AnimFlipper

Steve Tiffany

COLLABORATORS

	TITLE : AnimFlipper	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Steve Tiffany	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AnimFlipper

1.1 AnimFlipper Docs

```
AnimFlipper -- An example of a better way to string together
               Anims and IFFs in Amos Pro.  Freeware.
               ©1993, 1996 Steve Tiffany


TABLE OF CONTENTS

     What's It Do?

     About EXHIBITANIM[]
     About EXHIBIT[]
     About SCRNCENTER[]

     Limitations

     Author


AnimFlipper is heavily commented.  Read the comments!
```

1.2 AnimFlipper

```
WHAT'S IT DO?

AnimFlipper demonstrates a way to display anims that's easy to
use yet gives you a fair amount of control.  Use the program as a
starting point for your own presentations--Just Add  Anims !

The  EXHIBITANIM[]  procedure is the heart of the program, and it
was designed as a replacement for Amos Pro's 'Iff Anim' command.
On my system at least, the special screen that Iff Anim opens is
too low and too far to the right, and there's no way to relocate
it.  EXHIBITANIM[] uses screens that respond to Screen Display.
Where 'Iff Anim' inexplicably loops past the end and leaves you
looking at Frame 2, EXHIBITANIM[] correctly displays the anim's
```

last frame while the next anim is loading. Finally, EXHIBITANIM[]
gives you the option of displaying Frame 1 for a specific length
of time or until the user clicks the mouse.

The companion procedure EXHIBIT[] displays IFFs.

Even if you don't need to show anims, take a look at the
 SCRNCENTER[] procedure. It can prevent your programs from
looking off-center on other people's systems.

1.3 AnimFlipper

```
EXHIBITANIM["Work:Anim/MyFile.anim",0,0,1,1]
```

This is the procedure that does the important work of loading and
playing your anims.

Parameters for EXHIBITANIM[] are as follows:

 AN$........ The path and file name of the anim you want to play, in quotes.

 TRIGGER.... What to wait for before displaying first frame of this anim:
               If value is -1 then wait for mouse click.
               If value is 0 then display it as soon as it's loaded.
               If value is greater than 0 then Wait TRIGGER
                  (60 = 1 second in NTSC or 50 = 1 second in PAL).

 HOLDFRAME1 What to wait for with the first frame displayed before the
             rest of the anim starts to play:
               Same values as TRIGGER.

 MIDFRAME... How long to Wait between anim frames. Value of 0 plays
             fastest. Higher numbers play slower.

 NUMLOOPS... How many times the anim should play.
             Recommended value is 1. See Limitations .

1.4 AnimFlipper

```
EXHIBIT["Work:Picture/MyFile.iff",240]
```

AnimFlipper includes the EXHIBIT[] procedure for displaying IFFs
in the same environment that EXHIBITANIM[] uses.

Parameters for EXHIBIT[] are as follows:

 PIC$....... The path and file name of the IFF picture to show, in quotes.

 TRIGGER.... What to wait for before displaying this picture:
               If value is -1 then wait for mouse click.

```
                If value is 0 then display it as soon as it's loaded.
                If value is greater than 0 then Wait TRIGGER
                   (60 = 1 second in NTSC or 50 = 1 second in PAL).
                Here's more on the logic behind  TRIGGER .
```

Note that if you're writing a slide show with no anims, you're better off skipping EXHIBIT[] and the AnimFlipper environment entirely and just using 2 unbuffered screens and Screen To Front. AnimFlipper uses one unbuffered screen and one double buffered screen, which is a waste of chip ram if you aren't showing anims. (You want to use two unbuffered screens because slide shows which use a single double buffered screen may produce an ugly flash when pictures have different palettes.)

1.5 AnimFlipper

```
SCRNCENTER[SCRNWIDTH,SCRNHEIGHT]
               /          /
           320,352    200,240 NTSC (400,480 won't work w/OS 3.1)
           640,704    256,290 PAL
```

How often have you seen otherwise-enjoyable AMOS programs where the screen is way over on one side of the monitor? Now you can make sure your programs don't look that way on other people's systems. SCRNCENTER[] is a procedure that lets the user permanently center the screen for his or her particular machine. Here's how it works:

When your program calls SCRNCENTER[], the procedure looks for a tiny config file in the user's S: directory. If it finds it, the procedure opens it and reads two numbers, XOFFSET and YOFFSET, and then your program uses them to center the screen with the Screen Display command. If the file's not there, it presents the user with a screen they can center by moving the mouse. When they click the mouse, a requester suggests writing the values of XOFFSET and YOFFSET to the config file in the user's S: or Ram: directory. If the user picks S:, the program centers itself from then on. If Ram:, just til they reboot.

I'd like to propose a standard of sorts: Plug the SCRNCENTER[] procedure into your own programs without modification, and it will correctly select one of the following files to hold the values of XOFFSET and YOFFSET:

```
      AmosXyOffsetNTSC.config    if it's 320x200 or 640x200

      AmosXyOverscanNTSC.config  if it's 352x240 or 704x240

      AmosXyOffsetPAL.config     if it's 320x256 or 640x256

      AmosXyOverscanPAL.config   if it's 352x290 or 704x290
```

If everyone uses this version of SCRNCENTER[], the average user will only have to center his screens twice: once for standard

screens and once for overscan. From then on, all new programs
that use SCRNCENTER[] will center themselves by reading the
config file. Also, if we standardize to these four config file
names, people won't have a hundred different 11-byte files that
all do the same thing clogging up their S: directories.

If you use the config file to hold other information beyond
XOFFSET and YOFFSET, please pick a different name for the file.

1.6 AnimFlipper

LIMITATIONS

While EXHIBITANIM[] is a big improvement over 'Iff Anim,' it
still doesn't solve a couple of problems.

One is the hesitation at the loop point if you play an anim more
than once. I almost got rid of the NUMLOOPS parameter because of
this, but left it in for those anims that aren't hurt by a pause
at the loop (like when an object starts and ends off-screen).

Another limitation is that there's always a freeze-frame between
anims while it loads the next one. Try to design your programs
so the pause looks intentional.

EXHIBITANIM[] requires all the anims and IFFs to be the same
resolution and number of colors, and anims must be of the Anim-5
variety, the format used by DPaint.

As currently set up, you're limited to anims of 999 or fewer
frames, but you can increase that by changing the two 999's near
the top of the EXHIBITANIM[] procedure.

The most noticeable limitation of SCRNCENTER[] is actually a
deficiency of AMOS itself. AMOS screens adjust left/right in
16 pixel jumps, meaning users can't fine-tune the screen location
without adjusting a knob on their monitors. And if they do that,
it uncenters their Workbench screens.

SCRNCENTER[] assumes screens are standard sizes:
   320x200, 640x200, 352x240, 704x240 in NTSC
   320x256, 640x256, 352x290, 704x290 in PAL
It actually works with Laced screens as well (320x400, etc.), but
AMOS can't handle those once you upgrade to OS 3.1.

SCRNCENTER[] hasn't been tested with games that use hardware
coordinates directly.

1.7 AnimFlipper

Feel free to send comments by e-mail to stiffany@isd.net

If these procedures help you create something cool, let me know when you upload it to Aminet. If it's not going on Aminet, by all means copy it onto 880k floppies and send via snail mail, FedEx, bike messenger or mule train to:

Steve Tiffany
3255 14th Ave. S #4
Minneapolis, MN 55407
USA (addresses current June 1996)

1.8 AnimFlipper

TRIGGER

The 'Trigger' argument may be a little counter-intuitive. You see a Wait number right after an image file and it's natural to think that's how long you'll be displaying that file. But 'Trigger' tells the program how long to wait before displaying the called image or anim. So if the trigger is 5 seconds, that means you're looking at the previous image for 5 seconds, not the one you're calling.

I know it would have been more readable if the how-long-to-wait argument referred to the called image, but doing it like I have lets it pre-load the next file.

The advantage of this is apparent when the trigger is a mouse click: you click the mouse and the image appears instantly, versus you click the mouse, the file loads, and then the image appears.

1.9 AnimFlipper

ANIMS

AnimFlipper comes with 3 anims and 3 IFFs. These files need to be in the same directory as the program, and the program needs to be launched from that directory. Clicking its icon accomplishes this, as does running the program right after you load it into AMOS. If you move the files elsewhere, add the appropriate path information to the procedure calls.

For distributing your own programs, there's something to be said for omitting paths in the procedure calls, and just having all the anims and IFFs in the same directory as the compiled program. That way, no matter where the directory ends up -- Ram:, Work:, even immortalized on a CD-ROM -- the program can still find its anim and IFF files.