# INSIDE THE ULTIMA ONLINE GOLD DEMO
# - THE COMMAND LIST – PART 3

## GOAL

It's our goal to get a deep understanding of how the Ultima Online Gold Demo works. This demo is a representation of the rule set from the Ultima Online Second Age Era.

There is proof that some people have already reversed this demo partially or as a whole, however so far no tools or knowledge has been published.  This project is to overcome does shortcomings.

URL's with some proof for this:
http://www.runuo.com/forums/general-discussion/94767-help-m-files.html
http://azaroth.org/2008/12/31/your-topic/ (posting by Faust)

If we understand the demo there is a big chance we can alter the demo and even create our own demo. By default mounting horses is not possible in the demo, but what if we can alter the demo and unlock horses; can we then see how horses behaved during T2A?

This demo is 10 years old and I do not understand no one published his/her work. Maybe that DMCA thing is in the way?

## UTILITIES USED

IDA Pro, a very professional utility, definitely worth buying, Standard version is affordable.
BRAIN, it looks ugly on pictures but it's a beautiful device in the end

## ABOUT ME

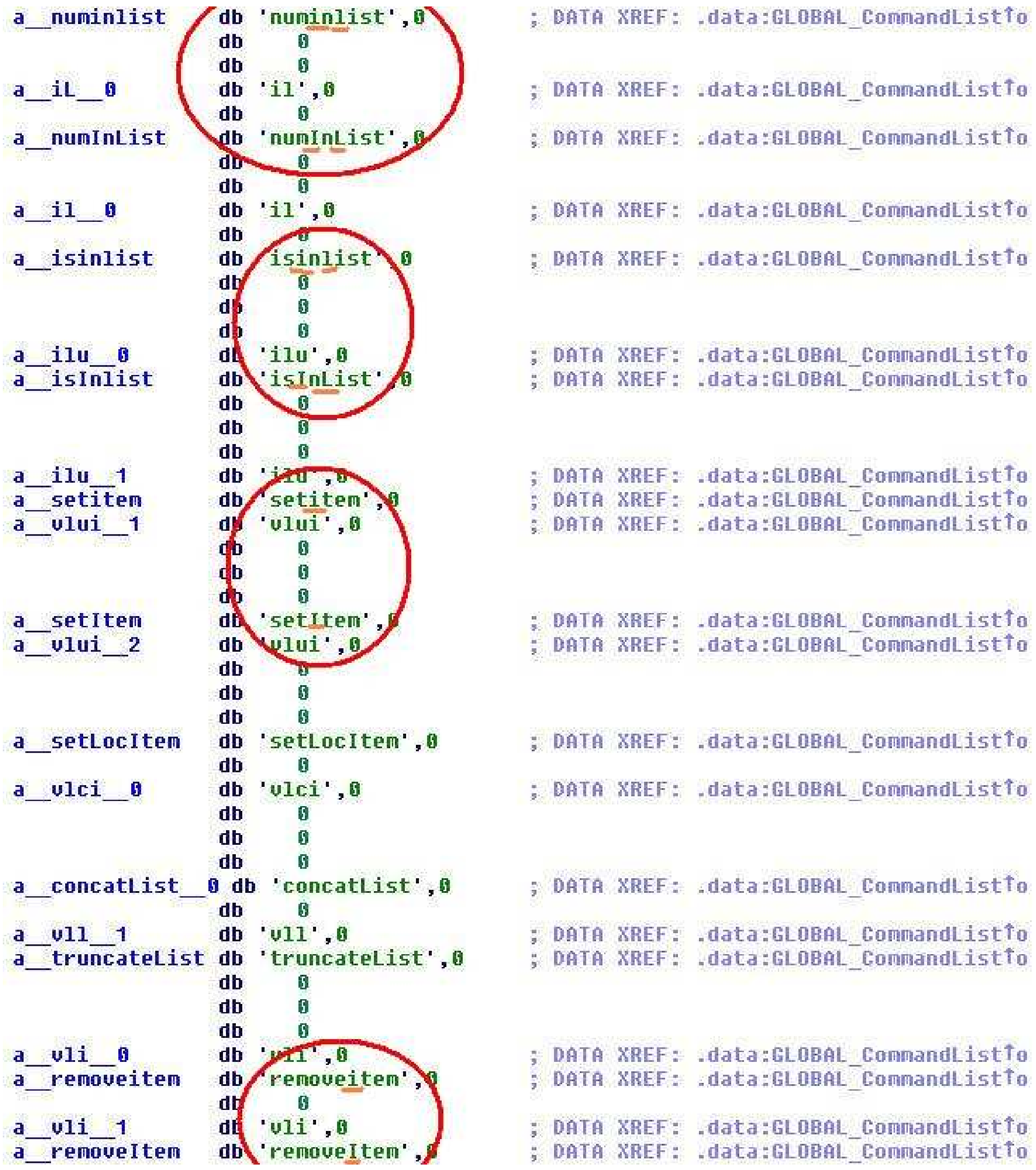
I'm just a guy who loves the Ultima universe and knows a bit assembler.  Why not combine the two? ☺  Back in the days, my first virus infection was by the tequila virus.  It didn't last long though, I noticed my C++ programs didn't start like I expected them to do, so I opened my EXEs with Turbo Debugger and I saw in horror my own programs were being messed with_  I should still have the infected base image (password protected) somewhere on my hard disk (ARJ.EXE).  If you never heard of the tequila virus then go to Google.  It was one of the first viruses to use a polymorphic engine.  A difficult term; but do some research please, pretty cool stuff in there.  Viruses; everyone hates them but you should study them if you're into reverse code engineering.  Much to be learned, even from old virii!  And when you're at it, also take a look at TBSCAN; it was the first scanner with a new approach to tackle to difficulties created by polymorphic viruses and I'll implement a similar approach for creating a universal patcher for the Ultima Online clients.

## CONTINUATION

In Part 1 the COMMAND_GlobalList became visible and in Part 2 the parameter passing was discovered. In this 3rd part I'll show you some interesting things found.

## FUNCTION CASING

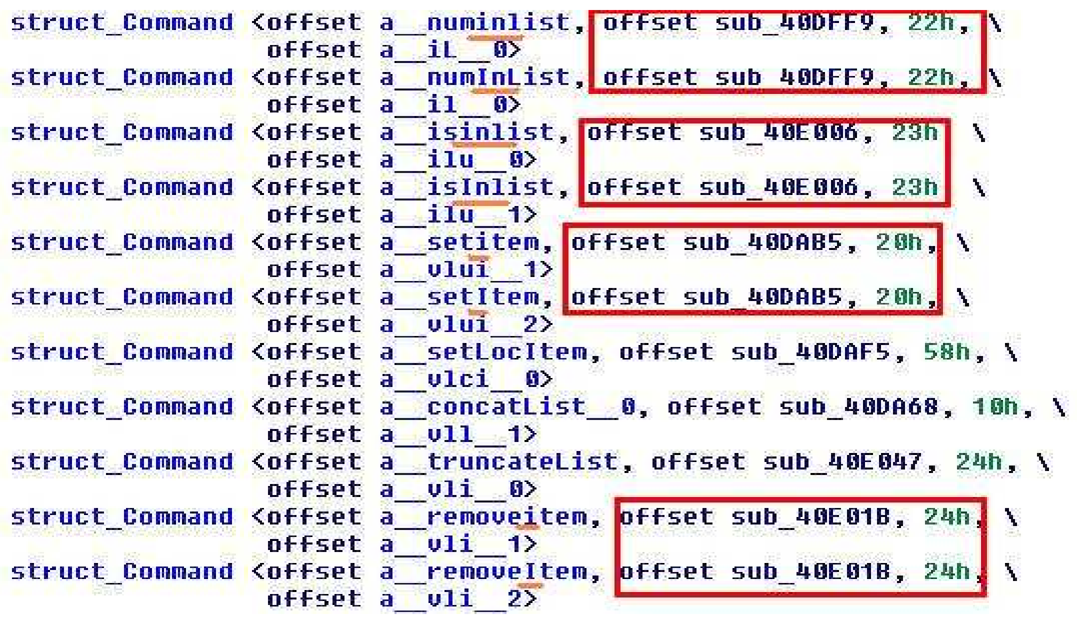
Take a look at this screenshot; it's in de data area where the function names used by the GLOBAL_CommandList are stored:



```
a__numinlist    db 'numinlist',0       ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
a__iL__0        db 'iL',0              ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
a__numInList    db 'numInList',0       ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
a__il__0        db 'il',0              ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
a__isinlist     db 'isinlist',0        ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
                db    0
a__ilu__0       db 'ilu',0             ; DATA XREF: .data:GLOBAL_CommandList↑o
a__isInlist     db 'isInList',0        ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
                db    0
a__ilu__1       db 'ilu',0             ; DATA XREF: .data:GLOBAL_CommandList↑o
a__setitem      db 'setitem',0         ; DATA XREF: .data:GLOBAL_CommandList↑o
a__vlui__1      db 'vlui',0            ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
                db    0
a__setItem      db 'setItem',0         ; DATA XREF: .data:GLOBAL_CommandList↑o
a__vlui__2      db 'vlui',0            ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
                db    0
a__setLocItem   db 'setLocItem',0      ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
a__vlci__0      db 'vlci',0            ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
                db    0
a__concatList__0 db 'concatList',0     ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
a__vll__1       db 'vll',0             ; DATA XREF: .data:GLOBAL_CommandList↑o
a__truncateList db 'truncateList',0    ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
                db    0
                db    0
a__vli__0       db 'vli',0             ; DATA XREF: .data:GLOBAL_CommandList↑o
a__removeitem   db 'removeitem',0      ; DATA XREF: .data:GLOBAL_CommandList↑o
                db    0
a__vli__1       db 'vli',0             ; DATA XREF: .data:GLOBAL_CommandList↑o
a__removeItem   db 'removeItem',0      ; DATA XREF: .data:GLOBAL_CommandList↑o
```

Many functions with the same name but only the casing is different. Does this mean that each casing has a different implementation?

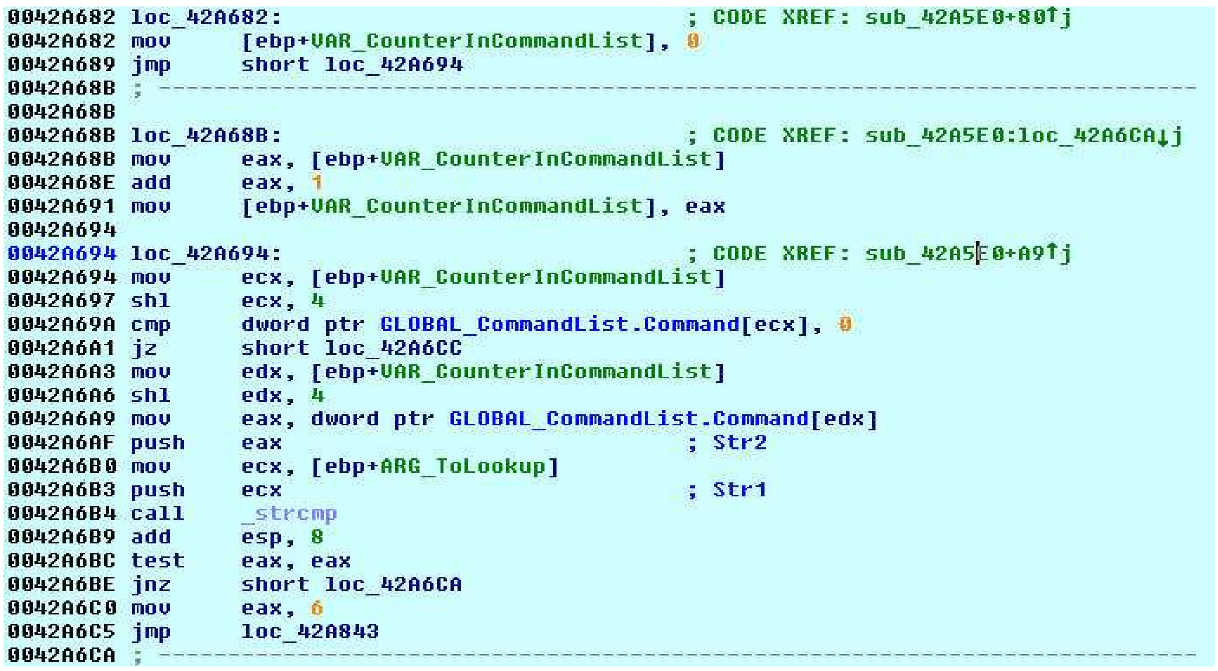
This is a screenshot from the GLOBAL_CommandList itself:



```
struct_Command <offset a__numinlist, offset sub_40DFF9, 22h, \
                offset a__iL__0>
struct_Command <offset a__numInList, offset sub_40DFF9, 22h, \
                offset a__il__0>
struct_Command <offset a__isinlist, offset sub_40E006, 23h  \
                offset a__ilu__0>
struct_Command <offset a__isInlist, offset sub_40E006, 23h  \
                offset a__ilu__1>
struct_Command <offset a__setitem, offset sub_40DAB5, 20h, \
                offset a__vlui__1>
struct_Command <offset a__setItem, offset sub_40DAB5, 20h, \
                offset a__vlui__2>
struct_Command <offset a__setLocItem, offset sub_40DAF5, 58h, \
                offset a__vlci__0>
struct_Command <offset a__concatList__0, offset sub_40DA68, 10h, \
                offset a__vll__1>
struct_Command <offset a__truncateList, offset sub_40E047, 24h, \
                offset a__vli__0>
struct_Command <offset a__removeitem, offset sub_40E01B, 24h, \
                offset a__vli__1>
struct_Command <offset a__removeItem, offset sub_40E01B, 24h, \
                offset a__vli__2>
```

Even though that the casing is different the underlying function calls are the same! This can mean that the scripting language isn't case-sensitive.

But, let's take a look somewhere deeper into the code where the GLOBAL_CommandList is being referenced/used:



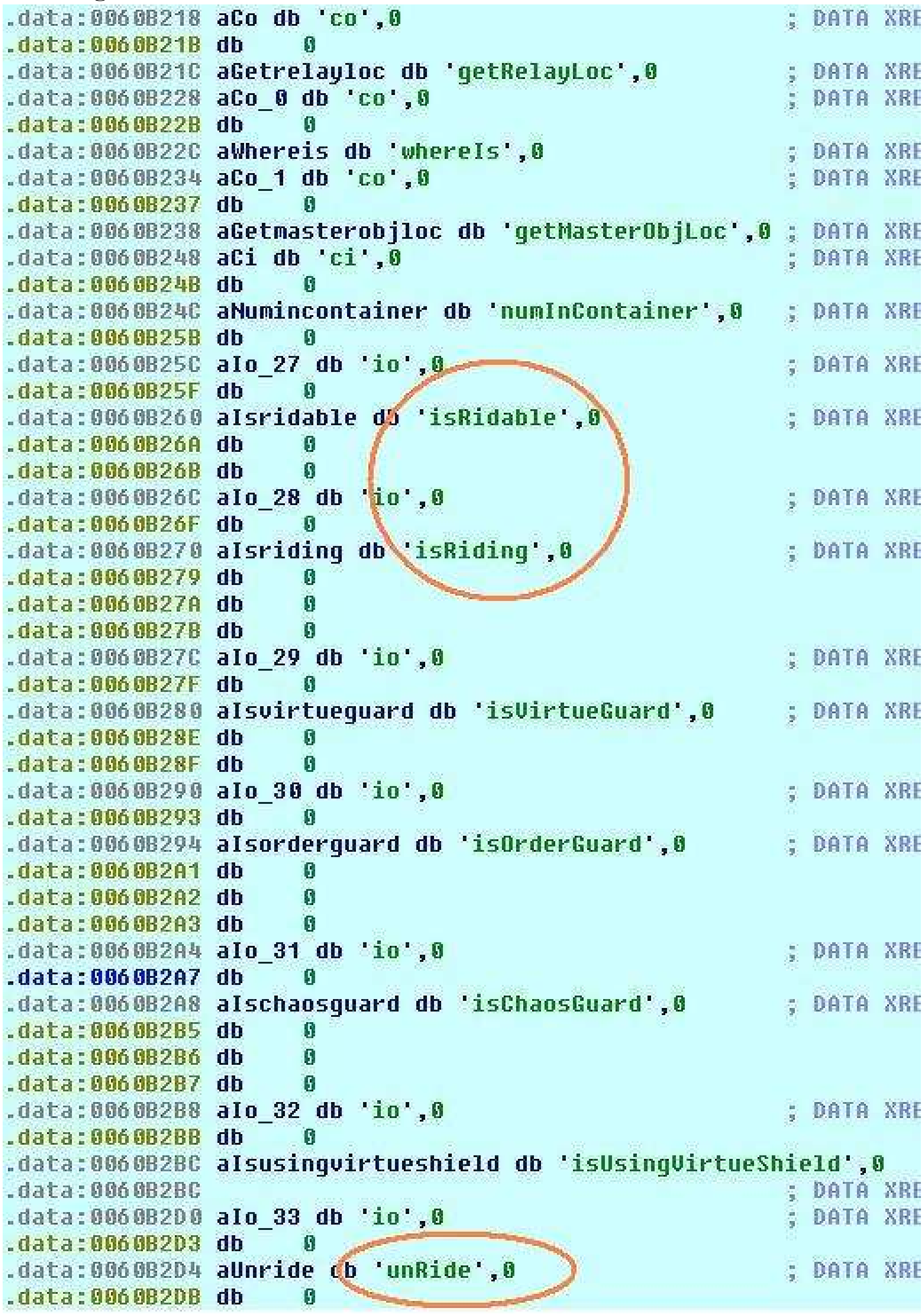
```
0042A682 loc_42A682:                            ; CODE XREF: sub_42A5E0+80↑j
0042A682 mov     [ebp+VAR_CounterInCommandList], 0
0042A689 jmp     short loc_42A694
0042A68B ; ---------------------------------------------------------------------------
0042A68B
0042A68B loc_42A68B:                            ; CODE XREF: sub_42A5E0:loc_42A6CA↓j
0042A68B mov     eax, [ebp+VAR_CounterInCommandList]
0042A68E add     eax, 1
0042A691 mov     [ebp+VAR_CounterInCommandList], eax
0042A694
0042A694 loc_42A694:                            ; CODE XREF: sub_42A5E0+A9↑j
0042A694 mov     ecx, [ebp+VAR_CounterInCommandList]
0042A697 shl     ecx, 4
0042A69A cmp     dword ptr GLOBAL_CommandList.Command[ecx], 0
0042A6A1 jz      short loc_42A6CC
0042A6A3 mov     edx, [ebp+VAR_CounterInCommandList]
0042A6A6 shl     edx, 4
0042A6A9 mov     eax, dword ptr GLOBAL_CommandList.Command[edx]
0042A6AF push    eax                            ; Str2
0042A6B0 mov     ecx, [ebp+ARG_ToLookup]
0042A6B3 push    ecx                            ; Str1
0042A6B4 call    _strcmp
0042A6B9 add     esp, 8
0042A6BC test    eax, eax
0042A6BE jnz     short loc_42A6CA
0042A6C0 mov     eax, 6
0042A6C5 jmp     loc_42A843
0042A6CA ; ---------------------------------------------------------------------------
```

The C function used here for string comparison is strcmp and NOT strcmpi (nor stricmp). This is important because it does mean that the functions are looked up in the array with case-sensitivity. So, somehow, some functions can be written with different casing, they do the same thing but you cannot apply other casing forms other than what OSI prepared.

## RANDOM SCREENSHOTS

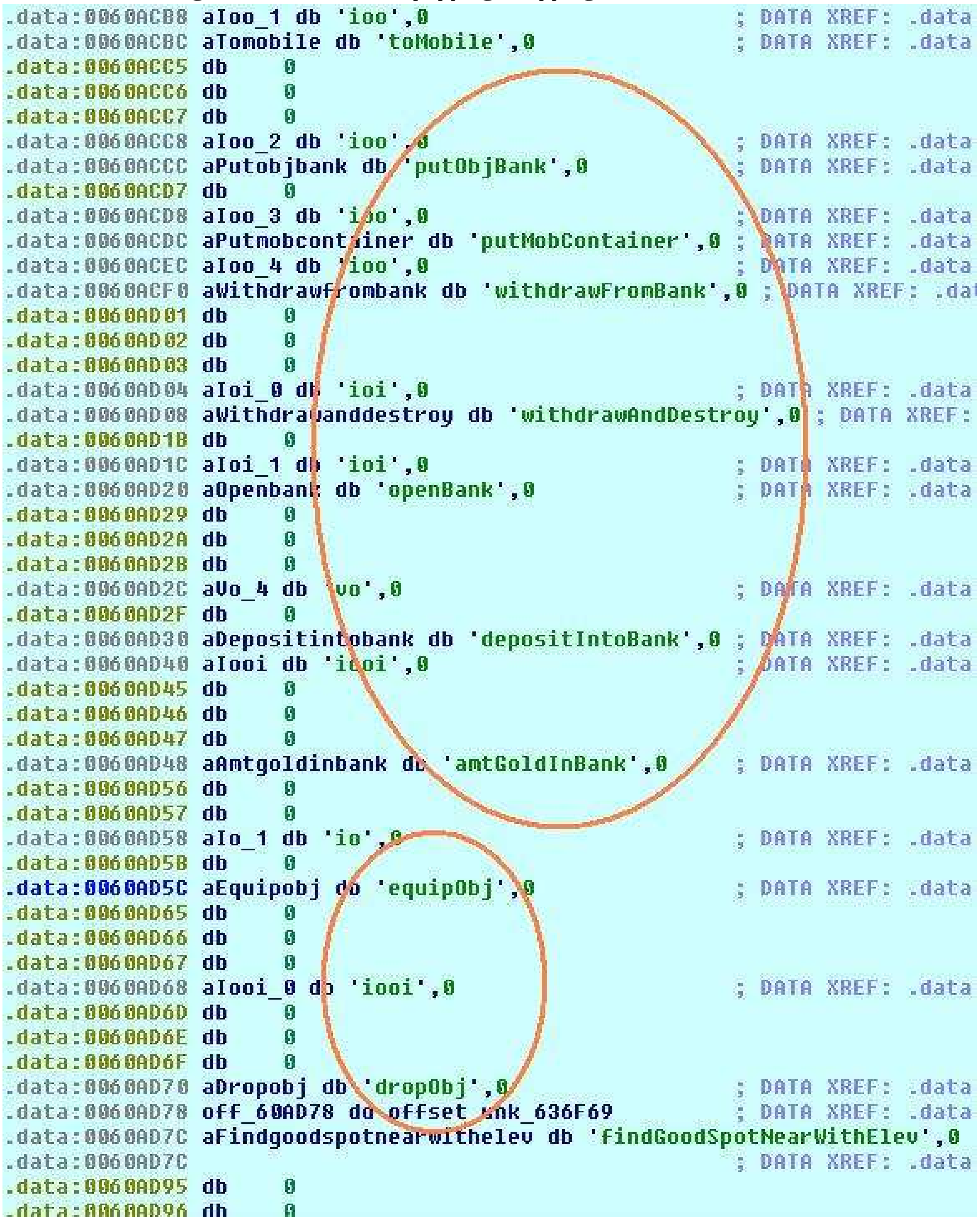
In this section I will just put some random screenshots that should waken your interest.

Mounting?



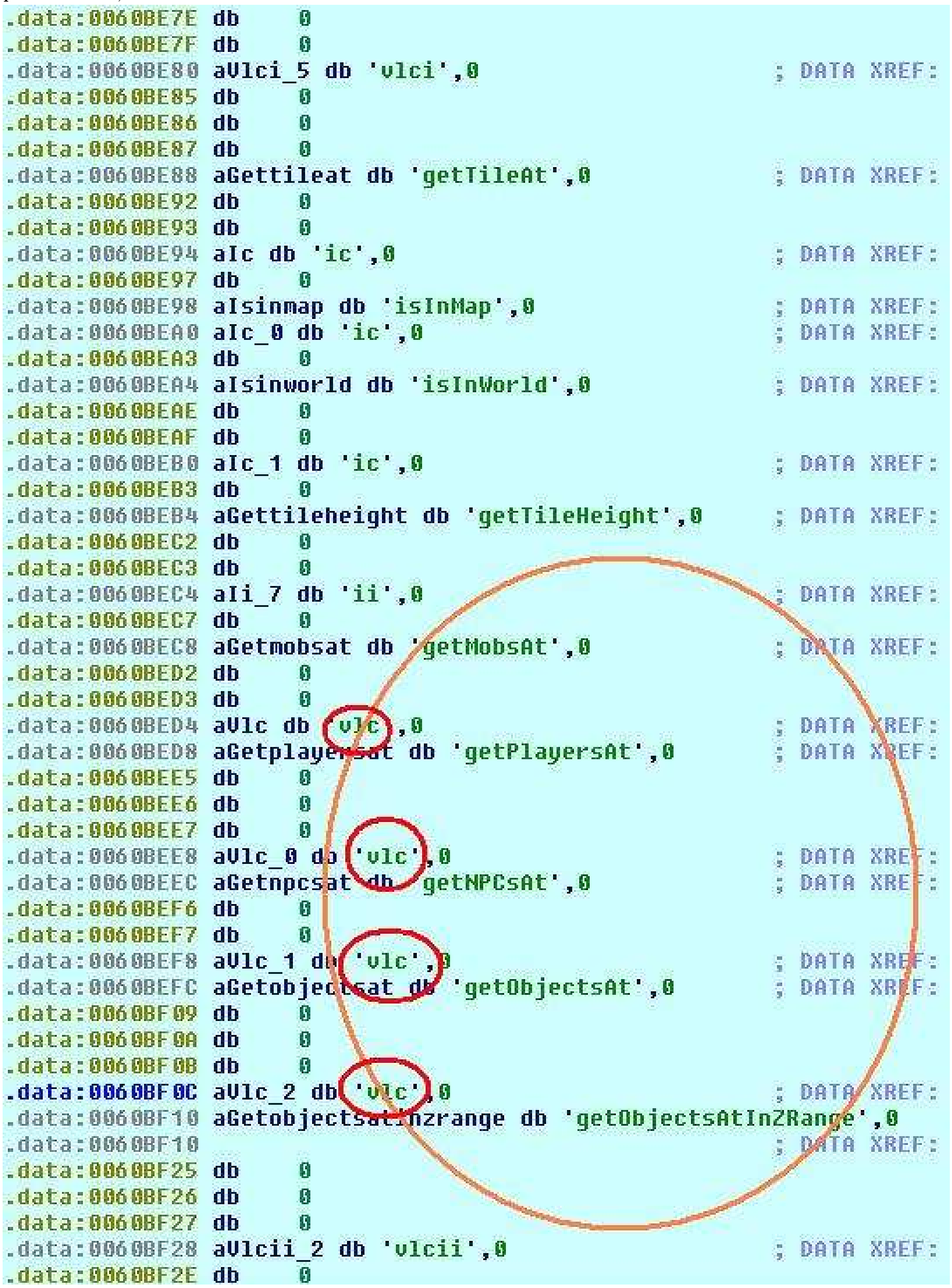
```
.data:0060B218 aCo db 'co',0                                  ; DATA XRE
.data:0060B21B db    0
.data:0060B21C aGetrelayloc db 'getRelayLoc',0                ; DATA XRE
.data:0060B228 aCo_0 db 'co',0                                ; DATA XRE
.data:0060B22B db    0
.data:0060B22C aWhereis db 'whereIs',0                        ; DATA XRE
.data:0060B234 aCo_1 db 'co',0                                ; DATA XRE
.data:0060B237 db    0
.data:0060B238 aGetmasterobjloc db 'getMasterObjLoc',0 ; DATA XRE
.data:0060B248 aCi db 'ci',0                                  ; DATA XRE
.data:0060B24B db    0
.data:0060B24C aNumincontainer db 'numInContainer',0          ; DATA XRE
.data:0060B25B db    0
.data:0060B25C aIo_27 db 'io',0                               ; DATA XRE
.data:0060B25F db    0
.data:0060B260 aIsridable db 'isRidable',0                    ; DATA XRE
.data:0060B26A db    0
.data:0060B26B db    0
.data:0060B26C aIo_28 db 'io',0                               ; DATA XRE
.data:0060B26F db    0
.data:0060B270 aIsriding db 'isRiding',0                      ; DATA XRE
.data:0060B279 db    0
.data:0060B27A db    0
.data:0060B27B db    0
.data:0060B27C aIo_29 db 'io',0                               ; DATA XRE
.data:0060B27F db    0
.data:0060B280 aIsvirtueguard db 'isVirtueGuard',0            ; DATA XRE
.data:0060B28E db    0
.data:0060B28F db    0
.data:0060B290 aIo_30 db 'io',0                               ; DATA XRE
.data:0060B293 db    0
.data:0060B294 aIsorderguard db 'isOrderGuard',0              ; DATA XRE
.data:0060B2A1 db    0
.data:0060B2A2 db    0
.data:0060B2A3 db    0
.data:0060B2A4 aIo_31 db 'io',0                               ; DATA XRE
.data:0060B2A7 db    0
.data:0060B2A8 aIschaosguard db 'isChaosGuard',0              ; DATA XRE
.data:0060B2B5 db    0
.data:0060B2B6 db    0
.data:0060B2B7 db    0
.data:0060B2B8 aIo_32 db 'io',0                               ; DATA XRE
.data:0060B2BB db    0
.data:0060B2BC aIsusingvirtueshield db 'isUsingVirtueShield',0
.data:0060B2BC                                                ; DATA XRE
.data:0060B2D0 aIo_33 db 'io',0                               ; DATA XRE
.data:0060B2D3 db    0
.data:0060B2D4 aUnride db 'unRide',0                          ; DATA XRE
.data:0060B2DB db    0
```

Hints for reversing bank access and equipping/dropping:



```
.data:0060ACB8 aIoo_1 db 'ioo',0                          ; DATA XREF: .data
.data:0060ACBC aTomobile db 'toMobile',0                  ; DATA XREF: .data
.data:0060ACC5 db    0
.data:0060ACC6 db    0
.data:0060ACC7 db    0
.data:0060ACC8 aIoo_2 db 'ioo',0                          ; DATA XREF: .data
.data:0060ACCC aPutobjbank db 'putObjBank',0              ; DATA XREF: .data
.data:0060ACD7 db    0
.data:0060ACD8 aIoo_3 db 'ioo',0                          ; DATA XREF: .data
.data:0060ACDC aPutmobcontainer db 'putMobContainer',0 ; DATA XREF: .data
.data:0060ACEC aIoo_4 db 'ioo',0                          ; DATA XREF: .data
.data:0060ACF0 aWithdrawfrombank db 'withdrawFromBank',0 ; DATA XREF: .dat
.data:0060AD01 db    0
.data:0060AD02 db    0
.data:0060AD03 db    0
.data:0060AD04 aIoi_0 db 'ioi',0                          ; DATA XREF: .data
.data:0060AD08 aWithdrawanddestroy db 'withdrawAndDestroy',0 ; DATA XREF:
.data:0060AD1B db    0
.data:0060AD1C aIoi_1 db 'ioi',0                          ; DATA XREF: .data
.data:0060AD20 aOpenbank db 'openBank',0                  ; DATA XREF: .data
.data:0060AD29 db    0
.data:0060AD2A db    0
.data:0060AD2B db    0
.data:0060AD2C aVo_4 db 'vo',0                            ; DATA XREF: .data
.data:0060AD2F db    0
.data:0060AD30 aDepositintobank db 'depositIntoBank',0 ; DATA XREF: .data
.data:0060AD40 aIooi db 'iooi',0                          ; DATA XREF: .data
.data:0060AD45 db    0
.data:0060AD46 db    0
.data:0060AD47 db    0
.data:0060AD48 aAmtgoldinbank db 'amtGoldInBank',0        ; DATA XREF: .data
.data:0060AD56 db    0
.data:0060AD57 db    0
.data:0060AD58 aIo_1 db 'io',0                            ; DATA XREF: .data
.data:0060AD5B db    0
.data:0060AD5C aEquipobj db 'equipObj',0                  ; DATA XREF: .data
.data:0060AD65 db    0
.data:0060AD66 db    0
.data:0060AD67 db    0
.data:0060AD68 aIooi_0 db 'iooi',0                        ; DATA XREF: .data
.data:0060AD6D db    0
.data:0060AD6E db    0
.data:0060AD6F db    0
.data:0060AD70 aDropobj db 'dropObj',0                    ; DATA XREF: .data
.data:0060AD78 off_60AD78 dd offset unk_636F69            ; DATA XREF: .data
.data:0060AD7C aFindgoodspotnearwithelev db 'findGoodSpotNearWithElev',0
.data:0060AD7C                                            ; DATA XREF: .data
.data:0060AD95 db    0
.data:0060AD96 db    0
```

These functions return lists of objects, but they are defined void and the list is passed as a parameter, remember “c” stands for location:



```
.data:0060BE7E db    0
.data:0060BE7F db    0
.data:0060BE80 aVlci_5 db 'vlci',0                   ; DATA XREF:
.data:0060BE85 db    0
.data:0060BE86 db    0
.data:0060BE87 db    0
.data:0060BE88 aGettileat db 'getTileAt',0           ; DATA XREF:
.data:0060BE92 db    0
.data:0060BE93 db    0
.data:0060BE94 aIc db 'ic',0                         ; DATA XREF:
.data:0060BE97 db    0
.data:0060BE98 aIsinmap db 'isInMap',0               ; DATA XREF:
.data:0060BEA0 aIc_0 db 'ic',0                       ; DATA XREF:
.data:0060BEA3 db    0
.data:0060BEA4 aIsinworld db 'isInWorld',0           ; DATA XREF:
.data:0060BEAE db    0
.data:0060BEAF db    0
.data:0060BEB0 aIc_1 db 'ic',0                       ; DATA XREF:
.data:0060BEB3 db    0
.data:0060BEB4 aGettileheight db 'getTileHeight',0   ; DATA XREF:
.data:0060BEC2 db    0
.data:0060BEC3 db    0
.data:0060BEC4 aIi_7 db 'ii',0                       ; DATA XREF:
.data:0060BEC7 db    0
.data:0060BEC8 aGetmobsat db 'getMobsAt',0           ; DATA XREF:
.data:0060BED2 db    0
.data:0060BED3 db    0
.data:0060BED4 aVlc db 'vlc',0                       ; DATA XREF:
.data:0060BED8 aGetplayersat db 'getPlayersAt',0     ; DATA XREF:
.data:0060BEE5 db    0
.data:0060BEE6 db    0
.data:0060BEE7 db    0
.data:0060BEE8 aVlc_0 db 'vlc',0                     ; DATA XREF:
.data:0060BEEC aGetnpcsat db 'getNPCsAt',0           ; DATA XREF:
.data:0060BEF6 db    0
.data:0060BEF7 db    0
.data:0060BEF8 aVlc_1 db 'vlc',0                     ; DATA XREF:
.data:0060BEFC aGetobjectsat db 'getObjectsAt',0     ; DATA XREF:
.data:0060BF09 db    0
.data:0060BF0A db    0
.data:0060BF0B db    0
.data:0060BF0C aVlc_2 db 'vlc',0                     ; DATA XREF:
.data:0060BF10 aGetobjectsatinzrange db 'getObjectsAtInZRange',0
.data:0060BF10                                       ; DATA XREF:
.data:0060BF25 db    0
.data:0060BF26 db    0
.data:0060BF27 db    0
.data:0060BF28 aVlcii_2 db 'vlcii',0                 ; DATA XREF:
.data:0060BF2E db    0
```