEOBJECT REMOVE ONE REMOVE ONE REMOVE ONE REMOVE ONE REMOVEVERTEX RESERVE PLOT RESERVING WOR RESET RESULT ROTATE 2.3.	2 2. OL EDITO	1 2.3 OR		2.3.	2 2.:	3.3 2		3.2 4.2.	1.2 	2.3.2 .3 \$01 4 4.2.\$! 	2.4.3.3.4.2.\$05.4.3.3.4.2.\$05.4.3.3.4.4.2.\$4.3.4.4.4.2.\$4.3.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$51 4.2.\$51 4.2.\$55 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.2.\$53 4.2.
	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINT CLEAR POINT INK POINT TABLE POINT TABLE POINT TABLE POINT TABLE POINT TABLE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVE MUL REMOVE ONE REMOVE VERTEX RESERVE PLOT RESERVING WOR RESET RESULT	2 2. OL EDITO	1 2.3 OR		2.3.	2 2.:	3.3 2		3.2 4.2.	1.2 	2.3.2 .3 \$01 4 4.2.\$! 	2.4.3.3.4.2.\$05.4.3.3.4.2.\$05.4.3.3.4.4.2.\$4.3.4.4.4.2.\$4.3.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.4.	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$51 4.2.\$55 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.2.\$53 4.
	PLOT TABLE PLOTFREE PLOTOBJECT PLOTTING MODE PLOTTER POINT POINTSD POINT CLEAR POINT INK POINT TABLE PROGRAMMING QUOS QUITTING THE REMOVELINE REMOVEOBJECT REMOVEOBJECT REMOVE ONE REMOVE ONE REMOVE ONE REMOVE ONE REMOVEVERTEX RESERVE PLOT RESERVING WOR RESET RESULT ROTATE 2.3.	2 2. OL EDITO	1 2.3 OR	1	2.3.	2 2.:	3.3 2		3.2 4.2.	1.2 	2.3.2 3 \$01 4 4.2.\$! 1 3.2 \$0E 4 \$4E 4	2.4.3 2.3.3 3 2.\$05 2.3.3 2.\$05 2.3.3 2.\$05 4.2.\$05 2.3 2.3 2.3 2.3 2.3 3	2.4.3 1 4.3.7 4.2.\$4A 4.2.\$36 4.2.\$51 4.2.\$55 2.4.2 2.4.2 4.3.5 4.2.\$39 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.4.3 3.2.\$53 4.2.\$50 6.

			-35												
ROTATECAMERA .	12													4.2.\$2D	
ROTATEINIT16 .		•		1		-	3.			•	•	•	•		
			•	•		•				٠	-	•		4.2.\$OA	
ROTATEOBJECT .		3 0 0							4.2	2.\$3	14	. 2 .	\$40	4.2.\$41	
ROTATEVERTEX .	12	20 0		920		20	92		•					4.2.\$4D	
ROTATEVERTEXX .				1920 1920	- 47	10	100		18	50		Č.	3 5 3	4.2.\$4E	
ROTATEVERTEXY .	•	•4						•	25	11.5	*	•	•	선거가 아내는 내용 아이들이 아니는 아이들이 되었다면 하는 것.	
	-	¥0 0				- 1 - 1				3. 	*	•	•	4.2.\$4F	
ROTATEVERTEXZ .														4.2.\$50	
ROTATEX16	26									11011	1	2	COD	4.2.\$OC	
ROTATEY16	•			•	3	•				•	• 7	. 4.	, \$ OD	4.2.300	
	•			•		•	•		•	•				4.2.\$OD	
ROTATEZ16	*	1941 0	20	•	4						. 4	. 2 .	\$OB	4.2.\$0E	
ROTATION	20	020 0	2 20		86		89	50 St	50					3.4 3.2	
ROUTINES	-		18			188	·		*	853	•	1.	2 2		
•	•	-		-	- 5	•	•		•	1655		•		. 4.2	
RUBBER LINE	-		100	•						4		9.5		2.3.3	1
CUNTIME EXTS .	4.				1		9	21 12	_		_	_	_	. 1.1	
GAVE ALL						1356	- 2	10 10 10 10		1758 2007			4 5	2.4.1	
SAVE DEFAULTS	•			- 5	*	•	•		•	•	•	•			
	•			•	•	•	•			•			•	2.4.3	
SAVE PICTURE .				12		59								2.4.3	
SAVEOBJECT	020		9 9999	12	20				2	4	1 3	2	4 1	4.2.\$3C	
SAVING	-			- 3	59	959	ē 1	•	1		• -		• • •		
73	•	* *	•	•	•	•	•		•	•	•	•	•	. 1.1	
CALE		¥ 9									2.4	. 3	3.2	4.2.\$28	
CREEN			1 124	-		14	20 10			32	8 7		12	4.3.1	
CREEN FACTOR .				- 8					1276		` -	٠,	2 2	2 4.3.1	
CREEN MODE .							•		•	•					
	•		1.5				•				2.4	. 3	3.2	4.2.\$24	
CREEN PARAMETER	S		9 9	4			9 9					4	-	. 3.2	
CREEN WINDOWS	9200								1550	10	5) () 			. 2.1	
EARCH	3.43				•	•	•	•	•	•	•	•	•		
					0.0	•	. 1		•			•		. 3.2	
ET					100									. 3.2	
F	2421		n 100										2	2 2.4.3	
				•	•		•	•	•	•		•			
	•		•	*			•		•	•		•		2.4.3	
IZE											. 2	. 4.	2 3.	2 4.3.3	
IZEOBJECT	36			5.4						<u> </u>			2	4.2.\$41	
PACE								5	(E)		2.3	1	2.3	글 경기를 하게 하면 하지 않는 것이 없었다. 그 없지 않는 것이 없다.	
PACE OBJ3D .	•					•		•	•		4.3		4.3	2 2.3.4	
TACE ODGOD .															
					•	-		•	•					. 1.1	10
TANDARD DRAWING	MOD	 E .	į		•	-		•	•					. 1.1 2.3.3	100
						-		•	•		•	•		2.3.3	\$8
TANDARD DRAWING TARTING A BASIC	PRO	GRAM				-		•	•		•	•	•	2.3.3	\$20
TANDARD DRAWING TARTING A BASIC TARTING THE EDI	PRO TOR	GRAM				-	• •	1	•			•		2.3.3 3.2 2.1 2.2	10
TANDARD DRAWING TARTING A BASIC TARTING THE EDITARTING THE EDITART	PRO TOR	GRAM				-		•	•			•	•	2.3.3 . 3.2 2.1 2.2 2.4.3	\$2.
TANDARD DRAWING TARTING A BASIC TARTING THE EDI	PRO TOR	GRAM				-				4.		. 2	\$00	2.3.3 . 3.2 2.1 2.2 2.4.3	10
TANDARD DRAWING TARTING A BASIC TARTING THE EDITARTING THE EDITART	PRO TOR TOR	GRAM	,	•					. 4.3	4.		. 2.	\$00	2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C	
TANDARD DRAWING TARTING A BASIC TARTING THE EDITARTING THE EDITART	PRO TOR TOR	GRAM	1	•					14 14	4.	l 4.	. 2.	\$00	2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1	
TANDARD DRAWING TARTING A BASIC TARTING THE EDITARTING THE EDITART	PRO TOR TOR	GRAM						•	•	4.	. 4	. 2 .	\$00	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1	**
TANDARD DRAWING TARTING A BASIC TARTING THE EDITARTING THE EDITART	PRO TOR TOR	GRAM							•	4.	l 4	. 2 .	\$00	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1	
TANDARD DRAWING TARTING A BASIC TARTING THE EDITARTING THE EDITART	PRO TOR TOR	GRAM							•	4.	l 4	. 2 .	\$00	2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TART TATION_OBJ3D TEP SIZE TIPPLE TOP	PRO TOR TOR	GRAM						•	•	4.	4.	2.	\$00	2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST)	PRO TOR TOR	GRAM						•	•	4.	. 4			2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM	PRO TOR TOR	GRAM							•	4.:	4	. 2 .	\$00	2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM	PRO TOR TOR	GRAM							•	4.:	4			2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION_OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE	PRO TOR TOR	GRAM								4.:	4			2.3.3 .3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION_OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC	PRO TOR TOR	GRAM						•		4.:	4			2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC TOOLK	PRO TOR TOR	GRAM								4.:				2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3.	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC WAPOBJECT	PRO TOR TOR	GRAM								4.:	4			2.3.3 . 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC WAPOBJECT	PRO TOR TOR	GRAM								4.:	4			2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .3.1.1 4.2.\$38	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC WAPOBJECT WAPPING PLOT LI	PRO TOR TOR	GRAM								4.:	4			2.3.3 .3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .3.1.1 4.2.\$38 .3.2	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC WAPOBJECT WAPPING PLOT LIST HOR	PRO TOR TOR	GRAM								4.:	4		2.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3. 1.1 4.2.\$38 . 3.2 . 2.2	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC WAPPING PLOT LIST HOR OOLKIT	PRO TOR TOR	GRAM								4	. 1.		2.3	2.3.3 .3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .3.1.1 4.2.\$38 .3.2	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC WAPPING PLOT LIST HOR OOLKIT	PRO TOR TOR	GRAM								4	1.1.	. 1	2.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3. 1.1 4.2.\$38 . 3.2 . 2.2	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC WAPPING PLOT LI HOR OOLKIT	PRO TOR TOR	GRAM								4.:	1.	. 1	2.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .3.1.1 4.2.\$38 .3.2 .2.2	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC WAPPING PLOT LI HOR OOLKIT	PRO TOR TOR	GRAM								4.:	1.	. 1	2.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .3.1.1 4.2.\$38 .3.2 .2.2	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION_OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3	PRO TOR TOR 	GRAM	2.4		3.1	3.2	4.2	: \$39		4.:	1.4.		2.3.	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.3.3 4.3.7 3.2.4.3 2.4.3 .3.1.1 4.2.\$38 .3.2 .2.2 3.1 4.1 .2.1 2.3.4 4.2.\$4E	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3	PRO TOR TOR 	GRAM			3.1	3.2	4.2			4.:	1.4.		2.3.	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.3.3 4.3.7 3.2.4.3 2.4.3 .3.1.1 4.2.\$38 .3.2 .2.2 3.1 4.1 .2.1 2.3.4 4.2.\$4E	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3	PRO TOR TOR 	GRAM	2.4		3.1	3.24.2	4.2 .\$4F	. \$39	4.2	4.:	1.4		2.3.	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 . 2.1 2.3.4 4.2.\$4E 5 4.3.6	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC TOOLK WAPPING PLOT LIST HOR OOLKIT URN PDATE ERTEXFLAG	PRO TOR TOR 	GRAM	2.4		3.1	3.24.2	4.2 .\$4F	: \$39	4.2	4.:	1.4.3.4.2.\$5		2.3.	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .1.1 4.2.\$38 .3.2 .2.2 3.1 4.1 .2.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC	PRO TOR 	GRAM	2.4		3.1	3.24.2	4.2 \$4F	: \$39	4.2	4.:	1.4.		2.3.	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .3.1.1 4.2.\$38 .3.2 .2.2 3.1 4.1 .2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LIST HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXINFO ERTEXX	PRO TOR TOR 	GRAM	2.4		3.1	3.24.2	4.2 .\$4F	. \$39	4.2.	4.:	1.34.		2.3 2.1 \$4D 4.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C .1.1 .2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 .1.1 4.2.\$38 .3.2 .2.2 3.1 4.1 .2.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LIST HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXINFO ERTEXX	PRO TOR TOR 	GRAM	2.4		3.1	3.24.2	4.2 .\$4F	. \$39	4.2.	4.:	1.34.		2.3 2.1 \$4D 4.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3.1 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC WAPPING PLOT LIST HOR OOLKIT URN PDATE ERTEX ERTEXFLAG ERTEXXINFO ERTEXX ERTEXXD	PRO TOR TOR 	GRAM	2.4		3.1	3.24.2	4.2 .\$4F	: \$399	4.2	4.:	1.34.		2.3. 2.1 34D 4.3.	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3. 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXXD ERTEXXD ERTEXXY	PRO TOR TOR 	GRAM	2.4		3.1	3.24.2	4.2 .\$4F	. \$39	4.2	4	1.4.3.4.	2.	2.3. 2.1 34D 4.3.	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3.1 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 . 2.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.4 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXY ERTEXYD ERTEXY ERTEXYD	PRO TOR TOR 	GRAM	2.4	. 2 3	3.1	3.24.2	4.2 \$4F	. \$39		4.:	1.4.3.4.		2.3 2.1 34D 4.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 . 3. . 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 . 2.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXY ERTEXYD ERTEXY ERTEXYD	PRO TOR TOR 	GRAM	2.4	. 2 3	3.1	3.24.2	4.2 \$4F	. \$39		4.:	1.4.3.4.		2.3 2.1 34D 4.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 . 3. . 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 . 2.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LITHOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXYLAG ERTEXYL	PRO TOR TOR 	GRAM	2.4		3.1	3.24.2	4.2 .\$4F	: \$39 4.2	4.2.\$50	4.:	1.34.	2.2.51	2.3 	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3.1 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 2.3.4 4.2.\$4E 5 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC UPERBASIC WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXINFO ERTEXX ERTEXYD ERTEXY ERTEXZ ERTEXZ	PRO TOR	GRAM	2.4		3.1	3.24.2	4.2 \$4F	3.\$39		4.:	1.34.	2.2.	2.3 	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 . 3.1 1.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 . 2.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXINFO ERTEXX ERTEXYD ERTEXY ERTEXZ ERTEX CLEAR	PRO TOR TOR	GRAM	2.4		3.1	3.24.2	4.2 \$4F	: \$39	4.2.	4.:	1.34.	2.2.	2.3 2.1 34D 4.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.3.\$2C 4.3.7 3 2.4.3 . 3.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXXD ERTEXYD ERTEXY ERTEXYD ERTEXZ ERTEXZ ERTEX CLEAR ERTEX TABLE	PRO TOR	GRAM	2.4		3.1	3.24.2	4.2 \$4F	: \$39	4.2.\$50	4.:	1.34.	2.	2.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3.1 4.2.\$38 . 3.2 2.2 3.1 4.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPOBJECT WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXINFO ERTEXX ERTEXYD ERTEXY ERTEXZ ERTEX CLEAR	PRO TOR TOR	GRAM	2.4		3.1	3.24.2	4.2 \$4F	: \$39	4.2.\$50	4.:	1.34.	2.	2.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 2.3.3 4.2.\$2C 4.3.7 3 2.4.3 2.4.3 . 3.1 4.2.\$38 . 3.2 2.2 3.1 4.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4	
TANDARD DRAWING TARTING A BASIC TARTING THE EDI TARTING THE EDI TART TATION OBJ3D TEP SIZE TIPPLE TOP TOP (PLOT LIST) TOP PROGRAM TORAGE DEVICE UPERBASIC UPERBASIC UPERBASIC TOOLK WAPPING PLOT LI HOR OOLKIT URN PDATE ERTEX 1.2 2.3 ERTEXFLAG ERTEXXD ERTEXYD ERTEXY ERTEXYD ERTEXZ ERTEXZ ERTEX CLEAR ERTEX TABLE	PRO TOR TOR	GRAM	2.4		3.1	3.24.2	4.2 \$4F	: \$39	4.2.\$50	4.:	1.34.	2.	2.3	2.3.3 3.2 2.1 2.2 2.4.3 4.2.\$2C . 1.1 . 2.1 2.3.3 4.3.\$2C 4.3.7 3 2.4.3 . 3.1 4.2.\$38 . 3.2 . 2.2 3.1 4.1 2.3.4 4.2.\$4E 5 4.3.6 4.3.4 4.3.	

$\overline{}$	WINDOW	FLAG	;	•		•			3.42		• 0		•									4.1
()	WIRE-FF	RAME	OBJ	JECT	S	*:												•				1.1
	WORK AF	REA		7.					1/40/	_		-			2.4	. 3	3.1	. з.	2 4	. 1	4.	2\$49
	WORKFRE	EΕ		1000 to				-	0.70	- 0			-	0-0								.\$49
	WORK TA		771 101	SE.	_				053		•		-	3.50		3.421		•	10.00		,	3.1
	WORLD		70		ē.	6.		•	•	•		•	•	24	•	•	•	• 1	1	1 2	2	.4.1
	WORLD U	INTTC	10		•	•		**		•		•			*	•	2	• т	. 1	1.2		
		פדדאי			•	•		•	•			•		•				•	114	1.1		.3.1
	х .	•	•		•		•	•	•		3					245		28		*	4	.3.7
	XOR .		•	3*	-			8.43	12	2	14	4	*1	12			2.	4.3	3.	2 4	. 2	.\$29
	Υ .	•	•					*3			1.2		40	- 13			*	94	14		4	.3.7
	YL .		(4)		•0	•			1. 14		(190)		20	- 12		1020	2.	4.3	3.	2 4	. 2	.\$28
	ZL .		•							2			102	100		192		- 15			220	3.2
	ZL .	•				•			12			Ū		-	-		2	4.3	З.	2 4	. 2	.\$28
	3D EDIT	'OR	16501	1 3 1	₹X	-	-			•		-	50 .	•	•			1.0	~ .	- 7	5	1.1
	3D_PREC		N P	TN		•		•		*		•	•		•		•	-		1 1	2	.3.3
	3D TOOL					•	•		٠	•	•	*	- 1		•	•	•	•		1.1		
							*	•	(4)	•	•	-	•	3	•	(40)	*	•	*	1	. 4	
	3D_TOOL	KIT_	RIN		•		•							12	•	•	•	•	9	•		1.1
	_ALL3D	•		•	•		•			-	(*)	-			•					1.1	2	.4.1
	_CAM3D	850													*1					1.1	2	.4.1
	DEF3D	•						27.4.01		•								1.1	2.	4.1	2	.4.3
~	_OBJ3D					_														1.1	2	.4.1
	PIC .	2.5		•	1.		•	()		•			•	•			•	•	2.4	.1	1000	4.3

1.50