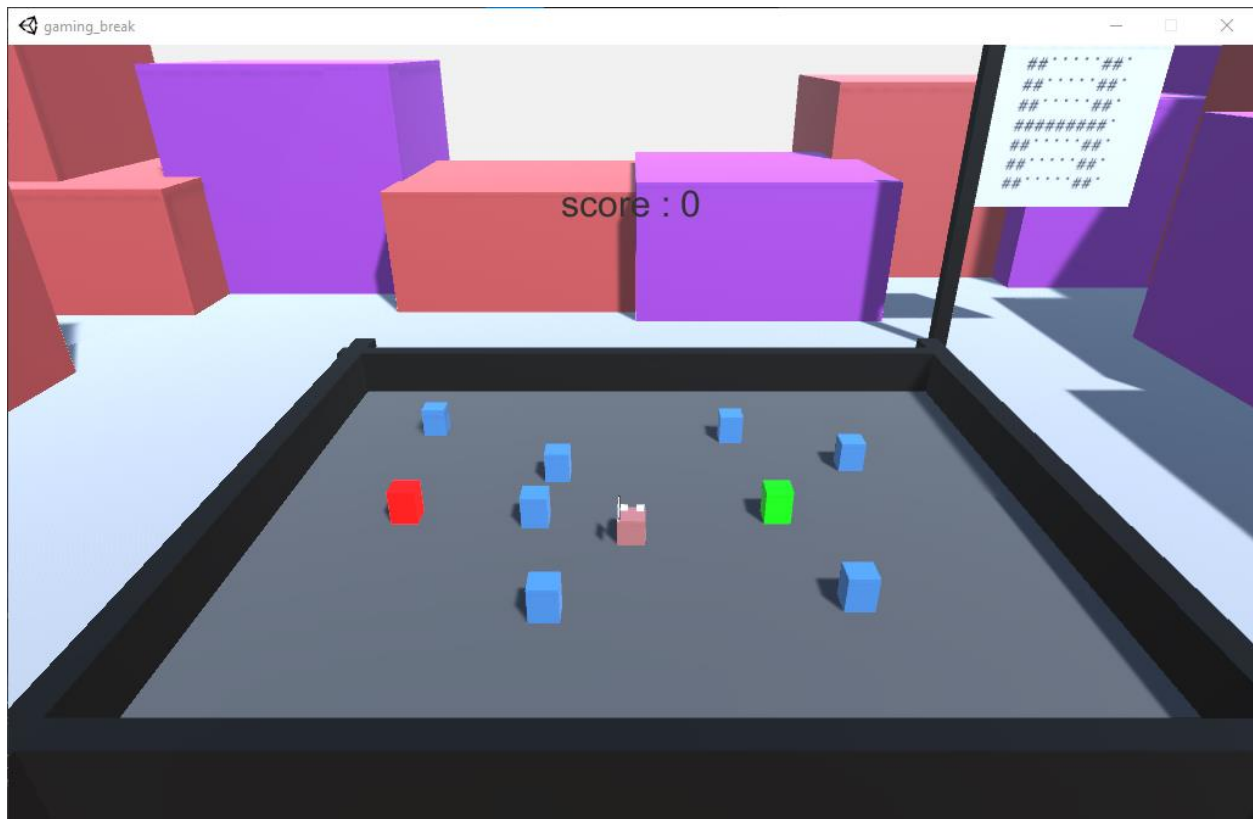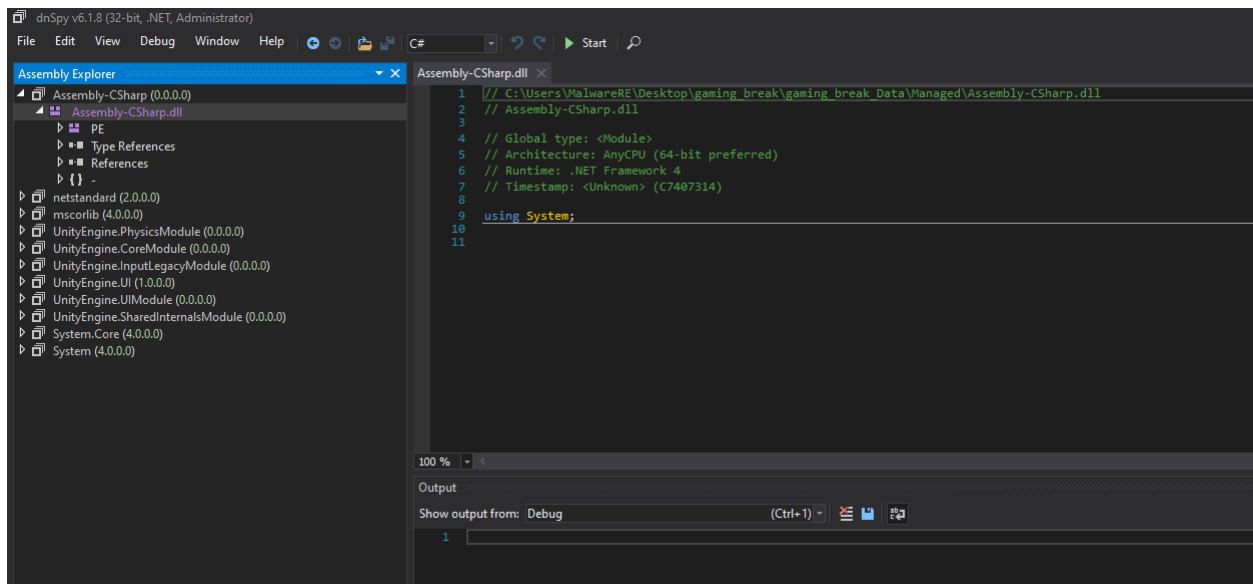The next challenge was called game break and as the name suggests we had to hack the game in order to win.
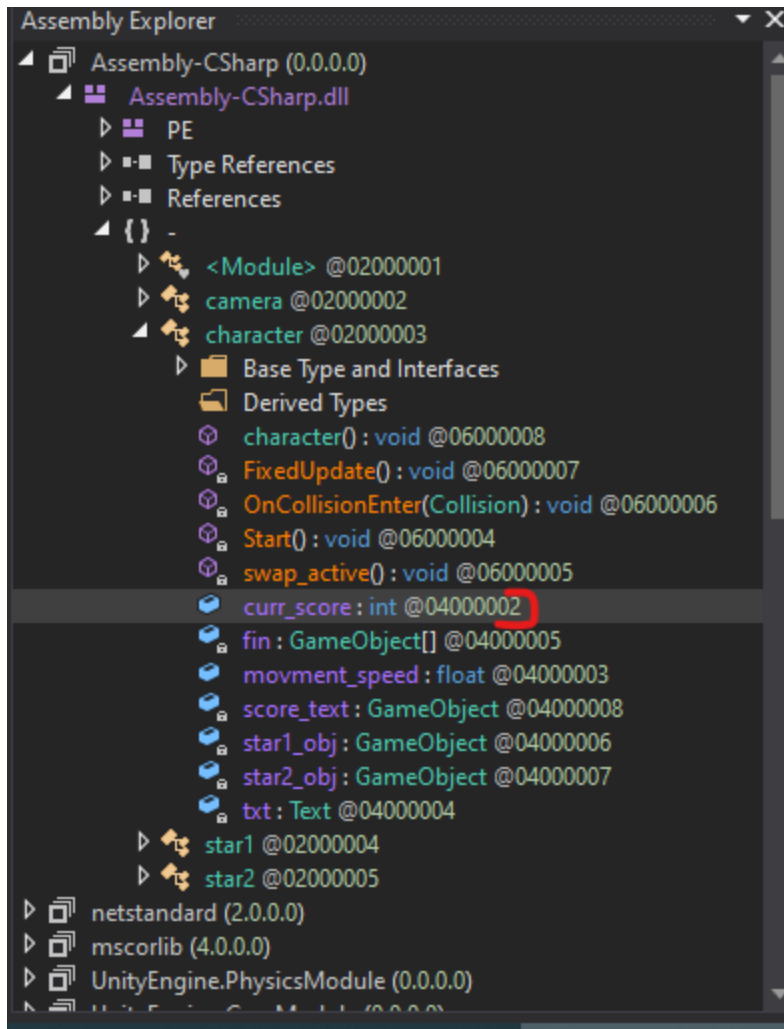

First, we start by opening the Assembly-CSharp dll in dnSpy:

We here have to paths the first one is to try and guess the score counter name which might be useful sometimes, but it wont work every time and the second path is to walk around the code and understand what it actually does.

After searching the code for a while we can see that the score counter name is "curr_score"

Assembly Explorer

- Assembly-CSharp (0.0.0.0)
  - Assembly-CSharp.dll
    - PE
    - Type References
    - References
    - { } -
      - <Module> @02000001
      - camera @02000002
      - character @02000003
        - Base Type and Interfaces
        - Derived Types
        - character() : void @06000008
        - FixedUpdate() : void @06000007
        - OnCollisionEnter(Collision) : void @06000006
        - Start() : void @06000004
        - swap_active() : void @06000005
        - curr_score : int @04000002
        - fin : GameObject[] @04000005
        - movment_speed : float @04000003
        - score_text : GameObject @04000008
        - star1_obj : GameObject @04000006
        - star2_obj : GameObject @04000007
        - txt : Text @04000004
      - star1 @02000004
      - star2 @02000005
- netstandard (2.0.0.0)
- mscorlib (4.0.0.0)
- UnityEngine.PhysicsModule (0.0.0.0)

A good place to start reading is the Start() function ,but also the function called OnCollisionEnter() caught my eye so I decided to take a look at it:

```
1    // character
2    // Token: 0x06000006 RID: 6
3    private void OnCollisionEnter(Collision other)
4    {
5        this.txt = this.score_text.GetComponent<Text>();
6        if (other.gameObject.name == "star1" && this.star1_obj.GetComponent<star1>().active)
7        {
8            this.swap_active();
9            this.curr_score++;
10       }
11       else if (other.gameObject.name == "star2" && this.star2_obj.GetComponent<star2>().active)
12       {
13           this.swap_active();
14           this.curr_score++;
15       }
16       else if (other.gameObject.tag == "negative_point")
17       {
18           int num = this.curr_score - 1;
19           this.curr_score = num;
20           Debug.Log(num);
21       }
22       if (this.curr_score < 0)
23       {
24           this.curr_score = 0;
25       }
26       if (this.curr_score >= 1337)
27       {
28           GameObject[] array = this.fin;
29           for (int i = 0; i < array.Length; i++)
30           {
```

100 % ▾ ◂
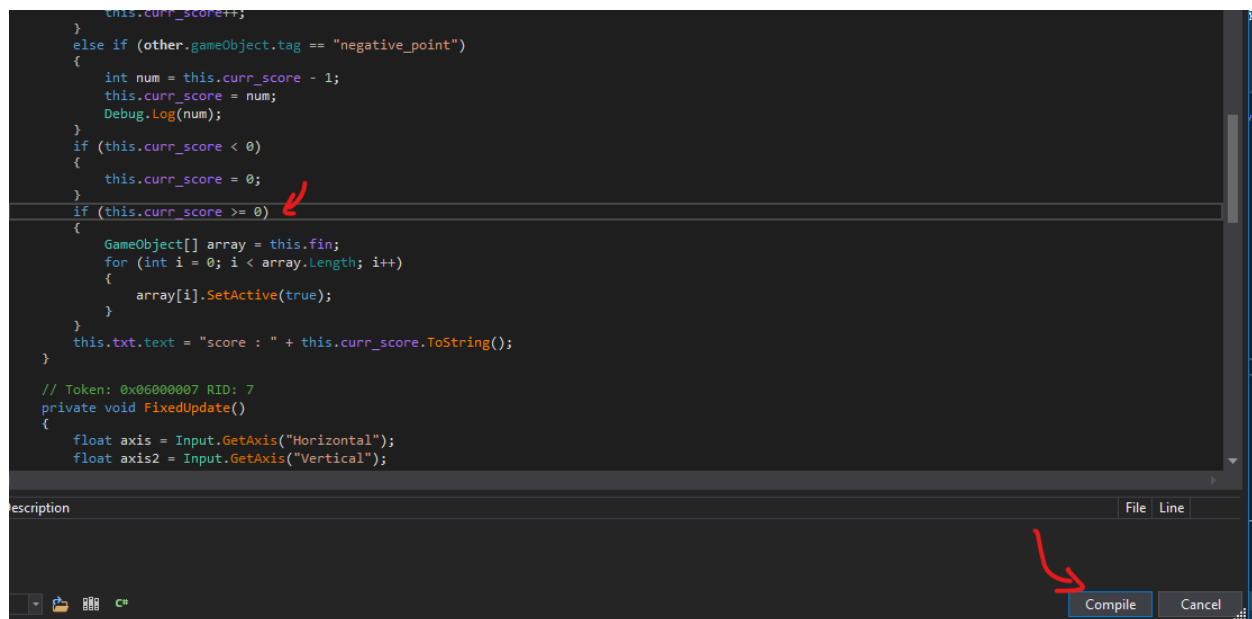
Locals Call Stack Output Analyzer

As I was reading it I understood the logic if the game, if you hit the green column you get a point else you get a point deducted from your score until you reach 0, this if statement seems weird doesn't it?

```csharp
if (this.curr_score >= 1337)
{
    GameObject[] array = this.fin;
    for (int i = 0; i < array.Length; i++)
    {
        array[i].SetActive(true);
    }
}
this.txt.text = "score : " + this.curr_score.ToString();
}
```

If our score reaches 1337 or higher it seems to print something, we could keep playing until we reach 1337, but obvesuly that's not the point so we go ahead and change the comparison value to zero by doing the following:

Right click on it and select edit class (C#):

```csharp
        }
        else if (other.gameObject.n
        {
            this.swap_active();
            this.curr_score++;
        }
        else if (other.gameObject.t
        {
            int num = this.curr_sco
            this.curr_score = num;
            Debug.Log(num);
        }
        if (this.curr_score < 0)
        {
            this.curr_score = 0;
        }
        if (this.curr_score >= 1337)
        {
            GameObject[] array = this.fin;
            for (int i = 0; i < array.Length; i++)
            {
                array[i].SetActive(true);
            }
        }
        this.txt.text = "score : " + this.curr_score.ToString();
    }
```

CollisionEnter(Collision) : void

	Start Debugging	F5
●	Add Breakpoint	F9
C#	Edit Method (C#)...	Ctrl+Shift+E
C#	Edit Class (C#)...	
C#	Add Class Members (C#)...	
C#	Add Class (C#)...	
	Merge with Assembly...	
	Edit IL Instructions...	
	Go to MD Token...	Ctrl+D
	Go to MD Table Row...	Ctrl+Shift+D
	Show Instructions in Hex Editor	Ctrl+X
	Find	Ctrl+F
	Incremental Search	Ctrl+I


```csharp
            this.curr_score++;
        }
        else if (other.gameObject.tag == "negative_point")
        {
            int num = this.curr_score - 1;
            this.curr_score = num;
            Debug.Log(num);
        }
        if (this.curr_score < 0)
        {
            this.curr_score = 0;
        }
        if (this.curr_score >= 0)
        {
            GameObject[] array = this.fin;
            for (int i = 0; i < array.Length; i++)
            {
                array[i].SetActive(true);
            }
        }
        this.txt.text = "score : " + this.curr_score.ToString();
    }

    // Token: 0x06000007 RID: 7
    private void FixedUpdate()
    {
        float axis = Input.GetAxis("Horizontal");
        float axis2 = Input.GetAxis("Vertical");
```

Description   File Line
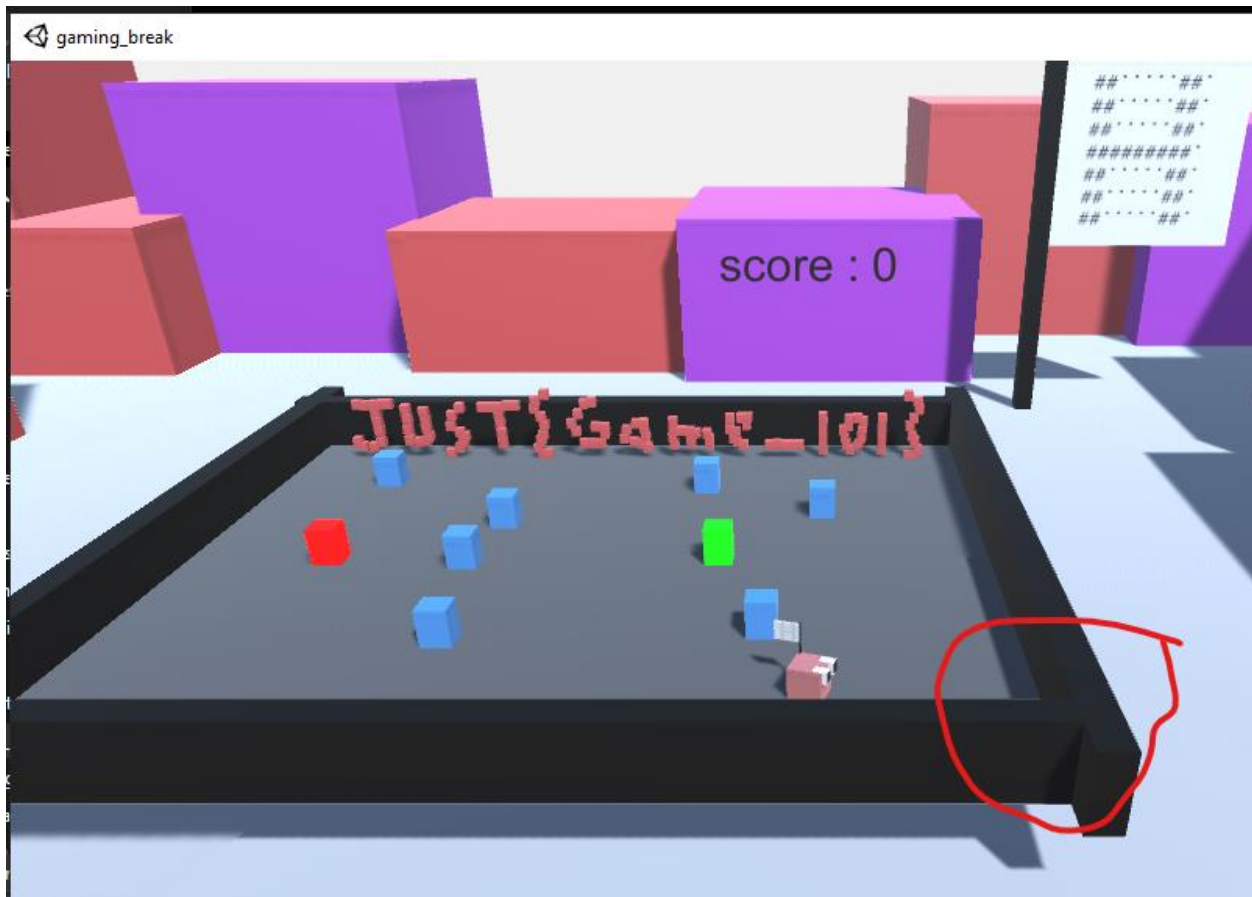
Compile   Cancel

Change the value to zero and hit compile, then from File tab chose Save Module.

Then start the game again and here is our flag:
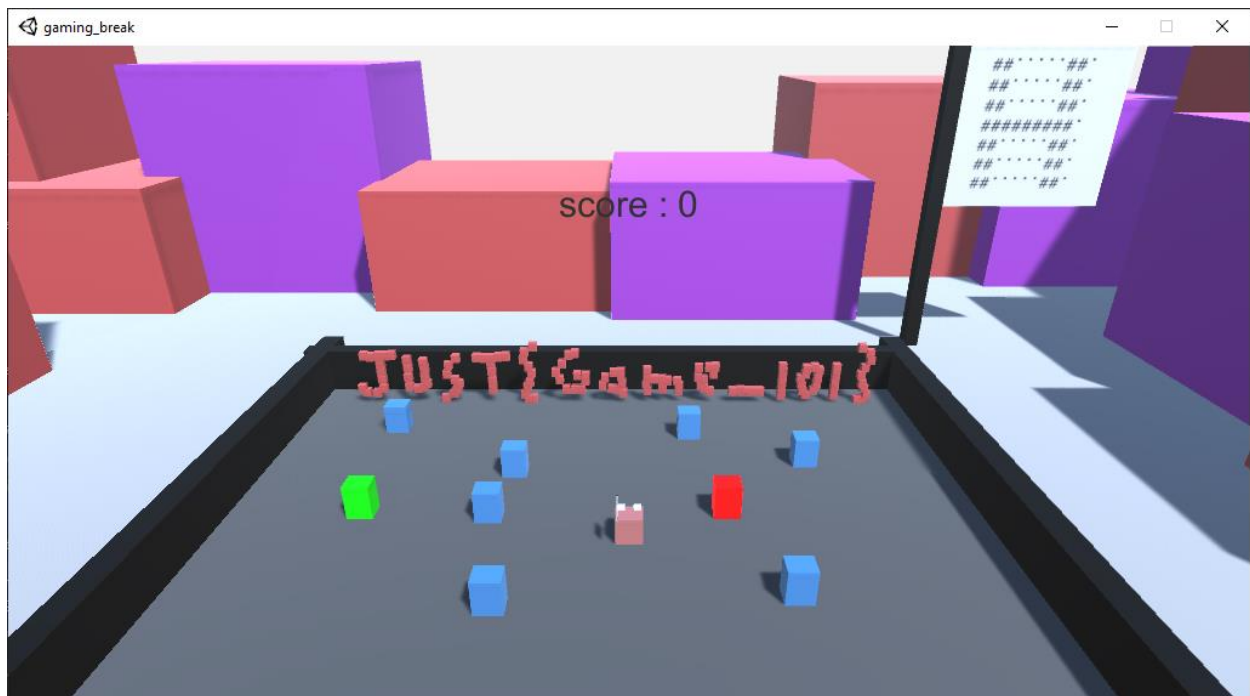
This challenge opened another challenge called easter egg, I didn't solve it at the competition but afterwards I figured it out we go to this corner:

and keep running into it until we fall off the map and then we walk around and find the hidden easter egg.