MainActor

a modular animation package
Version 1.2

31 December 1993

by Markus Moenig

1 Warm Up

MainActor is shareware. The program may be freely distributed and copied as long as the following conditions are acknowledged:

- The sales price must not be higher than the cost of an (empty) disk plus a nominal copying fee plus costs for shipping. The total price must not be higher than 6 US$ or 10 DM.
- All parts of the program and the documentation must be complete. The distribution of single parts is not allowed.
- MainActor or parts of it must not be sold in combination with commercial software without the written permission of the author.
- Program and documentation must not be changed in any ways. The repacking of this distribution with different packers is however allowed.
- Permission is granted to include this archive in Public-Domain collections, especially in Fred Fishs Amiga Disk Library (including CD ROM versions of it).
- The author is not responsible for misuse, or damage caused by MainActor.

2 Installation

Just copy the MainActor drawer to a device of your choice and add an assign to your user-startup, like

```
assign MainActor: <Path>
```

The `Locale`, `Rexx` and `Docs` directories should stay in the `MainActor` directory.

The default configuration file will clone your Workbench, the positions of the windows are however saved according to my setup and may look strange on yours.

If you have an older version of MainActor installed, make sure to delete it before installing the new version (except perhaps the preferences files).

Note that MainActor v1.2 is not able to read the preferences files of older releases.

3 Introduction

MainActor is a full featured animation program. You can create / edit / time / play animations of any size on your native amiga chipset or your graphic card (if supported).

3.1 What is MainActor ?

MainActor is a modular animation package for the `Amiga`. There exist four different kinds of modules:

1) Animation Loaders
2) Animation Savers
3) Picture Loaders
4) Picture Savers

this can change in the future.

You can handle up to five projects; every project can have an animation or picture loader, as well as an animation or picture saver. This makes it possible to convert everything in any ways, the number of available modules can be easily upgraded.

3.2 Features

- Intelligent caching/tracking, MainActor can cache your animations and picture lists, if you have not enough memory it will directly access your data from any device. This makes it for example possible to create/edit/play a 400MB animation on your 2MB Amiga.
- If you run OS 3.0 or higher, you will get an extra speed bonus on animations through the use of the new graphics routines. For OS2.0/2.1 users there are specific routines in the modules, which will give them the best playback speed possible on their system (and my knowledge :)).
- Under OS 3.0 you will get the extra features of showing pictures or playing animations in a scrollable, resizable window on your MainActor or Workbench screen. The colors will be properly adjusted to your screen attributes.
- The modules have play routines of their own, so it will be possible to redirect the input/output of special modules to graphic cards or other hardware.
- The play routines support a timecode per frame option. A Sound Module (per frame) option will be there in one of the next releases.

- MainActor has a totally user reconfigureable GUI. The settings of your sessions can be saved, this includes the size/position of windows as well as the status of the projects.
- MainActor has an arexx port, nearly all functions can be accessed through it. You can for example scale or dither whole animations on the fly through the use of an image processor, scripts are included.
- MainActor lets you setup and save nearly everything concerning the display mode for playing an animation or showing a picture. For example you can set the X/Y offsets, the display mode ID of your screen (view) and so on.
- MainActor supports localization.

3.3 New features in v1.1

- You can now map your arexx scripts to the function keys.
- New IFF-Anim8_32, IFF-Anim8_16, PCX modules.
- The new Universal modules allow you to easily load all of the supported graphic formats. The Universal_Picasso-II modules remap the output of all modules directly to the Picasso-II graphics board from Village Tronic.
- MainActor has now an powerful external player named 'MainView', it can view all animations/pictures for which MainActor has the proper loader module. It is executeable from the wb or cli and supports, per tooltypes or cli arguments, all of MainActors view options.
- MainActor now (optionally) saves icons for your newly created animations or pictures, enabling you to view the animations/pictures by selecting their icons. These icons support the same tooltypes as MainView.
- You can now control how many times MainActor/MainView play animations.
- The new 'Flash' option allows you to profile your animations.

3.4 New features in v1.2

- MainActor/MainView now have a center option which allows highly accurate centration of your images or animations.
- New BorderBlank option.
- MainActor has a new window called Project Information Window. The new functions included in the window are:

  An information window which will give you detailed descriptions of your current animation or picture list.

You can now view the size and time profiles of your projects as a graphical representation.

You can compare the decompression times of your projects through new timer functions.

- New GIF/FLI/FLC/DL loader modules and FLI/FLC saver modules.
- Chunky pixel modules (GIF/FLI/FLC) will fly if loaded through universal modules which support graphic cards (like Universal_PicassoII etc.). You can of course also view them on your native chipset as normal.
- Introduction of a new (light) registration package. It does not have the handbook or the disk based updates but is cheaper.
- You can now use the save/append functions again, even if you have not registered MainActor. You will have to deal with requesters though.

4 The Menus

You are able to access most of MainActor's functions through the various menu items and their equivalent shortcuts. Some functions can only be accessed through the menus.

4.1 Project Menu

`Open New Project`

will open a new project window. You can open up to five project windows, see Section 5.1 [Project Window], page 9.

The attributes of the project window will be set, according to the defaults selectable in the Section 5.6 [MainActor Preferences Window], page 21.

`Project`

Consists of several submenu items, their functions are identical to the gadgets of the Section 5.2 [Project Control Panel Window], page 11.

`Open Prefs`

Opens an asl-requester which lets you choose a MainActor preferences file, saved with one of the two following menu items. When you have choosen a prefs file, the display AND all projects and their contents will be closed, MainActor will then be reopened with the settings of the preferences file.

You can also activate MainActor by simply clicking on the icons of the preferences files.

`Save Prefs As ...`

Saves your current setup of MainActor; nearly everything will be saved : the projects informations(All view settings, the actual project name, the size and position of all windows, the settings of the system preferences ...).

An self-executing icon will be created for the preferences file. You should save every setup you use frequently !

`Save Prefs As Defaults`

This will save your setup as the default preferences file, which will be automatically loaded when you start MainActor. The default preferences file is `MainActor:Prefs/Default_Prefs`. Replace the default by your most used setup.

`Iconify`

MainActor will be terminated and the MainActor icon will be placed on your workbench. If you want to reactivate MainActor, just click on the icon.

`About`

Gives you some informations of the current version of MainActor and its author.

`Quit`

Will close all projects and quit MainActor.

4.2 Text Buffer Menu

MainActor uses the Section 5.7 [Text Buffer Window], page 21 for presenting feedback to the user.

`Print Buffer`

Prints the informations stored in the Text Buffer Window.

`Save Buffer As ...`

Will let you choose a file into which the contents of the Text Buffer Window will be saved.

`Clear Buffer`

Will clear the Text Buffer Window and delete its buffer.

4.3 Miscellaneous Menu

The menu items of this menu will only be accessable when a project is loaded in the source project. The `Set Timecode` item will only be activated if an animation project is loaded in the source project.

`Select All`

Will select all items frames/pictures in your source project.

`Select Range`

A requester will pop up, just enter the start/stop frame for the range of items to be selected.

`DeSelect All`

This item will deselect all frames/pictures in your source project.

`DeSelect Range`

A requester will pop up, just enter the start/stop frame for the range of frames/pictures to be deselected.

`Project Info`

This menu item is identical to the `Project Info` function of the Section 5.4 [Project Information Window], page 16.

`Project Profile`

This menu item is identical to the `Project Profile` function of the Section 5.4 [Project Information Window], page 16.

`Set Timecode`

Enter the timecode in the requester which you want to be applied to all selected frames in the source project. Remember to hit return after you typed a new value into the integer gadget. The new timecode will be set immediately. All timecodes are entered in 1/60 of a second, MainActor calculates the right timecode for your vertical beam frequency itself.

4.4 Arexx Menu

See Chapter 7 [The Arexx Interface], page 28,for more informations of MainActors arexx capabilities.

`Install Arexx Scripts`

This menu item will open a window through which you can map your most used arexx scripts to the function keys. The cycle gadget in the upper part of the window will let you choose the function key on which you want to install an arexx script. The function keys are displayed as F1 ... F20, F10 ... F20 mean the function key `+` one of the shift keys. You can choose the function key by clicking on the gadget or (the faster method) by simply pressing the appropriate function key. (Remember that you have to press F1 together with the shift key to get the F11 key).

Below the cycle gadget are the `Select Script ...` gadget and a string gadget. The arexx script which is currently associated to the actual function key displayed in the cycle gadget is visible in the string gadget. You can enter a new arexx script or edit the name of the existing one through the string gadget. You can use the `Select Script ...` gadget to choose the arexx script via an asl requester.

The three gadgets in the lower part of the window are labeled `Save`, `Use` and `Cancel`.

`Save` saves your definitions of the function keys, they will be automatically reloaded the next time you start MainActor.

`Use` will keep the definitions, but will not save them

`Cancel` ignores the changes you made to the definitions.

`Start Arexx Script`

Will let you choose an arexx script, which will then be executed.

4.5 Windows Menu

This menu will let you open/close the different MainActor windows by simply selecting the desired menu item.

5 The Windows

The windows are MainActors main interface to the user. Their use is however optional, you can close all of them (except the Project Windows) and only use the menus. The choice is yours. The position and status of the windows will be saved in the preferences files.

The following is a detailed description of MainActors windows and their gadgets.

5.1 Project Window

A new Project Window will be opened by selecting the `Open New Project` menu item in the Section 4.1 [Project Menu], page 6.

You can open up to five Project Windows, the number of your Project Window will be displayed in the title bar, in front of the project name, like `1: HAM8_Animation`. If no project is loaded, the title is `Unused Project`.

The Project Windows are mainly controlled by the Section 5.2 [Project Control Panel Window], page 11.

`Project Mode Gadget`

Depending on your projects mode, it will be labeled *Source Project*, *Destination Project* or *DeActivated Project*.

The Source Project
: You can only have one source project. This is the project on which all actions from the Section 5.2 [Project Control Panel Window], page 11 will be executed. The gadgets of the Project Control Panel and the Project View Settings will be sett up according to the source project.

The Destination Project
: This is the project into which all selected data from the source project will be appended through the `Append` gadget of the Project View Settings Window.

  There can only be one Destination Project.

The DeActivated Projects
: These projects are waiting to be switched into the source or destination mode. There can be more than one DeActivated Project.

You can change the project mode by clicking on the `Project Mode Gadget`, for example if you click on the source projects `Project Mode Gadget`, it will be changed into the destination project and the former destination project will changed into the source project.

If you click on a DeActivated projects `Project Mode Gadget`, it will change into the source project.

If only one Project Window is open, you cannot change the `Project Mode Gadget`.

Everytime you switch on the source project, the gadgets in the Section 5.2 [Project Control Panel Window], page 11 and the Section 5.3 [Project View Settings Window], page 14 will be adjusted to the settings of the new source project.

`Caching`

This is a simple toggle gadget. If it is checked, it means that your project is cached or is going to be cached in ram.

If you have not enough ram, you will have to disable it. That means that for example animations will be directly played from your harddisk, which will be much slower than playing them directly from ram.

You can only toggle the status of the `Caching` gadget when the project window is unused. As soon as you load a project, it will become *ghosted*.

`Select Load Module`

When clicking on this Button Gadget, a requester will appear in which you can select the loader module for this project.

The cycling gadget in the requester will let you switch between *Picture Loaders* or *Animation Loaders*.

When hitting the `Load` gadget in the Section 5.2 [Project Control Panel Window], page 11, MainActor executes the animation/picture loader you have selected for the source project.

The name of the current loader module is visible in a text gadget directly under the `Select Load Module` gadget. Next to the name is a small recessed box with an `A` or `P` in it. They stand for `A`nimation or `P`icture loader.

The `Select Load Module` gadget will be ghosted as soon as you load a project.

`Select Save Module`

When clicking on this button gadget, a requester will appear in which you can select the saver module.

The cycling gadget in the requester will let you switch between *Picture Savers* or *Animation Savers*.

When hitting the `Save` gadget in the Section 5.2 [Project Control Panel Window], page 11, MainActor executes the animation/picture saver you have selected for the source project.

The name of the current saver module is visible in a text gadget directly under the `Select Save Module` gadget. Next to the name is a small recessed box with an `A` or `P` in it. They stand for `A`nimation or `P`icture saver.

The `Select Save Module` gadget will only be accessable when a project is loaded.

`The Scrolling Frames/Picture List`

The scroll list will let you choose the frames or pictures you want to execute a certain operation on.

If you have an animation loader for this project window, the scroll list will show you the numbered frames of the animation along with their timecodes.

If you have a picture Loader, the scroll list will show you the names of the pictures you have loaded.

Just (de)select the frame/picture by clicking on its text. For multiple selection do not release the mouse button and move over the items to be selected. The list will scroll if you come to the top or bottom border.

See Section 5.6 [MainActor Preferences Window], page 21, you can select how many items shall be displayed in the scroll list. You can also select a special font for it. (The default is Topaz 8 for the font and 8 for the number of items which shall be visible).

5.2 Project Control Panel Window

This window executes its functions on the current source Section 5.1 [Project Window], page 9.

`Load`

If you have an animation loader selected for the source project, an requester will appear in which you have to select an animation of the appropriate type.

If you have a picture loader selected, the first requester will ask you about the first picture, after selecting this picture the second qequester will ask you about the last picture. MainActor will now alphabetically sort the pictures and load them. For example, if you have a picture list:

```
Amiga_Pic.00001
Amiga_Pic.00002
.
.
Amiga_Pic.00155
```

and select Amiga_Pic.00055 in the first requester and Amiga_Pic.00100 in the next, all pictures from Amiga_Pic.00055 to Amiga_Pic.00100 will be loaded. The second requester is optional, that means you can abort it by clicking on `Cancel` if you just want to load one picture.

MainActor checks the icon (if there is one) of the project for the following tooltypes: *Window*, *FrameInfo*, *NoCaching*, *BBlank*, *XOffset*, *YOffset*,

Repeat. If you saved the project with MainActor these tooltypes will restore the settings of the Project View Settings Window.

The `Load` gadget is only available if you have not loaded a project into the source project yet.

`Unload`

Is the gadget directly under the `Load` gadget. It will unload the current project of the source project.

It is only available when you have loaded a project in the source project.

`Play`

Is next to the `Load` gadget. It is only available when you have loaded an animation project in the source project.

If you play your animation on a screen, there are two routines available. MainActor checks automatically if you run an OS Version lower than OS 3.0 and will play it the old fashioned way. If you have OS 3.0 or higher you will profit from the new double buffering routines.

The animation will be played according to the settings of the Section 5.3 [Project View Settings Window], page 14.

The following keys can be used to control the playing of the animation:

`ESCAPE OR A MOUSEBUTTON`
: will abort playing the animation

`SPACE`
: will stop playing the animation. While the animation is stopped you can use the *TAB* key to flip to the next frame.

`RETURN`
: will enable the timecodes you see in the scroll list of the source project and which you can change (These timecodes are default).

`THE CURSOR KEYS`
: scroll the display, the offset changes will be shown in the Project View Settings Windows x and y offset gadgets.

`N`

`M`
: If you have activated the `Frame Info` option, you can cycle the color of the info text using `N` and `M` up and down.

`THE FUNCTION KEYS`
: are used to set the frame rate, they override the defaults. You can reactivate the defaults by pressing *RETURN*.

  `F1` sets the timecode to 1/60 Second (60 fps)

  `F2` sets the timecode to 2/60 Second (30 fps)

  `F3` sets the timecode to 3/60 Second (20 fps)

  `F4` sets the timecode to 4/60 Second (15 fps)

  `F5` sets the timecode to 5/60 Second (12 fps)

F6	sets the timecode to 10/60 Second (6 fps)
F7	sets the timecode to 20/60 Second (3 fps)
F8	sets the timecode to 40/60 Second (1.5 fps)
F9	sets the timecode to 60/60 Second (1 fps)
F10	sets the timecode to 120/60 Second (0.333 fps)

Show

Is the gadget under Play. It will display all the frames/pictures you have selected in the source projects scroll list. This gadget is available for both animation and picture projects.

The frames/pictures will be shown according to the settings of the Section 5.3 [Project View Settings Window], page 14.

The following keys can be used to control the showing of the pictures:

ESCAPE OR A MOUSEBUTTON
: will abort showing the pictures.

SPACE
: will show the next pictures you have selected.

THE CURSOR KEYS
: scroll the display, the offset changes will be shown in the Project View Settings Windows x and y offset gadgets.

Save

If you have an animation saver activated in your source project, it will ask you about the name of the new animation. Then it will build up the new animation in the appropriate animation Format. It will only save the frames you have SELECTED in the source projects scroll list. If you just want to convert or resave the animation, you have to select all the frames in the scroll list.

If you have a picture saver activated in your source project, a requester will appear in which you have to select the directory and the base name of the pictures.

Example: You have activated an animation loader and a picture saver and loaded an animation. The animation is called Example_Anim. Now you select the first ten frames of the animation and click on the Save gadget. In the requester you type Ram: for the Directory and Example_Pics for the base name. The ten pictures will then be saved as

```
Ram:Example_Pics.00001
Ram:Example_Pics.00002
.
.
Ram:Example_Pics.00010
```

This is also valid if you have a picture loader and a picture saver.

If you have an animation saver, some more requester will appear and ask you for some specific attributes of the new animation, like `Palette per Picture` and `Looping Animation`.

The following combinations are possible:

1. Animation Loader and Animation Saver
2. Animation Loader and Picture Saver
3. Picture Loader and Animation Saver
4. Picture Loader and Picture Saver

In this way, you can easily convert any picture/animation format into any picture/animation format.

`Append`

You can append the frames/pictures you have selected in the source project to the destination project. The destination project has to be an animation project.

The source projects animation saver has to belong to the destinations loader module. (For Example the IFF-Anim5 loader and saver). The destination project must `NOT` be cached.

Some animation types have to fulfill some other attributes, for example the Section 6.1.7 [IFF-Anim5], page 24 format can only append to non-looping animations. When you want to append frames to a loop Anim5 animation, just resave it and select `NO` in the loop animation requester, then append the frames.

If you want to add a loop to a non-looping animation, just append two pictures which are identical to the first two, then reload the animation.

See Chapter 6 [The Modules], page 22, for more information about the specific attributes of the modules.

5.3 Project View Settings Window

This window controls the output of the `Play` and `Show` gadgets of the Section 5.3 [Project View Settings Window], page 14. The values will be set individually for each project and will be saved with the preferences files.

`Select Display`

You can choose the view mode for your project with this gadget. This will override the old value, if you resave the animation/picture, the new view mode id will be saved. This gadget will be disabled if you view your project in the window mode.

`X/Y Offsets`

With these two integer gadgets, you can select the offsets for your project. If you scroll a frame (while playing an animation or showing a picture), the

values of the gadgets will immediately be set up to the new values. These gadgets will only be enabled if the `Center` option is disabled and you dont use the window mode.

`Display Mode`

This cycle gadget will let you choose between the screen and window mode. In the screen mode the animation/pictures will be shown on a screen (view) of its own.

In the window mode your project will be shown in a resizeable, scrollable window on your MainActor screen.

THIS OPTION IS ONLY AVAILABLE ON SYSTEMS RUNNING OS3.0+.

The more colors the MainActor screen has available (see Section 5.5 [System Preferences Window], page 19), the better is the dithering of your frame/picture. If you show a 256 color picture on a 4 Color screen, you will be disappointed by the result. The timecodes of the animation will be ignored, the animation will be displayed as fast as possible.

`Number of Repeats`

This integer gadget controls how often MainActor will play an animation. If it contains for example a `"1"`, MainActor will play it one time. Possible values are 1 to 9999.

This gadget is only enabled if the source project contains an animation.

`Frame Info`

This switch (when activated) will give you informations about the current frame while playing animations, it is only availaible if the source project contains an animation.

If playing on a screen you will see the info on the top right edge of the screen. When you see no info text, use the `M` and `N` keys while playing to cycle the color of the info text up and down.

When playing in a window, the windows title text will display the infos for the current frame.

This gadget is only enabled if the source project contains an animation.

`Center`

If you enable the center option, MainActor will automatically center the display for your project.The x and y offset gadgets will be disabled but will contain the values for centration. The gadget will be disabled if you use the window mode.

`BBlank`

If you enable the BorderBlank feature (BBlank), MainActor/MainView will set the color of the border of your display to black.

Example:

You have an 320*200 animation with blue as color 0. Without the BorderBlank option the whole screen has blue as color 0. If you enable the BBlank option only your animation (320*200) will have blue as color 0, the border (the rest of the display) will be black.

The BBlank gadget is only availabe if you view your images in the screen mode.

5.4 Project Information Window

This window will supply you with informations about your current source project.

`Project Info`

This gadget opens an information window which should give you all needed informations about your current source project. The window opens centered on the visible part of your working screen. Its title bar shows you which project you are informed about (the actual source project that is).

`WIDTH, HEIGHT :`
: This line shows you the width and height of your project in pixels.

`NUMBER OF PICTURES/FRAMES :`
: Shows you the number of frames in your animation or the number of pictures in your picture list.

`NUMBER OF COLORS :`
: Shows you the number of Colors of your project.

`SPECIAL FLAGS :`
: This line shows you special attributes of the project (if there are any). The flags can currently be:

`----`	for no flags
`HAM`	for HAM pictures or animations
`EHB`	for extra halfbrite images

`SIZE OF PROJECT IN BYTES :`
: Shows you the size of your animation or the total size of all pictures in your picture list in bytes.

`SUPPORTS CHUNKY PIXELS`
: This line will only appear if your source project is chunky pixel based. Chunky pixel based formats are for example : GIF/FLI/FLC. If you dont have a chunky pixel device (like Retina or the Picasso-II), decompression time of these formats are higher, as they have to be converted to the amiga's bitmap format. You can prove that for yourself through the BitMap/Chunky Pixel Time functions.

`ANIMATION TEXT`
: This line will only appear if your source project is an animation. It informs you if you have a color palette per frame or only one color palette per animation and if the animations has a loop or not.

`PICTURE/FRAME LIST GADGET`
: This gadget (the one under the black line) lists your frames or pictures. You can only select one at a time, the size and compression attributes of the currently selected picture are given in the two gadgets below. If you have a looping animation, this gadget will contain two more pictures which are not given in the `Number of Pictures :` line ! These two pictures represent the loop. Some animation formats (like the IFF-AnimBrush loader) need only one picture for the loop information.

`SIZE IN BYTES`
: This gadget shows the size of the currently selected picture or frame in bytes.

`COMP. METHOD`
: Informs you about the compression method used for the currently selected picture. For example *ByteRun* for IFF images or *Delta5* for IFF-Anim5 frames. Consider that the first frame of an IFF-AnimX animation mostly has an IFF-ByteRun image for the first frame.

`Size Profile`

This function opens a window and shows you the sizes of the images in your animation/picture list as a graphical representation. The largest picture has the biggest vertical image, the smallest picture the smallest one. The graphic is layed out to be 600 pixels in width, so the horizontal pixel width for every frame is calculated by : `600 / NUMBER OF PICTURES`. If there are more than 600 Pictures in your source project the pixel width will be 1 for every picture and you may have to scroll with the windows scroll gadgets to see the whole graphic.

`Time Profile`

This function opens a window and shows you the time needed to decompress the images in your animation as a graphical representation. The image which has the slowest decompression time has the biggest vertical bar, the fastest the smallest one.

The bars can have two different colors:

`BLACK`
: This indicates that this image could not be decompressed in the given timecode. If for example image nr.3 of your animation has a timecode of 3/60 of a second, its bar will be black if the

decompression time is bigger than 3/60. The first frame of your animations have mostly black bars, as these images can not be stored with delta compression.

`BLUE`

Everything is ok. This image could be decompressed within its timecode.

Things you have to consider:

If you have black bars, consider that these images will decompress fast on fast machines and slow on slow machines. Try to increase the timecode of these frames until they have blue bars.

If you want to be sure your animation runs smoothly on an A1200, remember to check the color of the bars on an A1200, your current machine (A3000 etc.) could be faster.

The Time Profile function will always decompress the animations in their native format : Bitmap based animations will be decompressed in bitmap mode, chunky animations will be decompressed in chunky mode.

The graphic is layed out to be 600 pixels in width, so the horizontal pixel width for every frame is calculated by : `600 / NUMBER OF FRAMES`. If there are more than 600 frames in your animation the pixel width will be 1 for every frame and you may have to scroll with the windows scroll gadgets to see the whole graphic.

The Time Profile function is only available if your source project is an animation.

`BitMap Time`

If you select this gadget MainActor will decompress the whole source project one time in bitmap format. That means that chunky formats like FLI will take more time due to the conversion to the amiga's bitmap format.

If the source project is an animation, MainActor will decompress the animation (without the loop frames) one time, if the source project is a picture list, MainActor will decompress each picture in the list one time.

The time needed for decompression is given in the text gadget below the `BitMap Time` gadget. The time is displayed in seconds.

Note that the time given for decompression has nothing to do with the time needed to actually play the animation on a screen. The decomp. time function will just decompress the frames as quickly as possible and does not care about time codes or vertical blank interrupts. It is therefore quiete accurate.

It is quiete useful to be able to compare the decompression time of animations who have different animation formats. Choose the format who has the best decompression time and compression ratio and suits you best.

`Chunky Time`

This gadget will only be enabled if the source projects module format supports chunky pixels (i.e. FLI/GIF etc.).

If you select it, MainActor will decompress the whole source project one time in chunky format.

If the source project is an animation, MainActor will decompress the animation (without the loop frames) one time, if the source project is a picture list, MainActor will decompress each picture in the list one time.

The time needed for decompression is given in the text gadget below the `Chunky Time` gadget. The time is displayed in seconds.

5.5 System Preferences Window

This window handles everything concerning the outer appearance of MainActor.

`Select Font`

Choose your font for MainActor. This font will then be used for all windows, gadgets, menus. Only the list items of the Section 5.1 [Project Window], page 9 will be displayed in a different font, you can select it in the Section 5.6 [MainActor Preferences Window], page 21.

The actual font and its size are displayed beneath the gadget. The `Select Font` gadget is not activated when you clone the workbench screen.

OS 2.0 will display the font requester only on the workbench screen and not on the MainActor screen.

`Select Display`

ONLY AVAILABLE AT OS 2.1 +

Choose your display mode for your MainActor screen. The current display mode is displayed beneath the gadget.

This gadget is only activated when

- MainActor has a screen of its own
- You dont clone the workbench screen

`Use own Screen`

If enabled, MainActor uses a screen of its own, if not, MainActor will use the workbench screen or the actual public screen.

`SHANGHAI Windows`

If enabled, MainActor uses the SHANGHAI public screen mode, that means that for example shell windows will automatically pop up on the MainActor screen.

This gadget is only activated when you run MainActor on a screen of its own.

`Screen Size`

This gadget is only available when you have enabled the `Use own Screen` gadget. You can choose MainActors screen size and its overscan mode with this cycle gadget. The screen size of the current mode is displayed in the two integer gadgets to the right of the `Screen Size` gadget. The screen size depends on the screen mode you selected with the `Select Display` gadget.

You can select the following screenmodes:

WorkBench
: Clones the workbench screen properties (screen size, overscan mode, display mode, font)

Custom
: Enter MainActors screen size in the two integer gadgets to the right.

Standard
: Use the standard screen size, without overscan (for Example 640 * 512) at Pal: HighRes Interlace).

Text Overscan
: The normal overscan mode which the workbench uses.

Video Overcan
: Tries to open as big a screen as possible. Flickers on normal displays at the right border.

Max Overscan
: Slightly smaller than the Video Overscan mode, but as useless on normal displays due to its flicker.

`Palette Size`

Select the colors of your palette with this gadget. The value left to the slider gadget shows the current colors available on your MainActor screen.

This gadget only makes sense if you display pictures or play animations in the window mode. More colors == better dithering of your frames/ pictures. (the window mode is only available under OS 3.0 +).

This gadget is only activated when

- MainActor has a screen of its own
- You dont clone the workbench screen.

`Use System Preferences`

If you want to enable the changes you have made to your display just click on this gadget.

5.6 MainActor Preferences Window

This window lets you choose/enter some special MainActor attributes.

`Select Default Load Module`

Lets you choose the default load module. Everytime you open a new project window, the default load module will be the one choosen for this new projects load module.

`Select Default Save Module`

Lets you choose the default save module. Everytime you open a new project window, the default save module will be the one choosen for this new projects save module.

`Select List Item Font`

Choose the font for your list items in your Section 5.1 [Project Window], page 9, this font must have a fixed with, the current font and its size are displayed to the right of the gadget, the default is topaz.font 8.

`Number of List Items`

You can enter how many list items will be displayed in the project windows with this integer gadget, the default is 8.

`Save Icon`

If you enable this gadet, MainActor will save an workbench-icon for every animation/picture you create. You can then view them by double clicking on their icons. The icons support the same tooltypes as MainView, see Chapter 8 [MainView], page 35. The initial settings of the tooltypes clone the settings of the source project (from which you create the new animation/picture).

`Use MainActor Preferences`

Will enable the changes made to the MainActor preferences.

5.7 Text Buffer Window

This window shows and buffers messages for feedback to the user. You can use the `CURSOR UP/DOWN` keys to scroll the text one line up/down. `CONTROL+CURSOR UP/DOWN` will move to the start/end of the buffer. You can also use the windows scrollbar to scroll through the buffer.

See Section 4.2 [Text Buffer Menu], page 7, for more functions concerning the text uffer.

6 The Modules

MainActor has animation and picture modules of which each has loader and saver modules.

6.1 The Animation Modules

6.1.1 Anim Universal

This module examines the animation format and executes the proper animation module. You should mostly use the universal modules as they are able to directly redirect the output of the modules to special hardware (PicassoII,Retina etc.).

Pecularities:

- The *Append* function will not work if the destination project has an Universal loader.

6.1.2 Anim Universal_PicassoII

This module examines the animation format and executes the proper animation module. It redirects the output of the modules directly to the Picasso-II board from VillageTronic.

Pecularities:

- It is faster to view animations of the bitmap formats (like Anim-5/7/8) on the native chipset as these formats have to be converted to the picassos chunky pixel mode.
- Animations of the chunky formats like FLI/FLC are of course much faster on the Picasso-II than on the native chipset.
- The frame info, display mode etc. options have of course no meaning for this module, the handling of the time codes is however correct and you can still stop the animation and flip to the next frame like normal. The function keys also control the speed of the animation.

6.1.3 Anim Universal_Retina

This module examines the animation format and executes the proper animation module. It redirects the output of the modules directly to the Retina board from MacroSystems.

Pecularities:

- It is faster to view animations of the bitmap formats (like Anim-5/7/8) on the native chipset as these formats have to be converted to the retinas chunky pixel mode.
- Animations of the chunky formats like FLI/FLC are of course much faster on the Retina than on the native chipset.
- The frame info, display mode etc. options have of course no meaning for this module, the handling of the time codes is however correct and you can still stop the animation and flip to the next frame like normal. The function keys also control the speed of the animation.

6.1.4 DL

DL animations are ChunkyPixel based and dont support data crunching. MainActor supports type 1 and type 2 DL animations. Type 1 animations are fixed to an resolution of 160*100, type 2 animations can contain resolutions of 320*200, 160*100 and 80*50. The image depth is fixed to 8 bit.

Pecularities:

- DL animations have a rather strange format, for example every frame has an odd offset. That is the reason why examining the animation and playing from hd is so slow : AmigaDos does not like odd offsets.
- Images are stored as raw data, no loop frames are needed.
- MainActor plays DL animations in a linear fashion, the playlist at the end of DL animations is ignored.

6.1.5 FLI

FLI animations are created by Animator Pro, an animation program only available on pc computers. It is quiete old, it can only contain 320*200 animations and its depth is fixed to 8 bit. The successor of FLI is FLC. I have included the FLI modules because a lot of animations (especially on CD-ROM) are in this format.

Pecularities:

- FLI animations always contain a loop, so it is not possible to append to FLI animations. The FLI saver is therefore not able to append to FLI animations.
- FLI animations have only one (global) timecode, it applies to all frames of the animation. The `Set Timecode` function will therefore only be able to set the global timecode and will not set a timecode per frame as usual.
- Because of the missing timecode per frame feature, a lot of animations contain 'Dummy' frames. This are empty frames who shall time the animations. N Dummy frames will for example delay the playback of the

animation for n * global timecode. You can recognice Dummy frames in the `Project Info Window`, they have a bytesize of 0 and a compression type of Dummy.

6.1.6 FLC

FLC animations are created by Animator Pro, an animation program only available on pc computers. FLC is the successor of FLI. FLC animations are not fixed to 320*200 resolutions like FLI and have faster DELTA chunks.

Pecularities:

- FLC animations always contain a loop, so it is not possible to append to FLC animations. The FLC saver is therefore not able to append to FLC animations.
- FLC animations have only one (global) timecode, it applies to all frames of the animation. The `Set Timecode` function will therefore only be able to set the global timecode and will not set a timecode per frame as usual.
- Because of the missing timecode per frame feature, a lot of animations contain 'Dummy' frames. This are empty frames who shall time the animations. N Dummy frames will for example delay the playback of the animation for n * global timecode. You can recognice Dummy frames in the `Project Info Window`, they have a bytesize of 0 and a compression type of Dummy.

6.1.7 IFF-Anim5

The IFF-Anim5 specification was created by `Electronic Arts`.

It is a "good at compressing but slow at playing" format.

Pecularities:

- You cannot append to looping animations. You can bypass this by resaving it to a non-looping animation and then appending the frames. You can make a non-looping Animation to a looping animation by appending two frames which are identical to the first two. (You have to reload the animation to make the loop appear)
- There are some bad animation packers out there, who do not save the first ANHD chunk before the BODY chunk. In this case MainActor cannot insert the timecode for the first frame and will give you an error message (the other timecodes will be set). if you want to time the first frame, just resave the animation with MainActor to avoid this problem.

6.1.8 IFF-Anim7

The IFF-Anim7 specification was created by `Wolfgang Hofer`.

It is a quit good compromise between speed and animation size. The IFF-Anim7_16 modules use words (16 bits) the Anim-7_32 modules longwords (32 bits) for compression. Therefore the 16 bit variant compresses slightly better than the 32 bit modules, but playback is slower.

The IFF-Anim7_32 modules are, in my opinion, the best choice for timeable, fast animations.

Pecularities:

- You cannot append to looping animations. You can bypass this by resaving it to a non-looping animation and then appending the frames. You can make a non-looping Animation to a looping animation by appending two frames which are identical to the first two. (You have to reload the animation to make the loop appear)
- There are some bad animation packers out there, who do not save the first ANHD chunk before the BODY chunk. In this case MainActor cannot insert the timecode for the first frame and will give you an error message (the other timecodes will be set). if you want to time the first frame, just resave the animation with MainActor to avoid this problem.
- The modules do NOT support animations with a DLTA chunk instead of a BODY chunk for the first frame.

6.1.9 IFF-Anim8

The IFF-Anim8 specification was created by Joe Porkka of `ASDG`.

Pecularities:

- You cannot append to looping animations. You can bypass this by resaving it to a non-looping animation and then appending the frames. You can make a non-looping Animation to a looping animation by appending two frames which are identical to the first two. (You have to reload the animation to make the loop appear)
- There are some bad animation packers out there, who do not save the first ANHD chunk before the BODY chunk. In this case MainActor cannot insert the timecode for the first frame and will give you an error message (the other timecodes will be set). if you want to time the first frame, just resave the animation with MainActor to avoid this problem.
- The modules do NOT support animations with a DLTA chunk instead of a BODY chunk for the first frame.

6.1.10 IFF-AnimBrush

The IFF-AnimBrush specification was created by `Dan Silva` of Electronic Arts. Anim Brushes are used/saved by DPaint.

Pecularities:

– There is no saver module at the moment.

6.1.11 Picasso

This animation format was especially created for the Picasso graphic board from VillageTronic. The modules can handle resolutions of 8/16/24 bit. These modules are only available to registered users and legal owners of the Picasso graphic board.

Pecularities:

– The functionality of the Picasso modules are the same as that of the IFF-AnimX. That means that you can add a loop to animations, select if you want to have a color palette per frame (of course only in 8 bit mode), set the timecode for each frame etc.
– If you want to create a 24-Bit animation you will be asked if the resolution shall be dithered down to 16-Bit. 16-Bit is faster than 24-Bit and offers nearly the same image quality so you should give it a try.

6.2 The Picture Modules

6.2.1 Picture Universal

This module examines the picture format and executes the proper picture module. You should always use the universal modules.

6.2.2 Picture Universal_PicassoII

This module examines the picture format and executes the proper picture module. It redirects the output of the modules directly to the Picasso-II board from VillageTronic.

Pecularities:

– The frame info, display mode etc. options have of course no meaning for this module. The only options you have is to press space for the next picture or press escape to abort viewing.

6.2.3 Picture Universal_Retina

This module examines the picture format and executes the proper picture module. It redirects the output of the modules directly to the Retina board from MacroSystems.

Pecularities:

– The frame info, display mode etc. options have of course no meaning for this module. The only options you have is to press space for the next picture or press escape to abort viewing.

6.2.4 GIF

A widely spread image format, designed by `CompuServe`. Revisions 87A and 89A are supported.

Pecularities:

– This format is chunky pixel based, so it will be fastest if viewed through the universal modules which support chunky pixel devices (like Universal_PicassoII etc.), otherwise the module will convert the image to your native chipsets bitplane mode, which will take somewhat longer.
– No saver module yet.

6.2.5 IFF

This is the standard Interchangeable File Format (IFF), the specification was designed by `Electronic Arts`.

Pecularities:

– Starting from MainActor Version 1.02, the IFF modules can load/save IFF24 (24-Bit) pictures.

6.2.6 PCX

PCX is a picture format heavily used on pc's, it was designed by `ZSoft`.

Pecularities:

– No save module yet

6.2.7 Workbench-Icon

These modules support the workbench icons.

Pecularities:

– The saver module will append `".info"` to the save name.

7 The Arexx Interface

MainActors arexx port name is `MAINACTOR`.

Format:

`<>`	Parmaters in these angle brackets have to be supplied
`[]`	Parameters in square brackets are optional
`|`	Vertical bars mean OR and are used to seperate options of which you can choose one : [ON | OFF]

The result variable contains the result of the operation, its possible contents are listed in the command reference.

Look at the `MainActor:rexx` directory for examples. Note that MainActor does not support blanks in filenames which are arexx arguments.

The arexx commands integrated in MainActor 1.2 are:

7.1 Append

`Format:`

Appends the selected frames/pictures of the source project to the destination project.

result: -

7.2 DeSelectAll

`Format:` DeSelectAll

Deselects all frames/pictures in the source project.

result: -

7.3 DeSelectRange

`Format:` DeSelectRange <from> <to>

Will deselect all frames/pictures in the source project from the integer value `from` to `to`

result: -

7.4 DeSelectRangeRequest

`Format:` DeSelectRangeRequest

The DeSelectRange requester will appear,letting you choose the frames or pictures to deselect.

result: -

7.5 GetSPInfo

`Format:` GetSPInfo

Returns information about the current source project:

The width of the source project, followed by its height, followed by the amount of Colors (`HAM6`,`HAM8`,`EHB` will be given back literally for AdPro compatibility), followed by the number of frames/pictures in your project, followed by `CACHING=NO` or `CACHING=YES` depending if you have caching en/disabled, followed by `LOOP=YES` or `LOOP=NO` depending on the loop status and at last `CPF=YES` or `CPF=NO` depending on your color per frame status.

Picture lists have `LOOP` disabled and `CPF` enabled.

result: width height colors #ofpics caching loop cpf

`Examples`:

result=320 256 HAM6 220 CACHING=YES LOOP=YES CPF=NO

result=320 256 16 10 CACHING=YES LOOP=YES CPF=YES

7.6 GetSPLoader

`Format:` GetSPLoader

Returns two arguments, the first is `ANIM` or `PIC` depending of the loader module type of the source project. The second argument is the name of the module itself.

result: ModuleType ModuleName

7.7 GetSPName

`Format:` GetSPName

If the source project has an animation loaded, it returns the full animation name with path, else it returns two arguments, the first is the name of the first picture with full path, the second is the name of the last picture without its path.

result: Name of animation or name of first and last picture.

7.8 GetSPSaver

`Format:` GetSPSaver

Returns two arguments, the first is `ANIM` or `PIC` depending of the saver module type of the source project. The second argument is the name of the module itself.

result: ModuleType ModuleName

7.9 LoadProject

`Format:` LoadProject <Name1> [Name2]

LoadProject will load an animation or picture list, according to the type of the load module of the source project.

`Name1` is the name of the animation or the name of the first picture of the picture list.

`Name2` is only needed when you want to load a picture list, it is the name of the last picture.

`Examples`:

LoadProject Graphics:CountAnim.00001 Graphics:CountAnim.00010

LoadProject Graphics:CountAnim.00001 CountAnim.00010

LoadProject Graphics:Test/mandel7_32

result:-

7.10 OpenNewProject

`Format:` OpenNewProject [CACHING]

Opens a new project window (up to five) If you supply the `CACHING` parameter, the project window will have caching enabled, otherwise it will be disabled.

result: -

7.11 PlayProject

`Format:` PlayProject

Plays the animation contained in the source project, will fail if the source project is a picture list.

result: -

7.12 Quit

`Format:` Quit

Closes all projects and quits MainActor.

result: -

7.13 PrintTxt

`Format:` PrintTxt <text>

Displays `text` in the Section 5.7 [Text Buffer Window], page 21.

result: -

7.14 PrintAndStoreTxt

`Format:` PrintAndStoreTxt <text>

Displays and buffers `text` in the Section 5.7 [Text Buffer Window], page 21.

result: -

7.15 RequestFile

`Format:` RequestFile <text>

This command pops up an ASL-File requester, `text` will be displayed in the requesters title bar.

`Example:`

RequestFile "Select File Name"

result: full path of selected file

7.16 RequestInteger

`Format:` RequestInteger <InitValue> <text>

This command pops up an integer requester, `text` will be displayed above the integer value which was initialized with `InitValue`.

`Example:`

RequestInteger 640 "Enter new Width"

result: integer value

7.17 RequestSaveFile

`Format:` RequestSaveFile <text>

This command pops up an ASL-Savefile requester, `text` will be displayed in the requesters title bar.

`Example:`

RequestSaveFile "Select Save File Name"

result: full path of selected file

7.18 Save

`Format:` Save <Name> [CPF=YES|NO] [LOOP=YES|NO]

An animation saver will save the selected frames into a new animation under the name you put in the `Name` parameter. `CPF` decides wether you want a color palette per frame or not, `LOOP` decides if you want to have a looping animation. `LOOP` and `CPF` overrule the normal behaviour, if you dont supply them, everything will be saved as though you had activated the `Save` gadget.

If you have a picture saver, `Name` is the base name for the pictures, `LOOP` and `CPF` have, in this case, no meaning.

result: -

7.19 ScreenToBack

`Format:` ScreenToBack

Flips the MainActor screen to the back.

result: -

7.20 ScreenToFront

`Format:` ScreenToFront

Flips the MainActor screen to the front.

result: -

7.21 SelectAll

`Format:` SelectAll

Selects all frames/pictures in the source project.

result: -

7.22 SelectRange

`Format:` SelectRange

Will select all frames/pictures in the source project from the integer value `from` to `to`

result: -

7.23 SelectRangeRequest

`Format:` SelectRangeRequest

The SelectRange requester will appear, letting you choose the frames or pictures to select.

result: -

7.24 ShowPictures

`Format:` ShowPictures

Shows the selected pictures/frames of the source project.

result: -

7.25 SetSPLoader

`Format:` SetSPLoader <ModuleType> <ModuleName>

Sets the load module of the source project to the module type `ModuleType` with the module name `ModuleName`.

`EXAMPLES:`

SetSPLoader PIC IFF

SetSPLoader ANIM IFF-Anim5

result: -

7.26 SetSPSaver

`Format:` SetSPSaver <ModuleType> <ModuleName>

Sets the save module of the source project to the module type `ModuleType` with the module name `ModuleName`.

`EXAMPLES:`

SetSPSaver PIC IFF

SetSPSaver ANIM IFF-Anim5

result: -

7.27 SetTimecode

`Format:` SetTimecode <Timecode>

Sets `Timecode` (a value from 1 to 999) to the selected frames of the source project.

result: -

7.28 SwapDeActivatedToSource

`Format:` SwapDeActivatedToSource

Flips the first deactivated project to the source and the source project to the destination project.

result: -

7.29 SwapProjects

`Format:` SwapProjects

Flips the destination project to the source and the source project to the destination project.

result: -

7.30 UnloadProject

`Format:` UnloadProject

Unloads the current project of the source project.

result: -

7.31 Version

`Format:` Version

Will return the actual version string of MainActor.

result: MainActor xx.xx (xx.xx.xx)

7.32 WBToFront

`Format:` WBToFront

Flips the WorkBench screen to the front.

result: -

8 MainView

MainView is MainActors external player. It can display all animations/pictures for which MainActor has the proper loader modules.

The key commands while playing animations or showing pictures are the same as if you would view them directly with MainActor. Thats the reason for not listing them here again, have a look at the Section 5.2 [Project Control Panel Window], page 11 under `Play` and `Show`.

If MainActor saved an icon for the animations/pictures to view, MainView will take the view settings from this icon (overruling its own settings) ! You can change this behaviour through the `IgnoreIcons` option.

MainView will use MainActors center option (which centers the display of your animations or pictures) by default. If you specify an X or Y Offset the center option will be disabled and the specified offsets will be used instead.

You can start MainView from the workbench or form the cli.

`Starting MainView from the Workbench`

If you start MainView from the wb (per icon) you can select the file(s) to view in an asl-requester. For multi selecting just press one of the shift keys and select the files with the mouse. (Starting from OS 2.1+)

MainView supports the following tooltypes:

Window This tooltype enables the window mode, all animations or pictures will be displayed in an window on the current public screen. It is disabled by default.

FrameInfo FrameInfo enables the FrameInfo mode. It Is disabled by default.

NoCaching
Disables caching, with caching disabled MainView plays direct from the harddisk (or whatever). This allows you to play big animations even if you have not much ram. It is disabled by default.

IgnoreIcons
Ignores the icons of the animations/pictures It is disabled by default.

BBlank Enables the BBlank option. It is disabled by default.

XOffset=<Value>
You can directly set the start x offset. 0 is the left border. If you specify an XOffset centration will be disabled !

YOffset=<Value>
You can directly set the initial y offset. 0 is the top border. If you specify an YOffset centration will be disabled !

Repeat=<Value>
: Repeats the animation <Value> times. For example `Repeat=1` would play the animation one time. Defaults to 100.

Directory=<Path>
: This is the initial directory for the asl-requester.

Anim_Module=<Module>
: Chooses the default loader module for animations. Default is the `Universal` module.

Pic_Module=<Module>
: Chooses the default loader module for pictures. Default is the `Universal` module. If you want to view all pictures for example on the PicassoII gfx board, choose the `Universal_PicassoII` module.

`Starting MainView from the cli`

Starting MainView from the cli will give you enhanced functionality. You can for example define patterns for animations or picture lists.

Usage is : MainView [-<options>] <file>

You can enter as many files as you like.

MainView supports these options :

```
-w          : enables the window mode
-fi         : enables the frameinfo option
-nc         : turns caching off
-i          : ignore the icons for the animations/pictures
-bb         : enables the BorderBlank (BBlank) option
-xvalue     : value describes the initial x offset
-yvalue     : the same for the y offset.
-rvalue     : value describes how often the animation
              should be repeated
-amodule    : where module is the default animation loader
              for example -aUniversal selects the universal
              loader (default).
-pmodule    : where module is the default picture loader
              for example -aUniversal selects the universal
              loader (default).
              If you want to remap your pictures to the
              PicassoII board enter -pUniversal_PicassoII
```

Examples:

`MainView -nc -r1 Graphics:Pics/#? Animation/#?8_32` will show all files in the Graphics:Pics/ directory as well as all files in the Animation/ directory which end with '8_32' and all animations will be directly played from the harddisk (one time).

`MainView -pUniversal_PicassoII Graphics:#?_24Bit` will display all pictures which names end with '_24Bit' on the Picasso-II board.

9 Tutorial

This tutorial will try to (quickly) explain the tasks you can perform with MainActor. It is just meant to give you the idea behind MainActor, you have still to read all of the documentation to get the most out of MainActor.

`This tutorial will tell you how to`

9.1 Load Animations

You need an unused Project to load an animation into MainActor. Select the `Select Load Module` gadget of the Section 5.1 [Project Window], page 9 and select the proper animation format (if you, for example, want to load an Section 6.1.7 [IFF-Anim5], page 24 animation, select the IFF-Anim5 loader or the Universal loader). If you want to cache the animation (buffer the animation in memory), enable the `Caching` gadget. Now select the `Load` gadget of the Section 5.2 [Project Control Panel Window], page 11 and select the animation in the opening asl requester. MainActor will now examine the animation and load it into memory (if caching is enabled). After that the frames of the animation are listed in the scroll list area of the project window, also all gadgets (except the `Load` and perhaps the `Append` gadget) of the Project Control Panel Window are now enabled.

9.2 Load Picture(s)

You need an unused Project to load a picture or a picture list into MainActor. Select the `Select Load Module` gadget of the projects window and select the proper picture format (if you, for example, want to load an Section 6.2.5 [IFF], page 27 picture, select the IFF loader). If you want to cache the picture(s) (buffer the picture(s) in memory), enable the `Caching` gadget. Now select the `Load` gadget of the Section 5.2 [Project Control Panel Window], page 11 and select the first picture of the picture list in the opening asl requester, if you want to load only one picture, you can cancel the second requester, otherwise select the last picture of the picture list in the second requester. MainActor will now load the selected picture(s) and buffer them into memory (if caching is enabled). After that the picture(s) are listed in the scroll list area of the project window, also all gadgets (except the `Load` and `Play` and perhaps the `Append` gadgets) of the Project Control Panel Window are now enabled.

9.3 Play Animations

You have to load an animation first. Just click on the `Play` gadget of the Section 5.2 [Project Control Panel Window], page 11. For further details have a look into the documentation.

9.4 Show Pictures

You have to load an animation or a picture list first. Select the pictures you want to show in the source projects list view gadget, after that simply press the `Show` gadget of the Section 5.2 [Project Control Panel Window], page 11. For further details have a look into the documentation.

9.5 Create/Convert Animations

You have to load an animation or a picture list first. Select the pictures who shall be included in the new animation in the source projects list view gadget. After that select the `Select Save Module` gadget and select the proper animation format. Now click on the `Save` gadget of the Section 5.2 [Project Control Panel Window], page 11 and enter the name of the new animation. MainActor will then create the animation, you may have to answer some requesters (looping or non looping animation and so on). For further details have a look into the documentation.

9.6 Save Pictures From Animations

The procedure is exactly the same as before, just select a picture save module instead of an animation module. For further details have a look into the documentation.

9.7 Convert Picture Formats

The procedure is exactly the same as before, just load a picture or a picture list and select another picture formats save module.

9.8 Append To Animations

Load an animation or picture list into the source project and an animation into the destination project. The name of the source projects saver module has to equal the name of the destination projects saver module (Beware of the Universal loader !!) The `Append` gadget of the Section 5.2 [Project Control Panel Window], page 11 should now be enabled. Select the pictures you

want to append (to the destination project) in the source project. After that select the `Append` gadget, MainActor will now append the selected pictures of the source project to the animation in the destination project. For further details have a look into the documentation.

9.9 Time Animations

Load an animation and select the pictures you want to give a new timecode. Now select the `Select Timecode` item in the Section 4.3 [Miscellaneous Menu], page 7 and enter the new timecode into the integer gadget. MainActor will now write the new timecodes to the animation, you can check the new timecodes by playing the animation.

9.10 Execute Arexx Scripts

You can directly execute arexx scripts by selecting the `Start Arexx Script` item in the Section 4.4 [Arexx Menu], page 8 and selecting the arexx scripts. You can map arexx scripts to the function keys through the `Install Arexx Scripts` item (again in the Arexx Menu). You can really speed up your work by using them, take a look at their header to get a short description of what they do.

9.11 Show Pics/Anims On Graphic Boards

Just use the `Universal_PicassoII` or `Universal_Retina` loader modules, they remap the output of every module to the proper gxf board. Note that it is faster to view animations of the bitmap formats (like Anim-5/7/8) on the native chipset as these formats have to be converted to chunky pixel modes. Chunky pixel animation formats like FLI/FLC are on the other hand faster on the graphic boards as on the native chip set.

10 Tips and Tricks

Remember that the Amigas native display is bitmap based ! Chunky pixel based animation formats like FLI/FLC/DL or the GIF picture format have to be converted to bitmaps ! This is slow ! You have to have a chunky pixel based display (like the Picasso-II or Retina gfx cards) to be able to view them full speed (use the Universal_#? loader). On the other hand bitmap based formats like IFF-Anim5/7/8 are slow on these cards as they have to be converted from bitmap to chunky pixel.

Always try to create animations with only one color palette per animation (CPA). They are faster than animations with color palettes per frame (CPF), especially if you have deeper (8-bit) displays. CPF animations also have slower playback rates from hd.

Remember that FLI and FLC animations are always identified as CPF animations, even if they dont have a color palette per every frame. Try to convert them to an CPA animation first if you are not sure if the color palette changes inside of the animation.

The preferences files are *SNAPSHOTS* of your current settup of MainActor, that means that if you save a prefs file and start MainActor later with it you will have exactly the same environment as before.

11 Registration

If you want to register MainActor you can choose between two different packages :

`PACKAGE ONE`

The registration fee for this package is `$50` (85 german marks) or any aquivalent sum in your local currency.

What you get for this package:

A packet consisting of:

- A TeX set, printed handbook
- The latest version of MainActor
- A keyfile

Two free disk based updates

Additional updates are $20 (30 DM) with and $10 (15 DM) without a new handbook

`PACKAGE TWO`

The registration fee for this package is `$25` (40 german marks) or any aquivalent sum in your local currency.

What you get for this package:

A packet consisting of:

- The latest version of MainActor
- A keyfile

Additional updates are $20 (30 DM) with and $10 (15 DM) without a new handbook

You can send cash or international money order to

```
Markus Moenig
Im Johannistal 36
52064 Aachen
Germany
Tel.: (49)-241-71844
```

If you prefer direct transfer:

```
Markus Moenig
NO. 20020251
Stadtsparkasse Aachen
BLZ 390 500 00 (Germany)
Usage: MainActor
PLEASE INCLUDE YOUR NAME ON THE TRANSFER ORDER !!
```

If you transfer money to our account from the outside of europe, the price for MainActor is $55 or $30, bank cost included.

If you have questions/suggestions feel free to contact me through email:

```
Fido        2:242/7.13
Internet    moenig@pool.informatik.rwth-aachen.de
ADSP        markusm@tolkien.adsp.sub.org
```

12 The Future

Future enhancements which are on my list:

- Harddisk playback of the IFF-Anim5/7/8 formats has to become *much* better, planned for the next upgrade.
- If you want to play animations directly from hd, MainActor has to examine the whole animation to store the offset of frames and examine the loop status of the animation. This information should become saveable within MainActor. MainActor/MainView will then be able to directly start playback from hd, planned for the next upgrade.
- An animation loader/saver for VRLIs Vista Pro format (VANIM) and for Real 3Ds animatiom format.
- MPEG and JPEG loader.
- More picture and animation modules (in general).
- Far more arexx scripts (sinus (de)-acceleration scripts and more).
- German documentation.
- Syncing sound (through sound modules) to animations.
- Special modules for commercial hardware (digitizers,gfx cards ..) are possible.
- The modules are designed to be expandable, that means that future expansions like image processing modules or multimedia effect modules are not only possible but were planned from day one of the development.

Your suggestions are always welcome.

13 Credits

This package was developed by usage of the following products:

- Dice, from Obvious Implementation Corp
- Devpac 3.04, from HiSoft
- WShell2.0 and Arexx, from Wishful Thinking Corp
- TurboText from Oxxi
- ToolManager from Stefan Becker
- MultiTool II from Boris Jakubaschk

Thanks for producing high quality software.

Special thanks to:

- Stefan Becker for his Amiga knowledge, without him this project would not have been become possible
- all GAMMA and BETA testers
- Norbert Bogenrieder for the icons
- Jesper Juul for his suggestions
- all registered users for their support and trust in my work

Index

Q

R

S

T

U

V

W

Z

Table of Contents