

Jungle Slicker

COLLABORATORS

	<i>TITLE :</i> Jungle Slicker	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		August 4, 2022
		<i>SIGNATURE</i>

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Jungle Slicker	1
1.1	Cheata Guide	1
1.2	Introduction	2
1.3	Example	2
1.4	Size	3
1.5	Alignment	3
1.6	Search Value	3
1.7	Replacement	4
1.8	Sortmode	4
1.9	Update	4
1.10	Start Search	4
1.11	Listview	4
1.12	Comment	5
1.13	Replacement	5
1.14	Delete	5
1.15	New	5
1.16	Execute	5
1.17	Listview	5
1.18	Menü	6
1.19	SegTracker from a monkeys view	6
1.20	Register your sister	6
1.21	History	7
1.22	Tech	7

Chapter 1

Jungle Slicker

1.1 Cheata Guide

WARNING: The author is not capable of the english language. Reading this documentation can severley damage your mind.

[Intro](#)

[Example](#)

[Main window](#)

[Size](#)

[Alignment](#)

[Search Value](#)

[Replacement](#)

[Update](#)

[Start Search](#)

[Listview](#)

[Macro window](#)

[Comment](#)

[Replacement](#)

[Delete](#)

[New](#)

[Execute](#)

[Listview](#)

[Menu](#)

[SegTracker](#)

[----->Register](#)

[History](#)

[Copyright](#)

1.2 Introduction

Wanna cheat? Use Cheata!

Cheata is a program to cheat with some games running in a multitasking environment. A good example would be SomniaZity®.

Cheata runs best when cooperating with SegTracker. SegTracker can be found with newer versions of Enforcer. Enforcer can be found on newer fishdisks.

Unexperienced usage of Cheata may call big asian brother, so better read this documentation from the middle to the top.

1.3 Example

Next we will walk through a small example giving you some experience for handling the average case. Some general statements will be accompanied by a special case study: SomniaZity®.

Starting SegTracker now might be a good idea. Just open a shell, start SegTracker from within and close the shell. You might include SegTracker in your startup-sequence or user-startup (again, SegTracker is not really needed, it just makes things easier).

Now start the program (normally a game) which you wanna trick out. Let the game run until you can see the value you wanna (eh, sorry for the "wanna", learnt my english by listening to the Ramones...) change and stop/pause the game. Switch back to your workbench and start Cheata by doubleclicking on its icon.

Special example: Start SomniaZity@ and select the difficulty level which gives you \$10000. This value is the one we wanna change. Now set game speed to paused, change to Workbench and start Cheata.

Enter the value to be changed in the string gadget labeled Search Value. Right below you enter the value which should replace the other value.

Special example: Enter 10000 in the... ah, you know, enter 500000 (or whatever) in the gadget labeled Replacement.

Above you can see two cycle gadgets. In the first one you have to select the maximum the value you wanna change can have.

Special case: As SomniaZity's® \$-value can crawl above 65535 the only possible selection is Longword.

Same thing with the other cycle labeled alignment. Good programs always have longwordalignment for longwords and at least wordalignment for words.

Special case: SomniaZity® likes the longwordalignment.

Now you are ready to hit the Start Search gadget. There will come another window which you must not give any attention. Searching might take some time depending on the values you entered and the configuration of your Amiga.

Special case: nothing special.

The listviewgadget will show you all the addresses of your RAM in which the value to be searched was found. You now have to find out, which address is the right one. If there is only one... lucky. Maybe the SegTracker info which is displayed in the first three columns will help you identifying the only real one. If you do not get the trick, you have to take another strategy. Switch back to the first application and let it run for some while until the value to be changed changes. Now stop it again and switch to Cheata, press the Update button and wonder what happened. If you have a lot of entries you should now set the Sort Mode to Highlight. All highlighted entries have changed their values and will be at the top of the display.

Special case: Look for entries with the name SomniaZity® as Loadname (shows up in connection with SegTracker only). More than one? Switch to SomniaZity®, select Slow Game Speed, buy a donut factory for \$100 and set Game Speed to Paused. Now press Update in Cheata and set Sort Mode to Highlight. Look for a row with Value value 9900.

To change the value at a special address: doubleclick the row.

Special case: doubleclick the row you identified before and watch the Replacement jumping into.

You may now kill Cheata or let it be there for another round later. For a usage days, months, years later with the same application you should remember the Loadname, Hunknum and Offset of the position where you found the right value. This will always be the same, which is not the case with the address, which may change every time you load the application again. Hunknum and Offset might change if you install another version of your application.

Special case: Go on building LA in SomniaZity®. You can get more money by entering a new value in the Replacementgadget and doubleclicking the identified row in the listviewgadget again. For later usage you should remember Loadname, Hunknum and Offset of the row, because they will always be the same and help you identifying the right entry next time. This will not be the case if you switch from SomniaZity® to SomniaZityII® or SomniaZity2001®.

----- Addendum for V1.1

You can now save found values permanently and manipulate them much easier and faster at a later time. SegTracker is needed to do so.

Identify the value as usual, open the macro window and drag&drop the found entry into it. If you want to change several values at once in one program, then identify them one after another and d&d them all in one macro. Now enter an informative comment and save the macro (ending should be ".cheat"). You can load it back later and simply press the Execute button to poke all values at once into memory.

1.4 Size

The Size gadget is for selecting the maximum the value you wanna change can have. Byte means 0..255, Word means 0..65535, Longword 0..(2³²-1).

In fact you will have some problems knowing the maximum if have never seen the value getting above 100 or so. In this case it is possible that the value might be stored in a longword, too. If you choose Byte Size you have to choose Byte Alignment or you will never find such a longword value. Experience, guess or pray.

Searching for longwords is a little bit slower on computers with 16Bit RAM (500, 500+, 600, 1000, 2000) than searching for bytes or words.

There is something to know when working with 68000 or 68010 Amigas. This CPUs cannot read words or longwords from odd addresses. You may not set the Alignmentgadget to Byte when looking for such sizes. Cheata knows about these problems and will reject wrong selections. These combinations are possible on CPUs >= 68020 but should not be used as they are very unlikely.

1.5 Alignment

The Alignment gadget is for choosing the RAM positions in which Cheata will search. Bytealignment lets Cheata search at all positions, wordalignment means all positions which can be divided by 2 without a remainder, longwordalignment means all positions which can be divided by 4 without a rest.

You can expect that longwords have longwordalignment and words have at least wordalignment, but there might be some exceptions.

When selecting bytealignment Cheata has to search at 4times the positions than selecting longwordalignment. Time spent for searching will be influenced noticeable by this gadget.

There is something to know when working with 68000 or 68010 Amigas. This CPUs cannot read words or longwords from odd addresses. You may not set the Alignmentgadget to Byte when looking for such sizes. Cheata knows about these problems and will reject wrong selections. These combinations are possible on CPUs >= 68020 but should not be used as they are very unlikely.

1.6 Search Value

This gadget is for entering the value you are looking for. You may enter only positive integers and the maximum value depends on the selection you made in the sizedgadget.

The value you wanna look for is important when thinking about the time Cheata will spent for searching. As small values (like 0, 1 ...) will be very often in RAM opposed to big hairy numbers (like 9418), searching for these small numbers will take factors

longer. Everytime Cheat finds a number, the memory address has to be remembered, it has to be placed in the listview and SegTracker has to be questioned. So it is not recommended to search for the value 0.

Another problem is, that every time the value will be found, some memory has to be allocated in the listview to remember its address. No problem for a dozen entries, big one for a few thousands. If Cheata runs out of memory during searching, no further values will be remembered and the final list will be incomplete.

1.7 Replacement

Here you may enter the value which will replace the value to be searched at selected addresses. His max depends on your selections in the sizegad.

1.8 Sortmode

This gadget will change the sortmode of the listviewgadget.

Address: Entries will be sorted depending on their value in the address-column. Small ones first.

Name: Entries will be sorted depending on their alpha value in the loadname-column. With same names the values in the columns hunknum and offset are decisive.

Highlight: First ones are the highlighted entries, inside and outside of this group all goes like with name-sorting.

1.9 Update

Update will scan the address of each entry in the listview again and compare the found value to the formerly found value. If the value has changed, the entry will be highlighted. If sortmode is set to highlight, sorting will follow the update.

This little action is useful, if you cannot decide right after searching which entry is the right one to change. See example section for an example.

1.10 Start Search

Clicking this gadget will start the actual search. There may pop up something like an error window if you made some wrong entries in some of the gadgets (e.g. search value of 10000 with searchsize of byte).

If alls OK, a second window will materialize and inform you about the progress of the search. You can abort the search with the Cancelgadget at (nearly) any time. As only used memory will be searched, there may be some jumps in the progress indicator.

Searching has finished as soon as the second windows transforms itself back into a piece of unused memory. You may have to wait some more seconds for the result, because SegTracker will inspect the addresses after the search and MUI takes sometimes some time to display the entries if there are a lot of them, too.

1.11 Listview

The listviewgadget is for admiring the result of Cheatas work.

The first three columns of an entry will only show something useful if SegTracker is active:

Loadname: Loadname gives the name of the loadfile, in which the value to be searched was found. Sometimes path is included, sometimes not. (Note: A program may occupy much more memory than occupied by its loadfile, so this is column is only a hint (or so...))

Hunknum: Programms tend to consist of several hunks. Hunknum(ber) tells you, in which hunk the value was found.

Offset: This is the offset to the beginning of a hunk. (Note: Sometimes there will be some idiot values in this column, but this is of no importance to Cheata and its purpose. Handle them like normal values.)

Useful columns without SegTracker:

Address: This column shows the address where the value was found.

Value: This column shows you the value of an address. As Cheata is running in multitasking, the value might be changed by any other program at any time without Cheata noticing. Same valid, when you change a value with Cheata.

Changing a value can be achieved by doubleclicking its entry. This will ram the replacement value into the RAM address specified in the entry. Doubleclicking again will place the old value again into place.

Warning: Unbrained changing of RAM-values can knock out the system immediately or place a time-bomb for some fun in the evening. So...

1.12 Comment

Enter an informative description, including name of the program to manipulate, its version etc.

Example:

SomniaZity® 2001 V1.003 Egyptian version, \$ cheat

1.13 Replacement

Replacement value for the listview.

1.14 Delete

Deletes the selected entry in the listview.

1.15 New

Deletes all entries of the macro.

1.16 Execute

Pokes all values from the macro into memory. SegTracker needs to be active!

1.17 Listview

A doubleclick will transfer the value from the replacement gadget into the entry in the listview. The value is not poked into memory, hit the Execute gadget to do so.

1.18 Menü

New: Deletes all entries of the macro.

Open: Load a macro from disk.

Save: Save a macro under its old name.

Save As: Save a macro under a new name.

About: Information about Cheata.

About MUI: Informationen about MUI.

Quit Program: Quit Cheata.

1.19 SegTracker from a monkeys view

SegTracker tracks programmes being loaded from disk to memory. But it only knows about their static memory usage, these progs may allocate much more memory at a later time.

Why should you use it?

First, if you are searching for a value in program A, with the help of SegTracker you can include out addresses which belong to another program B or C (sometimes D).

Secondly, all addresses with a blank loadname are unlikely to be of any importance to you, as most the values you will be looking for (scores) are allocated in a static way at loadtime.

Thirdly, if you want to change a value several times, you can easily recognize it by its loadname, hunknum, and offset after you have identified it the first time.

1.20 Register your sister

Cheata is shareware. After some time (15 days) of evaluation you have to register. To register you have to send the author the fee of 10DM (Deutsche Mark) or \$7 (US Dollar) or the corresponding amount in your local currency (but only if its inflation is less than 20% a year, otherwise pay DM/US\$). Do not send coins, only banknotes. As you have already the full version, I will not send you a useless copy (would cost you much more money), but instead release new versions through the usual distribution channels.

Please fill out the order form (file: REGISTER.TXT). Fold it, put the notes(s) between it and send it in a closed envelope to:

Stefan Tiemann

Heisstr. 20a

D-48145 Münster

Germany

Annoy the author with your bugreports and comments at:

tiemant@uni-muenster.de

Extra Chance

In the order form you will find a special place where you can enter your personal obscure number (between 0 and 1,000,000). If this number matches the predrawn number then you will have won a sweet little donkey, which will be shipped to you for free. Additionally, you will receive fooder for the first three weeks.

Your donkey is named Calvin Klein, is house-trained, is a good listener and will answer all your questions with eyeah. He is fond of children and old people and likes to watch tv when you have no time for him.

He is also a perfect watchdonkey, because he needs almost no sleep and will never loose orientation even in the biggest garden. If you wish, we can train him before delivery to go on SM-DOS or WonderWhy95 users for free.

Force your luck, register now! (You can even register multiple times to maximize your chances)

1.21 History

V1.0

First public version

V1.1

- Better handling of "Out Of Memory"
- Macros
- Mui 3.x bubble-help

1.22 Tech

Cheata needs AmigaOS 3.0+ and MUI 3.0+

Cheata is written in Modula-2 and compiled with M2Amiga V4.3.

GUI by MUI, so read the following:

This application uses

MUI - MagicUserInterface

(c) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz

Eduard-Spranger-Straße 7

80935 München

GERMANY
