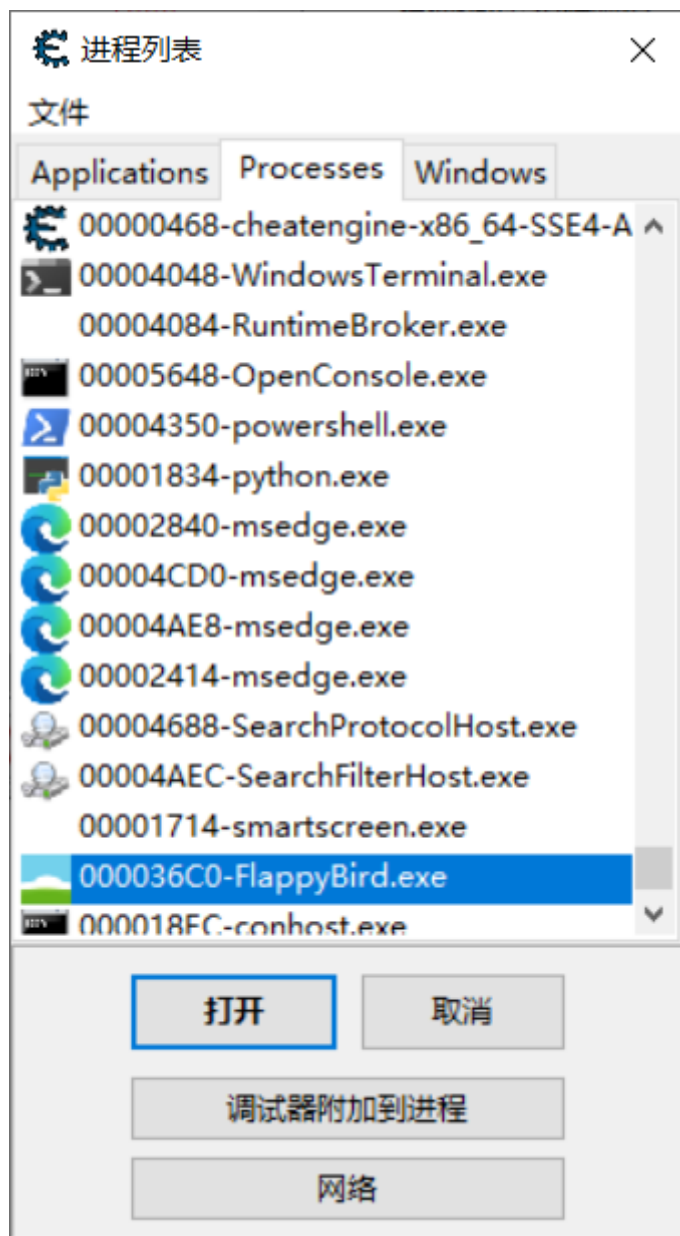


# 菁英班作业第4课

对FlappyBird游戏进行逆向分析，修改当局的分數

## 一、使用CE数据查询手动修改分数

### 1、将CE附加到FlappyBird上



### 2、搜索数值找到存储score的位置

通过score不同值变化进行多次搜索

当前score数值为5




搜索时只有一个结果，即为Score的存储位置

地址	当前值	先前值	First
09C546F4	5	5	0

将Score值修改为1000

激活	描述	地址	类型	数值
<input type="checkbox"/>	无描述	09C546F4	4 字节	1000

附加调试器，判断哪个地方代码访问了该地址

 下列操作码访问了 09C546F4

计.. 指令

替换

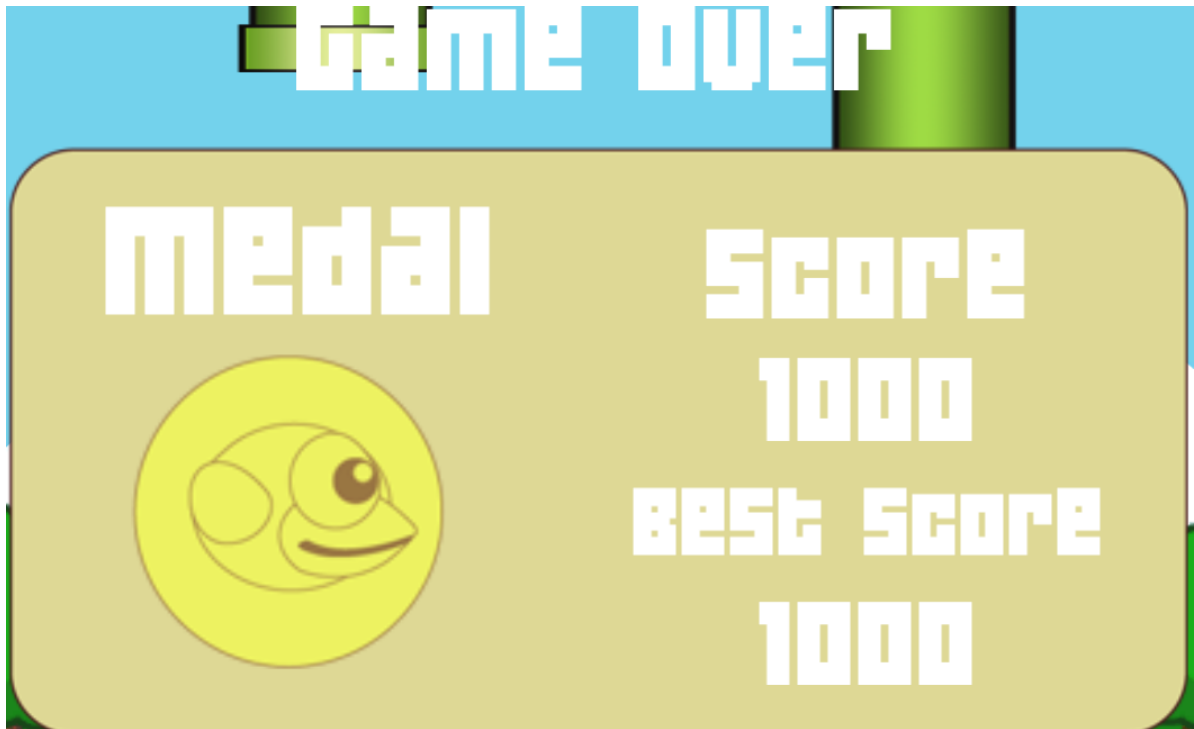
显示反汇编程序

添加到代码表

详细信息

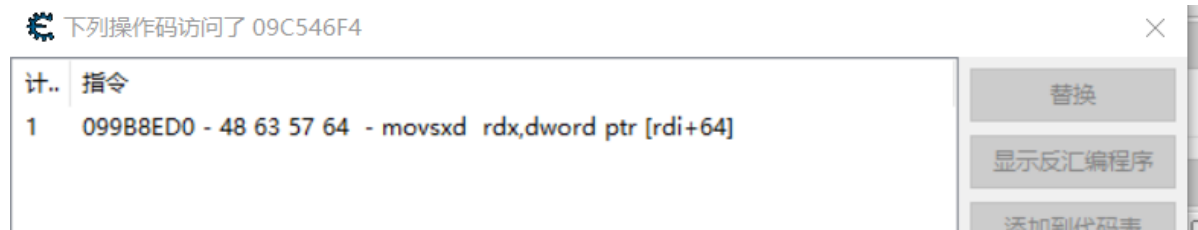
example

修改成功，死亡时分数值为1000。



### 3、固定化Score的位置

调试器发现此处地址0999BED0访问了该数据。

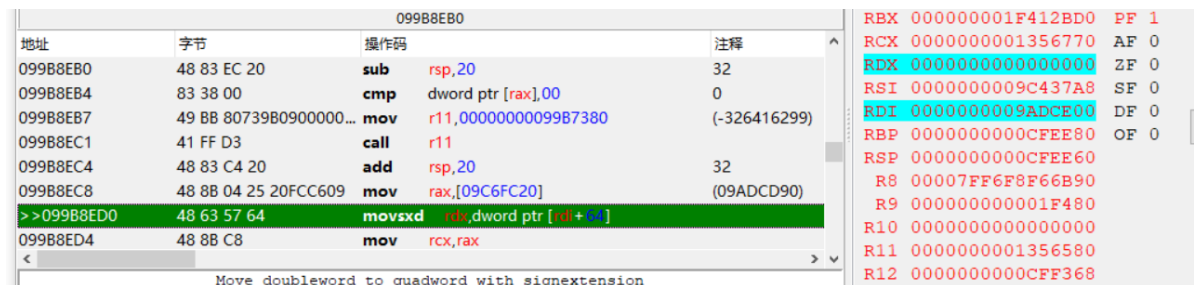


初步判断rdi存放有该对象的基地址，0x64为该score属性偏移地址。



在此位置设置断点，当程序下次访问时，在此暂停。

重开游戏，让小鸟死亡。



显示rdi寄存器为9ADCE00,应为此次游戏小鸟对象的基地址。

在内存空间查看这一片地址。

地址	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
09ADCE00	B0	5F	2C	1F	00	00	00	00	00	00	00	00	00	00	00	00	.....
09ADCE10	C0	15	AF	13	00	00	00	00	70	BE	B0	09	00	00	00	00	.....p.....
09ADCE20	58	BE	B0	09	00	00	00	00	A0	19	CB	09	00	00	00	00	X.....
09ADCE30	40	BE	B0	09	00	00	00	00	08	F7	C9	09	00	00	00	00	@.....
09ADCE40	E0	F6	C9	09	00	00	00	00	B8	F6	C9	09	00	00	00	00	.....
09ADCE50	90	F6	C9	09	00	00	00	00	00	00	40	40	00	00	80	40	.....@..@
09ADCE60	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00	.....
09ADCE70	B0	5F	2C	1F	00	00	00	00	00	00	00	00	00	00	00	00	.....
09ADCE80	40	9E	B7	13	00	00	00	00	30	BF	B0	09	00	00	00	00	@.....0.....
09ADCE90	18	BF	B0	09	00	00	00	00	00	00	00	00	00	00	00	00	.....
09ADCEA0	00	BF	B0	09	00	00	00	00	08	F7	C9	09	00	00	00	00	.....
09ADCEB0	E0	F6	C9	09	00	00	00	00	B8	F6	C9	09	00	00	00	00	.....

9ADCE64位置处恰好为1，验证判断正确。

接下来寻找内存中哪个地方存储了小鸟对象的基地址。

搜索数值为9ADCE00的内存位置

地址	当前值	先前值	First
09A6EB80	09ADCE00	09ADCE00	09ADCE00
09C1E9C0	09ADCE00	09ADCE00	09ADCE00
09C6FC30	09ADCE00	09ADCE00	09ADCE00
13AF15E8	09ADCE00	09ADCE00	09ADCE00

有四处均存储了该数据，先全部添加进备选列表内。

重启游戏，使小鸟Score 为1，暂停。

<input type="checkbox"/>	无描述	09A6EB80	4 字节	00000000
<input type="checkbox"/>	无描述	09C1E9C0	4 字节	09ADCE00
<input type="checkbox"/>	无描述	09C6FC30	4 字节	09ADC850
<input type="checkbox"/>	无描述	13AF15E8	4 字节	00000000

四个地址中有2个为0，2个发生了变化，分别查找两个不为0的区域。

0x09ADCE00位置处如下图所示，对应0x64偏移处值为1

09ADCE00	B0	5F	2C	1F	00	00	00	00	00	00	00	00	00	00	00	00	00
09ADCE10	00	00	00	00	00	00	00	00	00	70	BE	B0	09	00	00	00	00
09ADCE20	58	BE	B0	09	00	00	00	00	00	A0	19	CB	09	00	00	00	00
09ADCE30	40	BE	B0	09	00	00	00	00	00	08	F7	C9	09	00	00	00	00
09ADCE40	E0	F6	C9	09	00	00	00	00	00	B8	F6	C9	09	00	00	00	00
09ADCE50	90	F6	C9	09	00	00	00	00	00	00	00	40	40	00	00	80	00
09ADCE60	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00	00
09ADCE70	B0	5F	2C	1F	00	00	00	00	00	00	00	00	00	00	00	00	00
09ADCE80	00	00	00	00	00	00	00	00	00	30	BF	B0	09	00	00	00	00
09ADCE90	18	BF	B0	09	00	00	00	00	00	00	00	00	00	00	00	00	00
09ADCEA0	00	BF	B0	09	00	00	00	00	00	08	F7	C9	09	00	00	00	00

0x09ADC850位置处如下图所示，对应0x64偏移处值也为1

09ADC850	B0	5F	2C	1F	00	00	00	00	00	00	00	00	00	00	00
09ADC860	70	D2	B5	13	00	00	00	00	00	80	17	B6	09	00	00
09ADC870	68	17	B6	09	00	00	00	00	00	C0	18	CB	09	00	00
09ADC880	50	17	B6	09	00	00	00	00	00	08	F7	C9	09	00	00
09ADC890	E0	F6	C9	09	00	00	00	00	00	B8	F6	C9	09	00	00
09ADC8A0	90	F6	C9	09	00	00	00	00	00	00	00	40	40	00	80
09ADC8B0	00	01	00	00	01	00	00	00	00	00	00	00	00	00	00
09ADC8C0	B0	5F	2C	1F	00	00	00	00	00	00	00	00	00	00	00
09ADC8D0	30	F1	A9	13	00	00	00	00	00	40	18	B6	09	00	00
09ADC8E0	28	18	B6	09	00	00	00	00	00	00	00	00	00	00	00
09ADC8F0	10	18	B6	09	00	00	00	00	00	08	F7	C9	09	00	00

暂时无法确定，使小鸟死亡，二者值未发生变化。

重启一局，去掉两个0值，重新判断

发现两个地址处结果均为0x9A1C460,此时小鸟score为2，对应0x64偏移处结果也为2，判断这两个地址存放都是小鸟的基地址（存疑）

标识	描述	地址	类型	数值
<input type="checkbox"/>	base1	09C1E9C0	4 字节	09A1C460
<input type="checkbox"/>	base2	09C6FC30	4 字节	09A1C460
<input type="checkbox"/>	score1	P->09A1C4C4	4 字节	2
<input type="checkbox"/>	score2	P->09A1C4C4	4 字节	2

重开一局测试

<input type="checkbox"/>	base1	09C1E9C0	4 字节	09C400E0
<input type="checkbox"/>	base2	09C6FC30	4 字节	09C40310
<input type="checkbox"/>	score1	P->09C40144	4 字节	2
<input type="checkbox"/>	score2	P->09C40374	4 字节	3

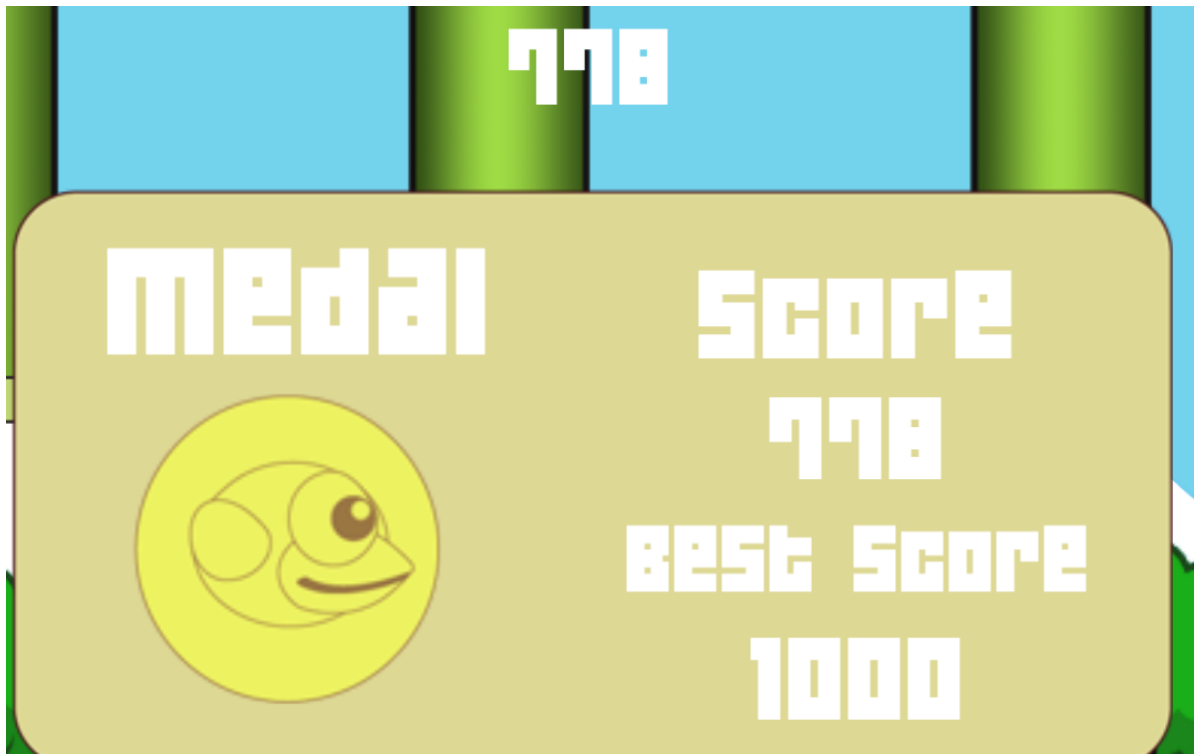
多次测试后，发现存在一定规律，每两局游戏时，其中一局这两个地址一样的，而其中另一局这两个数值不同，不同的这一局分数为上一局的分数。

<input type="checkbox"/>	base1	09C1E9C0	4 字节	00000000
<input type="checkbox"/>	base2	09C6FC30	4 字节	09C404D0
<input type="checkbox"/>	score1	P->00000064	4 字节	??
<input type="checkbox"/>	score2	P->09C40534	4 字节	15

多次测试后，第一个位置变为0，故base1应为临时地址，删去，base2才是真实地址。将score改为777进行测试

<input type="checkbox"/>	base	09C6FC30	4 字节	09CAAB60
<input checked="" type="checkbox"/>	score	P->09CAABC4	4 字节	777

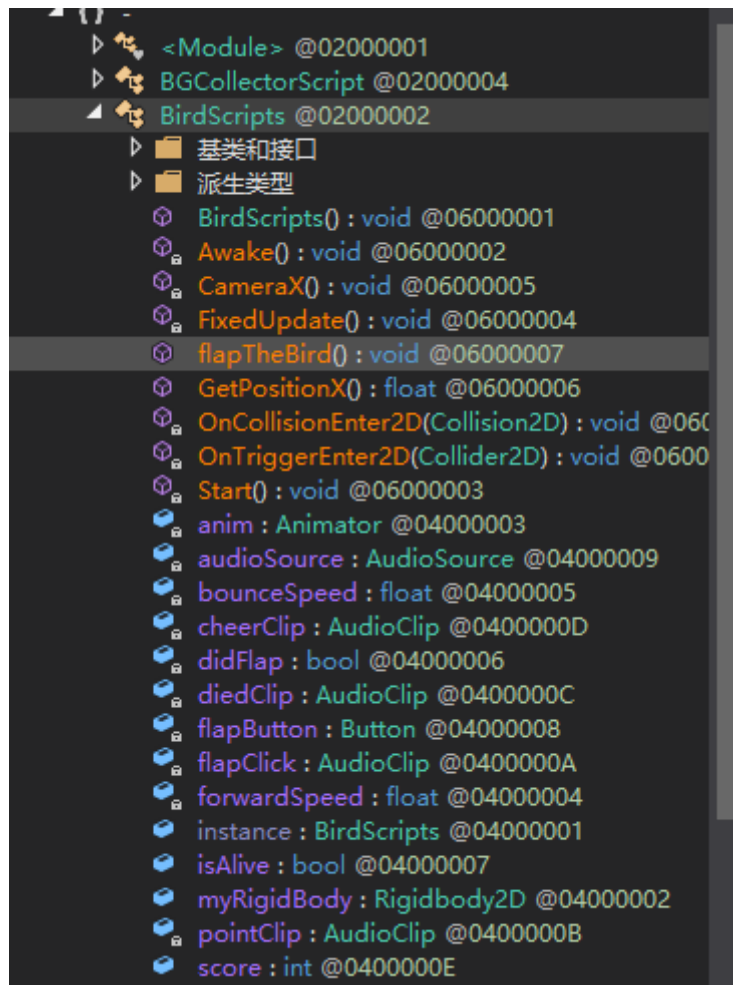
分数刷新后，数值改为778



## 二、使用CE修改汇编代码

### 1、使用dnspy找到计分规则函数

将Flappy Bird的Assembly-CSharp.dll拖入dnspy中进行反编译



发现其存在BirdScripts这一主要类

找到score属性字段

```
// Token: 0x0400000E RID: 14
public int score;
```

进行交叉查找看那一部分修改了此属性

赋值于

- ▶ BirdScripts.Awake() : void @06000002
- ▶ BirdScripts.OnTriggerEnter2D(Collider2D) : void @06000009

发现这两个函数修改了此属性。

```
private void Awake()
{
    if (BirdScripts.instance == null)
    {
        BirdScripts.instance = this;
    }
    this.isAlive = true;
    this.score = 0;
    this.flapButton = GameObject.FindGameObjectWithTag("FlapButton").GetComponent<Button>();
    this.flapButton.onClick.AddListener(delegate()
    {
        this.flapTheBird();
    });
    this.CameraX();
}
```

```
// Token: 0x00000009 RID: 9 RVA: 0x000230C File Offset: 0x000230C
private void OnTriggerEnter2D(Collider2D target)
{
    if (target.tag == "PipeHolder")
    {
        this.audioSource.PlayOneShot(this.pointClip);
        this.score++;
        GameplayController.instance.setScore(this.score);
    }
}
```

其中Awake函数为初始化该score为0

而OnTriggerEnter2D为分数增加。

故对OnTriggerEnter2D函数进行分析修改。

## 2、CE中修改OnTriggerEnter2D

在CE中找到该函数的代码段

nter2D+6f	48 83 C4 20	add	rsp,20	32
nter2D+73	48 63 47 64	movsxd	rax,dword ptr [rdi+64]	
nter2D+77	FF C0	inc	eax	
nter2D+79	89 47 64	mov	[rdi+64],eax	
nter2D+7c	48 8B 04 25 20CC4309	mov	rax,[0943CC20]	(0958BA80)
nter2D+84	48 63 57 64	movsxd	rdx,dword ptr [rdi+64]	
nter2D+88	48 8B C8	mov	rcx,rcx	
nter2D+8b	48 83 FC 20	sub	rsn,20	32

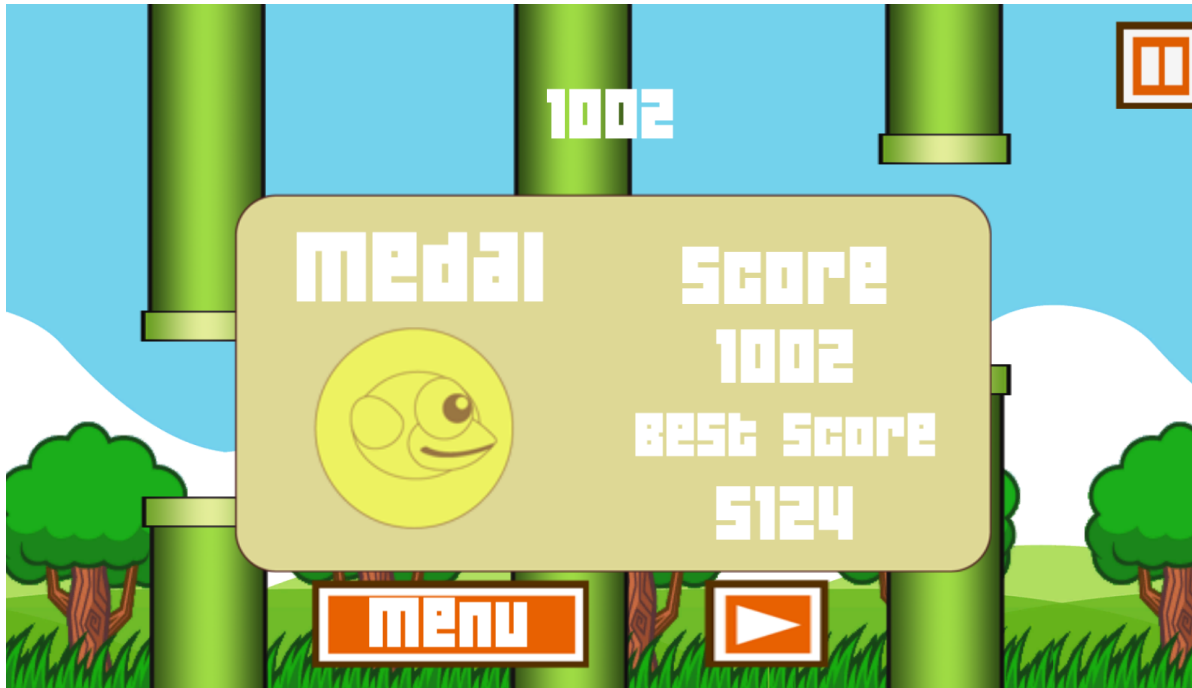
rdi为鸟对象实例，而rdi+64即为score地址

将score赋值给rax后，对eax自加，则赋值会rdi+64。

此处应为修改score部分，修改汇编指令inc eax为add eax, 500，即可实现每次通过管道后后加501分

地址	字节	操作码	注释
0F7D0000	05 F4010000	add eax,000001F4	500
0F7D0005	FF C0	inc eax	
0F7D0007	89 47 64	mov [rdi+64],eax	
0F7D000A	E9 FD6FADF9	jmp BirdScripts:OnTriggerEnter2D+7c	
0F7D000F	00 00	add [rax],al	

通过2个管道后，即可实现加1002分



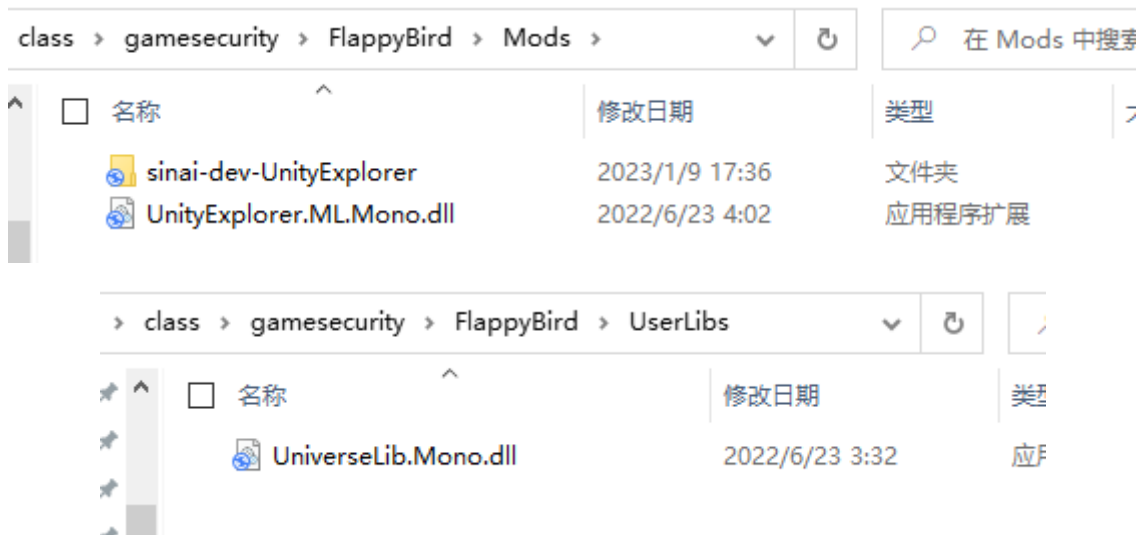
### 三、使用UnityExplorer调试游戏程序集

#### 1、使用MelonLoader对游戏进行修改



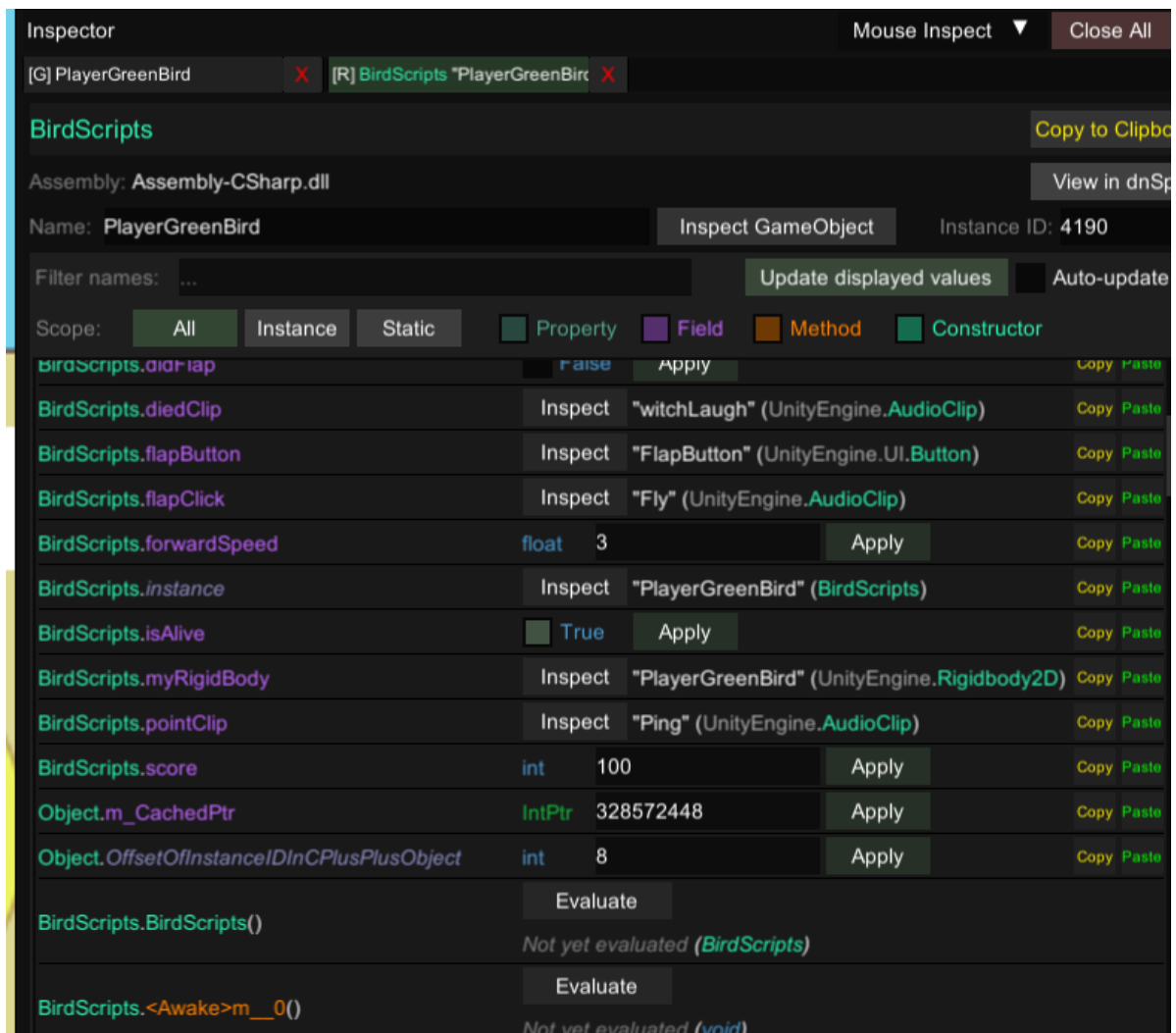


## 2、将UnityExplorer加入mods和libs中

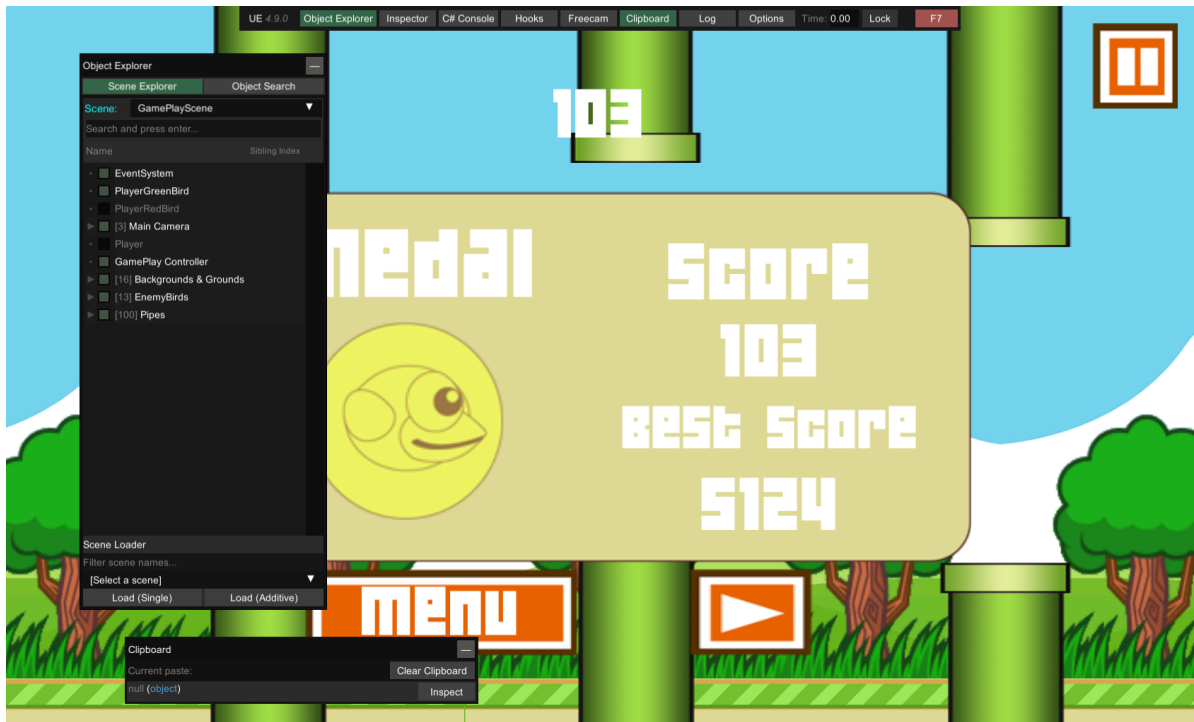


## 3、使用UnityExplorer进行修改分数

修改分数为100并应用



修改分数成功



## 四、使用程序集注入方式

### 1、编写注入函数

Loader.cs

```
namespace InjectDll
{
    public class Loader
    {
        static UnityEngine.GameObject gameObject;
        public static void Load()
        {
            gameObject = new UnityEngine.GameObject();
            gameObject.AddComponent<Cheat>();
            UnityEngine.Object.DontDestroyOnLoad(gameObject);
        }
        public static void Unload()
        {
            UnityEngine.Object.Destroy(gameObject);
        }
    }
}
```

cheat.cs

```
public class Cheat : UnityEngine.MonoBehaviour
{
    private void OnGUI()
    {
        UnityEngine.GUI.Label(new Rect(0, 0, 100, 100), "Hack!\nPress F1: score + 1000\nPress F2: 无敌\nPress F3: 取消无敌");
    }
    public void FixedUpdate()
    {
    }
}
```

```

{
    if (UnityEngine.Input.GetKeyDown(KeyCode.F1))
    {
        //分数 + 1000
        var bs =
UnityEngine.GameObject.FindWithTag("Player").GetComponent<BirdScripts>();
        if (bs != null)
        {
            bs.score = bs.score + 1000;
        }
    }
    if (UnityEngine.Input.GetKeyDown(KeyCode.F2))
    {
        // 无敌
        var Player = UnityEngine.GameObject.FindWithTag("Player");
        var bs = Player.GetComponent<BirdScripts>();
        Player.GetComponent<Collider2D>().isTrigger = true;
    }
    if (UnityEngine.Input.GetKeyDown(KeyCode.F3))
    {
        //取消无敌
        var Player = UnityEngine.GameObject.FindWithTag("Player");
        var bs = Player.GetComponent<BirdScripts>();
        Player.GetComponent<Collider2D>().isTrigger = false;
    }
}
}

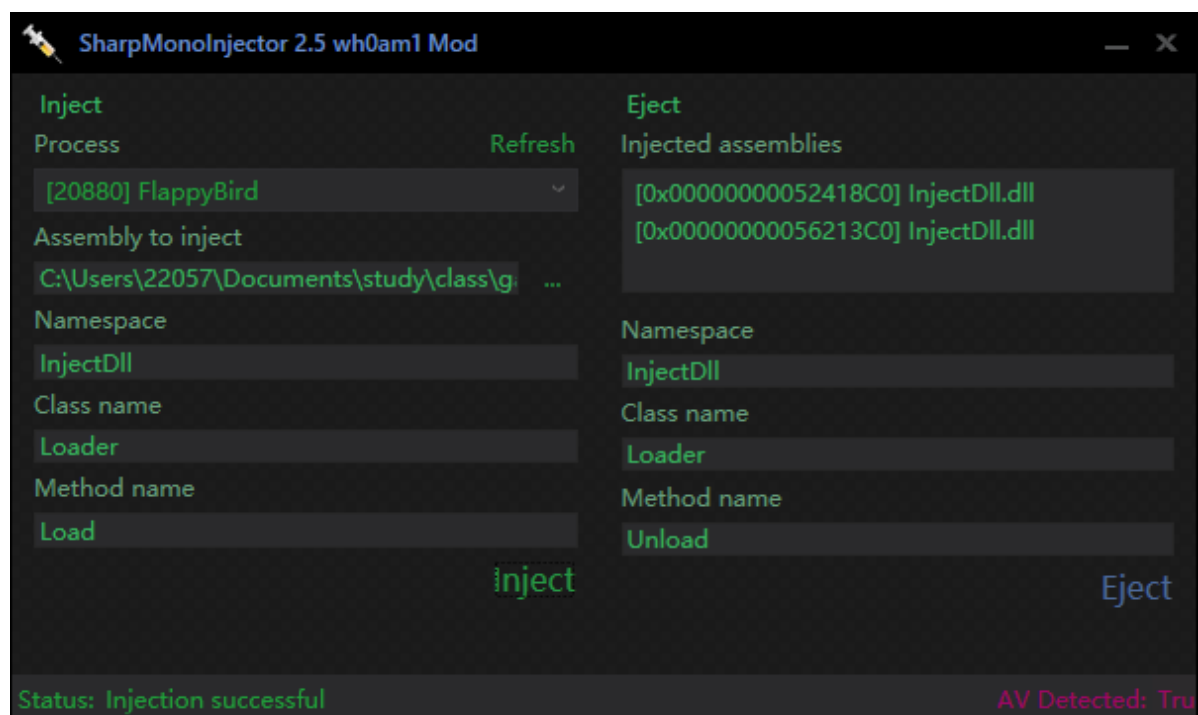
```

通过读取按键F1：来搜索Player对象，使对象的score+1000来达到修改分数的目的

## 2、使用sharpMonoInjector进行注入

将注入程序集编译为dll

使用sharpMonoInjector进行注入



注入成功，左上角显示输出信息

```
Hack!  
Press F1: score  
+ 1000  
Press F2: 无敌  
Press F3:  
取消无敌
```

### 3、测试结果

按下F1后，分数增加1000分，测试成功

