Emulator - Emulator Issues #7695

Rewriting Cheat engine - Simplifying Finding Ram Values

09/27/2014 07:18 PM - yasharnasirian

Status:	New	**% Done:**	0%
Priority:	Normal		
Assignee:			
Category:			
Target version:			
Operating system:	N/A	**Relates to performance:**	No
Issue type:	Feature request	**Easy:**	No
Milestone:		**Relates to maintainability:**	No
Regression:	No	**Regression start:**	
Relates to usability:	No	**Fixed in:**	

Description

Basically what is wanted is to implement ram searching into Dolphin.

The advantage being you don't have to worry about little/big endian, or converting virtual addresses, pointer addresses, etc when using Cheat Engine.

Having a way to conveniently display values at known addresses would be more useful than the way Dolphin users currently achieve this task.

This will be more convenient for tool-assisted speedrunners wanting to use Dolphin and speedrunners who want to understand their games at a deeper level

History

#1 - 09/27/2014 07:38 PM - yasharnasirian

For reference, please check this link out: http://tasvideos.org/EmulatorResources/RamSearch.html

#2 - 09/28/2014 06:24 AM - skidau

- Issue type changed from Bug to Feature request

#3 - 10/05/2014 11:06 PM - jesse_lun

Can I also request for a pointerscanner function? Certain games like Kirby Air Ride iirc has addresses that changes places every now and then, so a way to find out where it went would be nice.

An example of how Cheat Engine does it is here:
http://tasvideos.org/forum/viewtopic.php?t=13462

#4 - 02/16/2015 02:34 AM - reyvgm

I would love this option too.

And a way to add cheats while playing, instead of having to right-click the game on the list and add them there.

#5 - 08/18/2017 01:02 AM - aldelaro5

Hi, so I know a lot about the topic of RAM searching within Dolphin so allow me to state possible solutions.

First, altough it is definetely possible, having an integrated RAM search within Dolphin isn't that easy because of performance concerns (it would need to call functions that the emulator already uses a ton in a very short ammount of time, so the additional load would likely cause a drop in performance).

However, as I had been perfectly aware of the pain of the usage of Cheat Engine, I actually made a completely new RAM search specifically made for Dolphin, do note however that this isn't supported by the Dolphin team, I am currently the sole maintainer of this project. I solved in the first beta release most inconveniences of Cheat Engine such as:

- The inability to track dynamic memory (my RAM search has multilevel pointer supports and it works).
- The need to add extensions to have big endian types (you could on Cheat Engine, but you had to add custom types, for my RAM search, it's done automatically).
- No possibility to add 2 separate memory range to scan (this is mostly annoying for Wii games which has 2 separate memory region, for my RAM search, you can enable or disable the extra region depending on the console the game runs on).
- Finally, but definetely not least: the annoyance with the start address. You need to scan a limited part of the Dolphin process which coresponds to the emulated memory, but getting the address of this memory isn't easy, before 5.0-3981, it was only deterministic on Linux and KIND OF stable on Windows, but not a guarantee. After 5.0-3981, Dolphin supports ASLR which makes the start completely random because the process itself is mapped randomly. Even pressing stop and play can cause a change so you are forced to use the pointer method, which has many issues (it doesn't persist between revision and you need to compute a hex number for every address you add in the table). How my RAM search deals with this is it checks Dolphin's mapping info as it is running and detemrines the right one, once hooked, you shouldn't be bothered by the start address anymore.

Altough, reminder this is still currently in BETA, it;s very functional as a watcher and a scanner, it also supports file saving. You can get it in the release tab of this GitHub page: https://github.com/aldelaro5/Dolphin-memory-engine

For more information on its usage and a nice discussion points for TASers, I made a complete TASVideos thread on it which you can go here: http://tasvideos.org/forum/viewtopic.php?t=19437 Btw, this also explains how you can find pointers and add it to my RAM search. I need to say however that implementing pointerscan for my RAM search is pretty much insanely hard or impossible because I can't really read the original PowerPC instructions that the game Dolphin is running executes (you likely will run Dolphin behind a JIT which obsures them). However, in recent revisions of Dolphin, the debugger has been stable enough for you to basically do the same thing, you just need to use breakpoints upon reading or writting to the memory you want to know the pointer of. So, a read of the appropriate section in the thread should answer that question.

Alternatively, if you want more information on using Cheat Engine with Dolphin, I recommend this thread, I will still maintain it until my RAM search becomes stable enough for me to not recommend Cheat Engine anymore: http://tasvideos.org/forum/viewtopic.php?t=17735

With that, I hope this post will provide a good enough solution.