

c#游戏题目（其实之前也做过类似的，算是重新体验一波c#）

| | | | |
|---|----------------|--------|-----------|
|  Mono | 2019/11/7 0:51 | 文件夹 | |
|  Snake_Data | 2019/11/7 0:51 | 文件夹 | |
|  Snake.exe | 2019/11/7 0:51 | 应用程序 | 636 KB |
|  Snake.exe.i64 | 2021/9/5 10:08 | i64 文件 | 1,432 KB |
|  UnityCrashHandler64.exe | 2019/3/5 18:00 | 应用程序 | 1,424 KB |
|  UnityCrashHandler64.exe.i64 | 2021/9/5 10:09 | i64 文件 | 8,672 KB |
|  UnityPlayer.dll | 2019/3/5 18:00 | 应用程序扩展 | 22,352 KB |
|  WinPixEventRuntime.dll | 2019/3/5 17:54 | 应用程序扩展 | 42 KB |

首先看这个unity，可以知道是c#题目

tips1






（unity在打包后，会将所有的代码打进一个Assembly-CSharp.dll的文件里面，通过这个文件的反编译，就是详细看见里面的代码内容）

直接搜Assembly-CSharp.dll

Managed

共享 查看

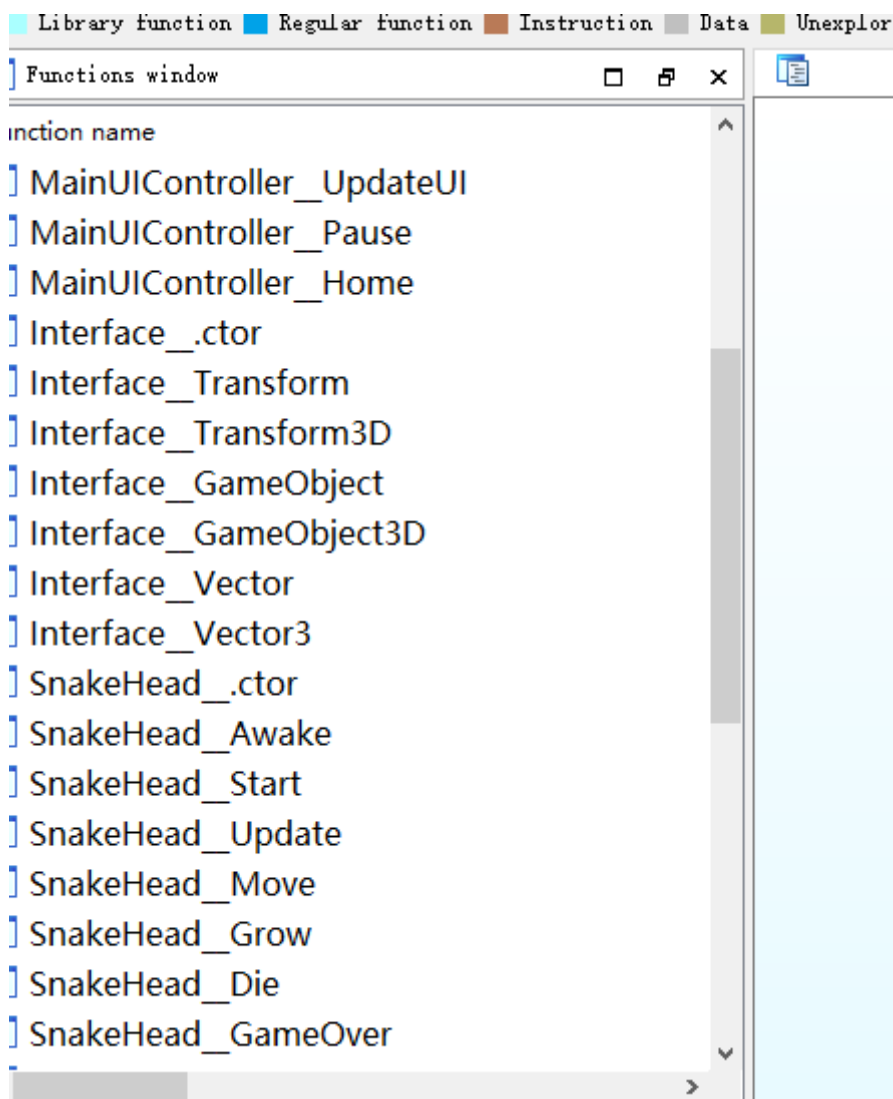
<< [2019红帽杯]Snake > Snake > Snake_Data > Managed

| 名称 | 修改日期 | 类型 |
|---|----------------|--------|
|  Assembly-CSharp.dll | 2019/11/7 0:51 | 应用程序扩展 |
|  Assembly-CSharp.dll.id0 | 2021/9/5 10:32 | ID0 文件 |
|  Assembly-CSharp.dll.id1 | 2021/9/5 10:32 | ID1 文件 |
|  Assembly-CSharp.dll.idb | 2021/9/5 10:28 | IDB 文件 |
|  Assembly-CSharp.dll.nam | 2021/9/5 10:32 | NAM 文件 |

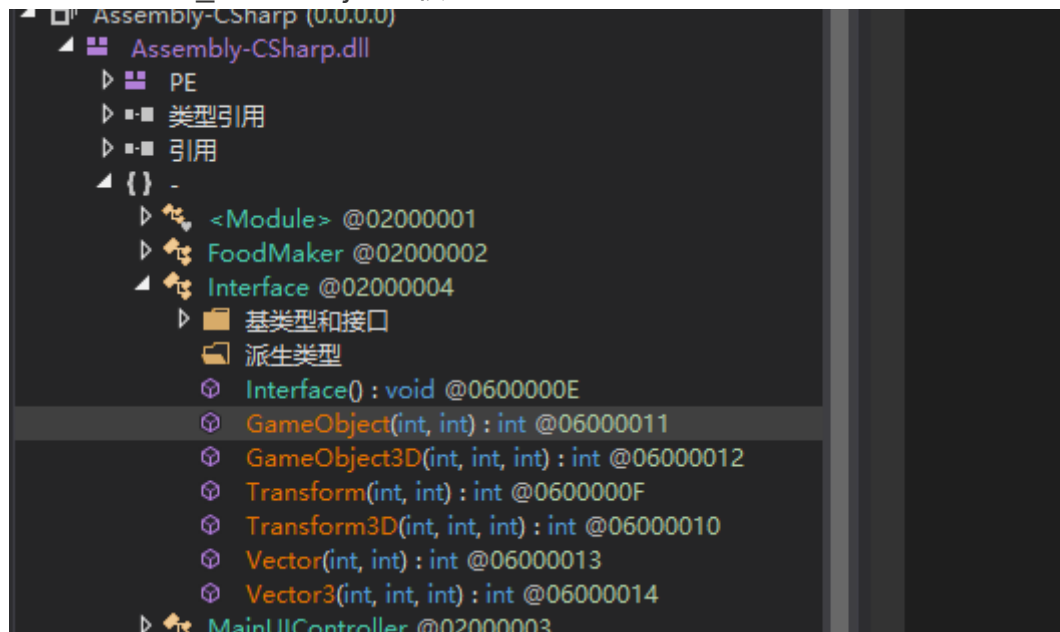
直接拖入ida看名字

tips2

直接拖入ida看是看不到代码细节的（对于c#dll而言），我们主要看得是名字。一般名称都是类型_函数名



看到interface_GameObject函数



```
GameObject(int, int) : int ×
1 // Interface
2 // Token: 0x06000011 RID: 17
3 [DllImport("Interface", CallingConvention = CallingConvention.Cdecl)]
4 public static extern int GameObject(int x, int y);
5
```

看得调用了interface库

```

1 __int64 __fastcall GameObject(int a1)
2 {
3     char v1; // di
4     __int64 *v2; // rbx
5     __int64 *v3; // rax
6     int v4; // er8
7     int v5; // er9
8     __int64 v6; // rax
9     _BYTE *v7; // rcx
10    __int64 v8; // rax
11    __int64 v9; // rax
12    __int64 *v10; // rdx
13    __int64 v11; // rax
14    __int64 *v12; // rcx
15    _BYTE *v13; // rcx
16    __int64 v15; // rax
17    int v16; // er8
18    int v17; // er9
19    __int64 v18; // rax
20    __int64 v19; // rax
21    __int64 *v20; // rdx
22    __int64 v21; // rax
23    __int64 *v22; // rcx
24    _BYTE *v23; // rcx
25    void *v24; // rcx
26    void *v25; // rcx
27    void *v26; // rcx
28    _BYTE *v27; // rcx
29    _BYTE *v28; // rcx
30    __int64 v29; // rax
31    _BYTE *v30; // rcx
32    __int64 v31; // rax
33    const void *v32; // rdx
34    bool v33; // bl

```

可以看到代码很长，但是a1只在0-100范围内（即a1只在0-100时，才有可能跑出正确答案），可以考虑爆破出结果

```
import ctypes
def burst(x):
    dll = ctypes.cdll.LoadLibrary('E:\\python\\virtual_environment\\Interface.dll')
    dll.GameObject(x)

for i in range(100):
    burst(i)
```