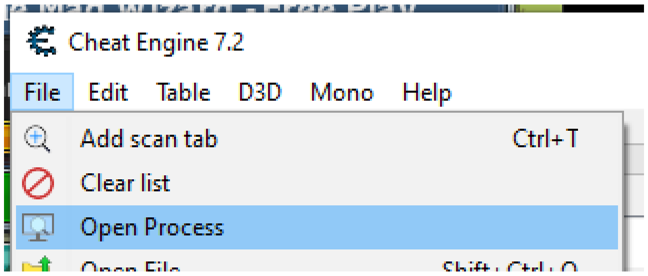
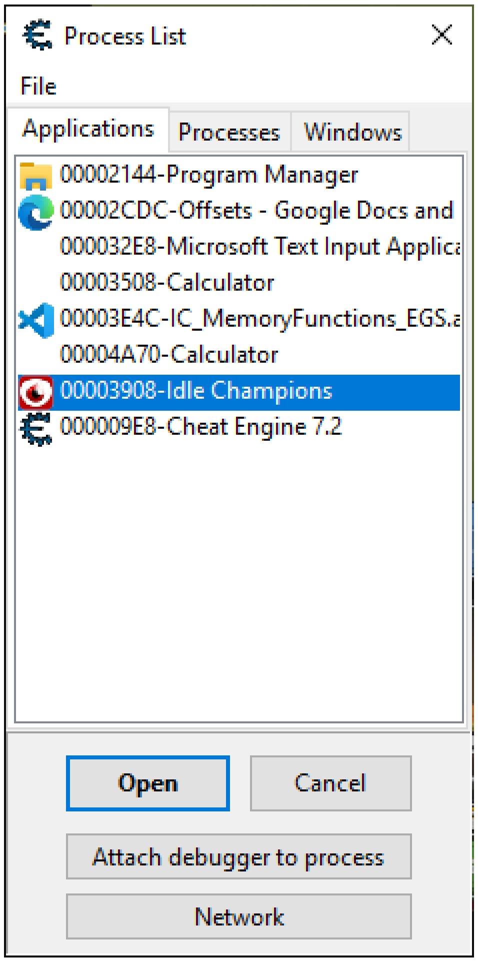
A Quick Run Down of Finding Idle Champions Pointers with Cheat Engine

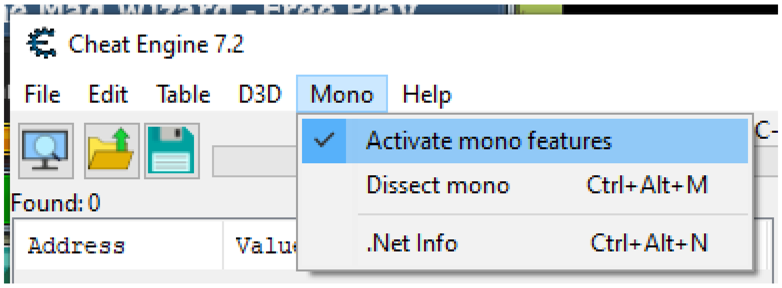
1. Restart your computer.
2. Open Idle Champions and Cheat Engine
3. Open the Idle Champions Process





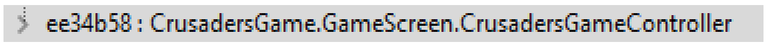
00002144-Program Manager
00002CDC-Offsets - Google Docs and
000032E8-Microsoft Text Input Applica
00003508-Calculator
00003E4C-IC_MemoryFunctions_EGS.a
00004A70-Calculator
00003908-Idle Champions
000009E8-Cheat Engine 7.2

4. Activate Mono Features (should be checked as in image)



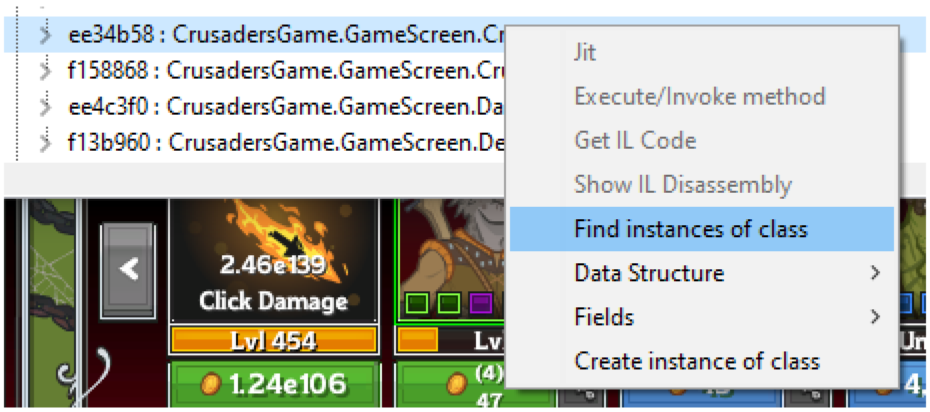
Tip: disable mono features before closing the opened process.

5. Dissect mono (Ctrl+Alt+M)
6. Expand tree and select Assembly-CSharp
7. Find (Ctrl+F) gamecontroller Tip: Uncheck Search entire file
8. Find Next until you find CrusadersGame.GameScreen.CrusadersGameController Class



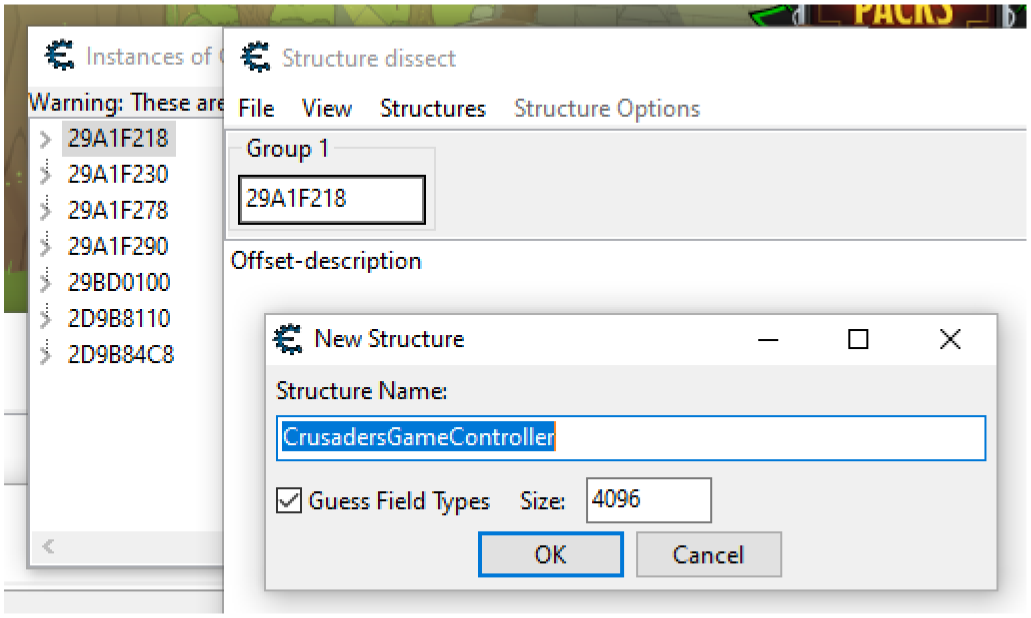
ee34b58 : CrusadersGame.GameScreen.CrusadersGameController

9. Right click and select find instance of class



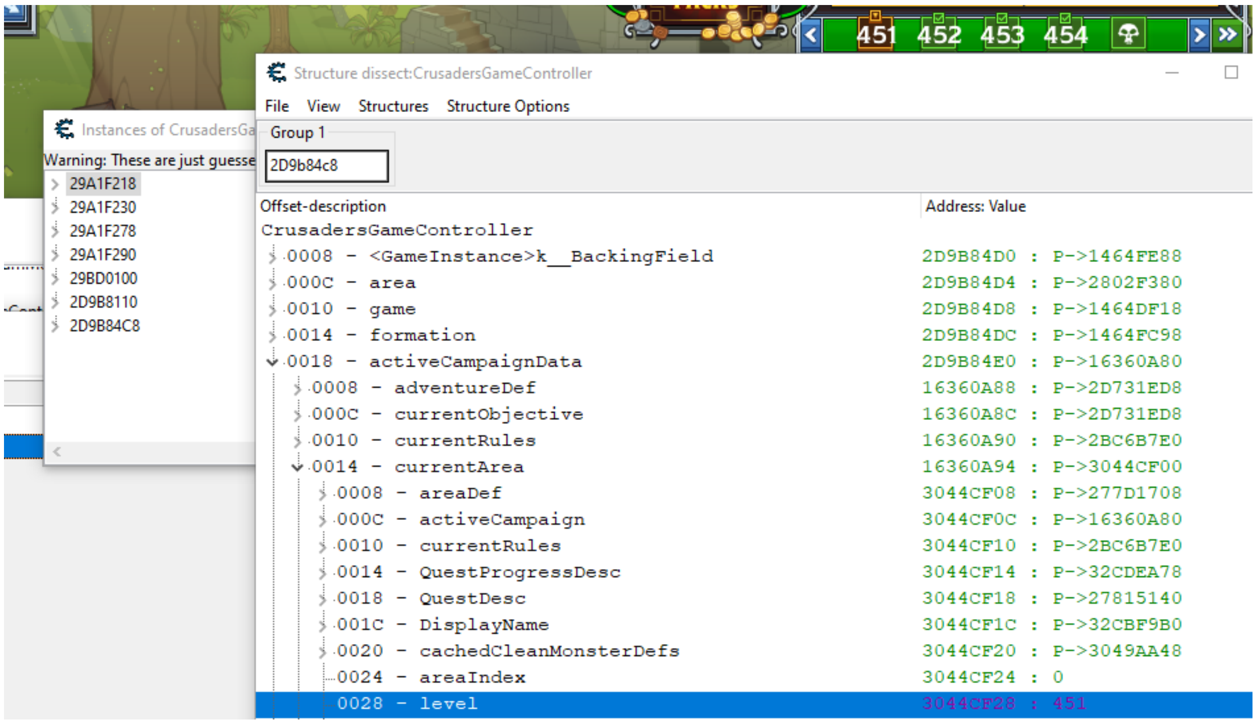
ee34b58 : CrusadersGame.GameScreen.Cr
f158868 : CrusadersGame.GameScreen.Cr
ee4c3f0 : CrusadersGame.GameScreen.Da
f13b960 : CrusadersGame.GameScreen.De

10. Double click any of the addresses in the Instances of… window that pops up to open the Memory Viewer
11. Open the Structure dissect window (Ctrl+D)
12. Enter an address from the Instances of… window in the field for Group 1.
13. Define a New Structure (Ctrl+N)



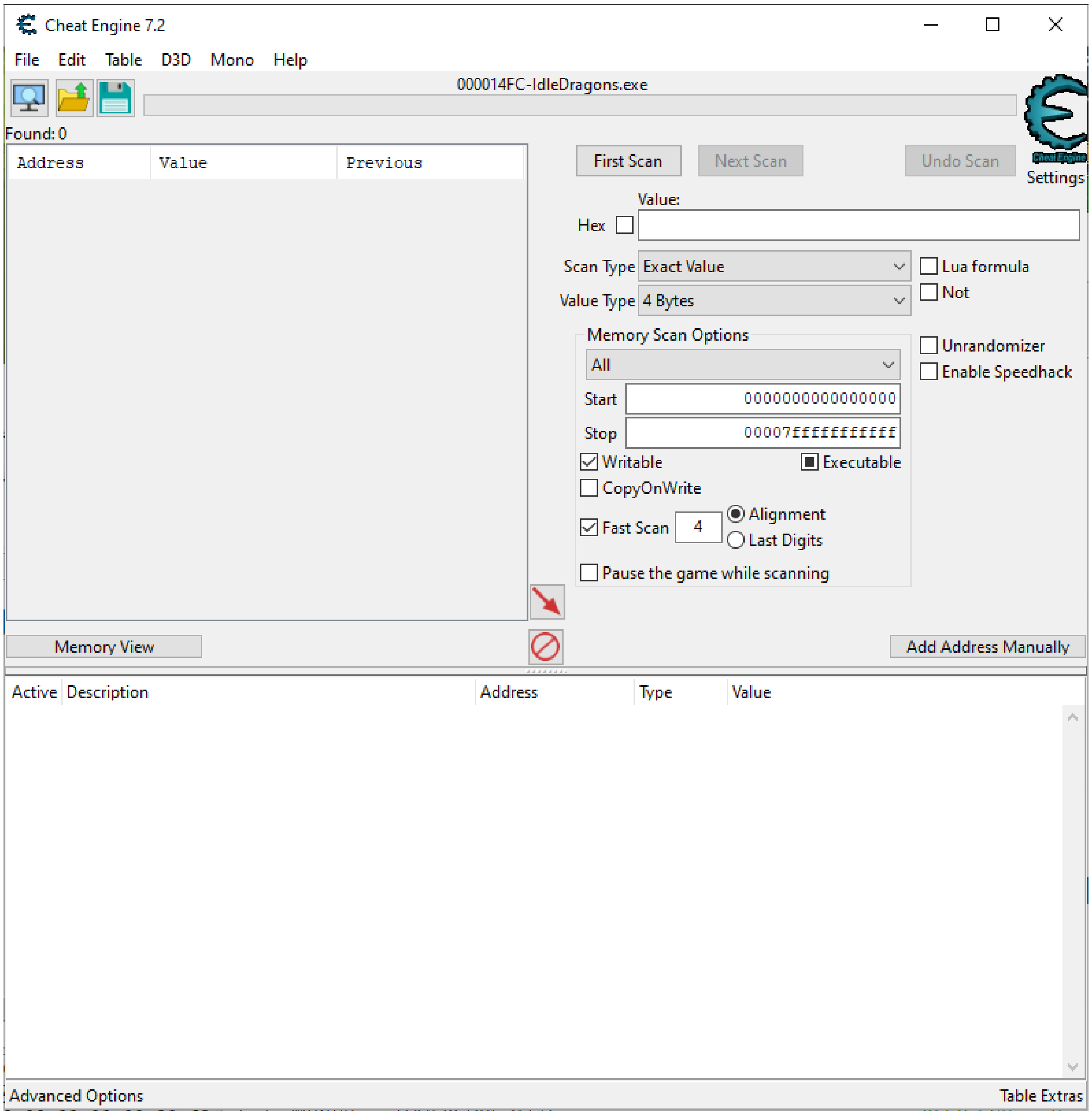
If the Structure Name is not consistent with the class, press cancel and try a new address from the list.

14. Check if the structure is correct.



```
CrusadersGameController
 0008 - <GameInstance>k__BackingField        2D9B84D0 : P->1464FE88
 000C - area                                 2D9B84D4 : P->2802F380
 0010 - game                                 2D9B84D8 : P->1464DF18
 0014 - formation                            2D9B84DC : P->1464FC98
 0018 - activeCampaignData                   2D9B84E0 : P->16360A80
   0008 - adventureDef                       16360A88 : P->2D731ED8
   000C - currentObjective                   16360A8C : P->2D731ED8
   0010 - currentRules                       16360A90 : P->2BC6B7E0
   0014 - currentArea                        16360A94 : P->3044CF00
     0008 - areaDef                          3044CF08 : P->277D1708
     000C - activeCampaign                   3044CF0C : P->16360A80
     0010 - currentRules                     3044CF10 : P->2BC6B7E0
     0014 - QuestProgressDesc                3044CF14 : P->32CDEA78
     0018 - QuestDesc                        3044CF18 : P->27815140
     001C - DisplayName                      3044CF1C : P->32CBF9B0
     0020 - cachedCleanMonsterDefs           3044CF20 : P->3049AA48
     0024 - areaIndex                        3044CF24 : 0
     0028 - level                            3044CF28 : 451
```

15. Add Address Manually (center left button), include a description.



000014FC-IdleDragons.exe

Found: 0

Value:

Hex ☐

Scan Type: Exact Value

Value Type: 4 Bytes

☐ Lua formula

☐ Not

Memory Scan Options: All

Start: 0000000000000000

Stop: 00007fffffffffff

☑ Writable ▣ Executable

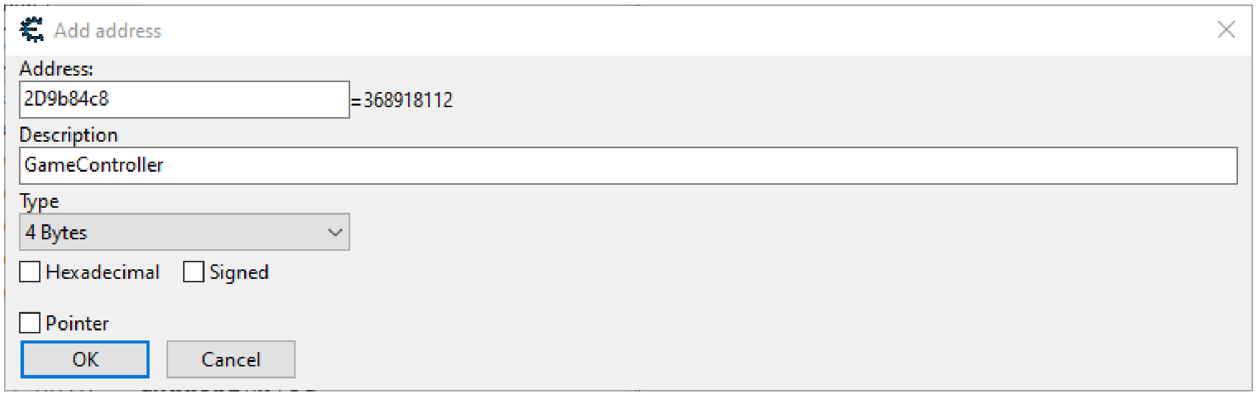
☐ CopyOnWrite

☑ Fast Scan 4 (● Alignment ○ Last Digits)

☐ Pause the game while scanning

☐ Unrandomizer

☐ Enable Speedhack



Address: 2D9b84c8 =368918112

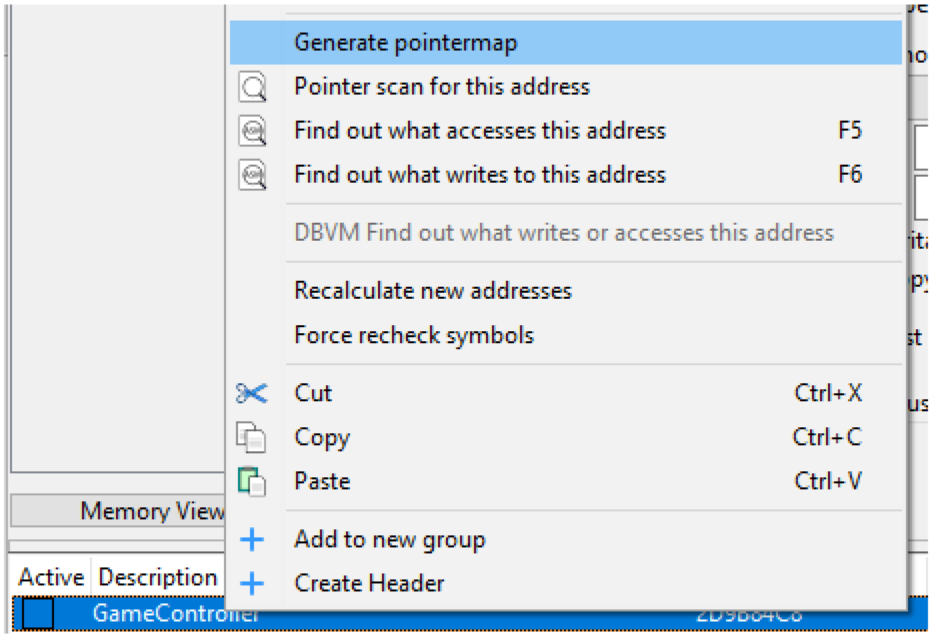
Description: GameController

Type: 4 Bytes

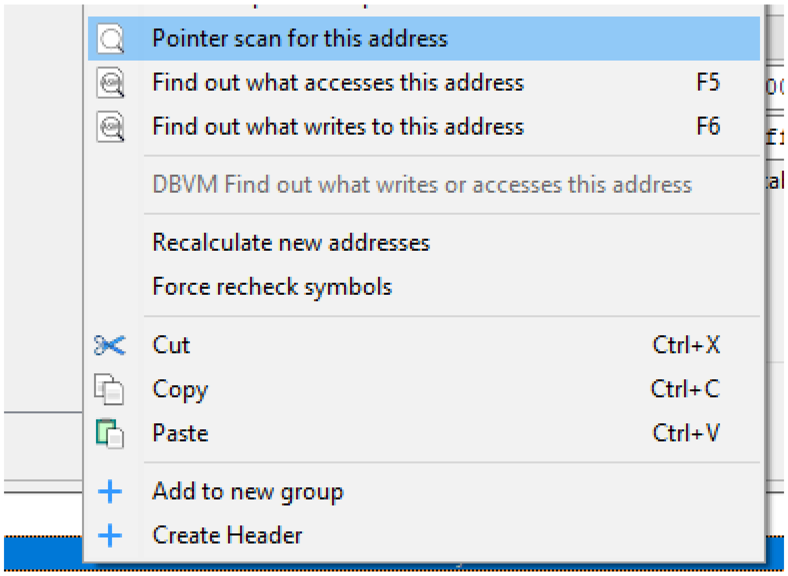
☐ Hexadecimal ☐ Signed

☐ Pointer

16. Right click the added address and select Generate pointermap and save the map.

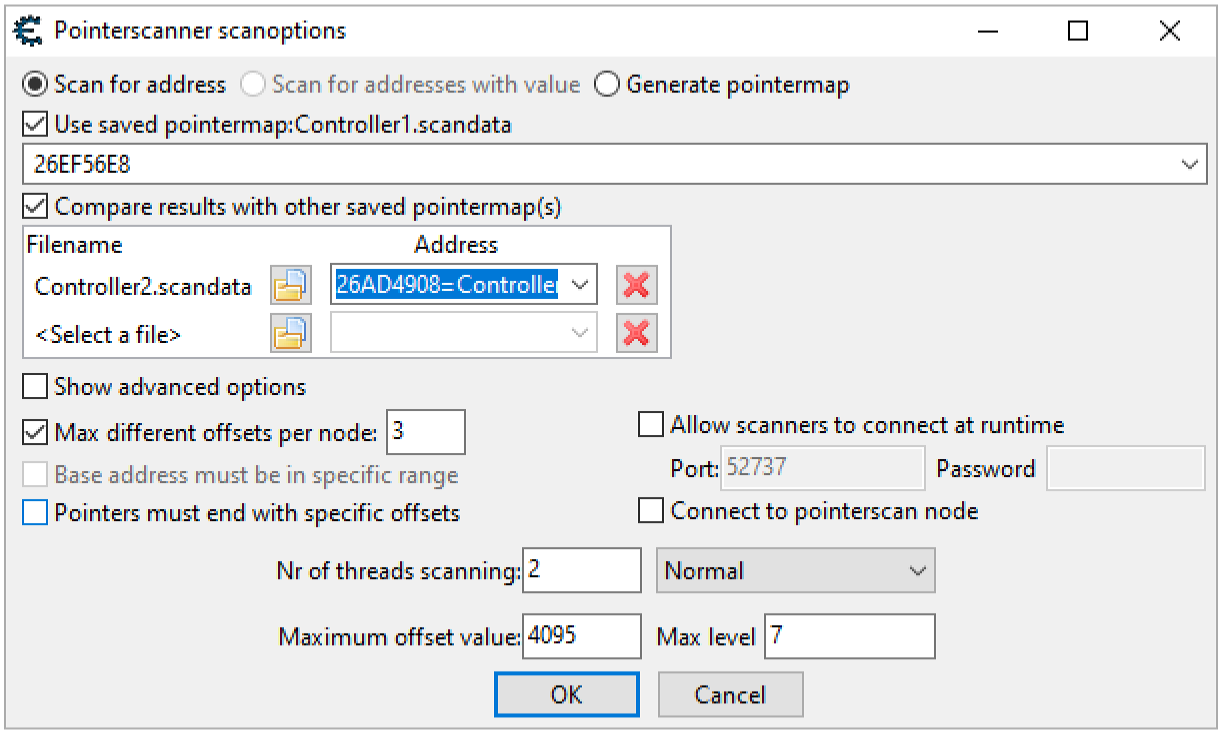

Generate pointermap
Pointer scan for this address
Find out what accesses this address F5
Find out what writes to this address F6
DBVM Find out what writes or accesses this address
Recalculate new addresses
Force recheck symbols
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Add to new group
Create Header
Memory View
Active
Description
GameController

17. Repeat steps 1 through 16 to generate a second pointer map.
18. Close Idle Champions.
19. Right click the second (or any) address and select pointer scan for this address


Pointer scan for this address
Find out what accesses this address F5
Find out what writes to this address F6
DBVM Find out what writes or accesses this address
Recalculate new addresses
Force recheck symbols
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Add to new group
Create Header

20. Check the Use saved pointermap box and select the first map you generated.
21. Use the pull down to select the correct address.
22. Check the Compare results with other saved pointermap(s) box.

23. Select a file and choose the second map you generated.
24. Use the pull down to select the correct address.
25. I generally start with these settings:


Pointerscanner scanoptions

[✓ Scan for address] Scan for addresses with value Generate pointermap

[✓] Use saved pointermap:Controller1.scandata

26EF56E8

[✓] Compare results with other saved pointermap(s)

Filename	Address
Controller2.scandata	26AD4908=Controller
<Select a file>	

[ ] Show advanced options

[✓] Max different offsets per node: 3

[ ] Base address must be in specific range

[ ] Pointers must end with specific offsets

[ ] Allow scanners to connect at runtime

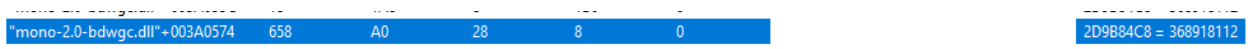
Port: 52737 Password

[ ] Connect to pointerscan node

Nr of threads scanning: 2 Normal

Maximum offset value: 4095 Max level 7

26. Restart Idle Champions, find the correct address for the Game Controller structure. You can use the steps above or look at the Pointer Scan results and start with an address you see there to skip using the mono dissector and instance scan.
27. Repeat step 26 until you have a handful of paths. If the number remains large and unchanging, consider restarting your PC, swapping parties back and forward in Idle Champions, restarting your adventure, etc. If you are left with no pointer paths you may need to change Max level setting in step 25 to a number higher than 7.
28. With a solid pointer to the structure we can then go back to our Structure dissect window and retrieve the remaining offsets to the actual memory locations we are interested in.


```
"mono-2.0-bdwgc.dll"+003A0574   658   A0   28   8   0        2D9B84C8 = 368918112
```

This structure has a module base address of “mono-2.0-bdwgc.dll”+003A0574 and offsets of 0x658, 0xA0, 0x28, 0x8, and 0x0. For our purposes with AHK the 0x0 can be omitted from our scripts.

The Structure dissect window lists the offsets on the left side. We see an offset of 18 to get to activeCampaignData, an offset of 14 to get to currentArea, and an offset of 28 to get to level. If this is the only read we want from this structure your offset array to level would look like: [0x658, 0xA0, 0x28, 0x8, 0x18, 0x14, 0x28]

If we were planning on reading multiple locations in memory based on this structure we could code AHK to read the our pointer to the structure which gives us the address of the structure, store that as a variable to be used as a base address for the subsequent reads with the offsets garnered from the Structure dissect window. In this example it would save four memory reads per any memory read after level. As of drafting these instructions, the script is reading Quest Remaining, Is Transitioning, and Time Scale Multiplier from this structure, saving us 12 memory reads per cycle.