

GFA VECTOR

A three-dimensional graphics program that
lets you easily add real-time animation to your
GFA-BASIC Interpreter programs.

for the Atari ST



GFA VECTOR

Three-dimensional graphics program


for the Atari ST series

(Requires the *GFA BASIC Interpreter*)

Written by GFA Systemtechnik

Distributed by MICHTRON, Inc.

**576 South Telegraph
Pontiac, MI 48053**

** (313) 334-5700
BBS: (313) 332-5452**

YOUR RIGHTS AND OURS: This copy of *GFA-VECTOR* is licensed to you. You may make copies for your own use or for archival storage. You may also sell your copy without notifying us. However, we retain copyright and other property rights in the program code and documentation. We ask that *GFA-VECTOR* be used either by a single user on one or more computers or on a single computer by one or more users. If you expect several users of *GFA-VECTOR* on several computers, contact us for quantity discounts and site-licensing agreements. Also if you intend to rent this program, or place this program on a BBS, contact us for the appropriate license and fee.

We think this user policy is fair to both you and us; please abide by it. We will not tolerate use or distribution of all or part of *GFA-VECTOR* or its documentation by any other means.

LIMITED WARRANTY: In return for your understanding of our legal rights, we guarantee *GFA-VECTOR* will reliably perform as detailed in this documentation, subject to limitations here described, for a period of thirty days. If *GFA-VECTOR* fails to perform as specified, we will either correct the flaw(s) within 15 working days of notification or let you return *GFA-VECTOR* to the retailer for a full refund of your purchase price. If your retailer does not cooperate, return *GFA-VECTOR* to us. While we can't offer you more cash than we received for the program, we can give you this choice: 1) you may have a cash refund of the wholesale price, or 2) you may have a merchandise credit for the retail price, which you may apply toward buying any of our other software. Naturally, we insist that any copy returned for refund include proof of the date and price of purchase, the original program disk, all packaging and documentation, and be in salable condition.

If the disk on which *GFA-VECTOR* is distributed becomes defective within the warranty period, return it to us for a free replacement. After the warranty period, we will replace any defective program disk for \$5.00.

We cannot be responsible for any damage to your equipment, reputation, profit-making ability or mental or physical condition caused by the use (or misuse) of our program.

We cannot guarantee that this program will work with hardware or software not generally available when this program was released, or with special or custom modifications of hardware or software, or with versions of accompanying or required hardware or software other than those specified in the documentation.

Under no circumstances will we be liable for an amount greater than your purchase price.

Please note: Some states do not allow limitations on how long an implied or express warranty lasts, or the exclusion or limitation of incidental or consequential damages, so some of the above limitations or exclusions may not apply to you.

UPGRADES AND REVISIONS: If you return your information card, we will notify you if upgrades to *GFA-VECTOR* become available. For minor upgrades and fixes, return the original disks to us with \$5.00. For major revisions, the upgrade fee is typically 15-20% of the original suggested retail price.

FEEDBACK: Customer comments are VERY important to us. We think that the use, warranty and upgrade policies outlined above are among the fairest around. Please let us know how you feel about them.

Many of the program and documentation modifications we make result from customer suggestions. Please tell us how you feel about *GFA-VECTOR* - your ideas could make the next version better for all of us.

COPYRIGHT NOTICE: The *GFA-VECTOR* program code and its documentation are Copyright (c) 1987 by GFA Systemtechnic.

Table of Contents

List of Diagrams	v
Using the Handbook	vii
Introduction	viii
General Comments	1
The Demonstration Program	2
CREATING YOUR OWN PROGRAMS	3
1.The structure	3
2.Preparations	3
3.Creating the object file	4
3.1 The coordinate system	5
3.2 Entering points	6
3.3 Entering lines	9
3.4 Defining an object	10
3.5 Displaying the object	10
3.6 Saving the object file	11
3.7 The graphics editor	12
4.The GFA-BASIC program	20
THE EDITOR	27
1. Disk	27
2. Edit points, lines and objects	28
2.1 Selection	28
2.2 List	28
2.3 Print	29
2.4 Add	29
2.5 Insert	29
2.6 Change	30
2.7 Delete	30
3. Graphics	32
3.1 Graphics editor	32
3.2 Display object	33
MACHINE ROUTINES AND ARRAYS	35
1. Machine routines	35
2. Arrays	37
Glossary	39
Specifications of GFA-VECTOR	42
Index	43

GFA SYSTEMTECHNIK

GFA-VECTOR

User documentation

All rights reserved, including those of translation, photomechanical reproduction and storage in electronic retrieval systems.

Worldwide distribution: GFA-Systemtechnik GmbH

Copyright (c)1986: Juergen Abel

Handbook Author: Hendrika Hilchner

The text of this manual is subject to alteration and no responsibility is accepted for its content.

1st Edition December 1986.

List of Diagrams

Fig. 1	5
Fig. 2	6
Fig. 3	8
Fig. 4	12
Fig. 5	15
Fig. 6	16
Fig. 7	17
Fig. 8	18
Fig. 9	19

USING THE HANDBOOK

The main chapter of this handbook is "Creating your own programs". This shows you by means of a simple example how to create a program diskette with a short three-dimensional (3-D) animation. After working through this chapter (not just reading it!) you should be familiar with the basic functions and possibilities of **GFA-VECTOR**.

The following chapters are meant to be used as a reference later:

The "Editor" describes systematically all the functions of the editor (which is used primarily to create objects).

In the next chapter all the 3-D machine routines and arrays with which GFA-BASIC can be expanded are listed and explained.

The "Glossary" explains some of the terms that are important to **GFA-VECTOR**.

In order to learn about **GFA-VECTOR** quickly:

Give yourself 2 or 3 hours on your *ATARI ST* after reading the introductory chapters to work through the chapter "Creating your own programs" directly on the computer.

INTRODUCTION

GFA-VECTOR is a software package which extends the graphics capabilities of **GFA-BASIC** considerably. With **GFA-VECTOR** even a relatively inexperienced user can create three-dimensional (3-D) graphics and animations in **GFA-BASIC**.

As the program that creates the pictures is written entirely in machine code, the screen can be updated very quickly, so that real-time animations can be made.

The basic element of any 3-D graphics or animation is an object which consists of lines and points. Objects can be defined by using the editor in one of two ways:

- Points and lines are defined by entering coordinates.
- Objects can be created interactively on the screen with the aid of the 3-D graphics editor.

GENERAL COMMENTS

Before using **GFA-VECTOR** for the first time you should create a security copy of the program diskette. Keep the original in a safe place. (If you are unfamiliar with the ATARI, and do not know how to make a security copy this can be found in the operating handbook under "Creating a security copy of your system diskette". Naturally the term "system diskette" must be replaced with the term "**GFA-VECTOR**".

The files on the program diskette are divided into 5 groups:

EXAMPLE, MODULES, OBJECTS, VECTOR\$, INITIAL

Under the heading "EXAMPLE" you will find a short example of **GFA-VECTOR**. This is started by "EXAMPLE.PRG".

The next group contains 5 modules which contain the machine code routines required to generate objects from a BASIC program. A finished 3-D program is also started by a module.

The group "OBJECTS" contains two object files: One for the example program and the other for the demo program which is started from the editor. Both files can also be used for your own programs.

From the group "VECTOR" the editor can be started by "VECTOR.PRG". The editor is used to create object files. It can also be used to customize modules.

Finally, the group "INITIAL" contains the file "INITIAL.LST" which must be "merged" into any **GFA-BASIC** program using the 3-D extension. Only if this has been done can a **GFA-BASIC** program call the 3-D routines.

THE DEMONSTRATION PROGRAM

Before you begin to learn how to write your own **GFA-VECTOR** programs, it is a good idea to see some of **GFA-VECTORs** capabilities by watching the demonstration program.

In order to start the demonstration program switch on your *ATARI ST*, put the program diskette in the drive and select "VECTOR" with a double click. With another double click start the program "VECTOR.PRG".

When it is done loading, the **GFA-VECTOR** editor appears. The demonstration program is in the window belonging to the item "DISK" in the menu bar. Move the cursor to this item and click the demonstration program to start it. After a brief description of some files some of **GFA-VECTOR's** capabilities are displayed.

When you think you have seen enough you can stop the program and return to the editor by clicking the mouse.

CREATING YOUR OWN PROGRAMS

After a short theoretical section on the basic components of **GFA-VECTOR** programs this chapter will show you, with the aid of a simple example, how to create a program diskette containing your own programs.

IMPORTANT!!! You should not just read through this chapter but work actively through it (i.e. actually do all of the following on your computer).

1.The structure of a GFA-VECTOR program.

A complete **GFA-VECTOR** program consists of at least 4 components:

1. A Module
2. **GFA-BASIC** (GFABASIC.PRG or GFABASRO.PRG)
3. A BASIC program (with INITIAL)
4. An Object file

The program is started via the module. After the 3-D routines (which are contained in the module) are loaded into memory **GFA-BASIC** is loaded automatically. This in turn loads the BASIC program which controls the object and which must contain the initializing procedure from "INITIAL.LST". After an Auto-start of a BASIC program the object file is loaded.

The object file contains the data which is needed by 3-D routines to create a wire model of the object and to manipulate this model.

2.Preparations

First format a new diskette and copy **GFA-BASIC** (GFABASIC.PRG) Version 2.0 onto what is to become the work diskette. From the **GFA-VECTOR** program diskette copy the file "MODUL1.PRG" from the "MODULES"

group and "INITIAL.LST" from the "INITIAL" group onto the work diskette (If you do not know how to format diskettes or copy files look this up in the operating handbook under the chapters "Formatting" and "Copying files").

Now load **GFA-VECTOR** (double click the "VECTOR" group followed by "VECTOR.PRГ"). When the menu banner of the editor appears put your work diskette into the current drive. It can now remain here for the rest of the session. From the "DISK" menu select the item "Create module". Change the name of the **GFA BASIC** shown here to "GFABASIC.PRГ".

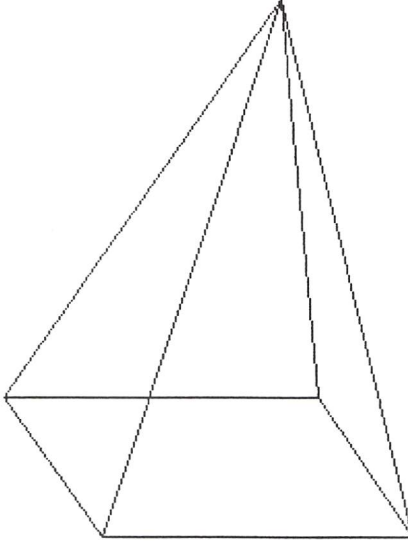
After pressing <Return> you will be asked for the name of the (yet to be created) BASIC program. The default name here is "MODUL.BAS". Change this to "TEST.BAS" and press <Return>. By clicking "MODUL1.PRГ" in the option box (followed by <OK>) the preparations are complete.

3.Creating the object file.

Now an important step: The object that is later to be drawn, moved, rotated, etc., must be defined. This is done by using the editor of **GFA-VECTOR** to create an object file.

We shall create a simple object: a pyramid with a square base. (see Fig.1)

Fig. 1



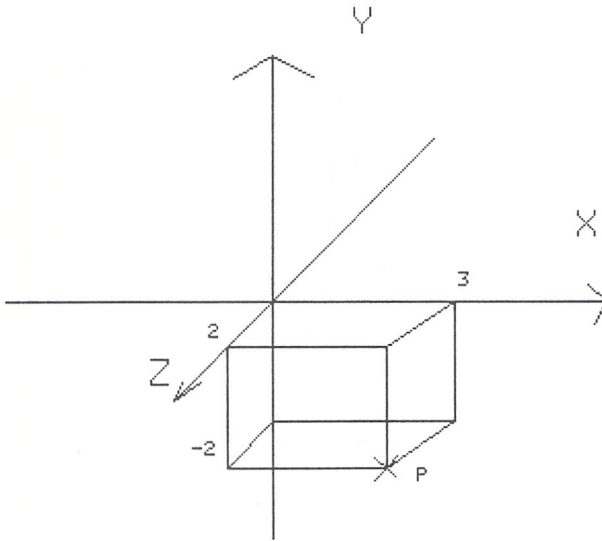
3.1 The coordinate system

In order to define a three-dimensional object such as our pyramid we require a system of coordinates. (see Fig. 2) A spatial coordinate system is defined by three axes:

The X-axis,
The Y-axis
The Z-axis.

All three axes are mutually perpendicular and intersect at one point known as the origin. According to convention the (positive) X-axis points to the right, the Y-axis points upwards and the Z-axis points out of the plane of the paper towards us.

Fig. 2



With this coordinate system any point in space can be defined by giving three values (coordinates). These coordinates are always given in the sequence x,y,z . Therefore the point P in Fig. 2 has the coordinates 3,-2,2.

3.2 Entering points.

Back to our object: the pyramid. In order to define this we first have to enter the coordinates of the 5 corner points. As later rotations of the object use the origin as their center it is a good idea to place the center of gravity of the object at the origin to make the rotation seem more natural.

We shall place the tip of the pyramid on the Y-axis and the base below the X-Z plane (see Fig. 3). This means that all four corners of the base will have negative Y-coordinates. . The 5 corners of the pyramid will have the following coordinates (see Fig. 3):

Tip: 0, 99, 0

Corners of the base:

front right: 40,-26, 40

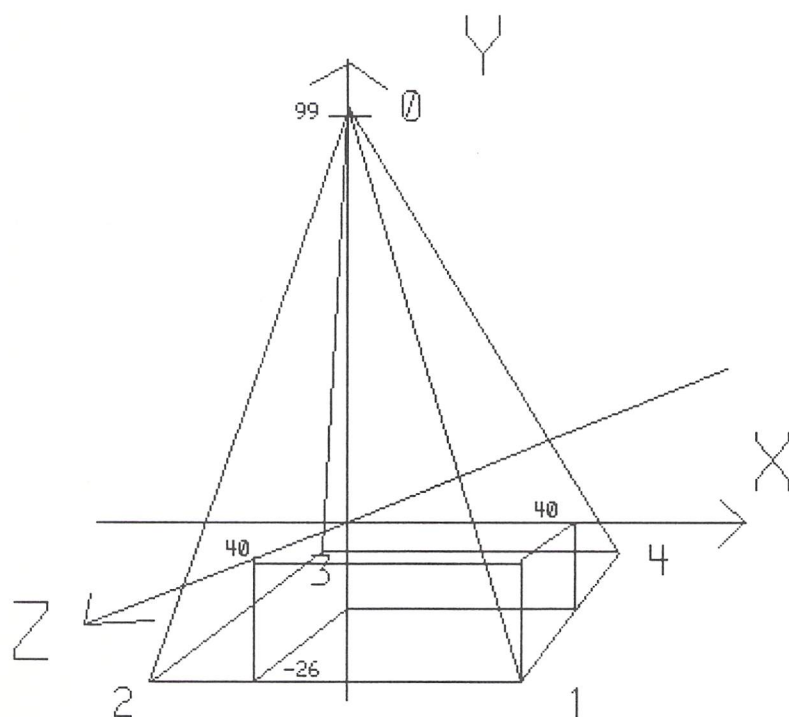
front left: -40,-26, 40

back left: -40,-26,-40

back right: 40,-26,-40

This gives the pyramid a side length of 40 units and height of 125 units. Also the center of gravity lies exactly at the origin.

Fig. 3



Now that we know the coordinates of the corner points they can be entered. You should have the menu bar of **GFA-VECTOR** On the screen.

From the "POINTS" menu select the item "Add points" (As no points have been entered up to now this is the only item that can be chosen; in all other cases the message "No points defined yet" is returned). Enter the 5 points with coordinates shown above (e.g. point number 0:0,99,0 <Return>). The entry of points is ended by simply pressing <Return> for point number 5. If after typing a line you notice you have made a mistake this can be corrected by using the function "Change points".

Notice that the first point entered is point number 0, the fifth point would have point number 4. Up to 1024 points (numbers 0 to 1023) can be entered in this way and the individual coordinates must lie in the range -1024 to +1023. This completes the point entry. Another method for entering points using the graphics editor will be explained later.

3.3 Entering lines.

To completely describe the pyramid we must next state which of the 5 points are connected by lines. In our example the tip of the pyramid (point number 0) must be connected to all four corners of the base (points 1,2,3 and 4). We must also form the base itself by connecting point 1 with point 2, 2 with 3, 3 with 4, and point 4 with point 1 to complete the square (There is no line connecting point 1 and point 3, for example, this would be a diagonal on the base).

Lines can be entered by selecting the item "Add lines" from the "Lines" menu. A maximum of 1024 lines can be defined. The lines are defined by entering the point numbers of the start and end points separated by a comma.

Therefore for our pyramid the following has to be entered:

Line number 0: 0,1
Line number 1: 0,2
Line number 2: 0,3
Line number 3: 0,4
Line number 4: 1,2
Line number 5: 2,3
Line number 6: 3,4
Line number 7: 4,1

The entry is ended by <Return>.

As for points there is another way of entering lines using the graphics editor which will be described later.

3.4 Defining an object.

The next and final step in creating an object file is to state which points and lines belong to a single object. An object file can contain many objects (up to 32). As our object file is to contain only one object all of the points and lines we have entered belong to this object. This is done by selecting the item "Add object" from the "Objects" menu.

Here the following must be entered:

Object number 0: 0,4,0,7

This states that the object number 0 consists of the points with the numbers 0 to 4 and the lines numbered 0 to 7.

3.5 Displaying the object.

Our pyramid is now completely defined and the results can be seen by choosing the item "Display object" from the "Graphics" menu.

Here the object number must be entered. Also the angle of any rotation about the X-, Y- or Z-axis and the object size (scale) by clicking the "+" or "-": Clicking the left mouse button alters the value by 1. Holding the right mouse button down causes the value to change continuously. When entering angles for rotation note the direction of rotation.

When looked at from the positive end of the axis towards the origin (the center for any rotation) all rotations seem to be anti-clockwise. For example, if a point lies on the positive z-axis a rotation of 90 degrees about the Y-axis will move the point to the positive X-axis.

With "Scale" the size of of the displayed object is determined. The normal value (1:1) is 64. The largest possible value is 511. Doubling the value causes a doubling of the size of the object.

Also a continuous rotation about any of the axis or a continuously repeating enlargement (i.e. zoom from value 1 to 128) can be selected by clicking the empty square. When selected an arrow appears in the square and the option can be switched off by clicking the square again.

To become familiar with the function "Show object" set a continuous rotation about the X-axis (click the box under "xrot") and look at the result by selecting "Graphics". (End by clicking the left mouse button). Take a little time to play with the various combinations of settings.

3.6 Saving the object file.

When you have finished looking at the object leave the function by clicking "Abort" so that we can now save the information in the form of an object file on the work diskette. To do this select "Write file" from the "Disk" menu and give your file a name e.g. "PYRAMID.DAT". After clicking the "OK" box the file is written on the work diskette.

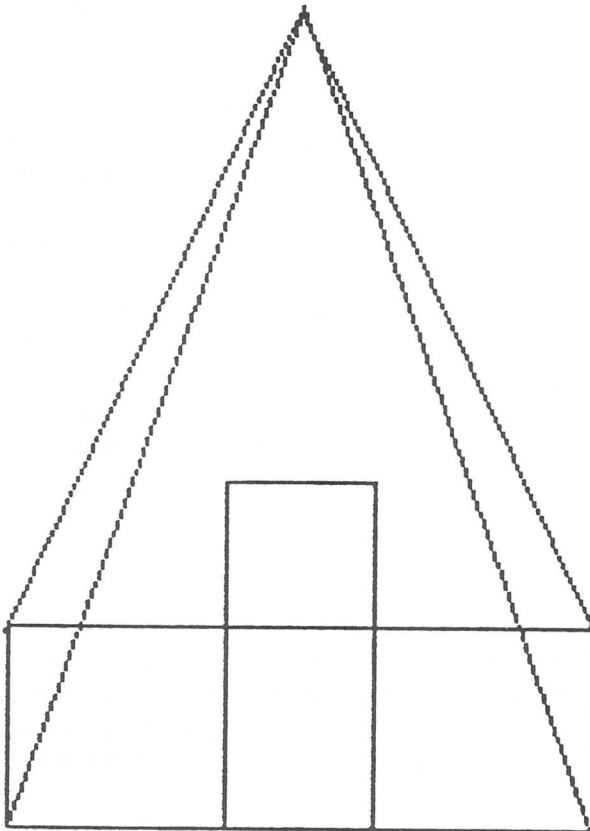
3.7 The graphics editor.

You are now, in fact, ready to write a **GFA-BASIC** program that manipulates the pyramid in any way you wish.

Before doing this, however, let us look at easier ways of entering data by using the graphics editor.

Suppose we want the pyramid to have a door on the front side.

Fig. 4



It takes a fair amount of geometry to calculate the coordinates of the two top corners of the door and on objects more complicated than our pyramid you would need to be a real mathematician. The graphics editor simplifies the entry of points and lines considerably.

First select "Graphics editor" from the "Graphics" menu. By clicking "-" change the object number to 0 (followed by "OK"). The editor consists primarily of three pictures: front view, top view, and side view (from the left). In each of these pictures the origin is located in the center. With a 1:1 scale the coordinate range in each picture is -99 to +99. The coordinates are shown in the top left area of the editor screen. The scale can be varied between 1:1 and 1:10.

Make sure that the option "Points" has been selected in the top right area of the screen and move the cursor around in the left-hand coordinate system (front view: X-Y-system).

Note: Cursor movements can also be entered with the keyboard. Pressing <Alternate> and one of the cursor keys moves the cursor 8 units in that direction. <Alternate>,<Shift> and a cursor key moves the cursor one unit at a time. <Alternate> and <Insert> together replace the left mouse button and <Alternate> with <Clr Home> replaces the right mouse button.

Important! Do not press the left mouse button at this stage, otherwise additional points will be placed on the object. If this has already happened, or if it should happen later, you should leave the editor and delete the extra points with the function "Delete points" from the "Points" menu. Then enter the editor again.

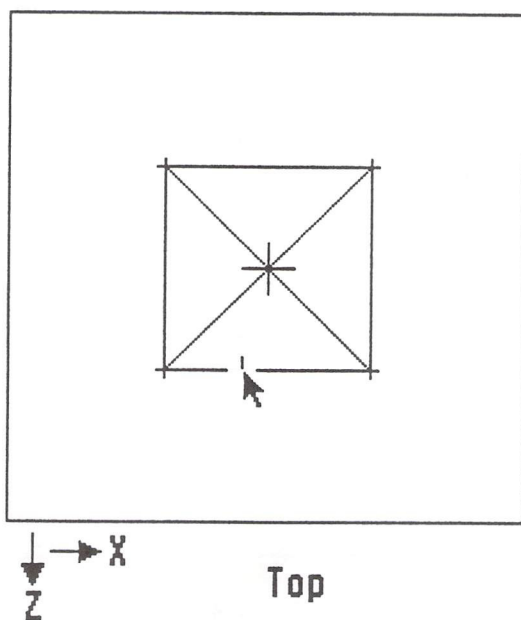
If the cursor is only moved in the left-hand system you will see that the Z-coordinate does not change. Movements in the center X-Z-system (top view) do not affect the Y-coordinate. The same principle applies to the side view.

Now try to find the origin of the complete coordinate system, i.e. try to set all three coordinates to zero. You will find that this is cannot be done. Try the following: First go to the center of one of the pictures, e.g. the top view. Here the X- and Z- values are zero. Now press the right mouse button and hold it down. This causes the two coordinates to hold their values even if the cursor is moved. Move over to the left-hand picture and release the right mouse button. You will now see that the Z-coordinate remains zero (The same would have happened to the X-coordinate if you had moved to the right-hand picture instead). If you now move to the center of the left-hand picture all three values will be zero.

Rule: Any point can be located by finding it in one of the pictures and, holding the right mouse button down, moving to another picture to find the same point again.

Now back to our task of putting a door on the pyramid. To do this we have to add 4 points and 3 lines to object number 0 (the pyramid). First we shall define the bottom left point of the door. As the door is to be 20 units wide this point is 10 units to the left of the center of the front base edge. In the top view (X-Z system) go to this point ($X = -10, Z = 40$). (see Fig.5)

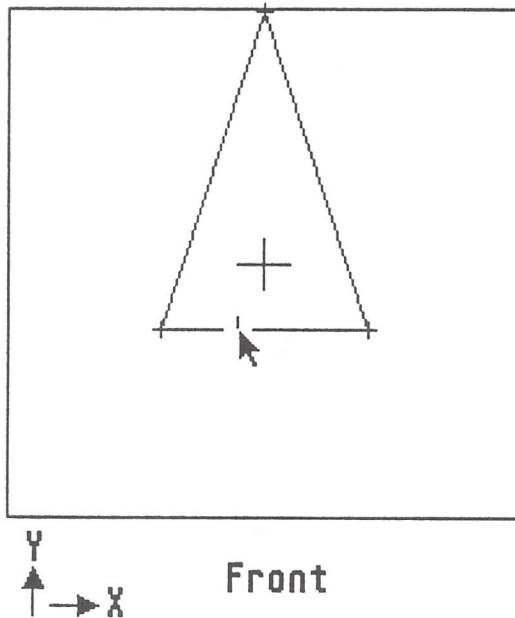
Fig. 5



When you are there press the right mouse button and, holding it down, move the cursor to the front view(You should only leave this picture after the point has been defined otherwise the procedure described above must be repeated). Here you can release the right mouse button.

The Z-coordinate is now constant and equal to the Z-coordinate of the front base edge. Move the cursor in the front view to the same place on the front edge (X= -10 ,see Fig.6) and define the point by clicking the left mouse button.

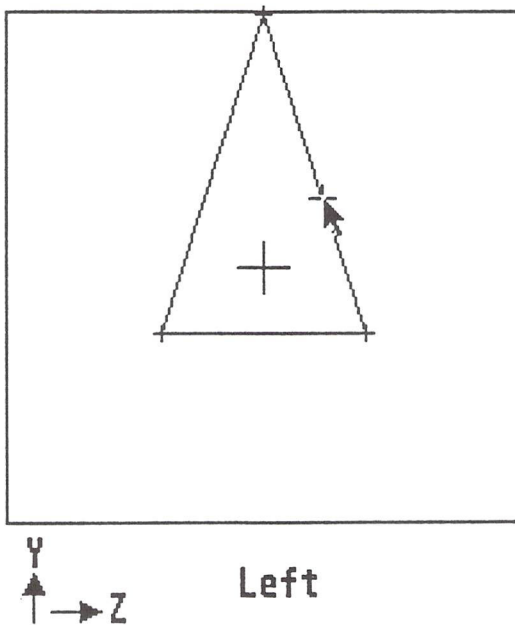
Fig. 6



The bottom right corner of the door can now be placed very quickly as it has the same Z-coordinate as the bottom left corner which we have just defined. Move the cursor in the front view to the correct position ($X=10$) and click once again.

Now we must define the top corners of the door. Move over to the side view and place the cursor as shown in Fig.7.

Fig.7

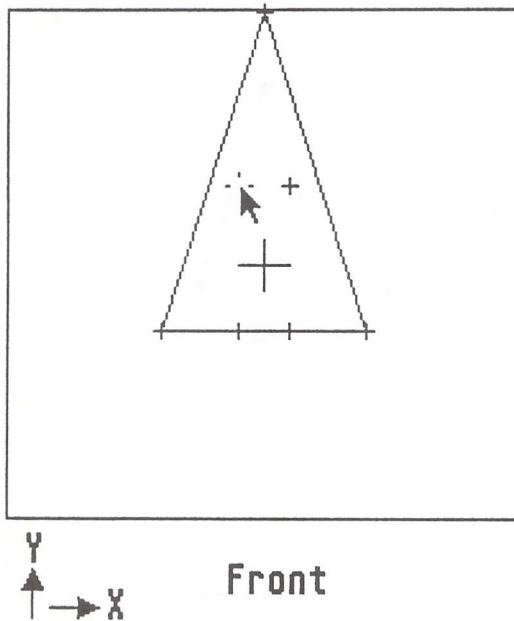


Now hold the right mouse button down and move to the front view. The button can now be released. Move the cursor to the point shown in Fig.8. For a vertical door the X-coordinate must be -10.

Important! When you are at the correct position in the X-Y system (front view) the cross in the side view is also in the correct position, i.e. it is on the side edge.

After the point has been clicked to define it the final point can also be defined in the front view in the same way (X=10).

Fig. 8



Now the 3 lines that make up the door must be defined. First click the small box labelled "Lines". Once again locate the bottom left corner of the door in the way described above. To tell the editor that this is the starting point for a line click the left mouse button while holding the <Shift> key down.

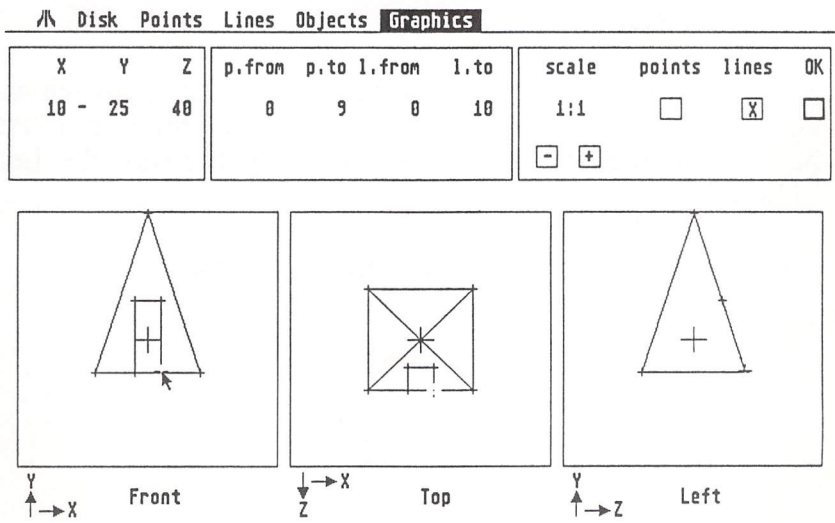
Note: The starting point of a line can only be where a point has been previously defined. When you hear a tone when defining start and end points of lines you know that a point has been located.

Now locate the top left corner of the door. (Remember that it is not enough to find the point in the front view. You must first locate the point in another picture and move with the right mouse button depressed back to front view.)

To define the end of our line (line number 8) click the left mouse button (*without <Shift>!*). If you have located the point correctly a tone is heard and the line appears in all 3 pictures.

The end point of a line is also automatically the starting point for a new line. Therefore you only have to locate and click the remaining two points to define the last two lines. The screen should then be as shown in Fig.9.

Fig. 9



The door of the pyramid is now completely defined as the object definition is handled automatically when working with the graphics editor. (Look at the point and line information in the top center of the screen.)

You can now leave the graphics editor (by clicking "OK") and inspect the new object with the "Display function" (see section 3.5 of this chapter).

After this you must save the object on the work diskette as an object file with a new name, for example, "PYRDOOR.DAT" (see section 3.6).

4.The GFA-BASIC program.

Leave the **GFA-VECTOR** program by clicking "Quit" on the "Disk" menu.

The work diskette now has the following files:

GFABASIC.	PRG
INITIAL.	LST
MODUL1.	PRG
PYRAMID.	DAT
PYRDOOR.	DAT

All that is missing now is a **GFA-BASIC** program which will move the object as desired. This BASIC program must have the name we chose when creating the module, i.e. "TEST.BAS".

Load "GFABASIC.PRG" from work diskette.

Now type in the following program:

```
Gosub Initialization
Gosub Clippingwindow(200,1,639,399)
Bload "PYRDOOR.DAT",Tbo%-6
Deffill 1,2,8
Pbox 200/X%,100/Y%,400/X%,300/Y%
Call Init%
Obj%(0)=1
Pos%(0,1)=200
Col%(0)=1
Rot%(0,0)=3
Rot%(0,1)=1
Rot%(0,2)=2
Yrot%(0)=330
Zrot%(0)=270
Call Linxor%
Do
    For I=60 To 800 Step 4
        Pos%(0,0)=1.25*I
        Xrot%(0)= I Mod 360
        If I<680 Then
            Scale%(0)=I*0.75
        Endif
        Call Vector%
    Next I
Loop
```

Make sure that the cursor is after the last program line and click the function "Merge" and load the file "INITIAL.LST" by clicking it. This causes the procedures Initialization and Clipping window to be added to the program.

The procedure Initialization must be executed in every program that is to use **GFA-VECTOR** before the first object is required. (The second procedure is used only if a clipping window (see below) is to be defined.)

The last step is to put the program onto the work diskette. Select the "Save" function and enter the name "TEST.BAS".

After clicking the "OK" box your program diskette is finally complete! (The file "INITIAL.LST is no longer needed as it has already been included in the BASIC program.)

Leave GFA-BASIC with "Quit".

You can start your program by double clicking "MODUL1.PRG". This loads **GFA-BASIC** and extends it by adding the **GFA-VECTOR** functions and then loads your BASIC program into memory which, in turn, after an automatic start loads the object file "PRYDOOR.DAT".

After you have seen your program running for a while we can finish this chapter by explaining the BASIC program. Interrupt the program (by simultaneously pressing <Control> and <Alternate>) and go back (after <Return>) to the **GFA-BASIC** editor.

Gosub Initialization

This executes the procedure called Initialization. Put simply, this procedure makes the **GFA-VECTOR** functions available to **GFA-BASIC**.

Gosub Clipping window(200,1,639,399)

If the effects produced by **GFA-VECTOR** are not to fill the entire screen a clipping window must be defined. The values in brackets are the coordinates of the top left and bottom right corners of the part of the screen where the action is to take place (as in the BASIC command "Box"). To give the effect of the pyramid coming out of the black rectangle a clipping window is defined in the right-hand area of the screen (from X=200, with high resolution).

Blod "PYRDOOR.DAT",Tbo%-6

Here the object is loaded. If the pyramid without the door is to be used then the file name "PYRAMID.DAT" is used instead. After the file name the address Tbo%-6 must always be given to ensure that the object file is loaded in the correct place.

Deffill 1,2,8 Pbox 200/X%,100/Y%,400/X%,300/Y%

A black rectangle is drawn in the center of the screen. The parameters X% and Y% are set by the Initialization routine and automatically allow for conversions of the coordinates when using low or medium coordinates. **GFA-BASIC** graphics commands should be located before the Call Init% command to keep the screen flicker-free.

Call Init%

This function first copies the visible screen page onto the not yet visible screen page. (**GFA-VECTOR** works with two screen pages which are continuously being exchanged.) This puts the same information onto both screens so that there is no flickering when the pages are changed.

Obj(0)=1

The object with the number 0 is activated. (Deactivate with Obj(0)=0.) If an object file contains more than one object they can be individually activated or deactivated.

Pos%(0,1)=200

The Y-coordinate on the screen of the object number 0 is set at 200. (The X-coordinate is set by Obj%(0,0).) The coordinates must always refer to the 640*400 resolution even if you are using one of the color modes.

Col%(0)=1

The color is assigned the value 1.

Rot%(0,0)=3

Rot%(0,1)=1

Rot%(0,2)=2

The sequence of the rotations about the axis are specified (Note that changing this sequence usually gives a different effect). The first rotation is about the Z-axis (axis 3) followed by the X-axis and then finally the Y-axis.

Yrot%(0)=330

Zrot%(0)=270

The object is rotated by an angle of 270 degrees about the Z-axis and an angle of 330 degrees about the Y-axis. (see above for the sequence)

Call Linnxor%

The Xor mode is activated. This allows the object to move over the background without destroying it. To study this function it is worth removing this line from the program (which leaves the Or mode set from the Init%) and running the program again.

Within an endless loop (Do...Loop) the value of I is increased from 60 to 800 in steps of 4 (For I=60 To 800 Step 4...Next I).

Pos%(0,0)=1.25*I

The X-position is calculated. As I is increasing the pyramid wanders to the right.

Xrot%(0)=I Mod 360

The object is rotated in degrees about the X-axis. As values greater than 359 (and less than 0) are not allowed here the "Mod" function must be used. ("Mod" calculates the remainder after a division; e.g. 361 Mod 360=1.)

If I<680 Then**Scale%(0)=I*0.75****Endif**

The size of the object number 0 is increased up to the value 510. (Normal size:64, Max.size:511)

Call Vector%

The object (or objects) are now drawn according to the values set in Pos%, Rot%, Obj%, etc. Usually the old object is first deleted but this can be suppressed if the function "Call Ofm%" is entered before "Vector%". Try this out too.

The work diskette is complete. As already mentioned the file "INITIAL.LST" is no longer required and can be deleted from the diskette.

You can also rename the four files "GFABASIC.PRG", "TEST.BAS", "MODUL1.PRG" and "PYRDOOR.DAT" (or "PYRAMID.DAT") to suit any conventions of your own.

Important note: Remember that a work diskette created in this way is for your own use only. If you wish to give your product to others you must replace "GFABASIC.PRG" with "GFABASRO.PRG". The module must also be changed accordingly.

After working through this chapter you should look at the example program on the program diskette under the heading "EXAMPLE". The file is named "EXAMPLE.PRG". This loads the run-only **GFA-BASIC** interpreter "GFABASRO.PRG" and the **GFA-BASIC** program "EXAMPLE.BAS". From the program the object file "EXAMPLE.DAT" (which contains a cube and a wire model of a "space shuttle") is loaded.

THE EDITOR

This chapter is a reference for the editor functions. The editor is used to create modules and object files and store, load or delete these to and from diskette. The demonstration program is also started from the editor.

There now follows a description of the single items on the menu banner.

1. Disk

Here you can load, write (store) or delete object files. If this item is selected a file select box appears. Here the name of the file must be entered (or if already present then just clicked). Clicking the name of a group causes the contents of this group to appear in the file select box. For example, if you want to load the object file "DEMO.DAT" or "EXAMPLE.DAT". These are in the group called "OBJECTS" and can be used in your own programs.

The next item in this menu is to create modules. Here you must first enter the names of the **GFA-BASIC** being used and the program that you are going to write to control the objects. Then a file select box appears in which you must click the module which is to contain the information you have entered.

The editor can be left by clicking "Quit".

2. Edit points, lines and objects.

An object in an object file is defined by three types of information:

Points.

Lines.

Statements about which lines and points belong to the object.

2.1 Selection.

For some items from this menu a range must be entered (from - to) for which the points, lines or objects are to be listed, printed or deleted. Others, such as insert or change, require just one number.

The range is selected by moving the cursor in the box below "+" and "-" and clicking the required values. A click of the left mouse button cause the value to change by one, the right mouse button changes the value continuously.

2.2 List

Points are listed in the normal way as three coordinates X, Y and Z for each point number.

Lines are listed as the point numbers of the start and end points.

Listing objects shows the numbers of the first and last points and the first and last lines that make up each object.

The list can be interrupted by clicking the left mouse button (click again to continue) and ended by clicking the right mouse button.

2.3 Print

Selecting this item causes the list described in 2.2 to be printed.

The printing can be interrupted by clicking the left mouse button (click again to continue) and ended by clicking the right mouse button.

2.4 Add

New points or lines are always added on to the end of a list. (Therefore a range is not required.)

New points, lines or objects can only be entered with this menu item.

A maximum of 1024 points and lines and 32 objects can be defined.

To add new points the coordinates must be entered in X-Y-Z sequence and separated by commas. All coordinates must lie in the range -1024 to +1023.

To add new lines the numbers of the start and end points must be entered.

To define a new object four numbers are required, separated by commas:

The numbers of the first and last points and the first and last lines.

The "add" function is ended by pressing <Return> without having entered a value.

2.5 Insert

This allows new points, lines, or objects to be inserted into existing ones.

The data is entered in the same way as for "add". After each insertion the numbers of the following items in the list are changed accordingly.

Inserting points could affect the definition of lines and objects.

Important! GFA-VECTOR corrects this automatically.

In the same way inserting lines can disrupt object definitions. This is also corrected automatically by **GFA-VECTOR**.

If you want to end the insert function simply press <Return> without entering any values.

2.6 Change

With this function the data of individual points, lines, or objects can be changed. The old data of the point, line or object is shown next to its number. New values can be entered as described in 2.4 above.

To leave this function press <Return> without entering any data.

2.7 Delete

All points, lines or objects within a given range are deleted by this function and any following items are renumbered.

Deleting points can affect a line or object definition. Likewise, an object definition can be affected by deleting lines. **GFA-VECTOR** recognizes when this happens and corrects accordingly.

Exception: If lines or objects refer to deleted points (or objects refer to deleted lines) it is not possible to correct this automatically and the appearance of the object can change unpredictably.

3. GRAPHICS.

3.1 Graphics editor.

The graphics editor allows you to create objects directly on the screen (interactively) instead of designing them on paper and then entering the data into the computer.

After entering the graphics editor you are asked for the number of the object to be edited. You can create new objects as well as changing existing ones. (The object number is set as usual by clicking "+" or "-" with one of the mouse buttons.)

The editor itself consists mainly of the three coordinate systems:

- The front view
- The top view
- The side view.

The origin of each of these systems is in the center and is marked by a cross. With the scale at 1:1 the coordinates are in the range -99 to +99.

The cursor can be moved over all three systems. The coordinates of the cursor position are shown in a window near the top left of the screen. Further information at the top of the screen includes which points and lines belong to the current object.

In the top right of the screen the scale can be chosen and here you can also specify whether you are creating lines or points. Clicking "OK" here ends the graphics editor.

To locate a particular point in space the small crosses in all three pictures must be in the correct position. This is achieved as follows: Move the cursor to the correct position in any one of the three systems. Press the right mouse button and hold it down as you move the cursor to

one of the other two systems. Release the mouse button and locate the point in this system as well. Points are set by clicking the left mouse button.

To define a line the start point is located as described above. (Do not forget to choose "Lines" in the top right window!) When you have located the starting point click the left mouse button while holding down the <Shift> key. As confirmation you will hear a tone (If you do not hear a tone this means that the point has not been located correctly).

The end point of a line is set by clicking the left mouse button. The end point of a line is automatically the start point for a new line. If this is not required then the new start point must be set by clicking the right mouse button and holding down the <Shift> key.

3.2 Display object

This function is used to display the created objects in various sizes or views. An objects can also be displayed while rotating about all three axis.

If the object file consists of several objects the object number must first be entered (Clicking "+" and "-": Left mouse button: single count; Right mouse button: counts through).

Now the object can be rotated by a fixed angle about each of the axes. The rotation is always first about the X-axis then the Y-axis and finally the Z-axis. A continuous rotation can be started by clicking the empty box for this function. An arrow appears in this box and to stop the rotation the box is clicked again.

The "Scale" option allows you to control the size of the object. The normal size (1:1) is preset and has a value of 64. If the empty box is clicked (arrow appears) the object automatically enlarges from 1/64 to double its original size. (This is an increase of the scale values from 1 to 128).

Finally the drawing mode can be selected. This can be set to OR or XOR mode.

When all the details have been entered click "Graphics". A "wire model" of the object appears in the form that has been specified.

The display of the object is ended by clicking the left mouse button.

The function is ended by clicking "Abort".

MACHINE ROUTINES AND ARRAYS.

GFA-VECTOR expands the graphics capabilities of the **GFA-BASIC** interpreter by making possible the animation of three-dimensional wire-models. This extension is written in machine code. Therefore, these routines are invoked by the **GFA-BASIC** program by the "Call" command.

The most important of these routines ("Vector%") draws the three-dimensional object (wire-model) on the screen. The parameters contained in arrays (fields) are used to do this (size, rotation, etc). .

There now follows a description of these routines and arrays:

1. Machine routines.

Init%:

When this routine is called the second screen page is deleted and the first page is copied onto the second. The routines, Tfrm% and Linor% are then called (see below). If a drawing has been created by "normal" **GFA-BASIC** commands (this is always created on the first screen page) the Init% function should be present before calling Vector%. This causes the drawing to be copied onto the second screen page which prevents the picture from flickering.

Vector%:

With the parameters contained in the arrays a new picture is drawn and (if Ofrm% has not been called previously) the old picture is deleted.

Linor%:

If this function is called then any background detail is deleted by the object. This function is set automatically if "Init%" is called.

Linxor%:

The command "Call Linxor%" causes the objects to be drawn (or deleted and redrawn) so that any background that may be present is preserved (or recreated).

Ofrm%:

When "Vector%" is called the old picture is not deleted.

Tfrm%:

When "Vector%" is called the old picture is deleted.

Page1%:

Puts the first screen page onto the monitor. One of the two screen pages must be put onto the monitor at the end of the program (or after an abort) so that the messages from GFA-BASIC can be seen after the end of the program or an abort.

Page2%:

Puts the second screen page onto the monitor.

Flip%:

Switches the screen page from page 1 to page 2 or vice versa.

"Flip%" has no effect when a page has been selected by "Page1%" or "Page2%". ("Init%" resets the page selection).

2. Arrays.

Obj% (Nr):

This allows the objects in the object file to be made active or inactive. "Nr" is the object number. This array can contain the value 0 or 1. 0 for inactive and 1 for active.

Col% (Nr):

The object with the number "Nr" is assigned a color. This is given by a value between 0 and 15 (depending on the resolution).

Scale%(Nr):

Determines the size of the object with the number "Nr". A value of 64 in this array gives the object in actual size (1:1). Doubling this value doubles the size of the object. The value must be in the range 1 to 511.

Pos%(Nr,0):

Specifies the X-position of the origin of object "Nr". Whatever the resolution the X-position must be given in the high-resolution range i.e. between 0 and 639.

Pos%(Nr,1):

Specifies the Y-position of the origin of object "Nr". Whatever the resolution the Y-position must be given in the high-resolution range i.e. between 0 and 399.

Note: the Y-position 0 is on the top edge of the picture.

Rot%(Nr,r):

This array allows the sequence of the rotations of the object "Nr" about the three axis to be specified. (Remember the results are different if the sequence

of rotations is changed). The array can contain the values 0,1,2, or 3. 1 means the X-axis, 2 means the Y-axis and 3 means the Z-axis. "r" is the sequence of the rotation where 0 is the first and 1 and 2 are the following rotations. Example: Rot%(0,1)=3 means that the second rotation for object number 0 is about the Z-axis.

Xrot%(Nr):

This array can contain an integer between 0 and 359. The object with the number "Nr" is rotated by this value (in degrees) about the X-axis. The direction of the rotation is always anti-clockwise when looked at along the positive axis towards the object origin.

Yrot%(Nr):

Rotation of the object "Nr" about the Y-axis. (See Xrot% above).

Zrot%(Nr):

Rotation of the object "Nr" about the Z-axis. (See Xrot% above).

Endpunkt%(Nr,a):

These arrays contain the X-position (a=0) and the Y-position (a=1) of the last point of object "Nr". These positions are always in the high-resolution range. This makes it very easy to move the object around the screen: An object consisting of just one point is defined (Actually, it is one point and line connecting this point to itself). This object is then moved around the screen in the desired fashion. After each call of "Vector%" the coordinates of the point can be obtained from "Endpunkt%" array and these values can be used as the position for the object to be moved.

GLOSSARY

Here is an explanation of some of the terms used:

Array

GFA-VECTOR uses one and two-dimensional arrays (or fields) to provide the machine code routine "Vector%" with the data required to draw the object or objects.

Screen page

To reduce the flickering of a graphical representation of an object **GFA-VECTOR** works with two screen pages. While one page is seen on the monitor with the previously created graphics a new picture is created on the second screen page.

Clipping window

Normally, the whole of the monitor screen is used to show the **GFA-VECTOR** graphics. With a clipping window the portion of the screen that is to be used for these graphics can be specified. (Always in the form of a rectangle however). A clipping window does NOT reduce the monitor screen to the given window size! Only the portion of the graphics that lies within this window is drawn on the screen.

A clipping window is defined in a **GFA-BASIC** program by the command:

Gosub Clipping window (X1,Y1,X2,Y2)

The procedure Clipping window is included in "INITIAL.LST" and X1,Y1 and X2,Y2 are the coordinates of the top left and bottom right corners of the clipping window respectively.

Direction of rotation

GFA-VECTOR allows an object to be rotated about any of the 3 axes. The direction of any rotation is determined by looking along the positive end of the appropriate axis towards the origin. The rotation is counter-clockwise when viewed in this way.

Editor

The part of the **GFA-VECTOR** program which enables you to create an object file more comfortably. It is started by "VECTOR.PRГ".

Coordinates

Values which describe the position of a point in a coordinate system. Object coordinates describe the (corner) points of the object. It is usual to place the center of gravity of the object at the origin of the system of coordinates. Screen coordinates serve to place the two-dimensional representation of the object at the required place on the screen.

The screen coordinates specify the position on the screen for the center of the object.

Machine routines

These are small programs written in machine code. The most important of these is "Vector%" which handles the drawing of objects.

Module

A module is the part of the **GFA-VECTOR** program package which contains the routines required for drawing objects. Completed 3-D programs are also started from a module. The module causes **GFA-BASIC** and the BASIC program to be loaded. In order that the module knows which program it is to load it must first be "created" by using the editor (menu item "Disk"). The **GFA-VECTOR** program diskette contains 5 identical (apart from their names) modules.

Object

In **GFA-VECTOR** an object is (usually) the 3-dimensional figure which is to be drawn and moved as a 2-dimensional representation with the aid of the information contained in arrays. An object is a "wire-model" of the figure to be drawn and is defined by points and connecting lines.

Object file

This is created using the editor and can contain up to 1024 points and up to 1024 lines. As it is specified in an object file which points and lines belong to an object, a single object file can contain several (up to 32) objects.

SPECIFICATIONS OF GFA-VECTOR

Editor:

Max. no. of points per object file: 1024

Max. no. of lines per object file: 1024

Max. no. of objects per object file: 32

3-D routines:

Scale: Objects can be scaled in 511 steps from 1/64th to 8x normal size.

Overlay modes: OR-mode or XOR-mode possible

Sequence of axis rotations: Can be defined

Min. angle for axis rotation: 1 degree

No. of screen pages: 1 or 2 selectable

Implementation: The drawing routines are written entirely in machine code.

Index

- 3-D graphics viii
- 3-D machine routines vii
- Abort 11
- Add object 10
- Add points 9
- Angle of rotation 11
- Animation viii, 35
- Array 39
- Bload 23
- Call 35
- Call Init% 23
- Call Linnxor% 24
- Call Vector% 25
- Center of gravity 40
- Changing 30
- Clipping window 21, 39
- Col% (Nr) 24, 37
- Continuous rotation 11
- Coordinate conversions 23
- Coordinates 13, 40
- Cursor movements 13
 - keyboard
 - <Alternate> and
 - <Insert> 13
 - <Alternate> with <Clr Home> 13
 - <Alternate>, <Shift> 13
 - <Alternate> 13
- Deffill 23
- Deleting 30
- Direction of rotation 40
- Disk 11
- Display 33
- Display function 20
- Display object 10
- Editor v, 40
- Editor screen 13
- Endpoint% 38
- Example program 1
- EXAMPLE.PRГ 1
- File 10
- First screen page 35, 36
- Flickering 23, 39
- Flip% 36
- Front view 32
- GFA-BASIC 1
- GFA-VECTOR 1, 2, 4
- GFA-VECTOR program 20
 - automatic start 22
 - Leave 20
- GFABASIC.PRГ 3, 20
- Gosub Clipping window 22, 40
- Gosub Initialization 22
- Graphics 11
- Graphics editor 12, 13
 - calculate coordinates 13
 - entering data 12
 - leaving 20
 - three pictures 13
 - front view 13
 - side view 13
 - top view 13
- Init% 35
- INITIAL 1
- INITIAL.LST 1, 4, 21, 40
- Initialization 21, 23
- Insert 29
- Lines viii, 9, 28
 - Add lines 9
 - adding 29
 - end point 19, 33
 - listed as 28
 - maximum of 1024 9
 - start point 33
 - starting point 19
- Linor% 36
 - background deleted 36
- Linxor% 36
 - background preserved 36
- List 28
- Machine code 35
- Machine code routines 1, 41

- Merge 1
- MODUL1.PRG 3, 22
- Module 1, 41
- No points defined yet 9
- Obj% (Nr) 23, 37
- Objects viii, 10, 41
 - 32 per file 10
 - activated 23
 - active 37
 - angle of rotation 11
 - created *interactively* viii
 - deactivated 23
 - Defining 10, 28, 29
 - Displaying 10, 33
 - draw 39
 - inactive 37
 - listed 28
 - loaded 23
 - moved 13
 - number 11, 23
 - object file 10
 - rotated 33
 - saving 20
 - size 11
 - system of coordinates 5
 - three axes 5
- Object file 10, 41
- Or mode 24, 34
- Origin 13, 14
- Page1% 36
- Page2% 36
- Parameters 23, 35
- Points viii, 9, 28
 - Change points 9
 - adding 29
 - entry 9
 - listed as 28
 - located 14
 - Maximum of 1024 9
- Pos% 23, 24, 37
- Printing 29
- Procedures Initialization 21
- Programs 3
 - A Module 3
 - BASIC program (with INITIAL) 3
 - creation 3
 - GFA-BASIC 3
 - Object file 3
- Range 28
- Rot% 24, 37
- Rotation 11, 24, 38
 - 90 degrees 11
 - continuously 11
- Scale 11
- Scale%(Nr) 37
- Screen page 39
- Screen updated viii
- Second screen page 35, 36
- Security copy 1
- Selection 28
- Show object 11
- Side (left) view 32
- Structure 3
- Tfrm% 36
- Three-dimensional wire-models 35
- Top view 32
- VECTOR 1
- Vector% 35, 38, 39, 41
- VECTOR.PRG 1
- VECTOR.PRG 40
- Write file 11
- X-axis 5
- Xor mode 24, 34
- Xrot% 25, 38
- Y-axis 5
- Yrot% 24, 38
- Z-axis 5
- Zoom 11
- Zrot% 24, 38

ALT

by J. Weaver Jr.

Alter the ALT keys!

Typing all those redundant phrases common in your daily forms and letters steals your energy and wastes your time. With MichTron's new *ALT* keyboard utility, you can convert whole strings of tedious typing into single, elegant keystrokes.

Redefine all 36 ALT keys. ALT-A to -Z, and ALT-0 to -9 can be replaced by often used text phrases, program commands, or any other string of up to 60 characters each. Pressing an ALT code will instantly summon your defined string, just as if you had typed it in by hand.

ALT goes on the GEM Desktop.

This handy accessory is available from almost any application. You can save and load your new key set-ups to and from disk. *ALT* can even be set to automatically load your key set-up when the Desktop is first initialized.

Whether to speed up office work or make home applications run smoother and easier...

***ALT* is your key to success.**



Available for the Atari ST

\$29.95

BACKUP!

by Dave Clemans

When a hard-drive crashes, there isn't any noise. No smoke, no fire, no broken glass or twisted metal. Physically, everything remains pretty much intact. But datawise...20 megabytes of amnesia? Well, you just start over. *From scratch.* And hope it never happens again...

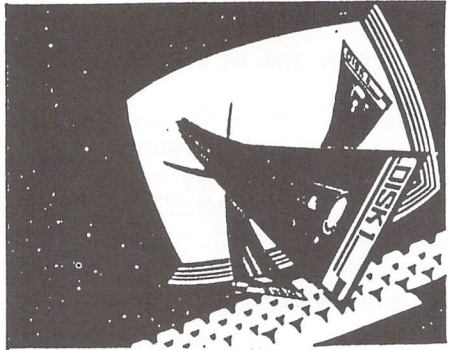
When a floppy crashes, most of us think it's the end of the world. But a hard-drive? You lose *everything*.

Backups, what about backups? How often do you make backups of a quarter billion bytes of computer memory? And even so, are they accurate and up-to-date? Would you stake a month's lost labor on it? Or even a week's? Probably not.

And now you don't need to. Now there's *Backup!* For those of us who like things quick and simple. *Backup!* duplicates the contents of your hard-drive to floppy disks, minimizing the risk of loss. It also lets you restore the floppy backups back to the hard-drive after an accident, reducing any actual loss to a minimum.

But much more than that, *Backup!*'s specialized routines are very fast, and the program orchestrates the entire backup procedure for you, minimizing time and effort as well. There's no longer any reason not to make backups. Especially when it's this easy.

Backup! uses pull-down menus to make things even more convenient. You can select numerous options and run different types of backups and restores with a click of the mouse.



"Intelligent" routines give you the option of making full backups of the entire hard-drive, backups of only newly created or modified files, or backups simply by date. You might even decide to make an "image" backup, transferring logical sectors to counterpart floppies, saving even more time.

Whatever method you need, the support is waiting for you in *Backup!* There's even a detailed Help menu available in the program.

- Backup your hard-drive to floppy disks.
- Restores your files from the backup after an accident.
- Specialized, fast routines.
- Automatically organizes the backup procedure.
- Pull-down menus for easy use.
- Make full or partial backups by use or date.
- Make file or image backups.
- Select whether to backup hidden files and system files.
- Option to segment large files to more than one disk.
- Includes Help function.

***Backup!* Before it's too late...**

Available for the Atari ST

\$39.95

BBS 2.0

by Timothy Purves

- **Easy to use and maintain**
 - Full featured
 - Affordable

A Bulletin Board System with all this? Only from MichTron!

This full-blown electronic mail and message system has everything:

Easy to set up - Have your own *BBS* running within an hour!

Versatile message base - Maintains up to 1024 messages, limited only by disk space and message size.

Longer messages - Up to 99 lines.

Up to 16 SIGs - Each Special Interest Group has messages, upload/download areas, and user access for posting public or private SIGs.

Message control language - Add customized menus, complete with user inputs, to message areas!

Multi-user capacity - The operator and one caller can both use the *BBS* at the same time, independantly! In the tracking mode, you see everything your callers see, and can even assist them from your terminal!

Four transfer modes - *BBS* supports XMODEM, XMODEM-CRC, DFT, and ASCII protocols.

Easy maintenance - *BBS* is practically self-maintaining. Updating user logs and download files can be done without shutting down! Just log-on through the host keyboard.

Remote sysop ability - Perform your duties with a remote call to your *BBS*! And by giving different levels of control, you can let other sysops help maintain your system.

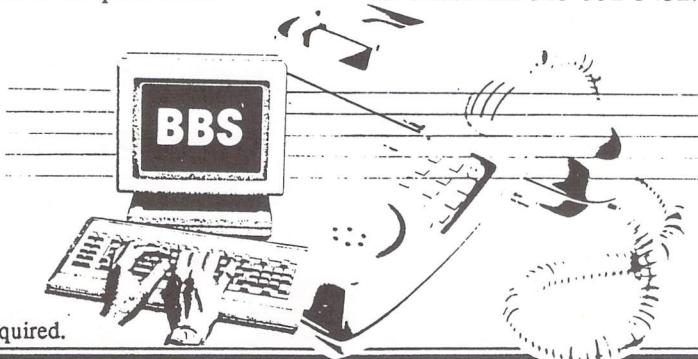
Adjusts to callers - Adapts to most users' baud and parity settings.

System security - *BBS* is not affected by accidental hang-ups. Users are isolated from disk I/O so that tampering is virtually impossible.

Reviewed by William Van Nest Sr, Compuserve OS9 SysOp:

"I've been running *BBS*, whenever I'm not using the ST, since the first beta version. I originally set it up without *any* documentation, and had it running within 1 hour! Now, several revisions and many enhancements later, it's even easier and more flexible. I can safely say that this *BBS* package is excellent, whether for a part time hobby or a full time, dedicated *BBS*. I highly recommend it."

AND MUCH MORE - Call our *BBS* for a test run: 313-332-5452.



Modem Required.

Available for the Atari ST and IBM PC

\$79.95

BUSINESS TOOLS

compiled by Fern and Paul Benson

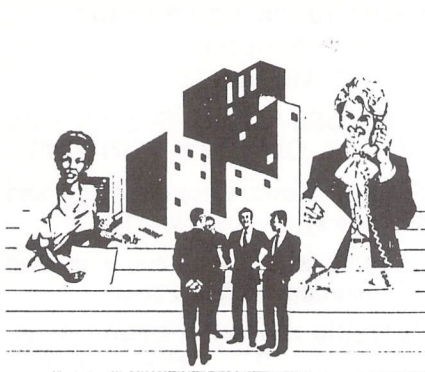
Every business has paperwork and legal concerns, but no one enjoys the labor and uncertainty in creating your own forms, or the cost of hiring a lawyer. With the 215 attorney-prepared business documents included in *Business Tools*, you can reduce the time, worry, and cost of paperwork, so that you can concentrate on making money instead.

77 Business Forms - When you need a form, don't waste time designing one from scratch: use one of the 77 forms stored on the two diskettes provided with *Business Tools*. You'll probably find just what you need already there. Simply fill in the blanks and print!

33 Contracts - If you need a simple contract, don't spend hard-earned cash on a lawyer when *Business Tools* can supply one for nearly any common situation small businesses face. Fill in the names and dates and print the contract.

105 Letters - Quit struggling to compose just the right letter to a customer or supplier. Instead, pick one of over a hundred letters already stored on your *Business Tools* disks, type in the name and date, start your printer and get back to business.

Customize - *Business Tools'* letters, forms and contracts for small businesses are saved as word processing templates you can use with your Atari ST and your favorite word processor. Use the forms just as they are, or customize them if you need to. You can even save the custom version to use again if you want.



Here's a sample - Fully indexed by subject, the Business Tools library contains documents for every common small business need:

- Standard form rental agreement
- Commission statement
- Apology for lost shipment
- Employee termination
- Daily sales call report
- Interview evaluation
- Auto expense report
- Request for accounting statements
- General bill of sale
- Bad check notice
- Application for credit
- Profit forecast

These and 203 other forms, letters and contracts can solve your paperwork woes.

An extra bonus - There are even eight business data appendices, with information like weights and measures, abbreviations and telephone area codes.

Available for the Atari ST, Sanyo 550, and IBM PC

\$49.95

CALENDAR

by J. Weaver Jr.

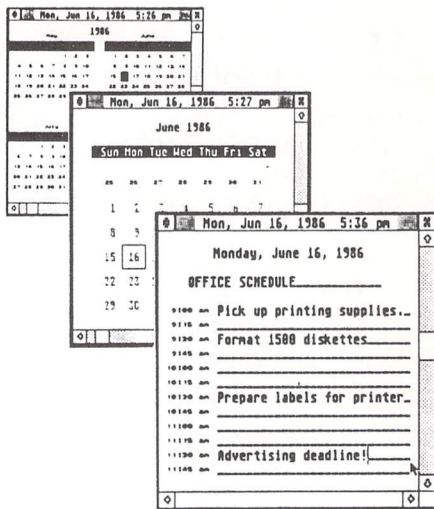
The complete appointment and reminder system

Available anytime - As a Desktop accessory, *Calendar* is ready at a moment's notice. From almost anywhere in the GEM system, you have instant access to *Calendar's* helpful features, *without* disturbing your current application.

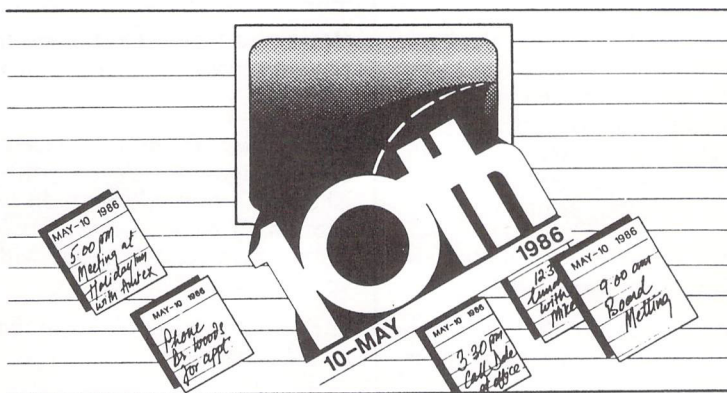
Calendars - Display or print calendars of any selected year and month.

Messages - The time-based message entry is perfect for schedules and time-keeping. Messages for any date and time (from 1980 to 2099) can be saved to disk giving you a complete appointment book!

Special Alarms - You can attach "alarms" to any or all reminder messages to notify you of special events as they occur! Alarms or hourly chimes appear automatically on the Desktop or during GEM programs--even when *Calendar* itself is not on the screen!



Easy to use - *Calendar's* superb user interface combined with the power of GEM lets you glide from screen to screen with the ease of "point and click". There are no commands to memorize, and no special procedures to follow; just one simple installation puts *Calendar* at your command.



* Sanyo users--see product notes, last page.

Available for the Atari ST, Sanyo 550, and IBM PC

\$29.95

CORNERMAN

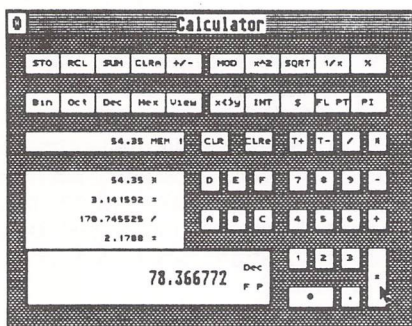
by J. Weaver Jr.

A 9-function utility to keep your desk top organized!

Sweep that clutter off your desk and into your computer! Ever the perfect servant, *Cornerman* waits patiently in memory while you use other software. But when you need to make a note or find your calculator, *Cornerman* is ready to serve.

Call him from the Desktop menu and he'll provide whatever you need:

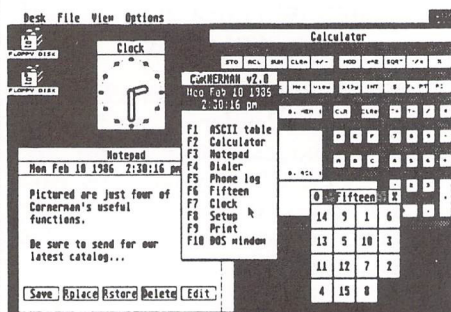
ASCII Table - Shows the ST's symbols with their decimal & hex values. Great for programmers!



Calculator - 16 digit calculator works in decimal, binary, octal and hex; floating point; 3 memory registers; standard math operations plus square root, percents, INT, \$ and Pi functions; mouse or keyboard controlled; even a "tape" printer output!

Clock - Real-time in digital or analog, large or small display.

Security - A special option prevents others from disturbing your computer while you take a break.



Note pad - Write notes for handy future reference and reminders. Search, save, and delete functions. Room for hundreds of 7-line memos.

Phone book/dialer - Contains a complete address book, and can even dial for you through your modem. Handles long-distance codes too!

Phone log - Records the date, time and duration of your calls.

DOS window - instant access to DOS commands (with separate DOS Shell utility).

15-squares puzzle - Simply for entertainment.

Set-up utility - Customize the format and location of your *Cornerman* windows.

Cornerman doesn't interfere with other programs. When you've finished with him, *Cornerman* will put everything back as it was before he came, so you can continue with what you were doing before.

Hire the perfect servant today - hire *Cornerman*.

* Sanyo users--see product notes, last page.

Available for the Atari ST, Sanyo 550, and IBM PC

\$49.95

D. F. T.

by Timothy Purves

Atari/IBM compatibility - Frustrated because you can't transfer programs or information between your IBM and your Atari ST? Since even the disk sizes are different, transferring has been a physical problem as well as one of programming. With MichTron's *DFT* (direct file transfer) utility, these problems are solved!

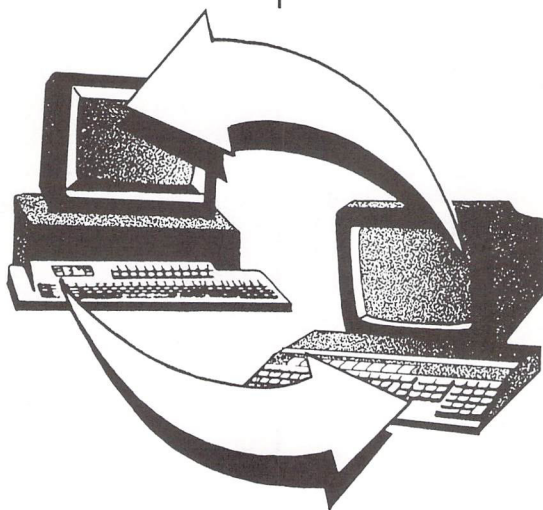
Easy to set up - Just run a null modem cable between the modem ports of your Atari ST and the IBM PC, load the *DFT* disks into each computer, and ship files between them almost effortlessly.

Priceless versatility - *DFT* lets you send files in either direction. Text and data files can be used immediately, of course. Program source code arrives ready to translate or compile on the second computer, saving days of retyping and proof-reading.

Simple, one-computer control - *DFT* runs your IBM as a host, and the Atari as a remote console, giving you access to all files on both computers from your Atari keyboard. View complete directories of either computer (including estimated time to transfer each file), change directories or drives on either machine, and transfer files--right from your Atari!

Multiple file transfer - *DFT* offers full wild card support to make it as easy to transfer multiple files as it is to send just one.

Convenient cable option - *DFT* includes one 5-1/4" IBM diskette and one 3-1/4" Atari diskette. The manual shows how to make the required null modem cable from parts readily available at any electronics/hobby store, or if you prefer, MichTron can supply a 10-foot null modem cable (female DB 25 at both ends) for an additional \$39.95.



Available for the Atari ST
\$49.95

DOS SHELL

by Timothy Purves

The better mousetrap!

Are you "clicking" away the days with the Atari's mouse? GEM has its special features, but becomes slow, tedious, even "pointless", when dealing with many programs or files.

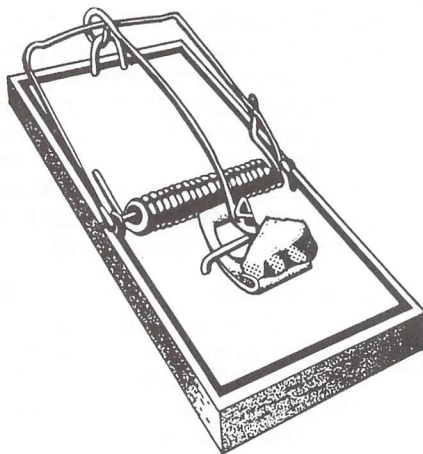
MS-DOS control - *Dos Shell* mimics the MS-DOS command format, one of the most popular disk operating systems in the business world. If you're familiar with MS-DOS, you already know how to use *DOS Shell*!

Easy to use - There's no more "dragging" your files (or your feet)! Simple, one-line text-commands give you refined control of all your disk actions. You can even use abbreviations to save on typing!

Faster than GEM - *DOS Shell* unleashes the full speed and power of your Atari ST. Commands run much more quickly under *DOS Shell* than with GEM mouse operations. After running this utility, you can get a complete directory listing faster than you could open your first GEM window!

Versatile file access - Global file-names let you copy or delete multiple files with a single command. With wildcards you can access files even when you're unsure of the full name!

Batch files - Run a whole series of DOS functions with one command, or automatically when you turn on the computer! Echo, Goto, If, Pause, Remark, and Shift commands give you even greater control.



Here are just a few of the commands featured in DOS Shell:

- **Dir** - Displays a file listing and the free space remaining on disk.
- **Change, Make, Remove** - For handling sub-directories.
- **Path** - Finds or runs programs, even from other sub-directories.
- **Tree** - Lists the file structure.
- **Type** - Display or print the contents of any file.
- **Check Disk** - Be sure that your disks are healthy.
- **Copy, Rename, Delete** - For standard file handling.
- **Pipe, Filter, Redirection** - Direct output into files or devices.
- **And much more!**

You'll be surprised by how quickly you can handle everyday disk operations. *Dos Shell* is ideal for use with your hard-disk too!

Available for the Atari ST

\$39.95

EASY RECORD

by Paul McLeod

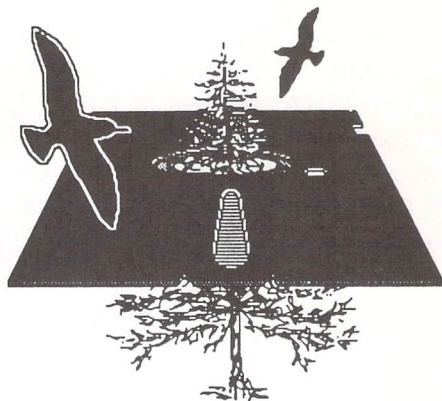
Worry free file management - *EasyRecord* means simple record management. This C programmer's utility handles all the monotonous details of file operations, letting you get back to the important work: creating your application. You can implement all the speed and power of a Binary-Indexed Record Director with the ease of calling up a function!

The *BIRD* that soars above the Trees - The *Binary-Indexed Record Director (BIRD)* takes the B-Tree a step off the ground. *EasyRecord* maintains a simple model of your file set-up at all times. This gives it a *BIRD*'s-eye view of all your records, effectively sorting, maintaining, and updating files instantly!

Handles Up to 16 files at once - *EasyRecord* applies its file handling power to up to 16 separate files at one time! It maintains up to 8 key fields for breaking data into major classes, 8 fields under each key field for minor classes, and over 65,000 individual records under each file.

Get started quickly, and easily - A simple, menu-driven file set-up utility helps you make a simple description of your file. The set-up is later used as the model for creating and accessing your data.

Everything you need - Includes the file set-up program, function library, the full source code, and complete manual--everything needed to use *EasyRecord* in your own C programs.



***EasyRecord* does it all -** It helps the C programmer in every aspect of file control and data handling:

- Sorts, maintains and updates files almost instantly.
- Creates files from a model held in memory or on disk.
- Clones open files and models.
- Deletes, updates, and inserts records easily.
- Accesses records through a full range of logical or step functions.
- Write portable C programs for a wide range of computers.

Portable - *Easy Record* works with virtually any C compiler, for easily portable code. Compilers tested include Alcyon C, Digital Research C, Lattice C, and Megamax C on the Atari, and Lattice C on the IBM.

MS-DOS 2.11 needed for use with the IBM.

Available for the Atari ST and IBM PC

\$79.95

GFA BASIC

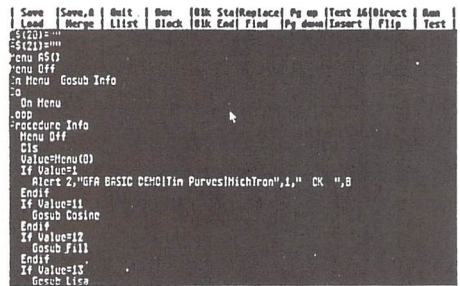
by GFA Systemtechnik

Which high level language for the Atari ST is as easy to program as BASIC yet offers full access to system features and an execution speed that rivals assembly language? The Answer? *GFA BASIC*.

GFA BASIC offers everything found in conventional BASICs, plus much more. The *GFA BASIC* resembles Modula-2 or Pascal code with ability to accept parameters from the main program as well as using local variables. Also, the various MENU commands lets you handle GEM drop-down menus with ease and efficiency.

Perhaps the most impressive functions of *GFA BASIC* are the EXEC and C: commands. EXEC lets you load and execute a non-Basic ST application from within a BASIC program. While the C: command calls a routine written and compiled in C.

GFA BASIC sports an abundance of new graphic commands, along with Unix-style DOS functions, and a host of additions to the keywords already offered by ST BASIC.

A screenshot of the GFA BASIC editor interface. At the top, there is a menu bar with options: Save, Load, Save As, Edit, Run, Block, Block End, Find, Replace, Print, Text, Insert, Direct, Run, Test. Below the menu bar, the editor window displays a BASIC program with menu-related code. The code includes comments like 'Menu Off', 'On Menu', 'Loop', 'Procedure Info', 'Menu Off', 'Cis', 'ValueMenu(0)', and 'Alert 2, "GFA BASIC DEMO! Tin Purves! HighTron", 1, " CK " , 8'. There are also conditional statements like 'If Value=1', 'Gosub Cosine', 'Endif', 'If Value=11', 'Gosub Cosine', 'Endif', 'If Value=12', 'Gosub Fill', 'Endif', 'If Value=13', 'Gosub Line'.

Other dynamic features of *GFA BASIC* include:

- Run Time Module
- Built-in Editor
- Uses only 55K of memory
- Structured Programming

The new *GFA BASIC* is able to reach beyond its own confines and incorporate routines from other languages. By permitting a BASIC programmer to call C routines and even execute other ST applications, *GFA BASIC* stretches the definition of BASIC itself.

Available for the Atari ST

\$79.95

GFA BASIC Compiler

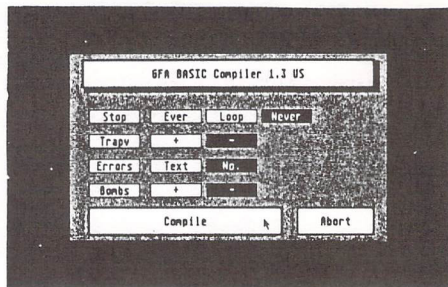
by GFA Systemtechnik

MichTron has now moved into the fast lane of the language race with an exciting new product from GFA. Presenting the GFA Compiler, a fast, 2-pass compiler which can convert your finished GFA BASIC programs into compact, faster running machine language files.

GFA BASIC has been drawing rave reviews from around the world, becoming the standard for basic programmers everywhere. Now with the new GFA Compiler you have one of the most powerful and easy to use languages ever developed.

GFA Compiler is totally compatible with the GFA BASIC Interpreter. The Compiler utilizes some of the most advanced programming techniques available to compile any GFA BASIC program in seconds, enhancing the speed and power of the already superb interpreter.

The self-contained files composed by the Compiler operate without the aid of a runtime module.



GFA Compiler utilizes all GEM features including mouse controls, windows and drop-down menus. In no time you will be converting your BASIC programs into faster running, compiled programs.

Another benefit that adds to the value of this package is that no royalties need to be paid when selling a program compiled with the GFA BASIC Compiler.

Requires GFA BASIC
Interpreter

Available for the Atari ST
\$79.95

GFA DRAFT

by GFA Systemtechnik

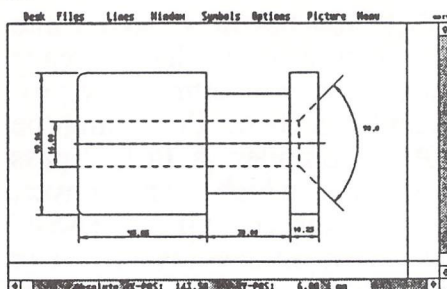
Presenting *GFA DRAFT*, a two-dimensional CAD program capable of satisfying all your drawing needs. Draw circuits, scaled technical drawings, designs and plans with ease and efficiency.

GFA DRAFT can make drawings of up to 255 layers, of which 10 can be displayed simultaneously. Layers can be printed or plotted together, each in a different color.

Scale drawing can be made easy because *GFA DRAFT* allows you to display cursor coordinates in inches, millimeters or meters; along with supplying a variable size grid. The use of the auto dimensioning function adds to the ease of scale drawing.

Your drawings can be changed and manipulated in several ways. Use the mouse to choose all, or part of the drawing. You are then able to copy, enlarge, reduce, rotate, stretch, or delete the chosen area of your drawing.

The many drawing functions and extensive use of symbols allow the user to make complete drawings in no time.



GFA DRAFT possesses the normal drawing functions such as rectangle, circle and ellipse, plus much more. Drop a perpendicular from a line, place a line at a specified angle, round corners with a variable radius, trim overhanging lines or break a single line in two. You can even shade any given area with various patterns.

For ease of use, *GFA DRAFT* is fully linked to GEM. Several help functions extend the ease at which drawings can quickly and effectively be created.

Available for the Atari ST

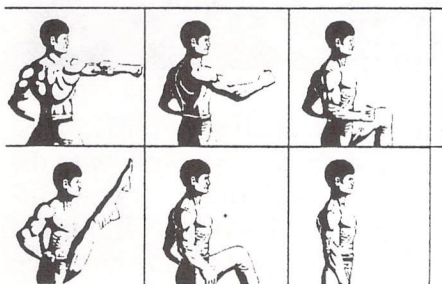
\$99.95

Make It Move

by Avila Associates

MichTron is pleased to announce the release of a dynamic new animation program, *Make It Move*, which allows you to breath life into all types of art work on your Atari ST.

Make It Move, combined with a paint program, creates an excellent, inexpensive graphics system for displaying colorful business reports, professional looking titles and animation sequences for home and industrial video recordings, as well as many other applications.



Make It Move is compatible with all popular paint programs. It has a set of utilities designed to enhance the pictures, and present them in a variety of dashing and spirited ways.

Operation of the program is graphically controlled entirely with the mouse, and it is user friendly.

You simply select a previously painted picture from any format and manipulate it using one of two programs.

The Objects program is a utility designed to let you first define an object by drawing a rectangle around it; this object can then be moved, copied to another picture, ZOOMED or DELETED. You can also move an object in front or behind another object in the picture. The objects velocity can also be controlled.

The Script function allows you to display your pictures in a series of events in any sequence and arrangement you choose. You can Cut, Fade to black or white, 2 Fade (go to black, then come up on another picture), Wipe (new picture scrolls over previous one) from any edge of the screen.

The remarkable simplicity that *Make It Move* brings to animation will astound those who have grown accustomed to the necessity of learning complicated programming commands to achieve even the most rudimentary screen movement. Prepare to be amazed, and to amaze those around you.

Available for the Atari ST

\$49.95

M-Cache

by Timothy Purves

If you're looking for ways to enhance the performance of your Atari ST, as well as increase the speed and efficiency of your hard drive; you'll want to own M-Cache, an incredibly handy disk utility.

M-Cache holds recently accessed disk sectors in your computer's memory, much like a RAM Disk, and eliminates the disk search-and-access time needed to reload old information.

M-Cache selectively saves programs that have only one sector reads (the type usually found in application programs). This prevents larger programs from flushing more frequently used procedures out of the memory cache. A vast improvement over competitive programs that indiscriminately cache everything loaded into the computer.

When M-Cache reaches its memory limit, it begins deleting sectors (the unit of measure for disk storage) based on a LRU (least recently used) algorithm. So the least used sectors are the first to be expelled.

Finally, M-Cache utilizes a write-through cache that saves your data to disk, even as it is being saved in RAM. Not only does this save you time in not having to manually rewrite everything to disk; it is also a great comfort in a power outage to know that your treasured work is saved to disk, and not lost with the normal RAM.

M-CACHE configuration program
(c) 1987 Nichtron, Pontiac, MI
Program by Timothy Purves

```
Cache drive C: (V,H) ? Yes
Cache drive D: (V,H) ? Yes
Cache drive E: (V,H) ? No
Cache drive F: (V,H) ? No
Cache drive G: (V,H) ? No
Cache drive H: (V,H) ? No
Cache drive I: (V,H) ? No
Cache drive J: (V,H) ? No
Cache drive K: (V,H) ? No
Cache drive L: (V,H) ? No
Cache drive M: (V,H) ? No
Cache drive N: (V,H) ? No
Cache drive O: (V,H) ? No
Cache drive P: (V,H) ? No
```

Sectors to be allocated for M-CACHE: 200

Settings saved in M-CACHE.M

Press RETURN

Access to small programs, files, and documents, as well as to directories and other system information, becomes lightning quick, effectively improving the transfer rate of both the reading and the writing operations.

M-Cache installs in seconds, and the user selects the number of sectors (1 - 1024) to be reserved for the memory Cache.

M-Cache can be used on any Atari ST equipped with a hard disk drive.

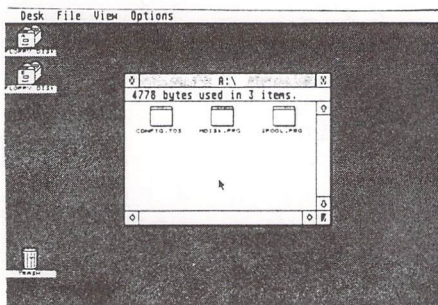
Available for the Atari ST
\$39.95

M-Disk Plus

by Timothy Purves

MichTron has now made vast improvements in their *M-disk* and *Soft Spool* programs; combining them together with even more new features, to give you a package with more practicality and convenience.

M-Disk is a RAMDisk: a software program that duplicates a hardware drive in memory. The result is that you get the equivalent of an Extra Disk Drive that behaves normally, except that it is incredibly fast and durable! The revised *M-Disk* is 20% faster than our earlier version, while it is 100% faster than most other RAMDisks sold on the market.



MichTron's RAMDisk features:

- Easy installation and use. The new *M-Disk* automatically loads into GEM when the system is initialized *without user intervention*.

- Customizing: you designate the size of memory disk you need.

- Makes two-drive operations much easier for single-drive owners.

- Faster than a hard-drive, and only a fraction of the cost!

Revisions have also been made in *Soft Spool*, a time saving printer utility that is as easy to use as it is useful. When printing files, your computer is just dead weight; waiting and waiting for the printer to finish the job. With *Soft Spool*, the long wait is over.

The revised *Soft Spool* program can now print 200 characters per second.

Now your computer can think and print at the same time! This software spooler builds a buffer into RAM which holds any printed data. You can then use your computer normally as the spooler feeds the data to the printer, saving hours of computer time.

Available for the Atari ST

\$39.95

MI-PRINT

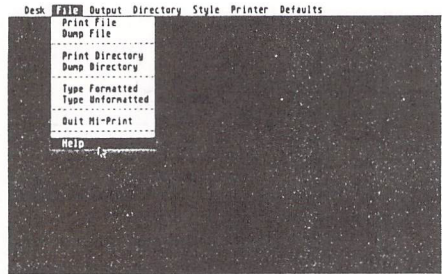
by Robert Fisher

Do you happen to be in need of your newly written program listings, on line user manuals, help files, or data from bulletin board systems printed in formatted styles? The new Mi-Print program from MichTron can do this once tedious task and much more!

Versatility- Mi-Print can print any text file you choose. It gives you command over paper size, all four margins, headers, page numbers, pitch and line spacing. The incredible advantage about this feature is that Mi-Print remembers the control codes and automatically sets up the printer to use the selected modes. Thus, saving you the valuable time of resetting the formats each time you print a file.

Convenience- If you happen to use several different styles of printers, Mi-Print lets you save them to disk and recall them later. Also, Mi-Print has the added advantage of drop down menus. This lets you learn how to use Mi-Print quickly, because there are no complicated commands to memorize; again saving you more valuable time.

Runs in color or monochrome



Simple to Control- To avoid the problem of loading your favorite printer and style sheet files every time you start Mi-Print, you can Save Defaults and they will be automatically loaded every time you run Mi-Print.

An Added Feature- You can also use Mi-Print as a typewriter for any small printing job, such as addressing an envelope, or writing a simple letter. This is of great advantage when the use of a word processor slows you down.

Still Another Added Feature- Mi-Print can give you a full disk directory listing. This means it will list every file on a disk or in a directory, including those in sub-folders! You can even change the way Mi-Print lists your directory--the way it's sorted, whether empty folder names are shown, the list of full or root directories, or even the indentation of the file or folder names.

Available for the Atari ST
\$29.95

MICHTRON UTILITIES

by Timothy Purves

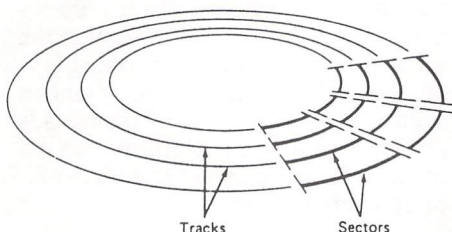
The most valuable utilities you might ever own!

What would utilities be worth that could turn back time and end frustration? Before you find out the hard way that they would be priceless, prepare yourself with *MichTron Utilities*.

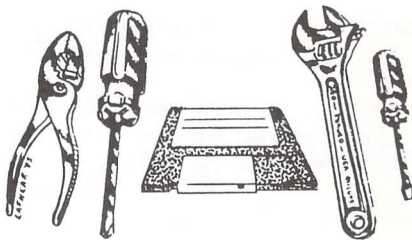
Keep control - Protect you files, even when things go wrong. This amazing set of utilities lets you recover lost data, repair damaged disks, and change individual bytes of information anywhere on your floppy disks or hard disks.

Take complete control of your disk files:

- Change file contents
- Change file and volume names
- Change file attributes
- Copy or verify individual sectors
- Format individual disk tracks
- Recover data from damaged disks
- Restore deleted files
- Repair damaged disks

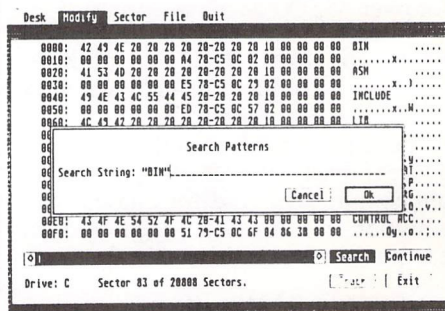


MichTron Utilities gives you more everyday flexibility to handle files and the potential to deal with trouble when it arises.



Change file attributes - *MichTron Utilities* lets you change the attributes of any file to hide it, make it a system file, make it read-only, change its date or name. You can even change the disk volume name.

Recover lost data - If you accidentally delete a file, you can recover it as long as you haven't physically overwritten it. Even a partially destroyed file may be recovered in part. Just use the mouse to accept or reject displayed data. It's that easy!



Repair damaged disks - If a disk has been damaged, you can isolate the trouble area and recover information from the rest of the disk. You can even try to reformat the affected area without harming the rest of the disk.

Available for the Atari ST and IBM PC

\$59.95

MIGHTY MAIL

by Timothy Purves

Mighty Mail does everything except lick the stamps!

If your address lists are messy and out-dated, or you sift endlessly over customer lists for the right addresses and even longer typing them out, *Mighty Mail* is a dream come true.

This utility ends hand sorting and printing. With bulk-mailings that must be sorted by zipcode, you'll save enough time and money on your first mailing to pay for the program! *Mighty Mail* knows *all* your mailing needs, and meets them conveniently.

Speed and power - Written in C for fast sorting and efficient data entry, *Mighty Mail* can maintain any address database. The number of addresses is limited only by disk size. A single-sided floppy can hold about 1,400 addresses, with the potential for over 65,000 on a hard-drive!

Easy to use - GEM's pulldown menus make operation and data entry a snap. There's room for a company name, a person's first and last names, address lines, city, state, zipcode, and phone number. And for entries with common information, *Mighty Mail* can put in the repetitive data for you!

16 user-definable flags - Special flags can be set with a click of the mouse. All entries have 16 flags. Any combination can be used as conditions for printing labels or reports. Use them to mark mailing classes or a customer's special interests. Flags can even be made to reset as an entry is printed, showing that a mailing has just gone to that customer!

Desk File Layouts Mails Printer SAMPLE

Company: HichTran, Inc.	Search Mail
Last: Hanner	Search Next
First: Gordon	Show By
Address 1: 576 S. Telegraph	Delete
Address 2:	Get Previous
City: Pontiac	Get Next
State: MI	Exit
Phone: 313-334-5700	Zip: 48051

First Record

Dealer	Dist.	Reviewer	Magazine	DEM	Author	Publisher	Misc

Versatile labels - *Mighty Mail* will make labels of any size or shape. Print labels for any range of entries in your database, using alphabetic ranges, flags, and the data fields themselves as selective criteria.

Visual layout system - Designing printed labels is easy with visual editing screens. Point-and-click on icons representing the data fields, drag them around to position them, and shrink or stretch them to the desired length! It's all done graphically, so you always know exactly what your printer prints. And of course you can save your layouts to disk for future use and editing.

Full-sized reports - Get full-page mailing lists that are always up-to-date. These useful reports feature the same versatile control and visual layout system as label reports. You can even create your own personal address/phone book! Reports can be made to fit any size paper with the 80 or 160 column option.

And more - Other features help you to delete duplicate or redundant entries, retrieve previous label or report formats, send selected records to a file, and even print test labels.

* Sanyo users--see product notes, last page.

Available for the Atari ST, Sanyo 550, and IBM PC

\$49.95

PERSONAL MONEY MANAGER

by Jonathan Kring

Trying to figure out just where you stand financially? You'd find it much easier if MichTron's *Personal Money Manager* were helping you out. It's almost like having your own personal accountant ready to go to work any time, day or night.

Find out how much you made from your brokerage account, from your weekend moonlighting, from your sales at art shows. Learn how much you've spent on groceries, on the kids' dental work, how much you've put into repairs on the old car.

Compare your budget to your actual expenditures at a glance, then decide how to deal with the differences.

And you don't have to be a computer genius or an accounting star to figure out what's going on. At the first of the year, type in your budget, one account after another. After that, you just enter your income and bills and let *Personal Money Manager* do the record keeping.

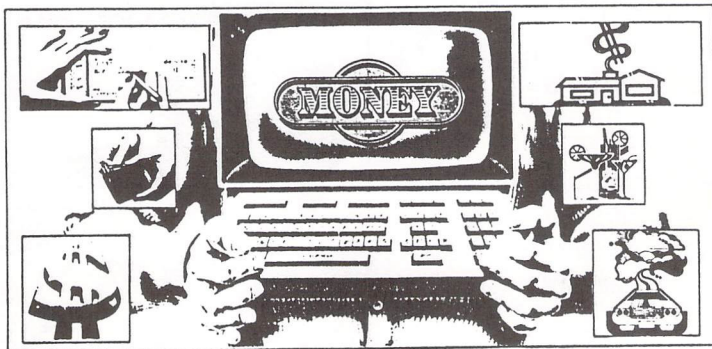
For every transaction, there's space to describe where the money came from, where it's going and what the purpose was.

Getting reports is even easier. Just use your mouse to point to the report you want, turn on the printer, click the mouse and watch the report roll out.

Personal Money Manager is your personal accountant:

- Keeps track of up to 999 separate budget and account categories.
- Automatically posts transfers that occur regularly, for up to eight income or expense categories.
- Balances your checkbook, and all your other accounts automatically.
- Uses your printer to get detailed reports on any or all financial categories.
- Makes summary reports without detail lines.

Personal Money Manager uses GEM's pull down menus, so you don't have to learn a set of complex commands.



Available for the Atari ST, Sanyo 550, and IBM PC

\$49.95

R.A.I.D.

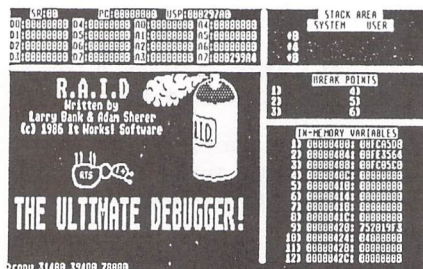
by Larry Bank & Adam D. Sherer

Introducing a powerful tool to help increase your productivity and efficiency. *R.A.I.D.* is an all new, complete debugger, with more features to help with the development of your applications.

R.A.I.D. features include:

- **Mini assembler/disassembler** - Lets you easily make temporary patches of bugs you may find in your target program.
- **Tracer** - A full set of options at your fingertips.
- **Copy, fill, and move block commands** - Copies a block of memory between the start and end addresses to the destination address. Also, fill address space between start and end with the specified byte.
- **Breakpoints** - maximum of six breakpoints in a table with full display given on the right side of the screen.
- **Symbolic debugging of all global symbols** - Includes all symbols defined in most C compilers.
- **Three methods of loading files** - Loads any type of file into memory and reports the start and end address after a successful load.

The most useful feature about *R.A.I.D.* is that it maintains two graphic displays. This allows the program screen being debugged, and those of *R.A.I.D.* to be completely isolated from each other; yet you may see either at any time.



To improve your programming efficiency, *R.A.I.D.* also allows for the inclusion of labels in your program. This will alleviate some of your debugging headaches since it makes the listings from the dump command more readable and will help you locate specific parts of your program more easily.



Available for the Atari ST
\$39.95

Realtizer

by Print Technik

Realtizer from MichTron, is a new utility that allows you to digitize images from a video camera or video cassette recorder! The high resolution produces pictures that are very similar to the original, perfect for reports, letters, advertisements, or just for fun.



You can digitize your pictures with a different number of graylevels and at four different speeds. This allows you to capture various shades and very rapid movements. Multiple grays produce excellent pictures of people, while fast movements are picked up best when using two or four grays.

Your pictures can be saved in NeoChrome, Degas, Doodle, Art Director, and Bit Image formats. This gives you the convenience of saving your pictures in whatever format is best suited for your needs.

If you would like to make dramatic changes to your picture you may use the editor in the program of the format you saved your image in.

Realtizer also allows you to make real color images instead of only grayscale ones. The program comes with three filters: red, green, and blue.

Using these filters to shoot your image, the software will calculate the colors in the original image and show them on the screen or printout.

Another feature of the *Realtizer* is the Toolbox. It lets you take pictures saved in NeoChrome or Degas formats, make changes to them, then save them back to disk without having to switch disks.

When printing, the size of your printout can be adjusted to fit your needs with the *Realtizer's* powerful, but simple to use, built in printer drivers.

Now you can have those pictures that can make all the difference, with the easy menu-driven, GEM-based utility, *Realtizer*.

Available for the Atari ST

\$199.95

STuff

by Timothy Purves & J. Weaver Jr.

Factory Programming and MichTron, Inc. are pleased to announce the release of STuff, for the Atari ST computers.

This package is the first collaboration between two outstanding Atari ST programmers: J. Weaver Jr., Senior Programmer for Factory Programming and author of MI-TERM, CORNERMAN, and CARDS; and Timothy Purves, MichTron's Senior Programmer and author of BBS v2.0, MIGHTY MAIL, and DOS Shell.

STuff is a package of 21 different utility programs for all ST users, from the novice to the professional programmer. The programs included are:

512K - "locks out" system RAM to run memory-sensitive programs

AUTODATE - sets the date and time without a clock card

AUTOFOLD - changes the execution order of AUTO folder programs

AUTOGEM - autoboots a selected GEM program from the Desktop

CAPSLOCK - avoids accidental activation of the CapsLock key

FC - compares binary data files, displaying the differences

FDEL - deletes files with no chance of unauthorized recovery
FILELOCK - encrypts and decrypts disk files.

GREP - searches text files for selected character strings

HARDAUTO - runs AUTO folder programs from the hard drive

HEADER - displays technical information about program files

HEX - displays binary data and program files in hexadecimal

HIGH - allows other AUTO folder programs to run in medium rez mode

KEYCODE - displays keyboard scan and ASCII codes for selected keys

KEYCOMBO - sets up 4 "hot keys" for screen, printer and reset functions

ONEHAND - allows one-handed use of Alternate, Control and Shift keys

RESET - flushes memory completely whenever the system is rebooted

STSELECT - selectively enables AUTO folder programs and desk accessories

TOUCH - sets the date and time of selected files

UNHIDE - sets or resets disk file attribute flags

VERIFY - speeds up disk writes when verification is not necessary

Available for the Atari ST

\$39.95

Super Conductor

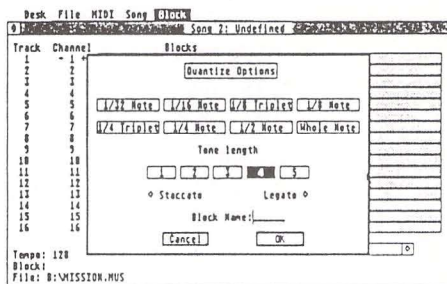
by Philip MacKenzie & Jeffrey Sorenson

Presenting *SuperConductor*, the professional 16 Track MIDI sequencing package that no musician should be without. Loaded with powerful features, *SuperConductor* can quantize (auto correct), filter, transpose, edit, and mix music played through a synthesizer. Comprehensive editing gives musicians precise control over their performances, including the alteration of controllers such as volume pedal, pitch bend, and modulation settings.

Using *SuperConductor's* unique "Block" song structuring, musicians can set up elaborate arrangements with minimal time and effort. A block of music can be repeated, interwoven with other blocks, placed on different channels at different times and more, all within the easy to use song edit mode.

The powerful song edit menu display allows musicians to visualize the structure of their music, and see how various blocks will be combined during a performance. This Block Structuring places *SuperConductor* a generation beyond the out-moded Punch In/ Punch Out tape style editing featured on many MIDI sequencers.

SuperConductor is ideal for performers allowing them to hold up to ten different songs in memory at once, and quickly change from song to song to play and edit.



Each song buffer has its own individualized settings for its tempo and song structure information. Blocks may be copied from one song buffer to another at will.

The *SuperConductor* main menu displays the names of the ten songs currently held in memory and gives access to the System Exclusive data transfer mode. This versatile feature allows MIDI synthesizer voices and data to be transferred and saved on disk. The simplified data storage method allows the System Exclusive data to be edited using a word processor.

Available for the Atari ST

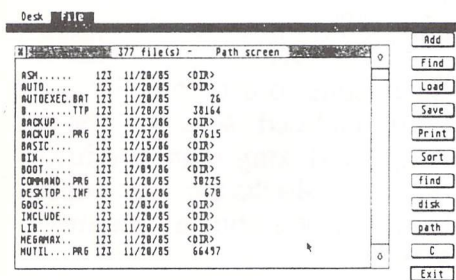
\$79.95

Super Directory

by Mark Feldman

Super Directory is a powerful but easy to use disk cataloging program which runs under the GEM operating system. It allows you to keep track of all your floppy and hard disk files in a very convenient format.

Super Directory has the capacity to add a remark of up to 25 characters to each disk file entry. The remark can be used to identify what each file contains, or what its function is.



Super Directory features many useful commands:

Find: Makes it easy to locate files in your collection. Finds the next occurrence of matching records using the Search masks. Super Directory is also capable of full use of wildcards and dummy characters.

Load: Loads specified data file from disk into memory.

Print: Sends all matching records to the printer instead of the screen.

Disk: Finds the next disk in the data file. Allows you to step through records by disk number.

Path: Allows you to step through your data files by pathname.

Sort: You can sort records by category, disk number, extension, filename, pathname, or remark.

Editor: Allows user to edit records.

And More!

End hours of headaches caused by the tedious search through your file collection. Super Directory makes finding and organizing your files fast and simple.

Available for the Atari ST

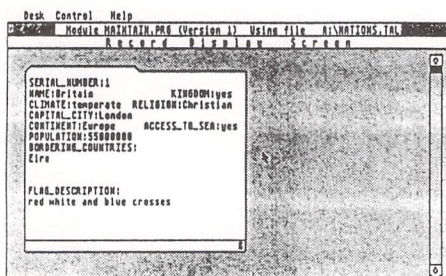
\$39.95

TRIMbase

by Talent

TRIMbase, a new Data Management system from MichTron, is so simple to use that you don't need to learn a special programming language; yet it still has the power needed to handle large amounts of complex data. While your application grows in size and in the demands you make of it, **TRIMbase** will continue to supply the efficient, quality support you need.

Convenience - Input and maintenance of data are provided by one simple command within the system. With **TRIMbase** you can also update and delete records, retrieve information, search, and sort. Not to mention altering the descriptions of files which have already been defined and loaded with data.



Customizing - **TRIMbase** is also capable of generating custom reports. Production of reports can be both as tables and as free text.

Versatility - **TRIMbase** is even capable of exchanging data between other software applications, along with having other valuable features such as help options, point-and-click controls, and pull-down menus.

- **Projection:** makes files smaller and easier to handle.
- **Selection:** used for partitioning a very large file into a number of smaller ones.
- **Merge:** form a single file from two files with identical record cards.
- **Subtraction:** subtracting a file from another.
- **Join:** constructs a new file with data collected from two or more files with different card and layout fields.

TRIMbase is a complete data management system.

Available for the Atari ST

\$99.95



Printed in Canada

GFA VECTOR

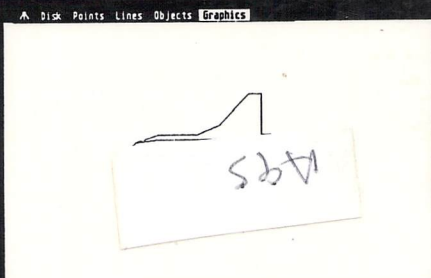
by GFA Systemtechnik

The incredible power of the GFA-BASIC Interpreter is increased dramatically with the introduction of the three dimensional graphics program, GFA-VECTOR.

Now you can create and manipulate astounding images and optical effects, and place them into your own GFA-BASIC programs. Since GFA-VECTOR creates pictures written entirely in machine language, you are able to rapidly update the screen, thus allowing the creation of real time animations!

GFA-VECTOR lets you create objects (The basic component of a 3-D graphic or animation) two ways. First, points and lines can be defined by entering their coordinates with the keyboard. Or the objects can be created interactively on the screen with the aid of the 3-D graphics editor. With this potent editor an object can be viewed from the top, front, or side, and modified effortlessly with the cursor.

GFA-VECTOR provides a concise tutorial that makes the program so easy to learn and simple to use, that even the most maladroit artist will soon be turning out exceptional graphics and animations.



Objects created with GFA-VECTOR can be revolved along any of the three axes in one degree increments through any order of rotation. These objects can be scaled in 511 increments from 1/64th to 8x normal size. Each object file can have up to 1024 defining points and 1024 defining lines. Up to 32 objects can be stored at once in a single file.

It's time to put action and adventure into those dull computer sessions, and GFA-VECTOR will make your BASIC programs live and breath in ways you never dreamed possible.

Requires GFA-BASIC Interpreter
Disk Enclosed



MichTron

576 S. TELEGRAPH, PONTIAC, MI 48053
ORDERS AND INFORMATION (313) 334-5700



GFA Vector

Atari ST 3-Dimensional Graphics Program
by GFA

MichTron

Copyright 1985



Nº 1373

