dnSpy

With Dreamhack Wargame

SCP | 이지훈 2024.11.19

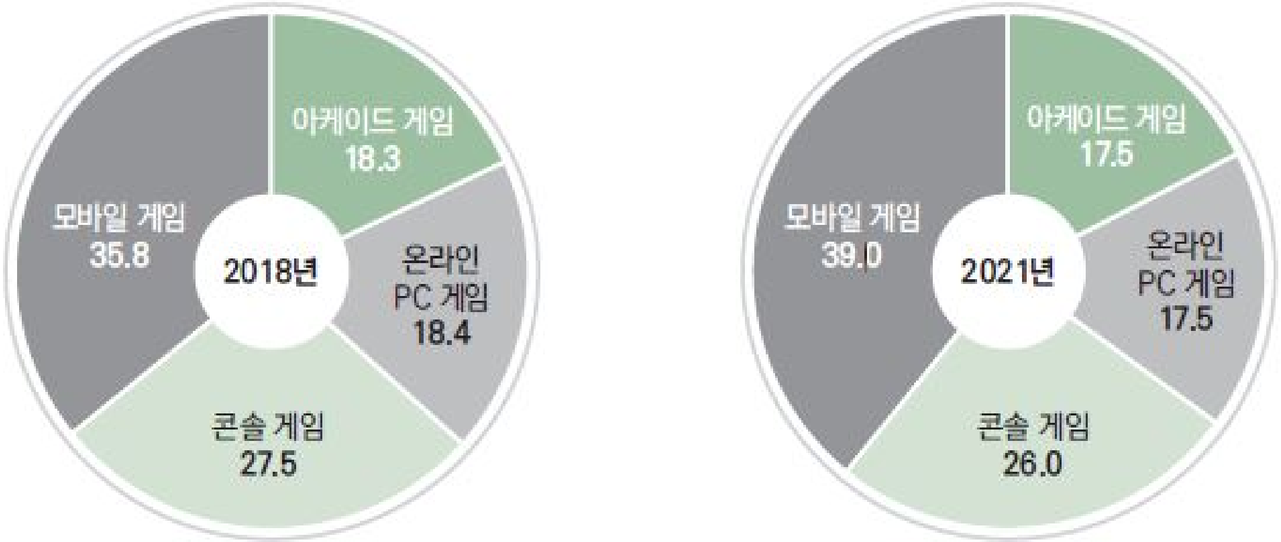
2024-2학기 SCP 내부 세미나 B조

INDEX

011 안드로이드 개발

[그림 6] 플랫폼별 세계 게임 시장 점유율(2018년/2021년)

(단위: %)



출처: PWC(2019), Enterbrain(2019), JOGA(2019), iResearch(2019), Playmeter(2016), NPD(2019)

01 | 안드로이드 개발

https://velog.io/@openhub/%EB%84%A4%EC%9D%B4%ED%8B%B0%EB%B8%8C-%EC%95%B1Native-App-vs-%ED%95%98%EC%9D%B4%EB%B8%8C%EB%A6%AC%EB%93%9C-%EC%95%B1Hybrid-App-vs-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%A0%88%EC%8B%9C%EB%B8%8C-%EC%9B%B9-%EC%95%B1PWA-%EC%A0%95%EC%9D%98%EC%99%80-%EC%9E%A5%EB%8B%A8%EC%A0%90

011 안드로이드 개발

	앱 설치 여부	개발 방식	디바이스 기능 접근	개발 지식
네이티브 앱	O	모바일 운영체제 별 전문 개발언어 사용 (Kotlin, Java, Swift, Objective C 등)	가능	iOS, 안드로이드 앱 개발 지식 필요
웹 앱	X	웹 코딩 기반 (HTML, CSS, Javascript 등)	불가능	웹 표준 웹 개발 지식 필요
하이브리드 앱	O	앱, 웹 기반 언어와 코딩을 모두 사용	가능	내부 페이지 개발을 위해 웹 개발 지식, 외부 앱 패키징을 위해 앱 개발 지식 필요

https://blog.hectodata.co.kr/app_kinds/

01 | 안드로이드 개발

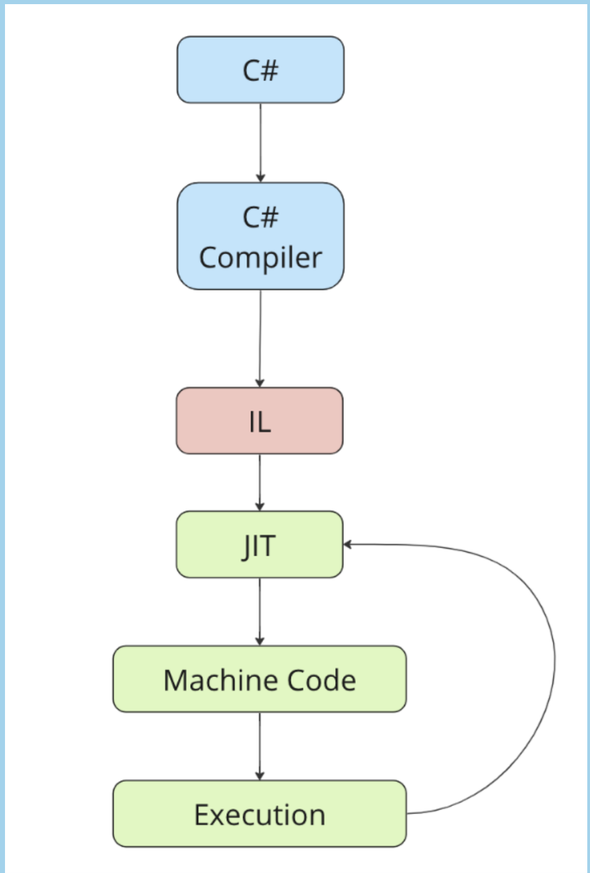
02 I Unity Engine

Unity Engine

C# 스크립트를 통해 게임을 개발하고,
게임을 개발할 수 있는 여러가지 기능들을 제공해주는 게임 제작 엔진

Window, Mac, Android, ios, Linux 등의 다양한 플랫폼 기반 게임 제작 및 배포가 가능하다. (테슬라 소프트웨어도 지원한다...)

03 I MONO VS IL2CPP



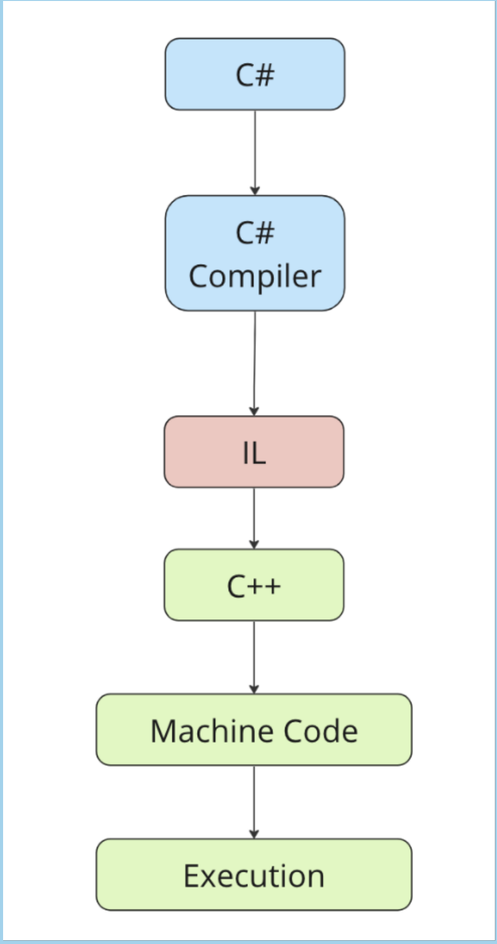
MONO

Unity에서 기본적으로 사용되는 스크립팅 런타임

.NET Framework과 유사한 구조를 가지고 있으며,
C# 스크립트를 IL 코드로 변환하고,
JIT컴파일러를 사용하여 IL 코드를 실행

다양한 플랫폼에서 실행할 수 있고 빌드가 빠르지만..
DLL 파일을 통해 쉽게 디컴파일이 가능하여 보안에 취약하다.

03 I MONO VS IL2CPP

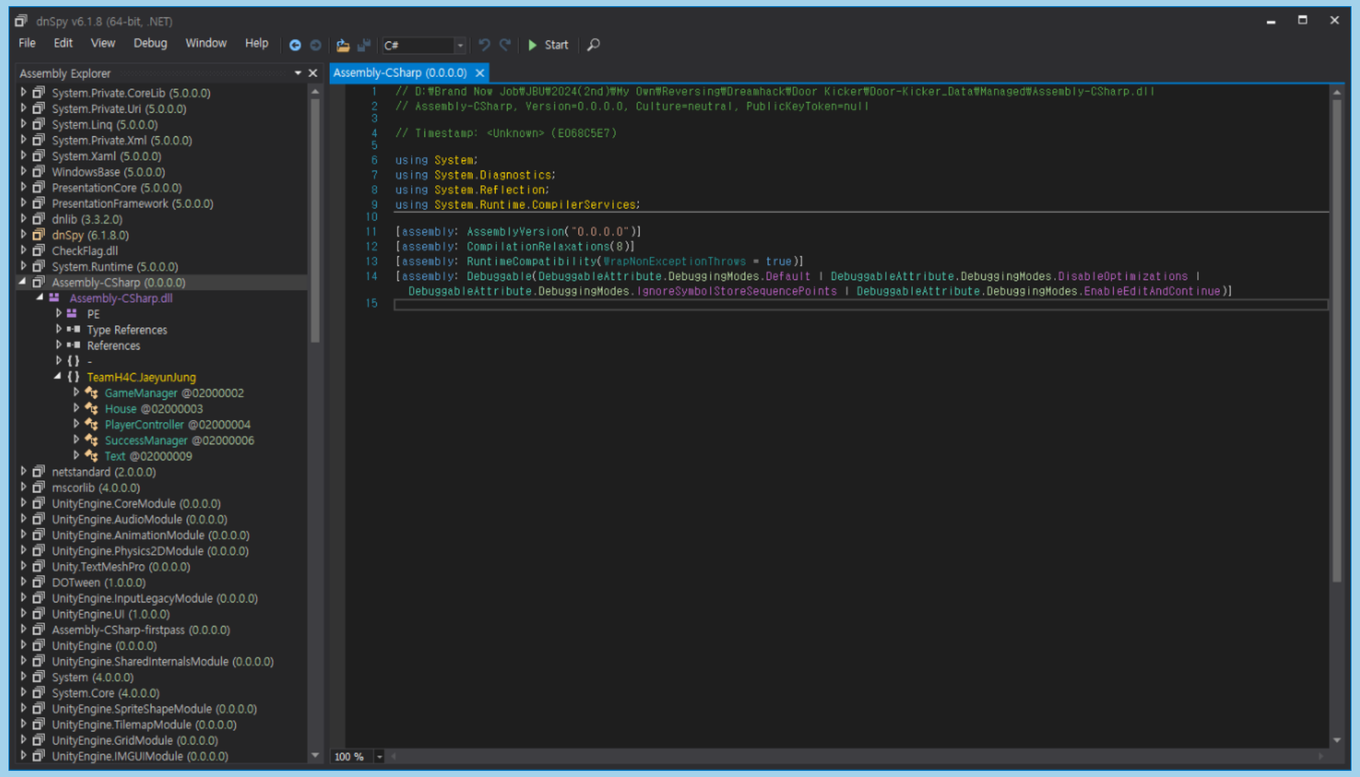


IL2CPP

AOT(Ahead-Of-Time) 컴파일 방식을 따른다. IL2CPP는 C# 코드를 컴파일하여 생성된 IL 코드를 C++ 코드로 변환한다.

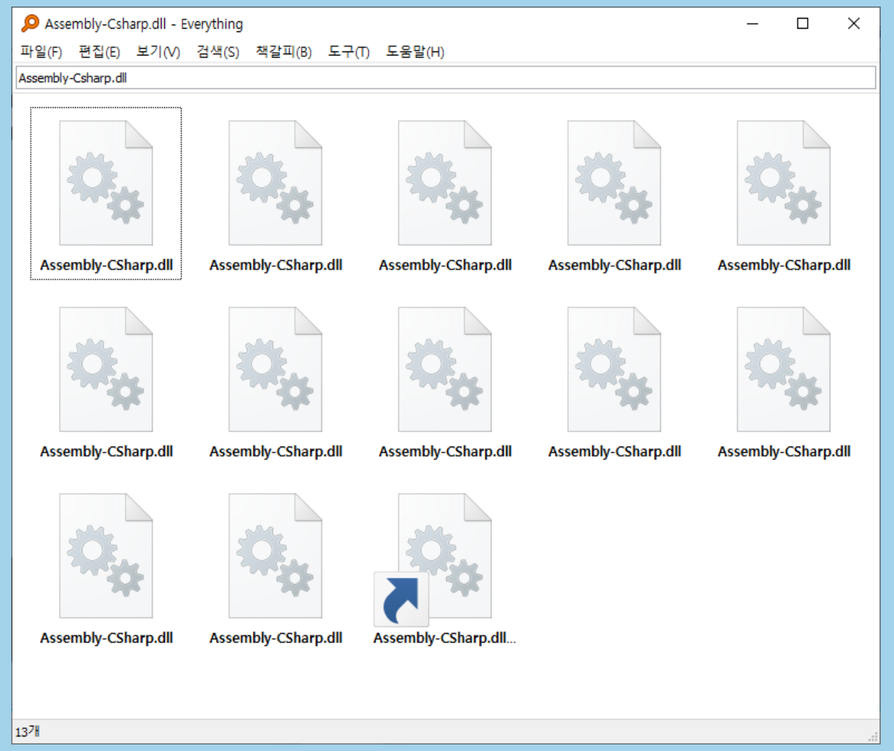
런타임 시점에 코드를 컴파일하는 JIT 컴파일러와 달리
미리 코드를 컴파일하여 사용하기 때문에 보안에 강하다.

04 I dnSpy



.NET 어셈블리 편집 및 디버깅 툴

05 I MONO 분석



Assembly-CSharp.dll 에 MONO로 빌드된 Unity 게임 코드가 담겨있다. 하지만..

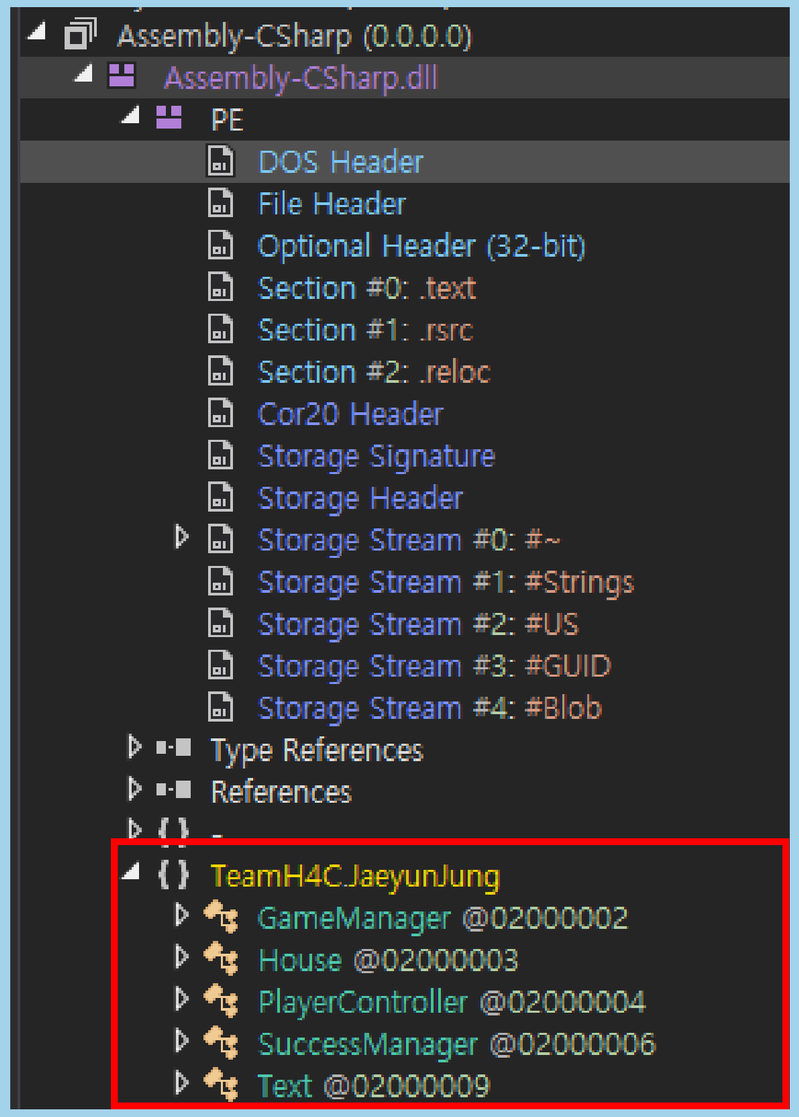
05 I MONO 분석

드림핵 워게임으로 실습

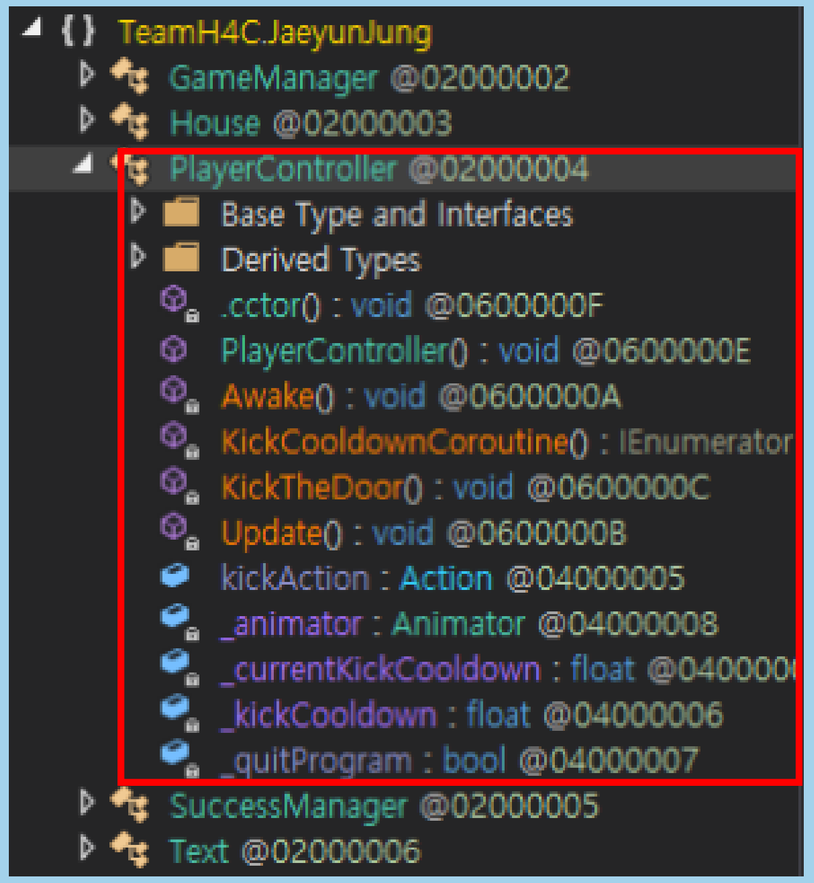
05 I MONO 분석


Door-Kicker
Press the space bar
to kick the door
Development Build

05 I MONO 분석


Assembly-CSharp (0.0.0.0)
Assembly-CSharp.dll
PE
DOS Header
File Header
Optional Header (32-bit)
Section #0: .text
Section #1: .rsrc
Section #2: .reloc
Cor20 Header
Storage Signature
Storage Header
Storage Stream #0: #~
Storage Stream #1: #Strings
Storage Stream #2: #US
Storage Stream #3: #GUID
Storage Stream #4: #Blob
Type References
References
TeamH4C.JaeyunJung
GameManager @02000002
House @02000003
PlayerController @02000004
SuccessManager @02000006
Text @02000009

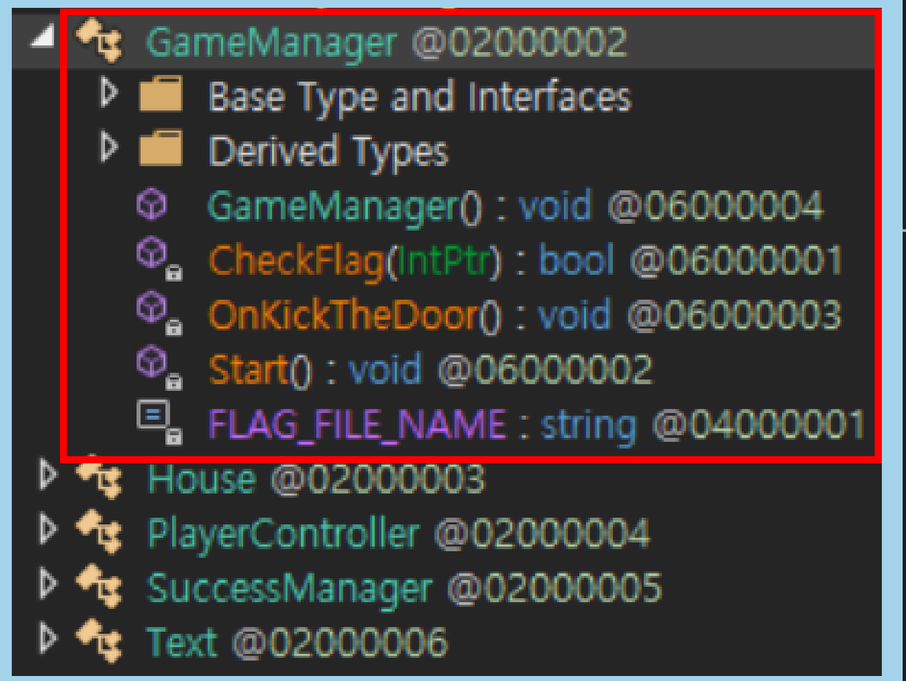
05 I MONO 분석




```
private void Update()
{
    bool flag = !Input.GetKeyDown(KeyCode.Space);
    if (!flag)
    {
        this.KickTheDoor();
    }
}
```

```
// Token: 0x0600000C RID: 12 RVA: 0x000021EC File Offset: 0x000003EC
private void KickTheDoor()
{
    bool flag = this._currentKickCooldown > 0f;
    if (!flag)
    {
        bool quitProgram = PlayerController._quitProgram;
        if (quitProgram)
        {
            Application.Quit();
        }
        else
        {
            this._animator.SetTrigger("Kick");
            PlayerController.kickAction();
            base.StartCoroutine(this.KickCooldownCoroutine());
        }
    }
}
```

05 I MONO 분석




```
private void Start()
{
    PlayerController.kickAction = (Action)Delegate.Combine(PlayerController.kickAction, new Action(this.OnKickTheDoor));
    bool flag = !Debugger.IsAttached
    if (!flag)
    {
        Application.Quit();
    }
}
```

```
// Token: 0x06000003 RID: 3 RVA: 0x00002094 File Offset: 0x00000294
private void OnKickTheDoor()
{
    bool flag2 = !File.Exists(Application.streamingAssetsPath + "/flag");
    if (!flag2)
    {
        string flag = File.ReadAllText(Application.streamingAssetsPath + "/flag");
        bool flag3 = GameManager.CheckFlag(Marshal.StringToHGlobalAnsi(flag));
        if (flag3)
        {
            SceneManager.LoadScene("Success");
        }
    }
}
```

05 I MONO 분석

```
private void OnKickTheDoor()
{
    bool flag2 = !File.Exists(Application.streamingAssetsPath + "/flag");
    if (!flag2)
    {
        string flag = File.ReadAllText(Application.streamingAssetsPath + "/flag");
        bool flag3 = GameManager.CheckFlag(Marshal.StringToHGlobalAnsi(flag));
        if (flag3)
        {
            SceneManager.LoadScene("Success");
        }
    }
}
```

```
[DllImport("CheckFlag.dll")]
private static extern bool CheckFlag(IntPtr flag);
```

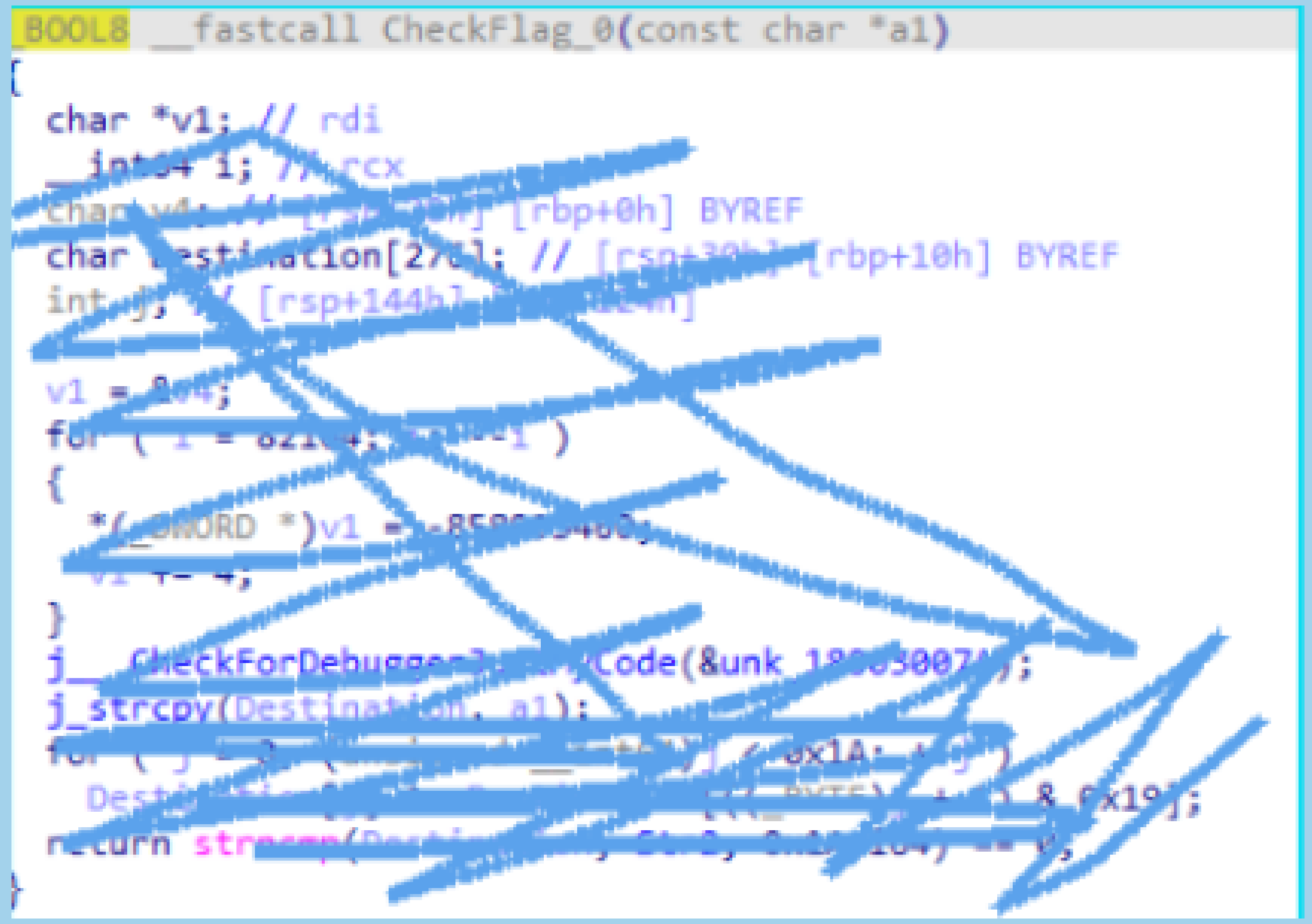
플래그가 스트리밍에셋폴더 내에 존재하고, CheckFlag 함수를 통해 문자열 비교 후 일치하면 성공

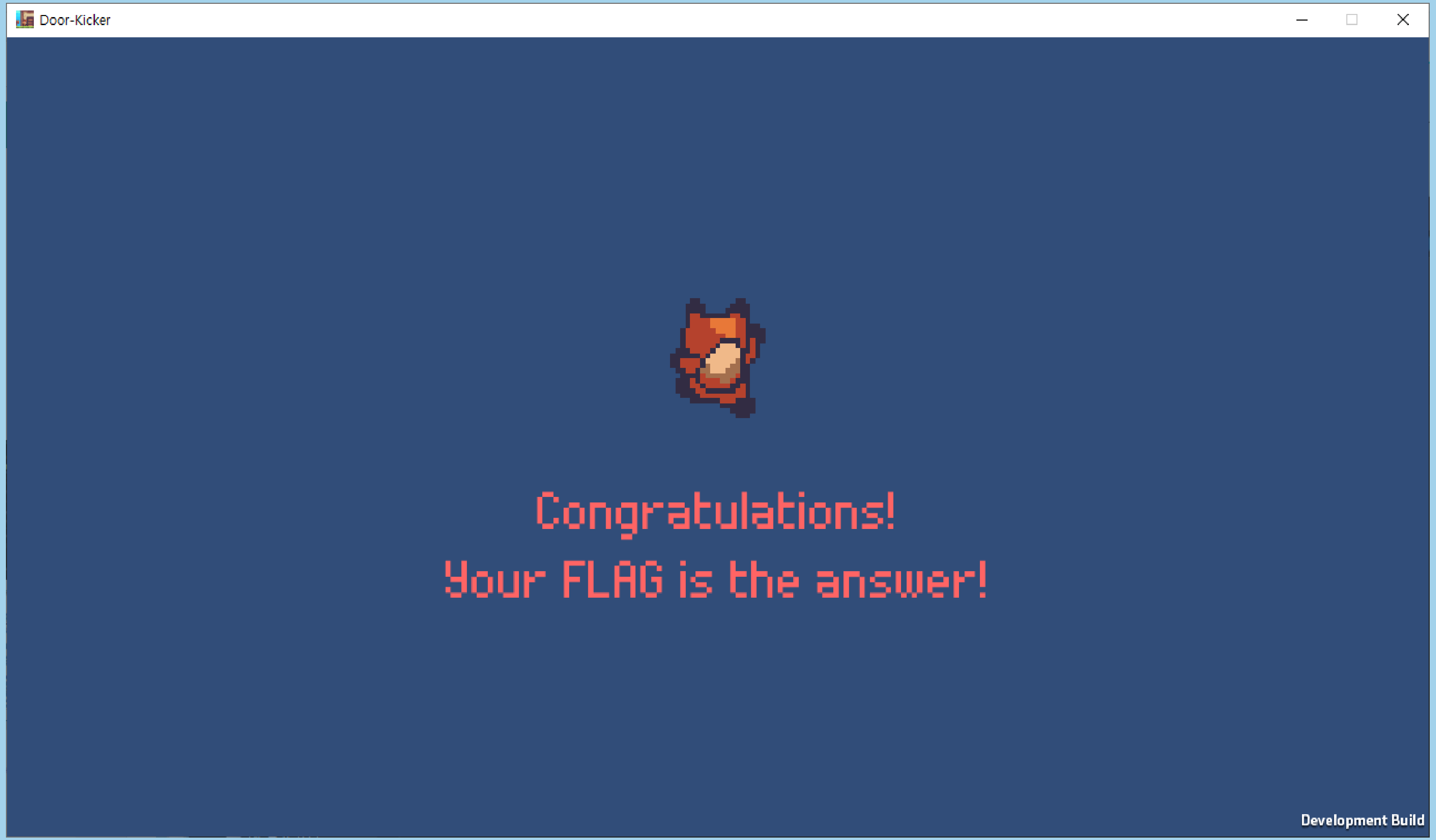
05 I MONO 분석

```
private void OnKickTheDoor()
{
    //bool flag2 = !File.Exists(Application.streamingAssetsPath + "/flag");
    if (true)
    {
        string flag = File.ReadAllText(Application.streamingAssetsPath + "/flag");
        bool flag3 = GameManager.CheckFlag(Marshal.StringToHGlobalAnsi(flag));
        if (flag3)
        {
            SceneManager.LoadScene("Success");
        }
```

강제로 파일을 불러오게 하여 스트리밍에셋폴더의 경로를 확인

05 I MONO 분석



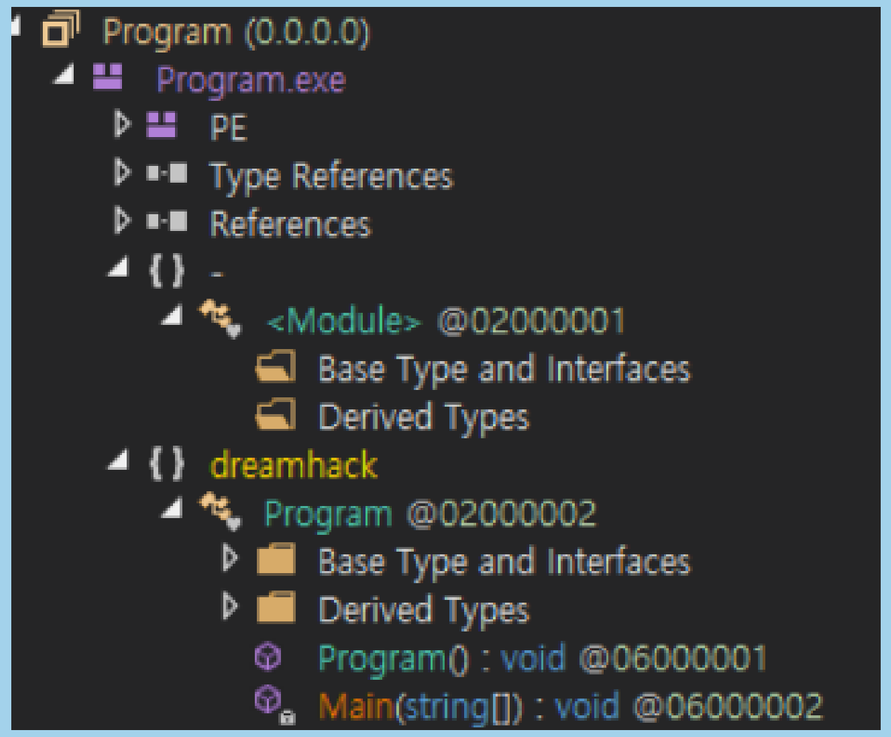


실제 플래그 값을 확인하는 CheckFlag.dll 분석후
정확한 경로 내 flag 파일에 연산한 값을 넣어주면 성공 화면 출력

06 I .NET Framework

드림핵 워게임으로 실습

06 I .NET Framework



.NET의 경우 바로 실행파일을 열면 디컴파일된 코드를 볼 수 있음

06 I .NET Framework

```
private static void Main(string[] args)
{
    int[] array = new int[]
    {
        148,
        27,
        14,
        27,
        34,
        25,
        10,
        30,
        48,
        33,
        23,
        15,
        19,
        43,
        46,
        30,
        23,
        15,
        19,
        43,
        33,
        34,
        60,
        54,
```

```
ator ilGenerator = methodBuilder.GetILGenerator();
LocalBuilder localBuilder = ilGenerator.DeclareLocal(typeof(char[]));
LocalBuilder localBuilder2 = ilGenerator.DeclareLocal(typeof(int[]));
LocalBuilder localBuilder3 = ilGenerator.DeclareLocal(typeof(int));
LocalBuilder localBuilder4 = ilGenerator.DeclareLocal(typeof(int));
LocalBuilder localBuilder5 = ilGenerator.DeclareLocal(typeof(int));
LocalBuilder localBuilder6 = ilGenerator.DeclareLocal(typeof(int));
LocalBuilder localBuilder7 = ilGenerator.DeclareLocal(typeof(int));
Label label = ilGenerator.DefineLabel();
Label label2 = ilGenerator.DefineLabel();
Label label3 = ilGenerator.DefineLabel();
Label label4 = ilGenerator.DefineLabel();
Label label5 = ilGenerator.DefineLabel();
Label label6 = ilGenerator.DefineLabel();
Label label7 = ilGenerator.DefineLabel();
ilGenerator.Emit(OpCodes.Ldarg_1);
ilGenerator.Emit(OpCodes.Callvirt, typeof(string).GetMethod("ToCharArray", new Type[0]));
ilGenerator.Emit(OpCodes.Stloc_0);
ilGenerator.Emit(OpCodes.Ldc_I4_S, 32);
ilGenerator.Emit(OpCodes.Newarr, typeof(int));
ilGenerator.Emit(OpCodes.Stloc_1);
ilGenerator.Emit(OpCodes.Ldarg_1);
ilGenerator.Emit(OpCodes.Callvirt, typeof(string).GetMethod("get_Length", new Type[0]));
ilGenerator.Emit(OpCodes.Stloc_2);
ilGenerator.Emit(OpCodes.Ldc_I4_0);
ilGenerator.Emit(OpCodes.Stloc_3);
ilGenerator.Emit(OpCodes.Br, label);
ilGenerator.MarkLabel(label2);
```

```
ilGenerator.Emit(OpCodes.Ldloc_S, 4);
ilGenerator.Emit(OpCodes.Ldc_I4_0);
ilGenerator.Emit(OpCodes.Bge, label4);
ilGenerator.Emit(OpCodes.Ldc_I4_0);
ilGenerator.Emit(OpCodes.Stloc_S, 5);
ilGenerator.Emit(OpCodes.Ldc_I4_0);
ilGenerator.Emit(OpCodes.Stloc_S, 6);
ilGenerator.Emit(OpCodes.Br, label5);
ilGenerator.MarkLabel(label7);
ilGenerator.Emit(OpCodes.Ldloc_1);
ilGenerator.Emit(OpCodes.Ldloc_S, 6);
ilGenerator.Emit(OpCodes.Ldelem_I4);
ilGenerator.Emit(OpCodes.Ldarg_2);
ilGenerator.Emit(OpCodes.Ldloc_S, 6);
ilGenerator.Emit(OpCodes.Ldelem_I4);
ilGenerator.Emit(OpCodes.Bne_Un, label6);
ilGenerator.Emit(OpCodes.Ldloc_S, 5);
ilGenerator.Emit(OpCodes.Ldc_I4_1);
ilGenerator.Emit(OpCodes.Add);
ilGenerator.Emit(OpCodes.Stloc_S, 5);
ilGenerator.MarkLabel(label6);
ilGenerator.Emit(OpCodes.Ldloc_S, 6);
ilGenerator.Emit(OpCodes.Ldc_I4_1);
ilGenerator.Emit(OpCodes.Add);
ilGenerator.Emit(OpCodes.Stloc_S, 6);
ilGenerator.MarkLabel(label5);
ilGenerator.Emit(OpCodes.Ldloc_S, 6);
ilGenerator.Emit(OpCodes.Ldloc_2);
ilGenerator.Emit(OpCodes.Blt, label7);
ilGenerator.Emit(OpCodes.Ldloc_S, 5);
```

ilGenerator함수가 엄청 많은 모습

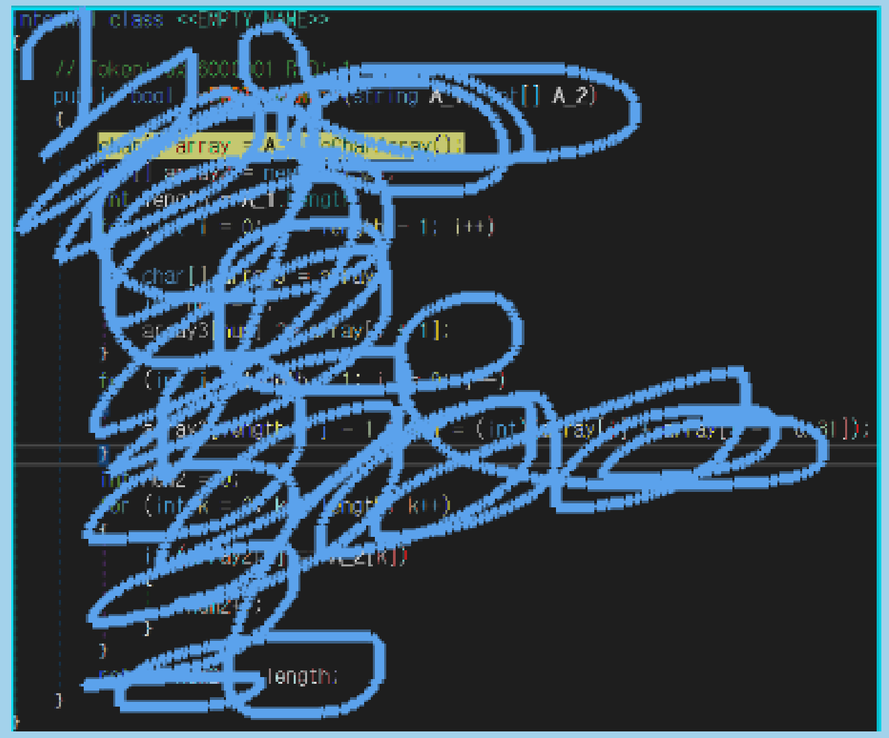
06 I .NET Framework

01 I IL 코드 분석
손발이 나쁘면 머리가 고생한다(?)

02 I IL 디스어셈블러
IL 코드들을 어셈블리어로 변환해주는 툴

03 I 디버깅
머리가 나쁘면 손발이 고생한다

06 I .NET Framework



대단해요! 정답을 맞히셨네요. 문제를 어떻게 해결하셨나요?

풀이 작성하고 포인트 받기 >

5 You shall not pass

3시간 전 · 가중치 100%

+ 200

dnSpy에서 제공하는 디버깅 기능을 사용하여 런타임 코드 진입

감사합니다.

자세한 디버깅 / 역연산 등의 과정은 개인적으로 질문 부탁드립니다