

# GOLDEN EGGS COLLECTION WRITEUP

This was the result I found by searching golden eggs in qdb.

[illegible]

so there are 9 goldeneggs which are subclasses of itempickup which is a subclass of actor, which matches with the hint given.

I have found a global variable `g_eggs` too, but couldn't find any data in it.

Then in the GameWorld object there is a function that lists the actors, using this function I was able to get the coordinates and names of all the objects currently, by typing a in the chat.

And the output was something like this

```
sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3...
Using binred.
F:\HaloCrash\overhead\ts\3780608 bytes
4.6.0-0+UE4 7038 3077 413 0
./src/intel/isl/isl.c:2105: FINISHME: ../src/intel/isl/isl.c:isl_surf_supports_ccs: CCS for 3D textures is disabled, but a workaround is available.

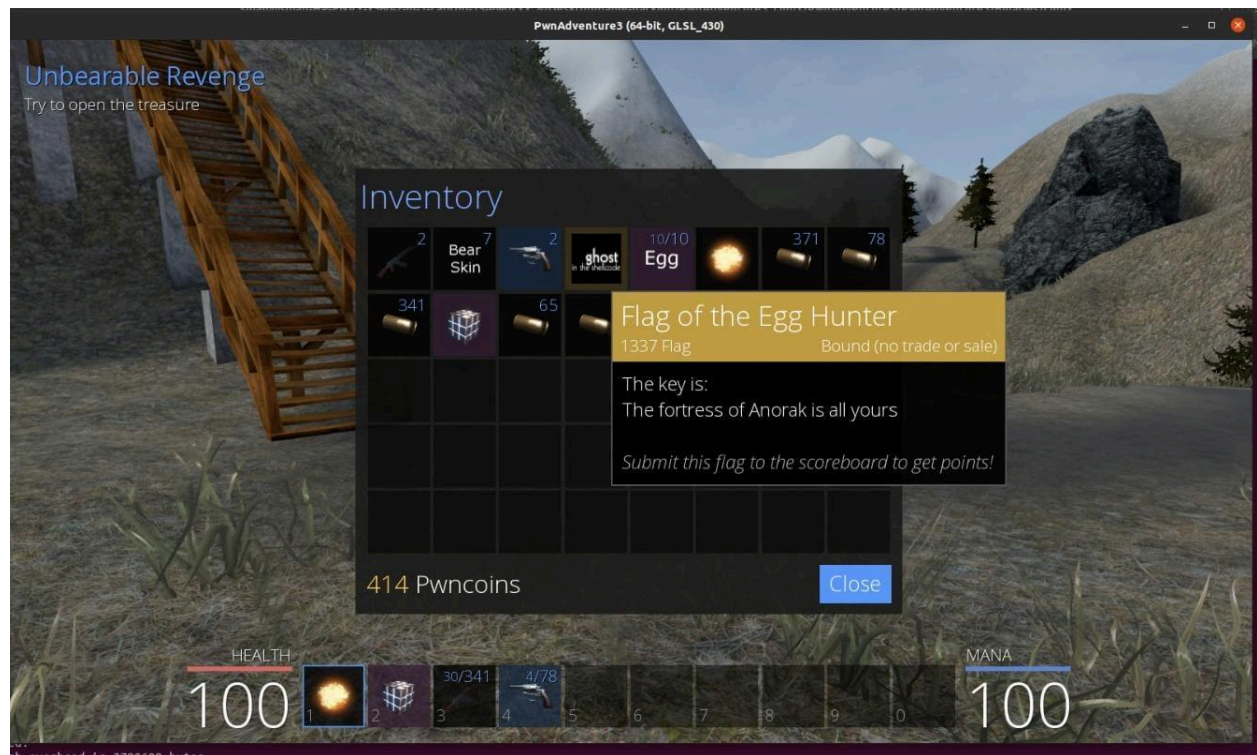
-10088.1 -9173.83 10129.1
65225 -5740 4928
-72667 -53567 1645
1522 14966 7022
Justin Tolerable
-41084 -16256 2270
-7894 64482 2663
-53539 -44246 358
252920 -245380 1170
50076 -5243 1523
-43655 -55820 322
60453 -17409 2939
Sum Ting Wong
21553 41232 2133
Michael Angelo
200255 -248555 1415
48404 28117 704
Major Payne
-37463 -18050 2416
11604 -13131 411
-3055 23005 2275
-25045 18085 260
-51570 -61215 5020
24512 69682 2659
-2778 -11035 10504
-6101 -10956 10636

AL lib: (WW) FreeDevice: (0xd9d7010) Deleting 2 Buffer(s)
```

Then I tp'd to all these and found 9 golden eggs and for then I used radare2 and searched for any symbols named egg,when I have found BallmerPeakEgg and poster and found a function related to damage to the poster With only fireball and ak i tried to shoot it, but it was not succesful Then on analysis of the Damage function in the poster object, I have found this string named “cowboy coder”

```
sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3... sujan@sujan-HP-ENVY-x3...
0x001bf624 [x86C]0 0x 265 ntlbGameLogic.so> a fs:pd $r @ sym.BallmerPeakPoster::Damage_IActor_IItem _int_DamageType_+4 # 0x1bf624
old syn.BallmerPeakPoster::Damage_IActor_IItem _int_DamageType_ (int64_t arg1, char *arg2, uint32_t arg3, int64_t arg4, int64_t arg5);
0x001bf624 48897dfe mov qword [var_10h], rdi ; arg1
0x001bf626 48897dfe mov qword [var_10h], rsi ; arg2
0x001bf628 48897dfe mov qword [var_10h], rdx ; arg3
0x001bf62a 48897dfe mov qword [var_10h], ecx ; arg4
0x001bf62c 48897dfe mov qword [var_20h], rax ; arg5
0x001bf62e 48897dfe cmp qword [var_10h], 0 ; BallmerPeakPoster.cpp:19
jnz 0x001bf643 0f8505000000 jne 0x1bf64e ; BallmerPeakPoster.cpp:10
0x001bf62f 48897dfe jmp 0x1bf741 ; BallmerPeakPoster.cpp:10
[ ] ; CODE XREF from BallmerPeakPoster::Damage_IActor_IItem* (int, DamageType) @ 0x1bf624(x)
-> 0x001bf630 488b45f0 mov rax, qword [var_10h] ; BallmerPeakPoster.cpp:11
0x001bf632 488b08 mov rcx, qword [rax]
0x001bf634 4889c7 mov rdi, rax
0x001bf636 4889c7 call qword [rcx + 0x20] ; [1] ; fcn.00000020 ; "g"
0x001bf638 4801 test al, 1
jnz 0x001bf65d 0f8505000000 jne 0x1bf668 ; BallmerPeakPoster.cpp:12
0x001bf639 48897dfe jmp 0x1bf741 ; BallmerPeakPoster.cpp:12
[ ] ; CODE XREF from BallmerPeakPoster::Damage_IActor_IItem* (int, DamageType) @ 0x1bf630(x)
-> 0x001bf63a 488b45f0 mov rax, qword [var_10h] ; BallmerPeakPoster.cpp:13
jnz 0x001bf670 0f8505000000 jne 0x1bf67b ; BallmerPeakPoster.cpp:15
0x001bf63b 48897dfe jmp 0x1bf741 ; BallmerPeakPoster.cpp:16
[ ] ; CODE XREF from BallmerPeakPoster::Damage_IActor_IItem* (int, DamageType) @ 0x1bf63a(x)
-> 0x001bf63c 488b45f0 mov rax, qword [var_10h] ; BallmerPeakPoster.cpp:17
0x001bf63e 488b08 mov rcx, qword [rax]
0x001bf640 488b4910 mov rcx, qword [rcx + 0x10]
0x001bf642 4889c7 mov rdi, rax
0x001bf644 4889c7 call rcx
0x001bf646 488d4d0 lea rcx, [var_30h]
0x001bf648 4889cf mov rdi, rcx
0x001bf64a 488945b0 mov qword [var_50h], rax
0x001bf64c 48894d0 mov qword [var_50h], rcx
0x001bf64e 48b138f6f call sym.std::allocator<char>::allocator() ; [2] ; std::allocator<char>::allocator()
0x001bf650 488d7dd8 lea rdi, [var_20h]
0x001bf652 488b75b0 mov rsi, qword [var_50h]
0x001bf654 488b550 mov rdx, qword [var_50h]
0x001bf656 488b0f5f call sym.std::basic_string<char>::basic_string(char const*, std::allocator<char>::allocator()) ; [3]
jnz 0x001bf6b5 0f8505000000 jne 0x1bf6b5 ; BallmerPeakPoster.cpp:18
[ ] ; CODE XREF from BallmerPeakPoster::Damage_IActor_IItem* (int, DamageType) @ 0x1bf656(x)
-> 0x001bf657 488d7dd8 lea rdi, [var_20h]
0x001bf659 488b3906f lea rsi, str.CowboyCoder ; 0x329d76 ; "CowboyCoder"
0x001bf65b 488b0f5f call fcn.0010c7b0 ; [4]
0x001bf65d 8845a7 mov byte [var_50h], al
jnz 0x001bf6c8 0f8505000000 jne 0x1bf6cd ; BallmerPeakPoster.cpp:19
[ ] ; CODE XREF from BallmerPeakPoster::Damage_IActor_IItem* (int, DamageType) @ 0x1bf65d(x)
-> 0x001bf65e 488d7dd8 lea rdi, [var_20h]
0x001bf660 488d7dd8 call sym.std::basic_string<char>::basic_string(char, std::allocator<char>::allocator()) ; [5] ; std::basic_string<char>::basic_string(char const*, std::allocator<char>::allocator())
0x001bf662 488d7dd8 lea rdi, [var_30h]
0x001bf664 488b43f5f call fcn.0010c7b0 ; [6]
0x001bf666 8845a7 mov al, byte [var_50h]
0x001bf668 4801 test al, 1
jnz 0x001bf6e4 0f8505000000 jne 0x1bf6ef ; BallmerPeakPoster.cpp:20
```

With a bit more analysis, i found out its a gun and after murdering a few bears for the skins, i bought it and shot the poster with it, and then found the final egg right outside the house.



With this I have achieved my first flag