GetMouseInput

COLLABORATORS			
	TITLE : GetMouseInput		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 25, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

GetMouseInput

1.1 Get Mouse Input V1.3: Contents

********** GetMouseInput V1.3 **********

Revision date: 30-May-1997

Contents...

1.2 What the heck is GetMouseInput?

GetMouseInput is a simple program to read the state of the mouse buttons.

It returns a value corresponding to the button(s) pressed which can be read

from an AmigaDOS® script.

1.3 What's new in this version?

V1.3

The following changes have been made since V1.2:

- GetMouseInput now has a LOCAL option to return the mouse button state to a local variable.
- Added GLOBAL option to return mouse button state to an environment variable
- Arguments now freed properly.
- A few small errors in the documentation have been corrected.

V1.2

The following changes have been made since V1.1:

- GetMouseInput now has an option to return the mouse button state to stdout instead of to an environment variable.
- Executable even smaller!
- Plain text docs no longer included. I have decided from now on to only include AmigaGuide documentation with GetMouseInput. If you REALLY want the plain text docs back, let me know.

1.4 What do I need to run it?

Any Amiga with WB2.04+. I think.

1.5 What files should I have?

In this distribution, there should be the following files:

GetMouseInput * The main program

GetMouseInput.guide * This document

GetMouseInput.guide.info

SampleScriptGLOBAL * Example startup-sequence using GLOBAL option

SampleScriptLOCAL * Example startup-sequence using LOCAL option

SampleScriptSTDOUT * Example startup-sequence using STDOUT option

Note that plain text docs are no longer included. If you REALLY want these back in, let me know!

1.6 Copyright & Distribution

GetMouseInput is FREEWARE. That means you can copy it and freely distribute it (including uploading to a BBS/Aminet etc.), as long as:
- You don't charge anything for it (except a nominal charge for disk and copying)
- You include all the files listed above in any distribution.
- You don't modify any files.
(For inclusion on compilation CDs or magazine coverdisks/cover CDs, contact me. Any other queries, contact me. No permission is necessary for inclusion on Aminet or Fred Fish CDs.)
The copyright of the program remains with me at all times.

1.7 Disclaimer

You use this program entirely at your own risk. The author will accept no responsibility whatsoever for any loss or damage caused by this program including but not limited to loss of or damage to data.

1.8 Why write GetMouseInput?

I wrote GetMouseInput as a quick and easy solution to a problem I had. Although some games will install onto a hard drive, some will not run from Workbench for various reasons (VGA screen modes, lack of memory etc.). Therefore they have to be either loaded from floppy or started from AmigaDOS before Workbench is loaded. This is inconvenient, so I decided to write a simple menu system as a script file. However, there remained the problem of how to start the menu. I decided a good way was to hold down a mouse button as the computer booted. So, I wrote GetMouseInput to enable me to do this.

1.9 OK, what does it do exactly?

When it is started, GetMouseInput will check the state of the mouse buttons (for the mouse in port 1, just in case you wondered...) and then return a number corresponding to the button(s) pressed in a variety of ways:
- to a global environment variable called MouseInput
- to a local variable called MouseInput
- to stdout
- any combination of the above
You can then check the returned number and perform an appropriate action.
See How do I use it? for information about how to do this.

1.10 How do I use it?

You need to call GetMouseInput from the CLI with the following template:
STDOUT/S,LOCAL/S,GLOBAL/S
If you call GetMouseInput without any arguments (as in V1.1), it will
behave as it did then; i.e. it will store the output in an environment
variable called 'MouseInput'.
- If you use the STDOUT option, the output will be sent to stdout.
- If you use the LOCAL option, the output will be sent to a local variable
called MouseInput. This is 'tidier' than the GLOBAL option because it
does not require any assigns.
- If you use the GLOBAL option, the output will be sent to a global
(environment) variable called MouseInput. This requires the ENV:
assign.
You can use any combination of STDOUT, LOCAL and GLOBAL. Therefore,
using:
GetMouseInput GLOBAL
is analogous to calling GMI with no arguments but the 'no arguments'
capability is left in for backwards compatibility.
GMI will probably need to called from within a script file if it is to be
of much use. You can create a script file in any text editor (e.g. Ed).
No. Button(s) pressed
--- -----------------
0 None
1 Left
2 Right
3 Left + Right
4 Middle
5 Left + Middle
6 Right+ Middle
7 Left + Middle + Right
You can check the number returned by using the AmigaDOS 'If' command, for
example:
; With no options, the GLOBAL option or the LOCAL option (storing to an
; env or local variable)
If $MouseInput EQ 1
; Left button was pressed
EndIf
; Using the STDOUT option (sending to stdout)

If `GetMouseInput` EQ 1
; Left button was pressed
EndIf
Please note: in the second example above, the apostrophes are 'backwards' apostrophes, not normal forwards apostrophes. You can get them by pressing Alt-apostrophe on a British keyboard.
So, as you can see, it is very easy to set up a whole range of functions each accessed by pressing a different combination of mouse buttons. You can therefore set up your startup-sequence to load different setups of Workbench or start menus etc. easily. To help you, I have included three sample scripts which you could use as your startup-sequence; they all do exactly the same thing but demonstrate GMI's three options. These files execute different startup files according to whether you press no buttons, the left button or the right button. You can of course modify them to include a middle mouse button etc.
However, all these examples on how to use GetMouseInput are only suggestions; there are many uses for it. You could even set up a loop in a script file and recursively call GetMouseInput until a certain mouse button combination is pressed, therefore allowing a 'wait for mouse click' function from AmigaDOS!
By the way, if you want to do things properly, you can unset the MouseInput environment variable (if you have been using this) after you have finished with it by using:
UnSetEnv MouseInput
Or, if you've been using the LOCAL option, you can use:
UnSet MouseInput

1.11 Future Improvements

I don't have any ideas for any future improvements to GetMouseInput as it's such a simple program. But then again, I said that two versions ago!! :)
So, if you have any further suggestions, I'd be glad to hear from you and implement them.

1.12 Known Bugs

None. But do let me know if GetMouseInput does anything naughty such as totally screwing up your system.

1.13 Version History

V1.00 Original version. Bit of a naughty bug which tended to cause a
software failure.
V1.01 FIRST PUBLIC RELEASE. Bug fixed.
V1.1 Same as V1.01 but with Style Guide compliant version number :)
V1.2 Added STDOUT option. Thanks to Andreas Boerner for this suggestion.
Removed plain text docs.
V1.3 Added LOCAL option. Thanks to Timo Ronkko for this suggestion.
Added GLOBAL option. Tidied up the code. Corrected a few small
errors in the documentation

1.14 Support and latest version

GetMouseInput is still under development, which means new versions will
appear from time to time. I will release new versions as soon as they are
in a fit state to be placed in public distribution.
The latest version of this program, along with some other programs I have
written, can always be downloaded from my program support web pages, which
can be found at the following URL:
http://www.radiolink.net/timjackson
New versions will also be uploaded to Aminet as soon as they become
available.
E-mail enquiries are always welcome, see my contact page for
details.

1.15 Contact me!

If you've found GetMouseInput useful, TELL ME ABOUT IT! The program is
freeware; there is nothing to be paid but I'd like to hear from you anyway.
Any bug reports or suggestions for improvement should also be sent to the
below e-mail address:
tim_jackson@bigfoot.com
I occasionally use IRC and when I do, my usual nick is Arcline. I tend to
hang around on IRCnet. A list of IRCnet servers can be found at:
http://www.irchelp.org/irchelp/networks/servers/ircnet.html

1.16 Index

You should be so lucky...

1.17 Sample script files

Here are copies of the sample script files included with this program:

Using environment variable:

```
Assign ENV: RAM:
Assign T: RAM:
GetMouseInput GLOBAL
If $MouseInput EQ 1
Execute S:Startup1 ; Left mouse button pressed
Else
If $MouseInput EQ 2
Execute S:Startup2 ; Right mouse button pressed
Else
Execute S:NormalStartup-Sequence ; No buttons pressed
EndIf
EndIf
UnSetEnv MouseInput
EndCLI >NIL:
```

Using local variable:

```
GetMouseInput LOCAL
If $MouseInput EQ 1
Execute S:Startup1 ; Left mouse button pressed
Else
If $MouseInput EQ 2
Execute S:Startup2 ; Right mouse button pressed
Else
Execute S:NormalStartup-Sequence ; No buttons pressed
EndIf
EndIf
UnSet MouseInput
EndCLI >NIL:
```

Using STDOUT option:

```
If `GetMouseInput STDOUT` EQ 1
Execute S:Startup1 ; Left mouse button pressed
Else
If `GetMouseInput STDOUT` EQ 2
Execute S:Startup2 ; Right mouse button pressed
Else
Execute S:NormalStartup-Sequence ; No buttons pressed
EndIf
EndIf
EndCLI >NIL:
```