

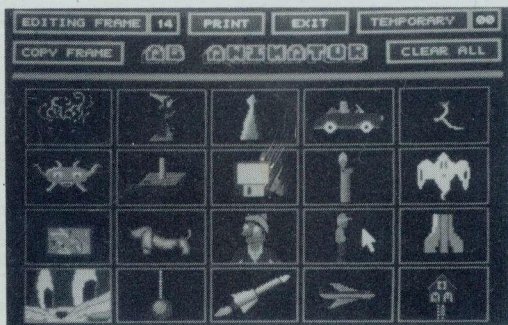


ADBROKE
COMPUTING
INTERNATIONAL

YOU'D LIKE TO WRITE YOUR OWN GAME IN BASIC,
BUT YOU CAN'T HANDLE THE ANIMATED GRAPHICS?
NOW WE HAVE THE ANSWER . . .

AB Animator

BY AB SOFTWARE
FOR THE ATARI ST



- * SUPPORTS GFA BASIC AND FAST BASIC
- * ANIMATE UP TO 20 BIG FRAMES (56 pixels wide by 33 pixels high)
- * FULLY ICON CONTROLLED
- * COMPATIBLE WITH DEGAS AND NEOCHROME PICTURE FILES
- * FULL DESIGN FACILITIES SUCH AS DRAW, FILL, GRID, COPY, etc.
- * MULTI-SCREEN LAYOUT
- * COLOUR SELECTION FROM DISK, OR CONTROLLABLE FROM AB ANIMATOR
- * VCR STYLE ANIMATION CONTROLS
- * SAVES FRAMES, COLOUR PALETTE AND ANIMATION DATA IN BASIC FORMAT
- * INCLUDES BASIC ROUTINES AND FREE DEMONSTRATION PROGRAMS

AB ANIMATOR

FROM AB SOFTWARE

(The people who brought you Quick-List ST)

INTRODUCTION

AB ANIMATOR is an icon driven graphics package for the ATARI 520 ST series of computers, which allows you to design up to twenty frames or pictures using 16 colours, and then, like a cartoon, replay them at various speeds to create a movie effect.

At your fingertips are controls for drawing, filling, flipping and inverting frames, use of 16 from a possible 512 colours, the ability to copy from frame to frame, put a frame into storage while you load/save frames, clear frames, animate at your choice of speed (including reverse), single step frames, move your animation about the screen and more besides.

And don't worry if you aren't the ST world's answer to Picasso, there are plenty of demo frames on the disks. and the program also has the ability to capture frames from picture files created with the two most popular art programs DEGAS (low resolution P11 files) and Neochrome (NEO).

Your movie will last only seconds, but with AB ANIMATOR, you can take your creations and control them with your GFA BASIC or FAST BASIC programs, and what is more, with the modules provided, you don't even have to understand what is going on!

On disk B, you will find BASIC demonstrations for bot GFA BASIC and FAST BASIC showing typical animations with frames created with AB ANIMATOR.

Also on the disk, is a series of files which you simply merge into your own programs to load picture files

and AB ANIMATOR frames, plus CONVERT.PRG a special picture file convertor which is explained in full later.

A full list of both disks contents can be found in the README.DOC.

What you produce in the way of programs with AB ANIMATOR is up to you (and your imagination) and some programs you need might even be of a standard to market.

If you do so, you need not worry about us grabbing a slice of your profits, just a little mention of AB ANIMATOR somewhere on the screen or packaging would be just great!

Whatever you do, the aim of the game is to have fun — so go to it and enjoy yourself.

SPECIAL NOTICE

AB ANIMATOR is compatible **ONLY** with **GFA BASIC** from GLENTOP PRESS LTD. and **FAST BASIC** from COMPUTER CONCEPTS if you wish to use the graphics in your own software.

To **RUN** this program you do not need either, but to make full use of it (in your own programs) you will need the **GFA BASIC INTERPRETER**, or **FAST BASIC**'s interpreter on cartridge or disk.

Programs created with **AB ANIMATOR** and its' associated modules will be 100% compatible with the **GFA BASIC COMPILER**.

AB ANIMATOR was written in **GFA** and then compiled.

If you do not own an extended BASIC such as **GFA BASIC** yet, pop down to your local ATARI dealer for a demo — your trip won't be wasted.

Thanks

Thanks to Andy Quayle for the **FAST BASIC** demonstration routines and loading modules, and to Graham from technical support at Glentop Press for his assistance.

LOADING

Ensure that your ST is correctly connected as shown in your manual, and is switched off.

AB ANIMATOR is on two half megabyte disks, and will run on any ATARI ST, but will run **ONLY** in low resolution, so you need either a colour television or colour monitor connected depending on which ST you have. (Some ST's do not have a socket for a standard television and require an RGB colour monitor for colour programs).

Put disk A into drive A and turn on the power.



ANIMATOR.PR

G When the green desktop appears, it will already be in low resolution, and you will see a small window with the PRG icon and the filename ANIMATOR.PR below it. Move the mouse pointer onto it and press the left mouse button twice as quickly as you can.

If you are successful, the 'busy bee' should appear to say that the program is loading.

If you have problems with this method, another way to load AB ANIMATOR is to click once on the PRG icon to turn it black, and then go to the 'pop-down' menu at the top of the screen and point at FILES and when the drop-down list appears select 'OPEN' and click it on.

It should be noted that AB ANIMATOR will also run from drive B if required, and as such, all subsequent disk access will be from drive B. For more details regarding loading PRG files, see the ST's owners manual.

A few seconds into the load, the screen should turn black. Don't worry it is supposed to! What you

shouldn't get is a row of bombs across the screen, but if you do, simply turn off your ST, wait a few seconds then power on and repeat the procedure again. If you suffer continual problems, check all connections again, and try another disk. If that one loads, pop back to the shop you bought AB ANIMATOR from and they will check it for you, and replace it should it be necessary.

Assuming all has gone well, the next thing you should see is the loading screen followed by the initial control panel. This is the central point of AB ANIMATOR and is where you return to after using the files, design and animation sections.

We have designed AB ANIMATOR to be as user-friendly and simple to use as possible, so if you want to get stuck in, you can now go and explore and return in a minute, or you can turn over the page and we will go through the different screens and features in more detail finally creating your first animation step by step.

USING AB ANIMATOR

THE MOUSE POINTER



AB ANIMATOR is controlled totally by the mouse using the 'mouse pointer' — a small arrow on the screen. Using the pointer, you tell AB ANIMATOR what you want to do.

All commands are found in boxes. Putting the tip of the arrow inside the box and pressing the left mouse button once will select that command.

The only time the keyboard is used is when you need to type in a filename when saving a set of frames or a colour palette.

DATA DISKS

Eventually, you will want to save data onto your own data disks. These **MUST** be already formatted, as AB ANIMATOR has no facility for formatting a disk.

FOLDERS

When formatted (to half megabyte capacity or one megabyte), it is strongly recommended that you put two folders, **FRAMES** and **PALETTES** onto the disk using the drop down menus on the desktop.



FRAMES **PALETTES** Use item **NEW FOLDER** to create the 2 folders.

The **FRAMES** folder will be automatically opened by AB ANIMATOR to save your frames into, and the **PALETTES** folder will be used for colour palettes. This keeps your disk un-cluttered and tidy, and you always know what is where.

For more information on creating folders or if you are unsure of the procedure, consult your Atari ST users manual.

THE MAIN COMMAND MENU

After a successful load you should see the Main Command Menu like the diagram (below).



On it you will see four boxes with **FILES**, **EXIT**, **EDIT FRAMES** and **ANIMATE SPRITE** in them.

On the diagram they are labelled with the letters **A** to **D**, and the instructions will refer to them for easy reference.

Your 'movies' are created in stages which are all accessed from here.

When you are finished with any section (you may move freely between sections during frame design), you will always return to this screen.

FRAME CREATION ORDER

As a rough guide to the order of creating an animated sequence from scratch, and there are many, you would probably use FILES (A) first to load colours and/or frames to modify, or even capture a frame from a Degas or Neo picture. Next would be EDIT FRAMES (C) to copy and edit the frames, and then back and forth to ANIMATE SPRITE (D) to see the frames in action. Finally, back to FILES (A) to save your completed frames or partially completed frames for finishing off in another session. Finally you would EXIT (B) from AB ANIMATOR to the desktop.

Points to remember . .

FILES (A) is the section where all the reading and writing to the disk is done.

EDIT FRAMES (C) is where you draw, copy, merge and modify the frames.

ANIMATE (D) is the section that gets your frames moving.

Let's assume you want to get straight into drawing a frame.

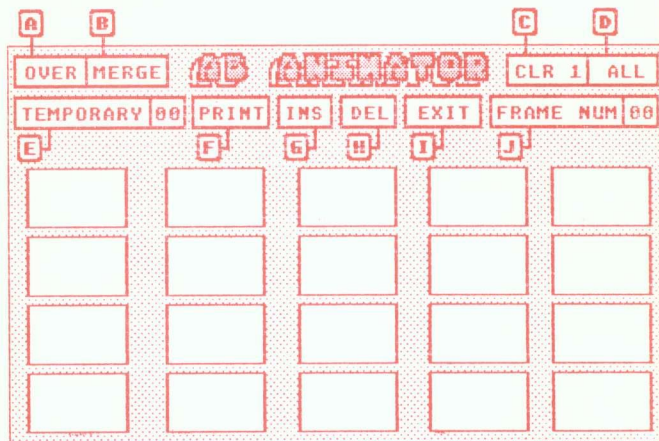
Take the mouse pointer on the screen so that the point is inside the box marked EDIT FRAMES (C) and press the left mouse button.

The screen should now have changed and you should be looking at FRAME EDITING SCREEN 1, so let's turn the page and take a closer look at it.

FRAME EDITING SCREEN 1

From this screen you can choose which of the twenty frames you want to design or edit. Each time you enter this screen all existing frames are displayed in their current state.

First of all, let's look at the screen layout . .



Each of the frames are displayed in the lower part of the screen, and the control panel is at the top. For easy reference, each part of the screen is labelled with a letter as shown below.

- A OVER — Copy frame using overwrite.
- B MERGE — Copy frame using merge.
- C CLEAR ONE — Erases a single frame from memory
- D ALL — Erases all twenty frames.
- E TEMPORARY — A protected temporary storage area for a single frame.
- F PRINT — Output series of frames to printer.

- G INS — Insert a NEW blank frame.
- H DEL — Delete frame and close gap.
- I EXIT — Returns you to the main menu screen.
- J FRAME NUM — As the mouse pointer passes over a frame, number here indicates the frame number.

If you have entered this section for the first time, and have not loaded any frames in, then obviously the twenty boxes will be empty. Turn the page and we will now go through each section in more detail.

In this state, the natural thing would be to start work on the first frame, although should you wish, you could use only frames seven through to fourteen for your animation. The choice is yours!

COPYING FRAMES

In the process of creating frames, the following two commands are invaluable. Having spent a while creating a frame, you don't want to have to remember how to redraw the object exactly the same way and then alter it slightly. OVER and MERGE allow you to make a carbon copy of your first frame or produce a copy of two merged frames and then work on your copy.

You may wish to put the destination frame into TEMPORARY before or after copying/editing it, so that you can restore the frame if you make a mess of it.

A OVER

To use OVER, just click on the OVER box once to turn the function on. The OVER panel will highlight. When highlighted, all you need to do is point to the frame

you want to copy (the source frame) and press the left mouse button to highlight the frame. Next point at the frame you want to put copy into (destination frame) and press the left mouse button again.

There you have it, a copy of your frame. Notice that the source frame is un-affected, and remember that anything already in the destination frame is permanently lost.

B MERGE

To carry out a merge, click on MERGE followed by the frame you wish to copy and again the chosen frame will be highlighted.

Next click on the occupied frame you want to copy over. (Note that in MERGE mode the destination frame is not deleted as the source frame is placed over it).

Repeat the process to copy other frames.

C CLEAR ONE

Clicking on this box will allow you to erase a frame from memory. After clicking on this box, you will be asked to confirm your intentions (just in case you click on the box by mistake). You must then point to and click on the frame to be deleted.

Once deleted there is no way to retrieve it.

D ALL

Clicking on this box will allow you to erase all frames from memory. After clicking on this box, you will be asked to confirm your intentions (just in case you click on the box by mistake). Once deleted there is no way to retrieve them.

E TEMPORARY

This allows you to put any ONE of the twenty frames

on the screen into temporary store, where it will be un-affected by future operations until you want to retrieve it. The ideal situation would be where you want to transfer a frame from one file on the disk, to another file, or when you want to use one frame from a set of twenty, yet delete the rest. An example is given later in the section entitled 'LET'S MAKE A MOVIE.'

Using the feature is simplicity itself. First, click on the TEMPORARY box, and the panel will highlight. Then click once on the frame you want to store. That frame number will appear in the box next to TEMPORARY, and the highlight should go off.

You now have that frame stored, and you are now free to delete all frames, or load in a new set of frames without the frame in TEMPORARY store being altered.

To retrieve it, you click on the TEMPORARY box again and it will begin flashing. This tells you the feature has been activated, and that you are to indicate which frame the stored frame is to be put into by clicking on the required frame in the lower section of the screen.

To retrieve is a simple 'replace' feature, and if the frame you are placing it into is not empty, then the contents will be lost completely. As the frame appears, the box will cease flashing, and the TEMPORARY store number will revert back to '00'.

TEMPORARY storage works in a 'once-in once-out' fashion, so only one retrieve is possible. Also, by checking the number you can tell if a frame is in store at a glance. Remember — 00 means storage empty.

Note that the frame stored in TEMPORARY is not immune from colour change or colour loading from palettes or GFR files, and the retrieved frame will be in the same colours as the current palette.

F PRINT FRAMES

If you have a dot matrix (graphics) printer which responds to the standard ALT-HELP screen dump correctly, you can output a SEQUENCE of frames to it with this feature. To be compatible with as many printers as possible, the printer setup is controlled from the desktop before loading AB ANIMATOR by the drop down menu under DESK by use of the CONTROL.ACC on the main disk.

Simply click on the print box to view frames, then click the mouse button again to continue.

The way it works may seem a little odd at first, but after a while you will realise its advantages.

The function can print any range of animation frames or all of them, along with the filename of the frames taken from the time they were saved/loaded. If the frames have neither been loaded or saved at the time of printing (newly designed), the heading is omitted, so SAVE before printing.

The number of frames printed is controlled from the ANIMATE SPRITE stage taking the values from the film start and end frame numbers. If these values are set to 1 and 20, then all frames will be output (even if unused). The advantage of this unusual arrangement is that if only 5 frames are needed for an animation, four separate 'movie sequences' can be fitted into the 20 frames. (1-5, 6-10, 11-15 and 16-20). Animating any of the four, you will have to set the start and end frames and then only that sequence will be printed.

If you carefully cut out the frames and staple them together along one edge in the correct order, you can make and amusing 'flick-book' cartoon of your animation. Even better results can be achieved if you plan out your frames and join up more than one set of 20 frames.

To exit printing before the end, simply hold down the Alternate key and press Help key once. (You may get a delay before the printer stops if you have a printer with an in-built buffer).

G INS

This feature enables you to insert a new blank frame within an existing set of frames, but not lose the frame chosen.

Simply click on INS, then point to and click the left mouse button on the frame where you want to insert a new blank frame. All frames to the right of your chosen frame are shifted one place to the right.

If you have used frame 20, then you will lose it. (That is why you have to confirm your actions).

H DEL

This feature enables you to remove a single frame, without leaving a gap in the sequence.

Simply click on DEL, then point to and click the left mouse button on the frame where you want to delete the frame.

All frames to the right of your chosen frame are shifted down one place to the left. Frame 20 will become empty.

I EXIT

As you would expect, this is where you click to leave this screen and return to the main menu. All frames remain intact when you exit. To animate your frames, you must exit this screen here to get to the ANIMATE SPRITE section.

J FRAME NUM

This feature merely shows you at a glance the frame number your mouse pointer is over.

Not exactly earth shattering, but it does get tedious having to count the frames up to the one you are going to work on.

THE FRAMES

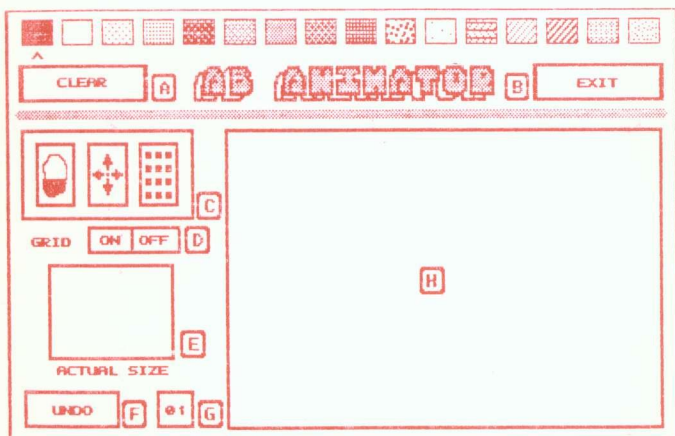
The remaining twenty boxes are the twenty frames you put your animations into and they are 56 pixels wide and 33 pixels high.

To create a frame or edit an existing one, you simply click the mouse pointer on the one you want to work on. The screen will then switch to EDITING FRAME SCREEN 2.

FRAME EDITING SCREEN 2

On this page we will look at EDITING SCREEN 2, so if you have been following the instructions and your frames are empty, put the mouse pointer on frame 1 and click on the left mouse button.

This is where you can put your artistic talent to work creating your frames of animation. Once again, let's look at the screen . . .



This time the controls are found on the left hand side of the screen and across the top. The major part of the screen is taken up by the enlarged view of the current frame. As before, to make things easier, parts of the screen are labelled as explained below.

A CLEAR

— Erase both large and small frame windows from memory

B EXIT

— Takes you back to Editing Screen 1

C CONTROL PANEL

— Frame creation utilities

D GRID CONTROL

— Large frame drawing grid control

E NORMAL FRAME

— Actual size frame window

F UNDO

— Reverts to previous state should you make a mistake

G FRAME NUMBER

— Indicates the number of the current frame

H WORK FRAME

— Enlarged view of the current frame, and the window that you work in.

When you enter this screen, any graphics previously created in the current frame will appear in both frame windows. The grid function will be off, and for obvious reasons, the UNDO function will not operate.

A CLOSER LOOK

We will now take a closer look at the 8 sections (A-H) as shown on the previous section in more detail.

If you are following the book, the frame number should read '01'.

Across the top of the screen are the 16 colour bars. Below the first box is a small white '^'. This mark always indicates the colour in current use, and upon entering this screen it is always set to the colour last used in this section or black when entering for the first time.

Selecting another colour couldn't be simpler — just point

at it and click on the left mouse button. The marker will now move under the newly selected colour.

A CLEAR

This box simply allows you to wipe the slate clean and start the frame from scratch. Both actual size frame and large frame are both cleared. The undo feature is cancelled.

If you have put the current frame into TEMPORARY storage from editing screen 1, then you can restore it when you EXIT as it will not have been affected by CLEAR. Naturally, if it was not stored, then you will have lost it for good. For this reason, the function has to be confirmed when selected to guard against you clicking on the box by mistake.

B EXIT

EXIT takes you back to EDITING SCREEN 1, storing the frame as it stands.

C CONTROL PANEL

Basically, the control panel is a set of utilities or tools for manipulating the frame as a whole.

With these commands, you can move objects around within the frame, mirror the image, and even turn your works of art upside down.

Also you have the capability to change the colour palette of 16 colours to suit your needs.

As it has its own set of commands, it is dealt with in detail in the next section.

So read on and let's take a look at it . . .

The control panel shows three icons. Reading left to right, the icons represent FLOOD FILL, FRAME

SHIFT and COLOUR PALETTE.

FLOOD FILL

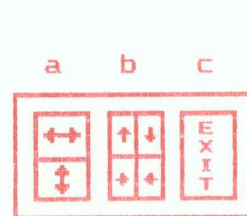
Click on this icon, and the box is highlighted to indicate FILL mode. From this point on, instead of drawing points on the large frame, AB ANIMATOR will change the colour of all points connected OF THE SAME COLOUR up to the border with another colour. Use this to say for example change the colour of a cartoon characters nose from one colour to another.

The ability to change the current colour is not affected, and the FILL function works with the grid mode on or off, although it is faster with GRID turned off.

FRAME SHIFT

When you click on this icon, the whole control panel changes to show you a second control panel. Try it and see.

The panel should now look like this:



a FRAME FLIP

This is divided into two icons

At the top:

HORIZONTAL FLIP — objects facing left change to face right and vice versa

And at the bottom:

VERTICAL FLIP — objects are turned upside down

b FRAME SHIFT

The middle icon is divided into four quarters, each with an arrow in it. Clicking on the arrows will shift the contents of the frame in the direction of the arrow

chosen by one pixel (dot).

SPECIAL NOTE

After using this section of AB ANIMATOR, the UNDO (F) feature does not restore the frame. Should you wish to guard against accidents, use the TEMPORARY function on Editing Frame screen 1.

c EXIT

You must use this icon to return you to the original control panel. Should you forget, it will flash to remind you.

COLOUR CONTROL

The last icon of main control panel is colour palette control. The colours set up when AB ANIMATOR is loaded may not be the choice of colours you need for your graphics. If you want to change any of the 16 colours follow the steps outlined here:

Choose the colour you want to change FIRST from the colour bars at the top of the screen.

Click on the colour palette icon on the control panel.

Another panel will appear in the centre of the screen with R G and B and three numbers.

The R G and B stand for Red, Green and Blue respectively, and the numbers are the current settings for each of the red, green and blue constituents in the chosen colour. (All 512 colours on the ST can be produced by a combination of these three values).

Pointing the mouse at these numbers, the LEFT mouse button increases the value, and the RIGHT button decreases the value.

As you alter the numbers, the colour bar chosen at the

top of the screen will change, as will all parts of the frame in that colour.

The values are in the range 0 (off) to 7 (full on). If you want a full green for example, set the G value to 7 and the R and B to 0. Why not try it!

When you have the colour you want, click on EXIT.

NOTE

Obviously, with 16 colours in this screen mode, when you change any of them, the AB ANIMATOR control panels will change colour also as they have to use the same 16 colours. But, unless you change all 16 colours to black let's say, you should not have any problems.

When you exit to any screen which does not display the frames, the palette reverts temporarily back to the original AB ANIMATOR one.

When you come to SAVE your frames, YOUR colour palette is saved with them and naturally loaded with them when you carry out a 'LOAD'. For differences in the way GFA and FAST BASIC colours are saved, see the sections on the relevant BASIC's later in the manual.

This way your own programs can be set up with the same colour palette for animations.

D GRID CONTROL

The grid mode is selected or deselected by clicking on the boxes ON and OFF respectively.

The feature is especially useful when you want to create mirrored objects, as the individual square pixels (dots) in the large frame appear as solid blocks of colour when the grid is off.

Switching grid mode on, as the name suggests, creates a grid effect whereby the pixels can be easily counted

to allow the creation of symmetrical objects.

FILL mode operates with grid mode on or off.

E NORMAL FRAME

This is an actualsize representation of the frame you are currently working on. It is updated in real time as you work on the large frame.

The mouse pointer has no effect on this part of the screen.

F UNDO

This feature will no doubt get you out of quite a few awkward situations while getting used to AB ANIMATOR, and possibly even when you ARE used to it!

The pixels (dots) you draw inside the large frame between first pressing the button and the moment you release the button are known as a pixel event. Using the UNDO (F) feature, the frame is restored to the state it was at the end of the previous pixel event.

So if you forget that you are in FILL mode and wipe out half of the frame, you can restore it to how it was ... instantly!

G FRAME NUMBER

This box just displays the number of the frame you are working on.

H WORKFRAME

This is where it all happens!

Enlarged to a ratio of 4:1, this part of the screen is where you draw your pictures after you have selected the colour and features you want to use.

LET'S GET DRAWING!

We can draw something on the workframe (H) by moving the mouse pointer over where we want a dot and pressing the left button. (Lines are drawn by holding down the button).

Now make a few squiggles on the large frame (H) by pressing the button only once and holding the button down. When you have done that release the button and move off the work frame.

As you did that, did you notice the same pattern appearing on the 'actual size' frame (E)?

At this stage your doodle is the first pixel event, and before the screen was empty. Go now and click on UNDO.

Done it? Good — the screen should now be empty.

Re-draw another doodle. This time, after using one colour, choose another one and add a bit more to your masterpiece. Then UNDO (F) again.

Note that this time (assuming only one button press for each colour on the frame) the second colour disappears as the frame reverts to the state after the previous pixel event.

Continue and make up a small design using as many colours as you like and using UNDO (F) whenever necessary.

To start a new design click on the CLEAR box (A).

To see what the GRID (D) looks like, click on the grid ON box. Like this you can quickly check how symmetrical your design is by counting the dots. Note that the actual size frame is unaltered. Now turn OFF the grid (D).

Now then . . . what else can we do? Oh yes, the CONTROL PANEL (C).

Have a go at vertical and horizontal flipping and moving the frame around.

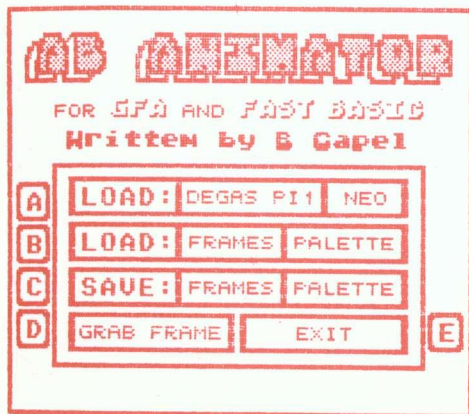
Feel free to experiment with drawing in different colours, using the control panel options, moving and flipping the frame until you are used to the way they work.

When you have messed around for a while, I will assume that you don't want to keep your results so clear the frame and exit this section to EDITING FRAME 1, then exit again and you should find yourself back at the Main Command Screen.

We will now go and look at the section on disk access, so go and click on FILES and read on.

FILES

This is the section you come to when you want to access the disk, so let's take a look at the screen . . .



As you can see, the screen has 8 icons in 5 main groups labelled A to E.

Of the 5 groups, 2 are for loading, 1 for saving, 1 for capturing frames loaded with group A and the exit icon.

Here you can load and save frames and colour palettes and load Degas and Neo picture files and then grab frames from them. The options will now be covered in more detail.

A LOAD (i) DEGAS

With this function you can load a picture file into AB ANIMATOR from the popular graphics and art program Degas Elite. When you click on this box, a file selector appears for you to choose the picture to load, and after loading, an alert will ask you if you want to use the picture files own colour palette or not. If you say no then the frame is recoloured with the current palette.

The picture stays in memory to allow further frames to be captured until another picture is loaded in.

NOTE — Degas files must be Low Resolution (PI1 files), and must have been saved in UNCOMPRESSED mode.

A LOAD (ii) NEO

This is the same as LOAD DEGAS as far as using is concerned, but allows the loading of picture files in other most popular picture file standard — NEOCHROME.

Once again, the picture remains until another is loaded.

Note

Both LOAD DEGAS and LOAD NEO only LOAD the

picture file into a screen in memory. They are accessed by option D (GRAB FRAME).

Following a LOAD, the text in the GRAB FRAME box will turn white to say that the feature is now available.

B LOAD (i) FRAMES

This is where you load your previously saved frames.



AB ANIMATOR will save frames in either GFA BASIC or FAST BASIC format and both are completely different formats, so you must first select which format you wish to load. GFA frames use the extender .GFR (Gfa FRames) and FAST BASIC frames .FFR (Fast basic FRames). The frame style selector is shown left.

Complete the operation by clicking on OK.

When loading frames, the colour palette for the frames are loaded in at the same time along with information about the frames animation. Also note that with a frame load you do NOT have the option to cancel the loading of the palette, and any existing files (with the exception of the TEMPORARY storage frame) are lost.

B LOAD (ii) PALETTE

This feature will load a colour palette, and recolour the existing frames with the new colours.

When you click on this box, AB ANIMATOR will automatically open the folder PALETTES on the disk and present you with a directory of the paletter files

inside it. If you create your own data disks, see the note on data disks on page 6.

All palette files have the extender .PAL and there are a superb range already on the AB ANIMATOR disk. You can also load your own palettes saved previously with SAVE PALETTE.

If you are creating frames for use with GFA BASIC, there is however another useful way to load a colour palette.

Assuming you want to create a new set of animations for the same program as a previously created set, you can load the colour straight out of the previously stored frame (with the extender .GFR). This method ignores all frame and animation data, and just loads the colours.

Whichever way you do it, each method displays the current palette at the top of the screen, loads the new palette, and changes the palette as you watch. This way you can see straight away if the palette loaded is the one you want without having to go to the screen displaying the colour bars.

FAST BASIC format automatically saves the frame colour information as a separate file on the disk, and so naturally the colours cannot be loaded from a FAST BASIC frame (.FFR) file.

C SAVE (i) FRAMES

When you have created your frames, this option allows you to save them to disk for future use either in your own programs or if incomplete, to finish off in another session.

AB ANIMATOR needs to know first, which BASIC format you wish to save your frames in, so click on GFA or FAST BASIC, and then on OK. (See frame selector in previous section).

When you save your frames, AB ANIMATOR also saves the current colour palette, and details of the animation you have set up such as start and end frames, speed, etc.

GFA colour and animation data is saved in the same file as the frame data and has the extender .GFR.

FAST BASIC frames are saved with the extender .FFR and the animation data is saved along with the colours in a separate file with the extender .COL which can be merged with your BASIC program in one of the FAST BASIC modules.

The number of frames saved to the disk is dependant on the numbers set on start and end frames from the ANIMATE SPRITE section.

In the same way that the PRINT function works on EDITING SCREEN 1, the SAVE FRAMES feature is unusual, but has its advantages. For instance, if you only create 7 out of the 20 frames, you only want to save the 7 frames NOT the 13 empty ones. To do this simply go to ANIMATE SPRITE and set the start and end frame numbers. In practice, however, you will most likely find you have done this already to see your animation, and so no problems should arise. Another advantage of this method is that you could, if you wish, make a set of 20 frames containing 4 five-frame sequences into 4 separate files on disk, each with only the 5 frame animation.

You are given a reminder when you save frames if you forget to set them.

When saving frames, do not use an extender as it is done for you.

C SAVE (ii) PALETTE

If you create a useful colour palette, with this feature

you can save it for future use. That way, when you design your frames from scratch, you can load the palette of your choice into AB ANIMATOR before you start.

When you save a palette you need only type the palette's name. Do NOT type an extender, as AB ANIMATOR automatically puts the extender .PAL on the end for you.

D GRAB FRAME

This feature is used in conjunction with A -LOAD DEGAS/NEO, and if a picture file has not been loaded, this function will not operate.

If the words 'GRAB FRAME' are in grey, then this tells you no picture file has been loaded. As soon as you load a Degas or Neochrome picture in, the text changes to the same colour as the rest of the icons.

When you click on this item, you are asked to confirm your intentions, and if you continue, you will see the picture you have just loaded in.

Also on the screen is a frame the same size as your animation frames which can be moved over the picture with the mouse.

Position the frame on the section you want to capture, and then press the left mouse button. The part of the picture inside the frame will then be placed inside frame 1 of your 20 frames.

The picture will be intact, and you can now go to the edit frame section and copy frame 1 into another frame and if necessary go back and 'grab' another frame. You only lose the picture file if you switch off your ST or load another Degas or Neo screen.

E EXIT

This takes you back to the MAIN COMMAND SCREEN.

To summarise the FILES screen, you can load picture files from Degas and Neochrome and grab frames from them. Also you can load and save frames and colour palettes from the disk in this section.

It might be worth mentioning here that some of you may not own the two art programs compatible with AB ANIMATOR. If not, the program you do own might well be the one that saves pictures in Degas or Neochrome format, so try to load a low resolution file first.

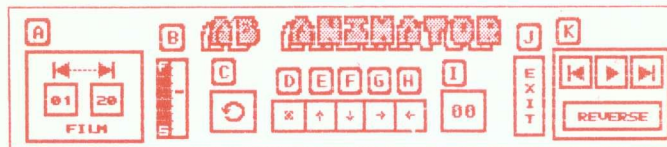
For this reason, no check is made to see if the picture file being loaded is actually a P11 or NEO file, so in fact you could load anything - possibly with fatal results, so it is advisable to try this when you first load AB ANIMATOR in, with no frames in memory, because the result may possibly be all colours being black and you will have to switch off and start again.

The only screen we have not looked at up to now is the ANIMATE SPRITE screen, and as it is not possible to see what it does exactly without frames to animate we will look at the controls now and then load a set of frames to see them in action.

ANIMATE SPRITE

From the FILES menu, click on EXIT and then when on the MAIN COMMAND screen click on ANIMATE SPRITE.

The screen should clear and you will see a control panel at the bottom of this screen like the one here:



Following the usual letter labels, the functions are:

- | | | |
|---|----------------------|---|
| A | FILM START/END FRAME | — Set first and last frame of the animation |
| B | ANIMATION SPEED | — Rate at which frames are displayed |
| C | CONTINUOUS LOOP MODE | — Animate film continuously. |
| D | DIRECTION CANCEL | — Cancels modes E, F, G and H |
| E | DIRECTION UP | — Upwards movement |
| F | DIRECTION DOWN | — Downwards movement |
| G | DIRECTION RIGHT | — Right movement |
| H | DIRECTION LEFT | — Left movement |
| I | DIRECTION RATE | — Direction of movement speed |
| J | EXIT | — Exit the animation screen |
| K | ROLL FILM CONTROLS | — See frames in action |

As you cannot see the controls in action properly until you have loaded some frames in, follow the following instructions, and you will soon be back here again with a set of frames to experiment with. Just before you go, note the settings of the control panel.

1. Exit this screen to the main menu by clicking on EXIT.
2. Click on FILES.
3. Now click on LOAD FRAMES.
4. Select GFA format and then on OK.
5. Select the frame file called WALK.GFR by clicking on the filename and then on OK.
6. When loaded, EXIT the FILES screen.
7. Select the ANIMATE SPRITE section.

Ah good, back to where we were before. Pretty easy huh? Now that's done let's go through the controls and see what they do.

Before we start, did you notice that the control panel has changed slightly since you loaded the frames in? Before, in the bottom left corner were the numbers 01 and 20. Now it reads 01 and 11. Also, the little line on the indicator that looks like a petrol gauge has moved. This is because when you LOAD frames, you load the settings for animation as they were at the time of saving as well as the frames.

There is also another loaded piece of information but we'll see that in a bit.

If you want a quick look at the frame animated now, go to the right hand side of the control panel and find the small buttons that look like a video cassette players controls, and click on the one with a triangle pointing to the right with the left mouse button. The animation will play through one single sequence, and then return control back to you.

Now you have seen a sequence, we'll continue with the control panel features in more detail.

A FILM CONTROL PANEL

This panel contains two numbers in boxes. The left number is the film START FRAME and is set to the number of the frame you want to be displayed at the beginning of your animated sequence.

The right number is the film END FRAME number and is set to the frame that you want to be the last frame in your animation sequence. When the film is run, all frames between these two numbers (inclusive) are displayed. To alter the values, simply point at the number you want to change and press the correct mouse button.

* LEFT button increases the value in the box being pointed at.

* RIGHT button decreases the value in the box being pointed at.

Please note that you are prevented from setting the start frame larger than the end frame, or the start and end frame numbers at the same value. Therefore, the least number of frames you can save or print is 2.

A secondary use is that the start frame can be used as a 'single step' facility by the use of the left mouse button, as the start frame is always displayed on the screen.

B ANIMATION SPEED

This bit, mentioned earlier that looks like a petrol gauge, is in fact the part of the control panel which you can use to control the speed of animation. The speed of animation is the length of time a frame stays on the screen before being replaced by the next one. It alters a delay in frame updating, and the small line gives you

a visual representation of this delay. The nearer the line is to 'F' at the top of the box, the faster the speed.

C CONTINUOUS LOOP

If you ran the frames in the previous section, you will have seen that the 'film' runs, then returns control to you after the last frame. You may not want to keep clicking on the PLAY button, but want it run continuously, so that is what this button does.

When this feature is operating, the icon is highlighted.

Once started with 'LOOP' on, you stop the film by pressing the right mouse button.

The icon acts as a 'toggle' and turns off when you click on it a second time.

D DIRECTION CANCEL

This cancels the use of items E, F, G and H. It also sets the value of 1 back to zero. When this icon is lit, the animated frame remains stationery on the screen.

E MOVE UP

Click on this icon, and the frame is set to move in an upwards direction on the screen.

F MOVE DOWN

Click on this icon, and the frame is set to move in a downwards direction on the screen.

G MOVE RIGHT

Click on this icon, and the frame is set to move across the screen from left to right.

H MOVE LEFT

Click on this icon, and the frame is set to move across

the screen from right to left.

I DIRECTION RATE

With the exception of having just loaded frames from disk, when the following operation will have been done automatically, (the other piece of animation data mentioned earlier) you will have to set up the direction rate after setting items E to H. For example, click on function H (move right). As you have already loaded WALK.GFR the number in the direction rate box will alter to the value loaded in with the frames.

This figure represents the number of pixels in the chosen direction the animation will move PER FRAME. If no frames are loaded, this value must be set up before running the frame, or no movement will take place.

The value can be altered simply by pointing at the value and pressing the relevant mouse button.

The LEFT button increases the value.

The RIGHT button decreases the value.

The maximum rate in any direction is 10 pixels per frame, and the value is set back to zero when you use item D (direction cancel).

When the frames are moving, should they go off the screen they will 'wrap around' and enter on the opposite side. Also, as you will by now have noticed, the control panel scrolls down off the screen to give you the maximum area for your animations to move in.

J EXIT

Click on here to exit back to the MAIN COMMAND SCREEN. Your animation details will be left as set, ready for saving with the frame details, and the continuous loop icon stays in force. The direction indicators are switched off.

K ROLL FILM CONTROLS

This is where you get the frames going! You will already know that the controls are like those from a video player and which one is the play (the triangle).

To the left of the play is the rewind-to-start-frame control, and to the right of the play is the forward-to-last-frame control. These are used if you stop an animation half way through and you want to go to the first or last frame.

The final icon is REVERSE. This, as it suggests, runs your film backwards. It is a toggle, and is a case of click on — click off.

An amusing setting for the walking man frames is moving left 1 pixel per frame playing the animation forward fairly slow.

Not exactly Michael Jackson, but Hey Presto — Moonwalking!

Play with the controls as much as you want to get used to them.

If you like, you can go and get another set of frames to animate. Don't forget that you are already loading the animation data, so when you get back, all is set up for you, just click on PLAY!

The most you will have to do is set on continuous loop mode and if required the direction and direction speed.

In the next section we will go through the whole package creating your first movie using the facilities in AB ANIMATOR.

LET'S MAKE A MOVIE!

If you have been following the book, you may find it easier at this point to switch off your ST and re-load AB ANIMATOR.

Done it? . . . Good — you should now be looking at the main command screen, so go and click on FILES.

Next click on LOAD FRAMES, and when the file selector and directory appear click on BITS.GFR to highlight it and then on OK. After a few seconds, the mouse pointer re-appears and the highlighted LOAD FRAMES box reverts back to normal. When this happens, EXIT back to the main screen and click on EDIT FRAMES.

You will see on this screen 20 frames from which to choose a single frame that you can make a 'movie' from. All you need to do is choose one, although if you want you could create your own. Either way it's up to you.

If you want to use one of those on the screen now, go up and click on TEMPORARY followed by the frame you want to use. That frame number should be in the TEMPORARY box. Next go and CLEAR all frames.

When they are gone pop back up to TEMPORARY and click on it. It should now start flashing. Next click anywhere inside frame number one, and your chosen frame will re-appear.

Next, COPY frame one into frame two using OVER, and click on frame two. The screen will then change to editing screen 2.

Your first job, if you feel it necessary, is to set the colours you want. Then, using the draw and fill functions, slightly alter the enlarged frame in the same

way as a cartoon drawing would be drawn slightly different from the last one.

When you are happy with it, go to EXIT. Naturally, the more you carry out this operation, the better you will get at it — practice makes perfect as the saying goes!

When back to Frame Editing Screen 1, repeat the COPY process to transfer frame two into frame three, and then click on frame three to slightly alter the latest one.

Carry on until you have created enough frames to produce sufficient movement when you animate the frames.

All that is needed in the creation side of AB ANIMATOR is to go to ANIMATE SPRITE and test them out. You will probably need to make several visits back and forth before the sequence is how you like it.

The last operation is to SAVE your frames to disk in the FILES section.

You can quite easily create some good animated sequences with very little artistic talent by just using your imagination and making frames up as you go along. It does, however, make things a lot easier if you work to a story board. This gives you a pattern to work to, plus you have an end sequence to aim for.

You do not HAVE to be an artist to be able to create animations. The frames I have created on the disk prove that! (The only art I have ever done was at school over 12 years ago!)

USING THE FRAMES IN BASIC

Running the frames in AB ANIMATOR is all very well, but limiting. What you probably bought it for is to incorporate the frames in your own BASIC programs.

There are basically three ways to use the frames from AB ANIMATOR in your BASIC programs:

1. A static animated sequence on plain background.
2. A moving animated sequence on plain background.
3. Either of the above two on a multi-coloured background.

Examples of these can be seen in demos 1, 2 and 3 in the GFA and FAST BASIC folders on the AB ANIMATOR data disk. (They are explained in detail later).

When you design your frames, you will need to give a little thought to the screen you intend to animate them on. You may have in mind a screen which has a plain area for the animation to move over (coloured or black). If coloured, you simply fill the frame around the object with the same colour as your background screen. For example if your screen is red, fill the frame around your object red as well.

The method designed for placing the frames on the screen allows you to use the full frame size, so no worries about leaving borders around your objects to blank them out!

If you want to move frames over a scene, a little more thought is needed.

Your first job, before using AB ANIMATOR is to create your background picture. The easiest method

is to use either Degas or Neochrome, or alternatively, any package which can store its' pictures in UNCOMPRESSED P11 or NEO format will do.

You then load your picture into AB ANIMATOR and use the same colours to draw your frames. DO NOT ALTER ANY OF THE COLOURS during the frame creation, as this will alter the look of your picture when loaded into your basic program.

MASKS

The following instructions assume the user has created a picture with colour 0 (background colour) set to black. If your background is a different colour read your colour instead of the word black.

When your frames are completed, AND SAVED, you need to do two more things ready for your BASIC program.

The first is to create a second set of mask frames identical in shape to the first set, but FILLING all parts of your OBJECT with a single colour. White on black, if in your palette, makes the best masks although any colour on black will do. You would be advised to experiment.

Also, you must remember to 'colour out' any sections of your OBJECT you want to remain black (eyes, etc.) If left black, the result will be a transparent section, and the background screen will show through.

This set of frames should be saved with a suitable name identifying it as the mask file.

To see what a mask should look like, load FLYMASK.GFR into AB ANIMATOR and take a look at them by going to the FILES section, choosing LOAD FRAMES (GFA format) and typing FLYMASK.GFR and then clicking on OK.

CONVERT.PRg

The second and final task is to convert your picture ready to load into BASIC. You will find on DISK A a program called CONVERT.PRg which simply and effectively does this for you. The resulting 32k file contains no colour information, so that is why the colour data for your BASIC program is always extracted from your frame files.

CONVERT.PRg is only for use to AB ANIMATOR users to create background screens, but you may like to know that it is only one small part of another useful AB SOFTWARE title you might like to check out called PICSTRIP, that allows the creation of loading screens, colours, alerts and icons for BASIC programs from Degas and Neochrome picture files.

ANIMATING IN BASIC DEMONSTRATIONS

The method of animating in demonstration 1 is the simplest form, whereby each frame of animation purely 'overwrites' the previous frame. Where screen movement of the frames is not required, the result is of reasonable quality, although close inspection may reveal very slight flicker with darker colours.

Demonstration 2 is with a moving set of frames. All operations on the screen in demo 2 are carried out in OVERWRITE mode like in demonstration 1 because there is no screen underneath the frame to restore it. If, in your program, your frames leave a trail, you can always use the technique in the following demonstration – Demo 3.

Demonstration 3 is the most sophisticated program, as it contains page flipping, grabbing sections of the screen ready for replacing later, masks and logical operations for the placing of frames.

Page flipping creates two screens in memory which are pointed to by the variable names Tv1% and Tv2% (% denotes integer variables). This enables you to look at one screen, while putting your new frame into the other. The two screens are then 'flipped'.

Covering logical operators and truth table in detail is beyond the scope of this manual, but put as simply as possible, a frame sized rectangle of the background is captured where the frame is to be placed. The mask frame is then put onto the screen with XOR (compare screen and frame pixel/pixel and where both lit make black and where one lit leave 'as was'), resulting in a black 'hole' the same shape as your object. The frame is then put onto the screen with OR (compare screen and frame pixel/pixel and where only one lit leave 'as was'). As the frame moves away the captured rectangle of background is replaced, blanking out the frame.

The result is a very impressive and rock steady animation that at the time of writing we have not seen beaten in any BASIC animated program.

MOVEMENT

Demo 3 is controllable by the mouse, but in your programs you may control the movement with the variables X% and Y% if you wish.

MORE THAN ONE SET OF FRAMES?

There is no reason why you should not use many sets of frames in your BASIC programs, although you must bear in mind that the more frames there are to manipulate, the slower your program will be. If you are using GFA BASIC, the procedures can be compiled along with your program with no problems should you own the GFA COMPILER, which should speed things up a bit.

As you know, AB ANIMATOR is written to work only with GFA BASIC from Glentop and FAST BASIC from Computer Concepts. It will not work with standard ST BASIC because it is too slow, and it uses special commands found only in GFA BASIC and FAST BASIC to run the frames.

On the main disk, you will find BASIC programs which need GFA or FAST BASIC loaded in to use. These include BASIC demonstrations and files to merge into your programs which load AB ANIMATOR frames, loading screens and background picture files.

These will now be covered on the following pages and to make it easier for you, the information for GFA BASIC users and the information for FAST BASIC users has been kept in separate sections.

GFA BASIC

AB ANIMATOR and use with GFA BASIC.

Contents of folder GFA-BAS.

DEMO1.BAS DEMO2.BAS DEMO3.BAS
LOADFRME.LST LOADSCRN.LST GFABASRO.PRG

In the GFA folder, you will find the procedure LOADFRME.LST which is used for loading your frames. This procedure is merged onto the end of your programs

GFA BASIC is very friendly with screen data, as it allows you to store the data in strings. All you need to do is name the string after the frames. For example, you could load a set of dog frames into the subscripted variable Dogf\$(), i.e. 6 dog frames will be loaded into Dogf\$(1) to Dogf\$(6). However, before calling the

procedure you must tell GFA that you intend using subscripted variables by dimensioning them with:

```
Dim Dogf$(20)
```

You use '20' because that is the maximum allowed in AB ANIMATOR, and the procedure takes care of lesser numbers. You must dimension other variables for storing more sets of frames in the same way.

The procedure is called once for each set of frames to be loaded in, with the line:

```
@Loadframe ("A: /FRAMES/DOGS.GFR", *Frame-  
name$( ))
```

The "A: FRAMES/DOGS.GFR" loads DOGS.GFR off drive A: and from inside the folder FRAMES, while the *FrameName\$() part is the subscripted variable name the frames are to be loaded into. You are responsible for altering both parts to suit your own frames filename, variable name and folder situation for each set of frames.

The procedure uses local variable, which means you may use the same variables in your own programs with no ill effect, and the colour data from the frames is extracted in this procedure, being altered on screen accordingly with the Setcolor command.

Upon exit from the procedure, the following variables are retained:

Fe% — Frame End (last frame of animation sequence)
Dy% — Delay factor from ANIMATION section of AB ANIMATOR
Mf% — Move Frame rate (in pixels) from AB ANIMATOR

These variables may be used if required in your program. For example, your sequence may be set up with AB ANIMATOR to look the best animating at a medium

delay rate (frame speed) and moving across the screen at 3 pixels per frame. At this setting, the animation data taken from the frames leaves the variable Mf% containing 3 and Dy% containing a value to be used in a delay loop as follows:

```
For A% = 1 to 60*Dy%  
Next A%
```

THE LOADFRAME PROCEDURE

```
Procedure Loadframe(Framename$,P.sarr)  
  Local Bjc%,X%,A%  
  Dim Fm1$(20)  
  Open "I",#1,Framename$  
  Bjc%=Lof(#1)  
  Buffer$=Space$(Bjc%)  
  Bget #1,Varptr(Buffer$),Bjc%  
  Close #1  
  Pal1$=Mid$(Buffer$,Bjc%-47,48)  
  Dta$=Mid$(Buffer$,Bjc%-57,10)  
  Dy$=Mid$(Dta$,1,2)  
  Mf$=Mid$(Dta$,3,2)  
  Fe$=Mid$(Dta$,9,2)  
  Fe%=Val(Fe$)  
  Dy%=Val(Dy$)  
  Mf%=Val(Mf$)  
  For L%=0 To 15  
    Nd$=Mid$(Pal1$,L%*3+1,3)  
    Cv%=256*Val(Mid$(Nd$,1,1))+16*Val  
      (Mid$(Nd$,2,1))+Val(Mid$(Nd$,3))  
    Setcolor L%,Cv%  
  Next L%  
  Nd$=""  
  X%=1  
  For A%=1 To Fe%  
    Fm1$(A%)=Mid$(Buffer$,X%,1062)  
    Add X%,1062
```

```

Next A%
Swap *P.sarr,Fml$( )
Erase Fml$( )
Clr Dta$,Dy$,Mf$,Buffer$,Pal1$
Return

```

LOADING SCREENS

The LOADSCRN.LST procedure also loads into a string of variables, but this time is not subscripted and therefore does not need dimensioning. By changing the variable name, the only limit to the number of frames loaded is the free memory in your ST. This number is aided by the fact that the .PIC screens are reduced in size to 32000 bytes by the program on the data disk called CONVERT.PRG.

This reduction results in the stripping off of the colour data, so it is vitally important that you use the same colour palette as the DEGAS/NEO picture to draw your frame with.

If you want a proper loading screen for your program, load a set of frames before your screen to get the colour data, the SPUT the screen it onto the screen, deleting the screen variable (if not required again) and finally load the rest of the data for your program while the loading screen is in view.

Normally only one or two screens will be needed, so the following screen loading procedure has been deliberately kept small and simple enough to be repeated for each screen to be loaded. However, should many screens be needed, replace both occurrences of Screen\$ with Buffer\$, replace "A:PICNAME.PIC" with Picname\$ and call the procedure with:

```

Picname$="A:screenfilename"
Gosub Lodscren

```

On return, the procedure will have loaded off the disk, into Buffer\$, the screen saved as whatever name you put as 'screenfilename' (e.g. Picname\$="A:MYART.PIC"), so you merely have to transfer the contents to your own variable (variablename\$), empty Buffer\$ and go back for another screen with the lines:

```

Variablename$=Buffer$
Buffer$=""
Sput Variablename$

```

THE LOAD SCREEN PROCEDURE

```

Procedure Lodscren
Screen*=Space*(32000)
Open "I",+1,"A:PICNAME.PIC"
Bget +1,Varptr(Screen*),32000
Close +1

```

Return

The obvious advantage of GFA BASIC's friendly environment means you can load your graphics data into strings and then forget about them.

Animation is simply a case of using the PUT command inside a loop to display the contents of a subscripted string variable using the loop variable inside the subscript brackets to denote the frame number to display.

The Run only version of GFA BASIC has been included on the data disk so that non GFA users may at least run the demos and compare!

FAST BASIC

AB ANIMATOR and use with FAST BASIC

Contents of folder **FAST_BAS**

DEMO1.BSC DEMO2.BSC DEMO3.BSC
LOADCOLS.BSC MFDB.BSC SETFRAME.BSC

The basis of using frames designed with AB ANIMATOR in your own FAST BASIC program is:

First set aside (RESERVE) enough space for your screens.

You need one screen for the picture, and one screen for each of the frame files you want to animate as you cannot store screen data in variables.

If you want your frames moving over an existing background screen, you will need to use page-flipping and frame masks. To demonstrate this DEMO3.BSC is an example of all these techniques.

Suppose you want to animate a set of frames over a DEGAS or NEOCHROME (PI1 or NEO) screen.

First create you DEGAS or NEO screen and save it (Degas files — uncompressed PI1's).

Load it into AB ANIMATOR and use the colour palette to design your animation sequence and your masks. See the section on MASKS on page 40 for more details.

Next you need to turn your picture into a 32000 byte file using CONVERT.PRG on the data disk then you are ready to do you BASIC program.

YOUR BASIC PROGRAM:

```
RESERVE TV%, 32768:TV%=(TV%+256) AND  
$FFFF00
```

This reserves space for one screen in the array TV%, so if you need 3 screens copy this line 3-times using the variables TV%, TV1% and TV2% etc.

Next INSERT PROGRAM using the BASIC editor to load in the procedures supplied by AB SOFTWARE. (It is good practice to put your procedures at the end of your program, so you can do this last if you like).

LOADCOLS.BSC — Loadcolour PROC

MFDB.BSC — Set up MFDB PROC

SETFRAME.BSC — Set up frames PROC

Load the frames from ANIMATOR by:

```
BLOAD "FILENAME.FFR".TV1% (where TV is the  
screen number).
```

So if you have two sets of animation (1 set of frames and 1 set of frame masks) as in DEMO3, do the above plus:

```
BLOAD "MASK-FILENAME.FFR",TV2%
```

To set the colours from any file do the following:

```
PROCLoadcolour(filename) *please note no extender  
required!
```

```
PROSet-colours(A$)
```

If you require the data you used in the animator e.g. speed and start frame, just do PROCLoadcolour(filename).

On return the following values are available for you:

DY% — delay value you set up in ANIMATOR

MF% — number of pixels to move through (up, down, etc.)

FS% — first frame in sequence

FE% — last frame in sequence

Load in your converted picture file (.PIC extender) with:

```
BLOAD"FILENAME.PIC",TV%
```

(See description of page flipping for explanation of different screens).

Now 'flip' screen to look at picture with:

```
LOGBASE=TV%:PHYSBASE=TV%
```

Capture the picture into a buffer for later use to restore the background screen when the frames move away with GRAB 0, 0, 320, 200

Set up the (MFDB) screen blocks to be used in the BLIT command with the line:

```
PROCSetupMFDB
```

The MFDB array is the source block, and SCREEN array is the destination. (2 of each type in the procedure). If you need more, COPY the whole block and change the number in the array name.

Next you get the 'boxes' where the frames are stored with:

```
PROCSetupframes
```

Each of the animation sequences is stored on its' own screen as a set of 56 by 33 pixel boxes. This procedure tells you where those boxes are in the arrays FSX%(N) and FSY%(N) where N is the frame number.

FSX%(1) is the start X position of frame 1

FSY%(1) is the start Y position of frame 1

We can now go on to do the animation. This part depends on how you want to move your 'SPRITE' or frame.

In DEMO3 it is controlled by the mouse.

You have to move from the screen with the frames on it, one frame, putting it on to the screen with the picture on it while you are looking at another screen. You then 'flip' screens.

The idea is to work on screen 1 while looking at screen 2.

When you have put your animated sprite on the screen you then have to restore the screen before moving the sprite.

In DEMO3, we get the mouse X, Y positions and store them in X% and Y% starting the animation with frame 1, We also use a mask for each frame.

```
LOGBASE=TV%:PHYSBASE=OLDSCR%
```

— means that we are looking at physbase screen and working on logbase screen.

```
PUT 0, 0, 3
```

— puts the picture onto screen TV%

```
BLIT @MFDB2&(0), FSX%(F), FSY%(F), 56, 33,  
@SCREEN&(0), X%, Y%,4
```

Puts the mask onto the screen — Easy Eh?!!!

BLIT @MFDB2&(0) is the block transfer from TV2% (masks screen).

Position FSX%(F) is frame F top X position.

Position FSY%(F) is frame F top Y position.

56, 33 is size of frame in pixels. *Don't change these numbers.

@SCREEN& is the destination screen (oldscr%).

X%, Y% is the screen co-ordinates from the mouse positions.

,4 means AND the mask onto the screen.

Next we put the frame on top of the mask with XOR with the line:

```
BLIT @MFDB1&(0), FSX%(F), FSY%(F), 56, 33,
@SCREEN&(0), X% Y%, 6
```

(Similar to putting the mask on the screen but source screen is different and mode to BLIT is now 6).

Now we can look at what you just did!

```
PHYSBASE=TV%
```

Now we clear the old screen:

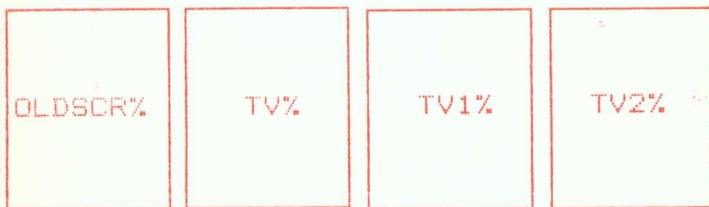
```
LOGBASE=OLDSCR%:PUT 0,0,3
```

and you do the animation again on the screen you can't see.

Remember — always work on screen 1 whilst looking at screen 2 and so on.

Looking at the boxes below as screens and the names we have called them may help to make page flipping easier to understand.

screen1 screen2 screen3 screen4



We have loaded frames into TV1%, masks into TV2%, picture into TV% and using TV% and OLDSCR% to flip between.

1 INTRODUCTION	1
2 SPECIAL NOTICE	3
3 LOADING	4
4 USING AB ANIMATOR	6
(i) Data Disks	6
(ii) Folders	6
5 MAIN COMMAND MENU	7
6 FRAME CREATION ORDER	8
7 FRAME EDITING SCREEN 1	9
COPYING FRAMES	10
(i) Over	10
(ii) Merge	11
(iii) Clr 1	11
(iv) All	11
(v) Temporary	11
(vi) Print	13
(vii) Insert	14
(viii) Delete	14
(ix) Exit	14
(x) Frame Number	15
(xi) Frames	15
8 FRAME EDITING SCREEN 2	16
(i) Clear	18
(ii) Exit	18
(iii) Control Panel	18
1. Flood Fill	19
2. Frame Shift	19
a. Horizontal Flip	19
b. Vertical Flip	19
c. Frame Shift	19
d. Exit	20
3. Colour Control	20
(iv) Grid	21
(v) Normal	22
(vi) Undo	22
(vii) Frame Number	22
(viii) Workframe	22

9 LET'S GET DRAWING	23
10 FILES	24
(i) Load Degas	25
(ii) Load Neochrome	25
(iii) Load Frames	26
(iv) Load Palette	26
(v) Save Frame	27
(vi) Save Palette	28
(vii) Grab Frames	29
(viii) Exit	30
11 ANIMATE SPRITE	30
(i) Film Control Panel	33
(ii) Animation Speed	33
(iii) Continuous Loop	34
(iv) Cancel Direction	34
(v) Up	34
(vi) Down	34
(vii) Left	34
(viii) Right	34
(ix) Direction Rate	35
(x) Exit	35
(xi) Roll Film	36
12 LET'S MAKE A MOVIE!	37
13 USING WITH BASIC	39
(i) Masks	40
(ii) The Convertor PRG	41
(iii) Animation in BASIC Demos	41
(iv) Movement	42
(v) Multiple Sets of Frames	42
14 GFA BASIC	43
(i) Load Frames Procedure	45
(ii) Loading Screens	46
(iii) Load Screen Procedure	47
15 FAST BASIC	48
16 INDEX	53
17 COPYRIGHT NOTICE	55
18 LIABILITY	56

COPYRIGHT NOTICE

This program is protected by copyright laws.

All copied software invariably comes from someones original disk. Please make sure that it is not from this one.

This practice is in fact illegal and is actually making software houses prefer not to produce software for the ATARI ST at all due to the lost sales incurred.

In the long run, the result of software piracy may be that there will be no new ST software available at all, or at the very least software houses like AB SOFTWARE will be forced to raise prices dramatically, so all copiers of software are simply cutting their own throats.

We at AB SOFTWARE hope you find AB ANIMATOR useful and easy to use.

As with most software, there is usually something that you want to do but can't, or something you have to do isn't really necessary.

We do not claim that AB ANIMATOR is perfect in any way, and we welcome suggestions.

If you feel that this software could be improved in any way, or if you have any comments about this software, please feel free to contact us at the address overleaf.

Happy animating!

LIABILITY

AB SOFTWARE are unable to accept responsibility for any subsequent use or mis-use of this software, the data created, or losses/damage incurred by the use of this product.

AB SOFTWARE
CREWE
CHESHIRE
ENGLAND

Distributed by

LADBROKE COMPUTING INTERNATIONAL
33 ORMSKIRK ROAD
PRESTON
LANCASHIRE

Telephone (0772) 203166

