

OpaqueMove

Steve Koren

COLLABORATORS

	<i>TITLE :</i> OpaqueMove		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Steve Koren	December 8, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	OpaqueMove	1
1.1	OpaqueMove Documentation	1
1.2	Introduction to OpaqueMove	1
1.3	Requirements for using OpaqueMove	2
1.4	Copyright Information	2
1.5	Installation of OpaqueMove	2
1.6	OpaqueMove User Interface and Program Operation	3
1.7	User Interface Window Gadgets	3
1.8	Drag Behaviors	4
1.9	User Interface Menus	4
1.10	Icon Tooltypes	5
1.11	UI vs. NoUI Program Versions	5
1.12	Factors Affecting Opaque Movement Speed	5
1.13	Quitting	6
1.14	Common Problems	6
1.15	Known Bugs And Limitations	7
1.16	Future Enhancements	7
1.17	Author	7
1.18	Credits	8
1.19	Revision History	8

Chapter 1

OpaqueMove

1.1 OpaqueMove Documentation

OpaqueMove: An Opaque Window Movement Commodity

for OS 3.0 or greater and ≥ 68030 CPU's

(recommended: accelerated graphics card)

Copyright 1994 Steve Koren

Version 1.0

@{ " Introduction " link Introduction } What is OpaqueMove?

@{ " Requirements " link Requirements } Requirements for using the program

@{ " Copyright " link Copyright } Copyright information.

@{ " Installation " link Installation } How to install OpaqueMove

@{ " Program Operation " link User_Interface } Using OpaqueMove

@{ " Common Problems " link Problems } If you have problems...

@{ " Bugs, Limitations, etc. " link Bugs_And_Limits } Known bug list

@{ " Enhancements " link Enhancements } Things to be added

@{ " Author " link Author } How to contact the author.

@{ " Credits " link Credits } Couldn't have done it without...

@{ " History " link History } Program version history

1.2 Introduction to OpaqueMove

Introduction to OpaqueMove

OpaqueMove is a commodity that provides opaque window movement for Intuition windows. That is, instead of simply moving the outline of a window, the entire window contents are moved as you move the mouse. The EGS window system provides this capability, but does not extend it to standard Intuition screens. Hence this commodity.

This commodity is intended for use with accelerated graphics cards such as the GVP Spectrum. Although it is possible to use it with standard Amiga AGA screens, it is likely to be impractically slow except possibly in single bitplane modes. However, there are various controls which allow selection of the conditions under which windows should be moved opaquely, and under which they should be moved as an outline.

1.3 Requirements for using OpaqueMove

Required: 68030 or greater CPU

Since OpaqueMove is not practical on slow systems, it has been compiled for 68030 or greater CPU's and will not run on 68000 or 68020 based systems.

Required: AmigaDos 3.0 or Greater

OpaqueMove makes use of some features which are not present under AmigaDos 2.04.

Required: reqtools.library

Some features of OpaqueMove require reqtools.library (Copyright Nico Francois).

Recommended: accelerated graphics card such as the GVP Spectrum

Opaque window movement requires very rapid window redraw speeds, and standard Intuition windows can only operate at these speeds using low bitplane depths, such as single or double plane screens. There is nothing in OpaqueMove which requires an accelerated graphics card, but it works much better with one. Furthermore, there are many other conditions which can create situations in which windows cannot be moved opaquely at acceptable speeds. OpaqueMove tries to allow control over when windows are and are not moved opaquely.

Required: AmigaDos libraries

OpaqueMove uses the following AmigaDos shared libraries, all of which must be available on the system in ROM or on disk:

- icon.library
- layers.library
- commodities.library
- gadtools.library
- asl.library
- graphics.library
- dos.library
- intuition.library

1.4 Copyright Information

Copyright Information

OpaqueMove is Copyright 1993 Steve Koren, All Rights Reserved

OpaqueMove is copyrighted freeware. It may be distributed on PD disk collections costing no more per disk than the price of a current Fish disk, or US\$6.00, whichever is the greater. OpaqueMove may be uploaded to BBS systems or public internet FTP sites. It may not be sold for a profit or modified without permission of the author, nor may authorship of OpaqueMove be claimed. Fred Fish is explicitly permitted to distribute OpaqueMove on his CD-ROM collection of software.

Further distribution of OpaqueMove may not be restricted. This applies to services such as CompuServe which otherwise attempt to restriction distribution of software from their archives. If OpaqueMove is distributed, please insure that the original archive is distributed, not a re-assembled version thereof.

1.5 Installation of OpaqueMove

Installation of OpaqueMove

To install OpaqueMove , simply drag the icon file to your WBStartup directory. There are two versions; a larger one with a UI window, and a smaller without.

For the larger version, it may also be desirable to set CX_POPUP to no in the icon tooltypes. Otherwise, the window will be displayed each time you start your system.

1.6 OpaqueMove User Interface and Program Operation

OpaqueMove User Interface and Program Operation

To see a picture of the OpaqueMove user interface, [click here](#).

@{ " Window " link UI_Window } Information on the main window

@{ " Menus " link UI_Menus } Information on the menus

@{ " Tooltypes " link UI_ToolTypes } Information on icon tooltypes

@{ " UI vs. NoUI Version " link UI_NoUI } Two program versions

@{ " Speed " link UI_Speed } Factors affecting speed of movement

@{ " Quitting " link UI_Quit } Exiting the program

1.7 User Interface Window Gadgets

User Interface Window Gadgets

Patterns (Screen)

This string gadget contains an AmigaDos pattern which is used to match against the title of the screen containing the window to be dragged. Opaque movement is only performed for screens whos title matches this pattern. Useful patterns are "#?", to match any screen, and "~(screen1|screen2)" to match any screen except screen1 and screen2, although any valid pattern may be used. This field has a maximum length of 1023 characters.

Patterns (Window)

Similar to the Screen pattern, this one is used for matching against the title of the window to be dragged. Opaque movement is only performed for windows whos title matches this pattern. This field has a maximum length of 1023 characters.

Screen Depths (1-8)

Windows will only be moved opaquely if the screen they reside on is of a selcted depth. For example, on most accelerated graphics cards, 8 plane screens are updated much faster than 4 plane screens. The reasonable values for opaque window movement might be 1, 2, and 8 plane screens, so the corresponding buttons should be checked. For a standard AGA Amiga, only low values should be used.

Max Windows on Screen

When a screen has a large number of windows, window updates slow down and opaque window movement is unreasonably slow. The value of this gadget is the maximum number of windows present on a screen before windows are always moved as outlines.

Max Window Size

Large windows may be updated too slowly to be effectively moved opaquely. This gadget sets a maximum size for the windows. Windows above this size will be moved as an outline. If the area covered by the window exceeds the area specified by these values, it will be moved as an outline. For example, if these are set to 300x300, then a 600x100 window will still be moved opaquely since its total area is less than 300x300, even though one of its dimensions exceeds the specified dimensions. It is the area which matters, not the width or height themselves.

Simple Refresh

On some accelerated graphics cards, simple refresh windows are moved much faster than smart refresh windows. If this gadget is checked, simple refresh windows can be moved opaquely. If it is not, they will always be moved as an outline.

Smart Refresh

Similarly, if this gadget is checked, smart refresh windows can be moved opaquely. If it is not, they will always be moved as an outline. If some windows appear to move much slower than others of the same size, experiment with these two buttons.

Drag Qualifier

The keyboard modifier that is used for window dragging (along with the left mouse button) is selectable via this gadget.

Drag Location

This option controls the location of the cursor for dragging windows. If a keyboard qualifier is used, the window can be dragged by putting the cursor at any location inside the window. Alternately, the window can be dragged from the titlebar either with or without a keyboard qualifier. Lastly, an option is provided called center which requires that the cursor be placed towards the center of the window. This option is provided for compatibility with other commodities which provide window resize capability if the cursor is placed near a window border. This lets the same qualifier key be used for both commodities; the window will be dragged if grabbed towards the center, or resized if grabbed near a border.

Note that the interaction of Drag Qualifier and Drag Location can provide different behaviors (see: [Drag Behaviors](#));

Save Button

This button saves the current preferences settings to both `envarc:Opaque.prefs` and `env:Opaque.prefs`. The changes will take effect even after your system is rebooted or shut down.

Use Button

This button saves the current preferences settings to `env:Opaque.prefs` only. These settings will only take effect until the next time your system is rebooted.

Cancel Button

This causes any changes made to be lost, the last preferences settings re-read from the `env:` directory.

1.8 Drag Behaviors

Drag Behaviors

By setting the drag qualifier and drag location options appropriately, different drag behavior can be obtained:

Drag Qualifier: unset, Drag Location: Drag Bar

This is the standard Intuition behavior. Windows can be dragged with no keyboard qualifier by grabbing them from the title bar.

Drag Qualifier: set, Drag Location: Anywhere

This will allow the window to be dragged from anywhere inside it, as long as the proper qualifier key is held down. Note that the window can also be dragged normally (via outline) using the standard Intuition convention of unqualified left mouse button.

Drag Qualifier: unset, Drag Location: Anywhere

This combination is not valid, because it would cause all mouse clicks to be intercepted by OpaqueMove, and ignored by the application windows. The user interface will not let this combination be chosen.

Drag Qualifier: set, Drag Location: Center

This combination is provided for compatibility with other commodities which allow window resizing by grabbing the window near a border or corner. By using the Center option, OpaqueMove will pass mouse events near the edge of the window on. (Right now the limit is 15 pixels from the edge, and is not configurable. If anyone has a real need to change it, I could make it configurable). This allows the resize commodity to use the same qualifier key as OpaqueMove.

Drag Qualifier: set, Drag Location: Drag Bar

Opaque dragging must be done from the drag bar, and a qualifier key must also be used. This allows windows to be outline dragged using the standard Intuition convention of unqualified left mouse button.

1.9 User Interface Menus

User Interface Menus

Project

Open...

Opens a new preferences file.

Save

Saves the current preferences to envarc: and env:

Save As...

Saves the current preferences to a named file.

About...

Presents a small requester with information about the program.

Quit

Exits OpaqueMove.

1.10 Icon Tooltypes

Icon Tooltypes

CX_PRIORITY=<n>

Sets the commodity priority. This should be set to a number greater than other commodities which care about double click events.

CX_POPUP={yes|no}

If set to yes, OpaqueMove will show its window when first invoked. If set to no, it will not.

TASKPRI=<n>

Sets the task priority for the OpaqueMove task.

DONOTWAIT

Tells Workbench not to wait for this commodity to exit when it is run from the WBStartup directory.

1.11 UI vs. NoUI Program Versions

UI vs. NoUI Program Versions

There are two versions of the program supplied in the archive. The standard version (OpaqueMove) provides a user interface window which is selectable from the Commodities Exchange window.

There is also a UI-less version (OpaqueMove_NoUI) which provides a smaller executable which can be used with less stack space as well. This version will read any preferences file saved from the UI version, so the UI version can be used as a preferences editor, and the UI-less version can be run to save memory.

1.12 Factors Affecting Opaque Movement Speed

Factors Affecting Opaque Movement Speed

Intuition was never designed for opaque window movement, and the fact that it can be done at all is a testimonial to its design and flexibility. There are other window systems on the Amiga which can provide faster opaque window movement than can Intuition; for example, the EGS system bundled with many graphics cards does so. However, Intuition can do a manageable job at this, especially on high end 68040 systems with accelerated graphics cards.

There are several factors which can affect window dragging speed during opaque movement:

o If you are using a graphics card and have a workbench backdrop picture, the PatchDT program will be almost a necessity for performing opaque window movement at reasonable speeds. It refreshes the datatype images from FAST instead of CHIP RAM, speeding background redraw speeds by about 5X.

- o If you set up a pattern instead of a picture as a workbench backdrop using WBPatten, some patterns (particularly non-solid) are redrawn quite slowly. This causes window movement over the workbench backdrop to be slow. For maximum speed, either use a solid background, or use a picture as a backdrop and use PatchDT to speed redraw. The latter case is almost as fast as the former, and windows moving opaquely over this background move very rapidly as long as no other windows need be redrawn in the process.
- o The number of windows on the screen, and the degree to which they overlap, affect refresh speeds. The more windows there are, the slower window redraws are. Also, moving a window in front of or behind another causes one or the other to be redrawn, which causes delays. The impact of this is dependent on the speed of your computer and graphics card. It isn't too bad on a 68040 system unless there are a lot of windows open. It might be a lot slower on a 68030 or with a slower graphics card.
- o The depth of the screen makes a big difference. On AGA Amigas, windows on deeper screens are updated slower than on more shallow screens. With most graphics cards, 8 plane screens are the fastest, and 4 plane screens are the slowest.
- o Bigger windows are refreshed slower than small. This effect is particularly noticeable for AGA screens.
- o Intuition moves smart-refresh windows in a manner which is not optimal. Especially on accelerated graphics cards, these windows are moved much slower than simple refresh windows. It may be necessary to toggle off smart-refresh window opaque movement to obtain reasonable speeds.

1.13 Quitting

Quitting

OpaqueMove can be shut down by any of the following:

- o Sending a control-C signal to the process.
- o Using the Commodities Exchange window to remove the commodity.
- o Selecting quit from the OpaqueMove menu.

1.14 Common Problems

Common Problems

Problem:

OpaqueMove exits without appearing as a commodity.

Cause:

OpaqueMove detected an error during startup. This is usually caused by running on the wrong version of AmigaDos (see [Requirements](#)), or a missing shared library, or an error in allocating memory or system resources.

Problem:

Windows are still moved as outlines even with OpaqueMove installed.

Cause:

One common cause of this problem is using the wrong qualifier keys for window movement, or not grabbing the window in the required location. Also, insure that OpaqueMove is really running and is not disabled in the Commodities Exchange window.

OpaqueMove also uses certain internal tests to determine whether a given window may be moved opaquely. If any of these tests fail, the window will be moved as an outline. Most of the parameters used in the tests can be set from the OpaqueMove [user interface](#).

Problem:

Certain other commodities which act on double-clicks are now doing so even for single button clicks.

Cause:

OpaqueMove must be run with a higher CX_PRIORITY than these commodities. For an explanation, see [Bugs and Limitations](#).

Problem:

OpaqueMove appears to intercept all mouse clicks.

Cause:

Inproper settings of Drag Qualifier and Drag Location. See [Drag Behaviors](#) for details.

Problem:

Windows are moved too slowly.

Cause:

Opaque window movement demands tremendous system resources, and is only practical on the most high end systems, and even then, only in some circumstances. For a list of factors which influence movement speed, see [Movement Speed](#).

1.15 Known Bugs And Limitations

Known Bugs And Limitations

- o Due to the manner in which the no-qualifier titlebar-drag option works, application gadgets placed into the window title bar may be inaccessible. I will try to find a way to fix this problem.

- o This commodity must both insert and remove events from the input stream. In order to avoid confusing other commodities which look for double clicks, it must run with a higher CX_PRIORITY than those commodities. Moderatly technical explanation: this commodity intercepts and removes certain input events, and sometimes inserts others after that happens. These are mouse events. If a mouse click goes by, and another commodity in the chain see is before OpaqueMove, and OpaqueMove removes this and inserts another, the former commodity will be fooled into seeing a double click where there was none. To avoid this, run OpaqueMove with a higher CX_PRIORITY. This will insure that the offending event is removed and re-inserted before the other commodity sees it, thus providing the correct behavior.

- o When OpaqueMove detects that the left mouse button has been released, it uses the current position of the mouse cursor to compute the final location of the window. However, if you are very fast, it is possible to have moved the mouse cursor after releasing the left button, but while the previous refresh is still redrawing. In this case, the window will be moved to the current location of the cursor, not the location the cursor was at when the mouse button was released. I hardly ever see this happen unless I am trying to make it happen, so I don't think it is a big problem, but it might be more of a problem on slower systems.

1.16 Future Enhancements

Future Enhancements

- Provide the ability to have non-qualified title-bar movement and qualified window movement at the same time.

1.17 Author

Author

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. Send comments, criticisms, and 68060 cards to the above address.

1.18 Credits

Software used in writing OpaqueMove:

- GadToolsBox (Copyright Jan van den Baard)
- ReqTools (Copyright Nico Francois)
- SAS/C 6.51 (Copyright SAS Institute)
- GNU emacs 18.58.1 (Free Software Foundation; Amiga port by David Gay)

Also:

- The icon for this AmigaGuide file was borrowed from the YAK commodity by Gael Marziou and Philippe Bastiani.
- The directory icon was borrowed from some MagicWB style collection from Aminet.
- I made the OpaqueMove icon myself, and it shows :-). If anyone comes up with a better one, send it along...

1.19 Revision History

Revision History

5 Sep 1994 - V1.0 - first public release