

AnimPoint

COLLABORATORS

	<i>TITLE :</i> AnimPoint		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 26, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 AnimPoint	1
1.1 AnimPoint Documentation	1
1.2 Introduction	1
1.3 System Requirements for AnimPoint	2
1.4 Copyright	2
1.5 Installation	3
1.6 AnimPoint User Interface	3
1.7 UI Top Section	4
1.8 UI Bottom Section	5
1.9 UI Buttons	5
1.10 UI Menus	5
1.11 UI Drag and Drop	6
1.12 Anim or Anim Brush File Requirements	6
1.13 Supported AnimBrush IFF chunks	7
1.14 Defining New Animated Pointers	7
1.15 Using OP-5 Anim files	7
1.16 Using AnimBrushes from DPaint	8
1.17 Submitting new pointers	8
1.18 File Naming Conventions	9
1.19 Suggested Text Description Format	9
1.20 Common Problems	10
1.21 The colors are not set correctly on my animbrush.	10
1.22 My 16 color pointers are displayed as only 4.	10
1.23 My pointer hotspot is not right.	11
1.24 AnimPoint doesn't work correctly on my display card.	11
1.25 AnimPoint fails to start.	11
1.26 AnimPoint crashes upon startup.	11
1.27 The pointer appears garbled.	12
1.28 The pointer won't move to the top of the screen.	12
1.29 AnimPoint bugs, limitations, etc.	12

1.30 Author Info	12
1.31 Credits	12
1.32 Change History	13
1.33 Notes on System Compliance of Animpoint	13
1.34 AnimPoint limitations	13
1.35 Known bugs in Animpoint	14
1.36 Warnings To Keep In Mind	14
1.37 Notes on CPU utilization	15
1.38 The AddText utility	15
1.39 The AddGrab utility	16
1.40 The AddColors utility	16
1.41 AnimPoint Preferences Files	18
1.42 Supported Tooltypes	18
1.43 Quitting AnimPoint	19
1.44 Future Enhancements:	19
1.45 ARexx Commands:	19
1.46 Index	20

Chapter 1

AnimPoint

1.1 AnimPoint Documentation

AnimPoint: An intuition pointer animator
for OS 3.0 or greater (>= V39 libs)

Copyright 1993 Steve Koren

Version 1.0

```
@{ " Introduction           " link introduction       }
@{ " Requirements          " link requirements      }
@{ " Copyright             " link copyright         }
@{ " Installation          " link installation      }
@{ " AnimPoint UI          " link user_interface   }
@{ " Defining new anim files " link anim_defs         }
@{ " Submitting new pointers " link submissions      }
@{ " ARexx Commands        " link arexx             }
@{ " Common problems       " link problems          }
@{ " Bugs, Limitations, etc. " link bugs_and_limits  }
@{ " Enhancements          " link enhancements      }
@{ " Author                 " link author            }
@{ " Credits                " link credits           }
@{ " History                " link history           }
@{ " Index                  " link index             }
```

1.2 Introduction

A long time ago, in a land far away, there was a program which would create animated intuition pointers. There were a few problems with this program:

- o It didn't work under newer versions of AmigaDos or on AGA systems.
 - o It was not free.
 - o It didn't accept standard IFF anim or animbrush files.
-

- o It had no UI.

AnimPoint is an attempt to remedy these things. AnimPoint tries to be system legal, although there are a few things which are on shaky ground. Also, by its very nature, AnimPoint is unlikely to work on 3rd party graphics cards.

AnimPoint has these features:

- o It is a commodity.
- o It has a graphical 2.04 style user interface for changing preferences settings. The AnimPoint window is an AppWindow for direct drag-and-drop. (Note: Drag-and-Drop).
- o It caches animation frames in FAST RAM, in order to take as little CHIP ram as possible. It takes 280 bytes of CHIP RAM total, no matter how many frames are in the animation.
- o It reads standard IFF anim or animbrush files, as written by DPaint 4.5, ImageFX, etc. The brush handle position is used as the pointer "hot spot". Some utilities are provided to add things to this file which DPaint (etc.) won't generate by itself, but these are optional.
- o It supports multi-palette IFF animations.
- o It attempts to coexist with programs which define window-specific pointers.
- o It is as system-legal as I could make it (although it is not 100% OS legal). (Note: @{"System Compliance" link system}).

1.3 System Requirements for AnimPoint

AnimPoint requires the following to run:

- o Any system with V39 libraries or greater. (These are mostly AGA systems. AnimPoint has not been tested on an ECS machine).
- o reqtools.library
- o A 68020 processor or better. There is currently no 68000 version, and this version will likely crash on a 60000 based system.

1.4 Copyright

AnimPoint is Copyright 1993 Steve Koren, All Rights Reserved

AnimPoint is copyrighted freeware. It may be distributed on PD disk collections costing no more per disk than the price of a current Fish disk. AnimPoint may be uploaded to BBS systems. It may not be sold for a profit or modified without permission of the author. However, you are free to give it away to friends, etc.

Fred Fish is explicitly permitted to distribute AnimPoint on his CD-ROM collection of software.

Further distribution of AnimPoint may NOT be restricted. This applies to services such as CompuServe which otherwise attempt to restriction distribution of software from their archives.

You may not claim authorship of AnimPoint or included pointers unless you are actually the author of the pointer.

AnimPoint may contain pointers copyrighted by others. The individual copyrights belong to the authors, but the pointers fall under the same distribution conditions as AnimPoint.

1.5 Installation

To install AnimPoint:

- o Unpack the .lha archive into a convenient spot on your hard drive using a command such as:

```
lha -a -m -x -F x AnimPoint1.0.lha
```

- o Assign "AnimPoint:" to the root directory where you have extracted AnimPoint. AnimPoint looks for certain things here. You can add a line such as the following to your s:user-startup file:

```
assign AnimPoint: sys:usr/appl/AnimPoint
```

- o If you do not have it, copy AnimPoint:libs/reqtools.library to your libs: directory.
- o If you wish to run AnimPoint each time you boot your system, Copy AnimPoint and AnimPoint.info to your sys:WBStartup directory. You can do this from workbench by dragging the AnimPoint icon, or from the CLI via:

```
copy AnimPoint AnimPoint.info sys:WBStartup
```

You might also wish to edit the icon to set CX_POPUP to "no".

1.6 AnimPoint User Interface

AnimPoint uses a 2.04 style GadTools user interface. The UI can be brought up by opening the Commodities Exchange window, selecting AnimPoint, and clicking on the "Show Interface" button. The UI will use the default system font and appear on the default public screen.

The AnimPoint window is divided into two parts. The top part lets you select information about the animation to be used, such as its filename and the desired frame rate. The bottom section displays information about this animation. There are also some buttons and menus available.

The window also supports direct drag-and-drop.

- UI Top Section
- UI Bottom Section
- UI Buttons
- UI Menus
- UI Drag-and-Drop
- AnimPoint Tooltypes

1.7 UI Top Section

- o AnimFile: A text entry field in which you can type the name of an animation file. AnimPoint will attempt to load the file immediately, and display an error message if it cannot.
 - o Pick: Brings up a file requester for animation file selection.
 - o Speed: A slider gadget with a range from 1 to 30. This is the frame rate in Frames/Sec to be used. Higher frame rates take more CPU time (Note: CPU Usage).
 - o Use CMap: If checked, AnimPoint will modify the sprite colors for the pointer sprite on new screens which are equal or less than 4 bitplanes. (It cannot modify deeper screens, since the sprite colors conflict with the screens colors. Even the intuition pointer suffers from this problem). However, it is possible to tell AnimPoint to ignore the screen depth (Note: @{ "Supported Tooltypes" link tooltypes }). AnimPoint will only change the palette on frames which have defined color maps, which may be as few as one per animation.
 - o Pong: If checked, AnimPoint will play the anim in a "pingpong" manner, going backwards to the beginning after it reaches the end. This doesn't work very well with animations which change their colormaps.
-

1.8 UI Bottom Section

- o # Frames: The total number of frames in this animation.
This number is limited (Note: @{"Limitations" link limitations }).
- o # Unique: The number of unique frames in the animation.
AnimPoint will detect frames reused later in the animation, and store them in a very small amount of memory.
- o Size: The approximate amount of FAST RAM used by the current animation. Does not include the fixed overhead of AnimPoint itself. CHIP RAM usage is insignificant and thus not presented.
- o # CMAPs The number of colormaps in this animation.
- o <text> The text scroll gadget shows the description, if any, from the animation file. This description must be present in an ANNO chunk. DPaint will not create this chunk, but a separate program included with AnimPoint will add the ANNO chunk (Note: @{"AddText" link addtext }).

1.9 UI Buttons

- o Save: Saves preferences files to envarc: and env:.. The next time AnimPoint starts, it will use the saved preferences file from env:.. If the system is rebooted, envarc: is copied to env: at startup time. (Note: @{"Preference Files" link preferences }).
- o Open: Load a new preferences file. Identical to the "Project/Open..." menu.
- o Hide: Makes the AnimPoint window go away. AnimPoint continues to run using the current settings.
- o Quit: Exits AnimPoint (Note: @{"Termination" link termination }).

1.10 UI Menus

- o Project
 - o Open... Opens a preferences file and loads indicated animation. (Note: @{"Preference Files" link preferences }).
 - o Save As... Save current preferences settings to the specified file. This will create a project icon with the same imagery as the AnimPoint program's icon. Double clicking on this icon will invoke AnimPoint with the indicated
-

- preferences file.
- o Save Saves current preferences settings to envarc:ANIMP.Defaults & env:AnimPoint.Defaults.
- o About... Displays "about" requester.
- o Help... Tries to run sys:Utilities/Multiview to display this help file.
- o Quit Exits AnimPoint (Note: @{"Termination" link termination}).
- o Edit
 - o Last Saved Loads the last preferences saved to envarc:
- o Options
 - o SmartScreen Toggles the SmartScreen option.
(Note: @{"Supported Tooltypes" link tooltypes}).
 - o IgnoreDepth Toggles the IgnoreDepth option.
(Note: @{"Supported Tooltypes" link tooltypes}).

1.11 UI Drag and Drop

The AnimPoint window is an "AppWindow". It supports direct drag-and-drop of the following file types:

- o AnimPoint preferences files
- o IFF anim brushes
- o IFF anim file

Icons can be dropped on the AnimPoint window from a workbench window (or other source). Only one icon at a time may be processed; multiple selection is not supported (and makes no logical sense for this program).

1.12 Anim or Anim Brush File Requirements

- o Animations are limited to 16 pixels in width.
 - o Animations are limited to 64 pixels in height
 - o Animations are limited to either 4 or 16 colors.
They must have exactly 2 or 4 bitplanes.
 - o Animations should have a "hotspot" defined by an GRAB chunk.
DPaint writes this according to the current brush handle position when the animbrush is saved. Animations without one
-

can have one added by the AddGrab utility (Note: @{"AddGrab" link addgrab }).

- o The animbrush must have an interleave factor of 1.
- o (Note: @{"Supported Animbrush IFF chunks" link chunks }).

For more information, see the ANIM.brush.doc section of the AmigaDos 2.0 Devices RKM (page 444).

1.13 Supported AnimBrush IFF chunks

ANHD	Animation header chunk: required per frame
GRAB	Hotspot chunk: optional but strongly recommended
CMAP	Color map chunk: optional. Up to one per frame.
DLTA	Delta chunk: required per frame.
BODY	IFF body chunk for 1st frame: required.
ANNO	Annotation chunk: optional (used for description).

Any other type of chunk may be present, but will be ignored.

1.14 Defining New Animated Pointers

Animated pointers can be created for AnimPoint by using DPaint 4.5 or any other program which can create IFF Anim brushes or OP-5 animation files in the format specified in the IFF standard. There are certain limitations for these files (Note: @{"Anim file requirements" link anim_requirements }).

(Note: @{"Suggested text description format" link text_description }).

```
@{ "Using AnimBrushes from DPaint" link def_animbrush }
@{ "Using OP-5 Anim files" link def_op5 }
@{ "Submitting new pointers" link submissions }
```

1.15 Using OP-5 Anim files

AnimPoint also supports standard OP-5 IFF anims, as long as they meet the size and depth requirements. However, these files do not typically have a hotspot "GRAB" chunk, and thus must have one added by the AddGrab utility (Note: @{"AddGrab" link addgrab }).

I highly recommend ImageFX for creating these files. It can take a set of input pictures, such as from a raytraced source, and scale them to, say, 16x20 pixels and crop them if needed at the same time that it creates the animation. It can also render 24 bit source files into 16 color animations with good results. I used this technique to make all of the 16 color animations that I have supplied with AnimPoint 1.0.

1.16 Using AnimBrushes from DPaint

The steps for creating an anim brush are as follows. These instructions are fairly DPaint specific, although you can probably use another program as well if it writes IFF animbrush files meeting the specification in the IFF registry.

(AnimBrushes are like regular OP5 anims, save for storing deltas from the previous frame, not two frames back. They also have a GRAB chunk indicating the "handle" position). Be aware that the first color of the brush will always be transparent when displayed by the Amiga sprite hardware.

1. Using DPaint, create an animated brush. This brush can be at most 16 pixels wide, and at most 63 pixels tall. It should be either two or four bitplanes, but be aware that there are some hardware problems when displaying 16 color (4 plane) sprites. (Note: 4 bitplane sprite problems).
2. After creating the AnimBrush, use the "Brush/Handle/Place" menu in DPaint, set the brush handle to the desired "hotspot" for the new pointer.
3. Save the animbrush file to disk.
4. If desired, add annotation text (Note: `@{ "AddText" link addtext }`).
5. If desired, add new color maps (Note: `@{ "AddColors" link addcolors }`). This is necessary if you wish to use color changes within your animbrush.
6. Load the brush into AnimPoint. If so desired, you can save a preferences file to encode the location of the brushmap and speed.

1.17 Submitting new pointers

There are a few animated pointers included with AnimPoint 1.0. However, I am soliciting submissions for others. If you wish to submit one, send it to me either by electronic mail or on disk by US mail. (Note: `@{ "Author Info" link author }`). (Email is preferred).

All submitted anim brushes must be freely distributable and will be included with future versions of AnimPoint. I also reserve the right to reject submissions for whatever reasons I see fit. You are free to retain a copyright on submitted pointers, but they will be distributed under the same distribution conditions as AnimPoint itself. (Note: `@{ "Copyright" link copyright }`).

Submitted animbrushes should also include a preferences file setting a good default playback speed and "pingpong" setting. Both the animbrush and the preferences file should follow the

normal filenaming conventions (Note: @{"File Names" link filenames }). The anim file name in the preferences file should fall under "AnimPoint:Anims", and under the proper subdirectory in that location.

The pointer should also contain a proper text description (Note: @{"Suggested Text Description Format" link text_description }).

1.18 File Naming Conventions

The following are file naming conventions for AnimPoint:

Anim brushes: Should use a "ANIMB." prefix. For example, "ANIMB.MyAnim".

Preferences: Should use a "ANIMP." prefix. For example, "ANIMP.MyAnimation". They should have the same name as the animbrush file.

Filenames should not contain spaces; it is unfriendly for CLI users. Replace spaces with underscores in filenames.

Anim files should live under either:

```
AnimPoint:Anims/4_Color  
or  
AnimPoint:Anims/16_Color
```

If a future version of AnimPoint supports AGA hires sprites (as I hope to make it do), I will create new "4_H_Color" and "16_H_Color" directories.

Preferences files (except the saved env: preferences) should live in: AnimPoint:Projects. This is where AnimPoint looks by default.

1.19 Suggested Text Description Format

I suggest a common format for included description text strings (Note: @{"AddText" link addtext }).

These strings are limited to 28 characters in width. The following should have no leading blank lines, but have a blank separator line between the initial fields and the text description lines:

```
Name : <name of animation>  
Author: <author's name>  
Date  : <creation date>  
Point : <short description of point, such as "center", etc>
```

```
<the rest of your description goes here, in lines of at
```

most 28 characters each>

For example:

```
Name : CoolPointer
Author: Joe Cool
Date  : 01-Jan-94
Point : upper left (arrow)
```

This is a really cool
pointer animation.

1.20 Common Problems

Problems:

```
@{ " The colors are not set correctly on my animbrush.      " link prob_colors }
@{ " My 16 color pointers are displayed as only 4.          " link prob_16color }
@{ " My pointer hotspot is not right.                       " link prob_hotspot }
@{ " AnimPoint doesn't work correctly on my display card.  " link prob_card }
@{ " AnimPoint fails to start.                              " link prob_start }
@{ " AnimPoint crashes upon startup.                       " link prob_crash }
@{ " The pointer is garbled.                                " link prob_hires }
@{ " The pointer won't move to the top of the screen.      " link prob_top }
```

1.21 The colors are not set correctly on my animbrush.

AnimPoint will not set colors when the cursor is over a screen containing 5 or more bitplanes, because the sprite colors would conflict with the colors of that screen. For example, if your workbench screen is 5 planes deep, the colors will be displayed incorrectly. This is a limitation of the Amiga hardware. However, you can tell AnimPoint to either ignore the screen depth and set the colors anyway, or to restore the default system pointer in those cases. (Note: @{ "Supported Tooltypes" link tooltypes }) ←

1.22 My 16 color pointers are displayed as only 4.

This is a problem with the Amiga video hardware. Due to bandwidth limitations, under certain conditions only a few sprites (as few as one) can be displayed. 16 color sprites are formed by attaching two 4 color sprites. If there is not enough video bandwidth available to display both, then only the first four colors will show up. There are several things you can do to fix this problem:

- o Reduce your overscan setting.

- o Slide your screen as far as possible to the right.
- o reduce the # of planes in your screen.
- o pick a monitor mode with a slower scan rate.

You may have to do several or even all of these things before your sprite becomes fully visible in 16 colors. Since this is a hardware problem, there is nothing I can do about it within AnimPoint. Different Amigas seem to have different tolerances.

Using the SUPER72 mode in 800x600 (not overscanned), it should be possible to adjust the picture position enough to get two sprites displayed on a stock A4000 to make a 16 color animated pointer.

1.23 My pointer hotspot is not right.

There are two possible reasons. One is that the program you used to save the animbrush did not write an IFF "GRAB" chunk into the file. The second reason is that the GRAB spot is not correctly set. You can set it via "Brush/Handle/Place" in DPaint.

1.24 AnimPoint doesn't work correctly on my display card.

AnimPoint assumes Amiga hardware. Some things it does are not possible without the Amiga sprite engine. Thus, AnimPoint will probably not work correctly on display cards, even though it tries to use standard system interfaces.

AnimPoint has also not been tested on a non-AGA system.

1.25 AnimPoint fails to start.

AnimPoint requires V39 libraries, as used by AmigaDos 3.0 or later. It will not run on earlier versions of AmigaDos.

1.26 AnimPoint crashes upon startup.

AnimPoint requires a 68020 or greater CPU, and will crash on anything less.

1.27 The pointer appears garbled.

AnimPoint cannot currently cope with hires pointers. Your pointer preferences must be set to low res in Preferences. Future versions of AnimPoint may support hires pointers when AmigaDos 3.0 programming information becomes publicly available.

1.28 The pointer won't move to the top of the screen.

This appears to be a limitation in the current Amiga hardware or software (I'm not sure which). AnimPoint does not do any movement of your pointer; all that movement is performed by normal system functions. Somehow, the way this movement is being performed does not move the sprite image to the very top of the screen if the hotspot is below the top of the cursor. I'll fix it if I can find a way to.

1.29 AnimPoint bugs, limitations, etc.

```
How OS friendly is AnimPoint?      : @{ " System Compliance " link system }
Current limitations of AnimPoint    : @{ " Limitations           " link limitations }
Bugs and other unpleasantness      : @{ " Known Bugs           " link bugs }
Things you should know:            : @{ " Warnings             " link warnings }
```

1.30 Author Info

The author (Steve Koren) can be reached at:

Email: koren@fc.hp.com

US Mail: 405 Pulsar St.
Fort Collins, CO 80525 USA

I will generally respond to all email, but not to all US mail.
I do it read it all though.

1.31 Credits

Software used in writing AnimPoint:

- o GadToolsBox (Copyright Jan van den Baard)
- o ReqTools (Copyright Nico Francois)

Software used in creating 16 color animated pointers:

- o Imagine 2.0
-

- o Essence
- o ImageFX
- o DPaint

1.32 Change History

29 Nov 1992 - V0.5 - initial creation

23 Dec 1993 - V1.0 - first public release (it sat around for a year and a month :-))

1.33 Notes on System Compliance of Animpoint

AnimPoint is as system legal as I was able to make it. There are several things it does which are not perfectly OS friendly, but which nonetheless work on OS 3.0. These are:

- o There is apparently no entirely system-legal way to change the pointer "hotspot" for the system pointer (although you can change it for a window's pointer). At least, there is no way which is publicly documented right now. Thus, AnimPoint writes a new preferences file in env:sys/Pointer.prefs to have IPrefs set the pointer hotspot itself. There is some danger here, as the IPrefs daemon gets quite upset if an incorrect file is written or the file does not agree with the current sprite definition. Also, the Pointer.prefs file format has been known to change. AnimPoint writes 3.0 style preferences files, NOT any other format.
- o In order to update the pointer image, AnimPoint "knows" that the pointer is really sprite #0. It also knows how to attach two sprites to make a 16 color sprite. This is not likely to work on 3rd party display cards.
- o To change the pointer colors, AnimPoint sets color registers 17 through 19. These are the color registers used by the amiga hardware for the intuition pointer. They are not necessary used by other graphics cards. (It does, however, use system-legal methods of changing color registers and pointer data).

1.34 AnimPoint limitations

Limitations in V1.0 of AnimPoint:

- o Animations may be no more than 64 pixels tall.
 - o Animations may be no more than 16 pixels wide. This limitation may be removed in future versions when I can get programming information on how to use the AGA 64 pixel
-

wide sprites.

- o Animations may not contain more than 512 frames. This limitation is also removable; if anyone runs into it, tell me, and I'll fix it. I just didn't consider it worthwhile at the moment.
- o Animations may contain either 4 or 16 colors.

1.35 Known bugs in Animpoint

- o AnimPoint does not gracefully handle busy pointers. They tend to stomp on each other. This is a display anomaly only, but looks bad. I'll try to fix it.
- o AnimPoint does not handle hires pointers. There is no public programming information on this topic yet. For now, the pointer MUST be set to lo-res mode in the Pointer preferences program. I'll add AGA hires pointers as soon as I can get programming docs on the new 3.0 sprite functions.
- o Sometimes when the original pointer is restored (if AnimPoint exits or is disabled), the IPrefs program will crash. I'm not sure why this is.
- o Since AnimPoint uses writes a Pointer.prefs file to tell IPrefs that it should change the pointer hotspot, there is a timing dependency between AnimPoint and IPrefs. AnimPoint has no way to know when IPrefs has actually finished its operation. AnimPoint waits a reasonable time for IPrefs to do its thing. However, it is quite possible that IPrefs got interrupted and was not able to change the sprite in time. In this case, AnimPoint will start changing the pointer sprite while IPrefs is trying to, which can cause a system crash. I do not know of any way around this situation, but it is best to only start or stop AnimPoint on an unloaded machine or where all CPU bound processes are running with a low priority.
- o If you use a lot of color cycling, there can be some flickering when AnimPoint updates the pointer.

1.36 Warnings To Keep In Mind

- o AnimPoint modifies the env:sys/Pointer.prefs file. It will restore it when it terminates. However, if another program (such as the Pointer preferences program) copies a new Pointer.prefs file into env:sys, the IPrefs program will read it, get confused when the current sprite definition doesn't match, and crash the system. AnimPoint tries to avoid this situation by detecting the new prefs file and shutting itself down. This seems to work, but may not be perfectly reliable.
-

If anyone has a better idea, let me know.

- o AnimPoint only works with OS 3.0 format Pointer.prefs files. Anything else is likely to crash the system.
- o If you use a recoverable RAM disk, and your machine crashes while AnimPoint is running, the pointer hotspot information contained in that file will be incorrect after your machine reboots unless you explicitly copy `envarc:sys/Pointer.prefs` to `env:sys`. The standard system startup-sequence only does the copy if the `ENV: ram disk` version is not there already, which it will be for a recoverable RAM disk.
- o AnimPoint 1.0 only handles lores pointers. Use the Pointer Preferences program to set the pointer to low res before using Animpoint.

1.37 Notes on CPU utilization

Approximate CPU usage by AnimPoint on a 4000/040 at 25 MHz, for a 4 color 20 pixel tall pointer with only one palette:

Frame rate	CPU Utilization
10 FPS	< 0.10%
30 FPS	0.16%

Having a lot of palette changes in the animation slightly increases the CPU usage. So do 16 color pointers. However, anything you do should be insignificant on a 68040 system.

Certain other operations, such as loading new pointers, displaying the UI, etc, take more CPU time, of course.

1.38 The AddText utility

The easiest way to make a compatible anim brush file is with DPaint 4.5. However, DPaint will not create the ANNO IFF chunk which AnimPoint displays in its information window. To do this, a small utility program called "AddText" is included. AddText is used from the CLI as follows:

```
AddText <in_animbrush> <out_animbrush> <text_file>
```

Where:

- o `in_animbrush` is the source anim brush file, with no ANNO chunk present.
- o `out_animbrush` is the anim brush file to be created, with a new ANNO chunk.

- o `text_file` contains the text to appear in the ANNO chunk.

The output file will be readable by DPaint or any program supporting IFF ANIM brushes. However, note that many programs, including DPaint, will NOT maintain the ANNO chunk if the file is saved and reloaded.

The text file should be limited to 28 character lines. A longer line will be truncated by the AnimPoint window. Any number of lines may be used.

Behavior if an ANNO chunk already exists is undefined.

1.39 The AddGrab utility

If a normal IFF op-5 animation is saved from a program such as ImageFX, ADPro, etc, it will not have a brush "GRAB" chunk, and thus will not have a handle or "hotspot" position.

In order to add this, a utility is provided:

```
AddGrab <in_animbrush> <out_animbrush> <x> <y>
```

Where:

- o `in_animbrush` is the source anim brush file, with no GRAB chunk present.
- o `out_animbrush` is the anim brush file to be created, with a new GRAB chunk.
- o `x` is the x offset of the hotspot
- o `y` is the y offset of the hotspot

Behavior if a GRAB chunk already exists is undefined.

1.40 The AddColors utility

AnimPoint supports multiple palettes per anim brush. However, DPaint is not able to create this form of anim brush directly. (other programs such as ImageFX can directly save anim files with multiple palettes).

The AddColors utility lets you add palette (CMAP) chunks to an existing anim file. Any number of CMAP chunks can be added. The palette of the original animbrush controls the palette of the first frame, and any frames following until another CMAP chunk. For example, if you had a 25 frame animation, and instruct AddColors to set a new palette on frame 10 and frame 15, then there will be three sections. Frames 1 to 9 will share the same palette from the original animbrush. Frames 10

to 14 will have the first defined palette (from frame 10), and frames 15 to 25 will have the second defined palette (from frame 15).

AddColors is used from the CLI as follows:

```
AddColors <in_animbrush> <out_animbrush> <cmap_file>
```

Where:

- o in_animbrush is the source anim brush file, with only one CMAP chunk for the first frame (as written by DPaint).
- o out_animbrush is the anim brush file to be created, with from 0 to <framecount-1> extra CMAP chunks inserted.
- o cmap_file is a text file containing color map information.

The output file will be readable by DPaint or any program supporting IFF ANIM brushes. However, note that many programs, including DPaint, will NOT maintain the extra CMAP chunks if the file is saved and reloaded. DPaint will issue a warning if you re-load such a brush to the effect that it does not support multiple palettes.

The format of the cmap_file is as follows. Blank lines, and lines starting with a '#' character are skipped. Other lines are formatted thusly, with white space being ignored:

```
<frame#> ( r1 g1 b1 ) ( r2 g2 b2 ) ( r3 g3 b3 )
```

There are only three colors, since the first color of any sprite is always transparent as displayed by the Amiga hardware.

For example, if you wished to have a palette change on frames 10, 15, and 20, you might use the following file:

```
# Test file for AddColor utility

10 (50 50 50) (45 46 96) (53 255 255)
15 (32 99 150) (123 32 32) (75 75 75)
20 (98 23 220) (221 200 198) (199 200 255)
```

Each color can have a range of 0 to 255. You can potentially change the colormap on each and every frame of the animation (save the first, which always uses the palette as defined in the input animbrush file). It costs about 16 bytes of fast RAM for each color change in the final animbrush.

Note that AnimPoint 1.0 currently displays some amount of flickering in the brush if both colors change frequently. For this reason it is currently best to use color changes only where necessary.

Behavior if CMAP chunks already exist is undefined.

1.41 AnimPoint Preferences Files

AnimPoint can save preferences files which store the following state information:

- o AnimFile name
- o Anim Speed (FPS)
- o Use CMap setting
- o Pong setting

These preferences files can be saved and restored. When AnimPoint starts up, it attempts to read a preferences file from env:ANIMP.Defaults, if it exists, or from AnimPoint:Projects/ANIMP.Defaults, otherwise. (This can be overridden with icon tooltypes). (Note: @{"Tooltypes" link tooltypes }).

Preferences files are more convenient than raw IFF anim files, since they also encode other state information such as playback speed, and in the future, hires/lace settings for the pointer.

Preferences settings should be named with the "ANIMP." prefix. The file requester will use this as a pattern when loading or saving these files.

1.42 Supported Tooltypes

AnimPoint supports the following tooltypes from its .info file. They can also be given as command line arguments if Animpoint is started from the CLI.

- o PREFS=<prefsfile> Points to a valid AnimPoint preferences file to be loaded upon startup. This overrides the env:ANIMP.Defaults default.
- o CX_PRIORITY=<num> Commodity priority.
- o CX_POPUP=<yes|no> "yes" to present AnimPoint window upon startup, else "no".
- o CX_POPKEY=<key def> Hotkey for popping up AnimPoint's window. Defaults to "lcommand a".
- o TASKPRI=<num> Task priority for the AnimPoint task.
- o SMARTSCREEN=<yes|no> If set to "yes", AnimPoint will try to restore the default pointer when it moves to a 5 bitplane or deeper screen in which the proper sprite colors cannot be set without messing up the screen. This option takes a little more CPU time because AnimPoint must re-write the Pointer.prefs file in env:.. It has also been known to cause IPrefs to crash, and

for this reason the default is "no".

- o IGNOREDEPTH=<yes|no> If set to "yes", AnimPoint will change the color palette even on screens deeper than 4 planes. This may garble other things on that screen, which is why the default is "no". However, if your screens are that deep but don't use those colors (workbench may be an example), you can turn this option on. Note that if SMARTSCREEN is set, this option will have no effect.

1.43 Quitting AnimPoint

There are several ways to terminate AnimPoint:

- o Send a CTRL_C signal to the process.
- o Copy a new preferences file to env:sys/Pointer.prefs.
- o Use the Quit button or menu item from the window.
- o Use the "Remove" button of the Commodities Exchange window.
- o Invoke it a second time when one copy is already running.
- o Send a QUIT ARexx command.

1.44 Future Enhancements:

Possible future stuff:

- o Support of OS 3.0/AGA hires sprites. These can be up to 64 pixels across instead of the current 16, and should look quite nice.
- o A GUI for AddColors, AddText, and AddGrab.

1.45 ARexx Commands:

AnimPoint supports a limited set of ARexx commands. The ARexx port name is "ANIMPOINT".

LOADPREFS <prefs_file>

Loads a new preferences file into AnimPoint, and loads the anim file pointed to by that preferences file.

SAVEPREFS <prefs_file>

Saves the current preferences settings into a preferences

file.

LOADANIM <anim_file>

Loads the specified anim file, leaving the rest of the preferences settings as they were.

SETSPEED <speed>

Sets the animation speed to <speed>, which must be between 1 and 30 inclusive.

VERSION

Returns the current version of AnimPoint.

SHOWGUI

Displays the AnimPoint user interface.

HIDEGUI

Hides the AnimPoint user interface.

DISABLE

Stops the animation, but leaves AnimPoint running.

ENABLE

Continue from the disabled state.

GET <info>

Obtains requested information. <info> must be one of:

HEIGHT - height of current anim, in pixels.
 FRAMES - # of frames in current animation.
 SPEED - speed of current anim, in FPS.
 COLORS - either 4 or 16.
 LOADED - true if an animation is currently loaded.
 ANIMFILE - the file name of the current animation.
 USECMAP - current setting of the UseCmap checkbox.
 PONG - current setting of the Pong checkbox.

QUIT

Causes AnimPoint to exit.

1.46 Index

Index to all nodes in this file.

```
@{ "Anim or Anim Brush File Requirements" " link anim_requirements ←
}
@{ "AnimPoint Documentation" " link main }
@{ "AnimPoint Preferences Files" " link preferences }
@{ "AnimPoint User Interface" " link user_interface }
@{ "AnimPoint bugs, limitations, etc." " link bugs_and_limits }
@{ "AnimPoint crashes upon startup." " link prob_crash }
@{ "AnimPoint doesn't work correctly on my display card." " link prob_card }
@{ "AnimPoint fails to start." " link prob_start }
@{ "AnimPoint limitations" " link limitations }
@{ "ARexx Commands" " link arexx }
@{ "Author Info" " link author }
```

```
@{ "Change History" " link history }
@{ "Common Problems" " link problems }
@{ "Copyright" " link copyright }
@{ "Credits" " link credits }
@{ "Defining New Animated Pointers" " link anim_defs }
@{ "File Naming Conventions" " link filenames }
@{ "Future Enhancements" " link enhancements }
@{ "Index" " link index }
@{ "Installation" " link installation }
@{ "Introduction" " link introduction }
@{ "Known bugs in Animpoint" " link bugs }
@{ "My 16 color pointers are displayed as only 4." " link prob_16color }
@{ "My pointer hotspot is not right." " link prob_hotspot }
@{ "Notes on CPU utilization" " link cpu_usage }
@{ "Notes on System Compliance of Animpoint" " link system }
@{ "Quitting AnimPoint" " link termination }
@{ "Submitting new pointers" " link submissions }
@{ "Suggested Text Description Format" " link text_description }
}
@{ "Supported AnimBrush IFF chunks" " link chunks }
@{ "Supported Tooltypes" " link tooltypes }
@{ "System Requirements for AnimPoint" " link requirements }
@{ "The AddColors utility" " link addcolors }
@{ "The AddGrab utility" " link addgrab }
@{ "The AddText utility" " link addtext }
@{ "The colors are not set correctly on my animbrush." " link prob_colors }
@{ "The pointer appears garbled." " link prob_hires }
@{ "The pointer won't move to the top of the screen." " link prob_top }
@{ "UI Bottom Section" " link ui_bottom }
@{ "UI Buttons" " link ui_buttons }
@{ "UI Drag and Drop" " link ui_drag_drop }
@{ "UI Menus" " link ui_menus }
@{ "UI Top Section" " link ui_top }
@{ "Using AnimBrushes from DPaint" " link def_animbrush }
@{ "Using OP-5 Anim files" " link def_op5 }
@{ "Warnings To Keep In Mind" " link warnings }
```