



2023 소프트웨어·인공지능 아카데미 게임핵 강좌

충북교육연구정보원 영재교육원 정보강사 박정진

게임핵 강좌

- 넷마블 틀린그림찾기 핵만들기 -

Starcraft Vision Hack v1.03



사용방법 :

1. 스타크래프트를 실행시킵니다.
2. Starcraft Vision Hack 을 실행시킵니다.
3. "Hack code was Successfully loaded" 라는 메시지가 뜨면, 해킹프로그램이 정상적으로 로딩된 것입니다.
4. 게임 화면에서 "Diplomacy" 버튼을 누릅니다.
5. 동맹 설정 화면이 나오면 바로 "Accept" 버튼을 누릅니다.
6. 잠시후 자신의 시야가 깨질 것입니다. 이 상태가 되면 테란의 건물을 내려놓음으로서 밑에 깔리는 건물을 부술 수 있게 되며, 모든 종족의 각종건물을 원하는 위치에 겹쳐서 지을 수 있게 됩니다.
7. 다시 시야를 켜고 싶다면 4,5번 과정을 반복하십시오.

Copyright(free) 2003 Initiative lab, <http://hackuser.net>
Programmed by dianox (짱~진)

Message :

Starcraft Vision Hack was Successfully Loaded.

☐ Easy mode (Vision hack이 동작중인 동안 맵핵을 켜줍니다.)

Load Vision hack code

Exit hack program



Progress bar for the Engineering Bay construction.



696/950

Terran Engineering Bay

Under Construction

Progress bar for the Engineering Bay construction.

MENU



Replay Progress

Progress bar for the replay.

Elapsed Time: 02:08

Speed: Fastest x 4

















Replay Progress

Elapsed Time: 14:24

Speed: Fastest

Slow Down

MENU

Starcraft vision hack

❖ 아이디어 구상

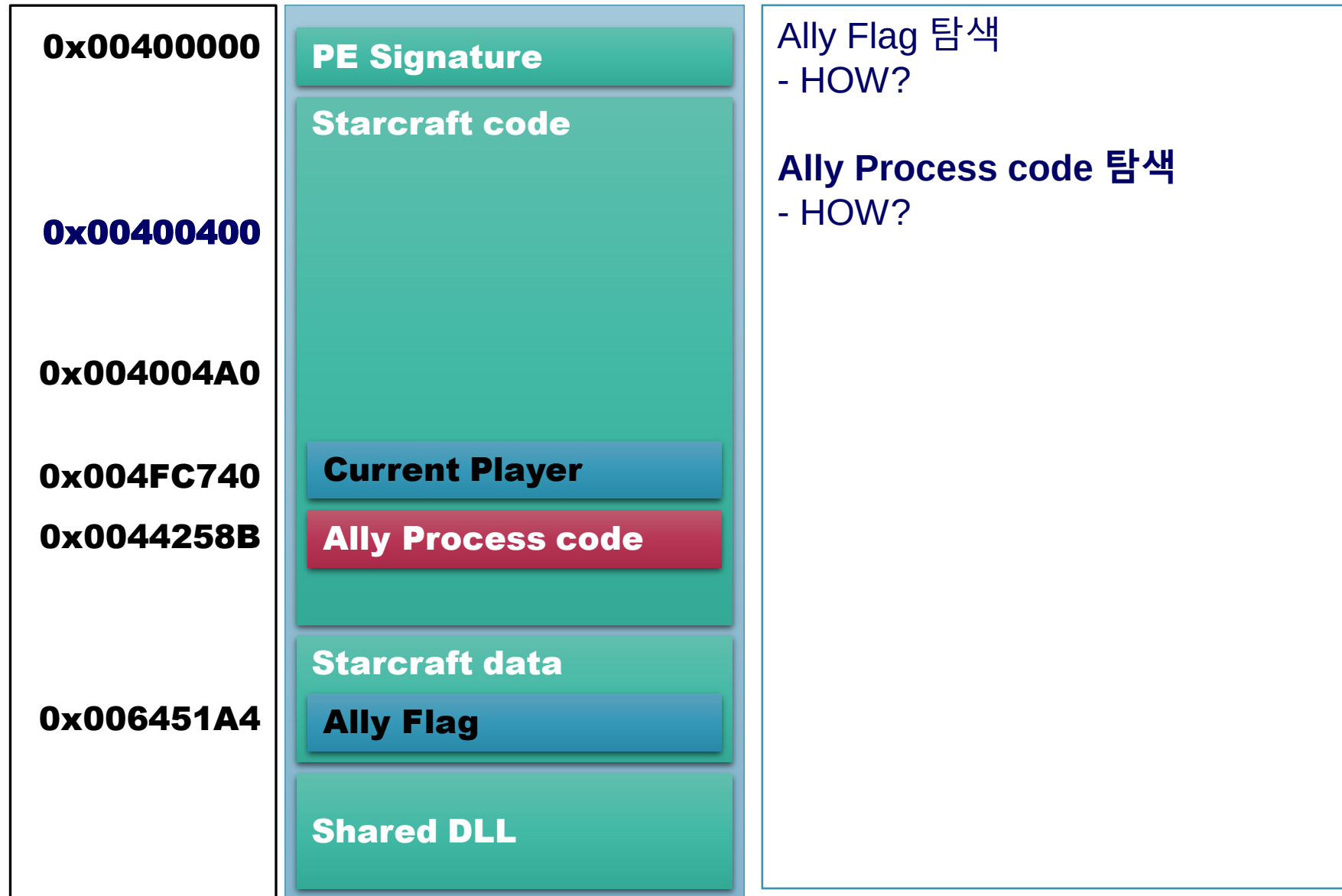
- 어떻게 저러한 플레이가 가능한가?
- Vision 을 조작하라.
- Use map setting 게임에서만 가능한가?
- Melee 게임에서 게이머의 의지대로 Vision을 조작하는 방법은?
- 방법을 찾을 수 없다면 프로그램으로 제작하라.

Starcraft vision hack

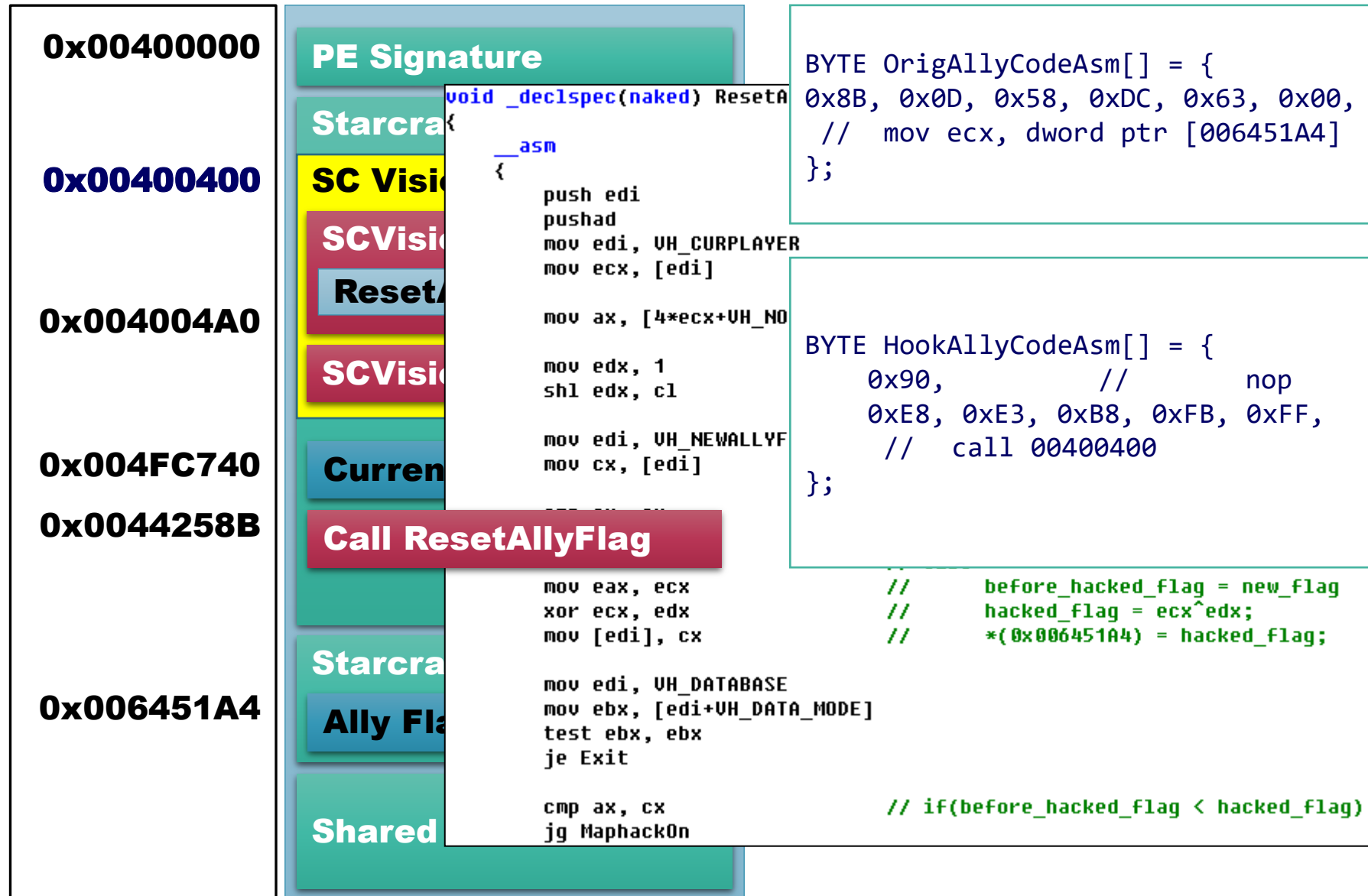
❖ 핵 프로그램 개발 순서

- Vision 정보가 저장되는 메모리를 찾아낸다.
- Vision 정보가 언제 업데이트되는가 포착한다.
- Vision 정보를 업데이트 하는 코드를 찾아낸다.
- Vision 정보를 업데이트 하는 코드를 후킹 하는 코드를 작성한다.
- Vision 정보를 적절하게 가공하는 코드를 작성한다.

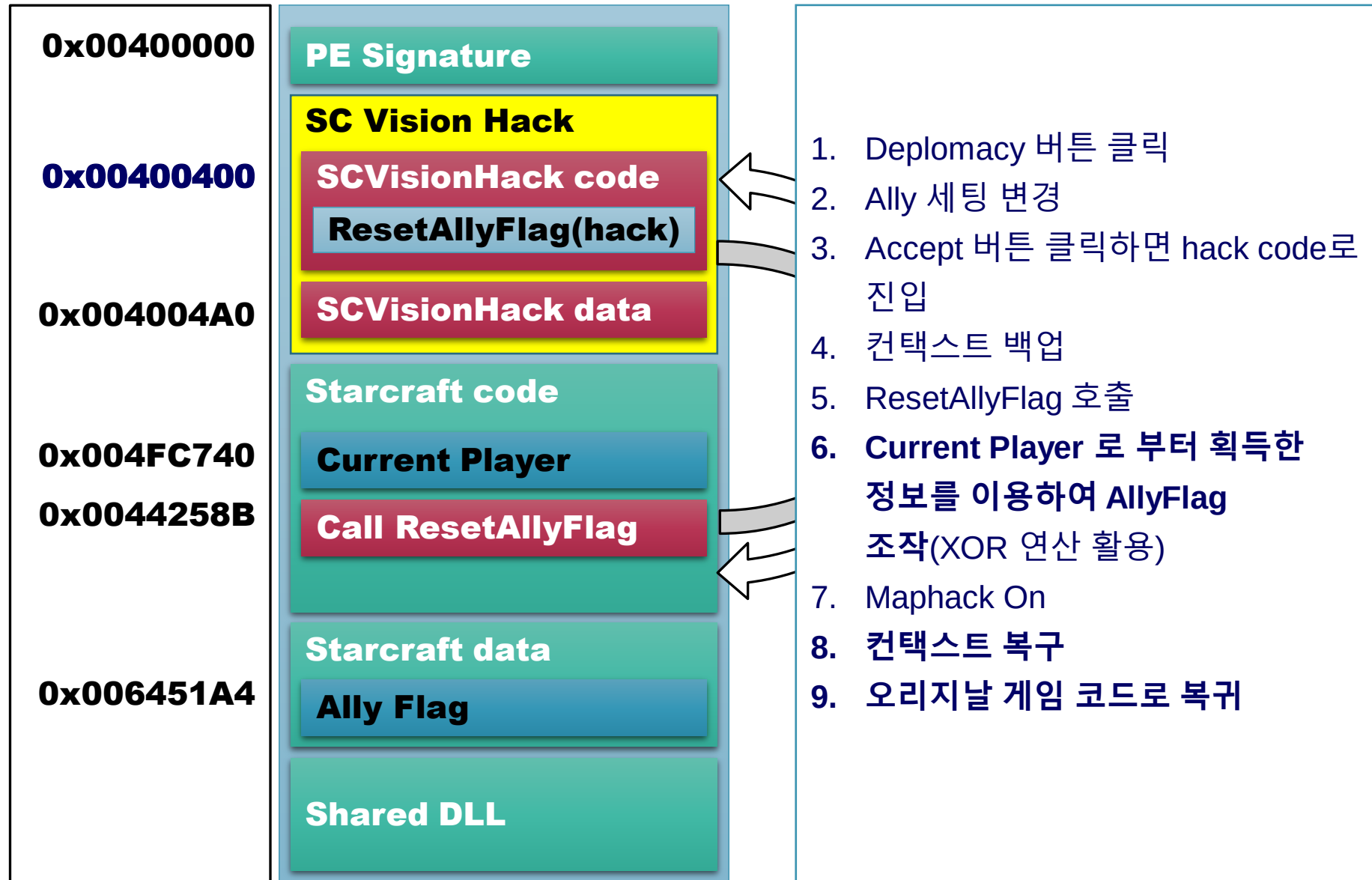
Starcraft.exe hacked memory map



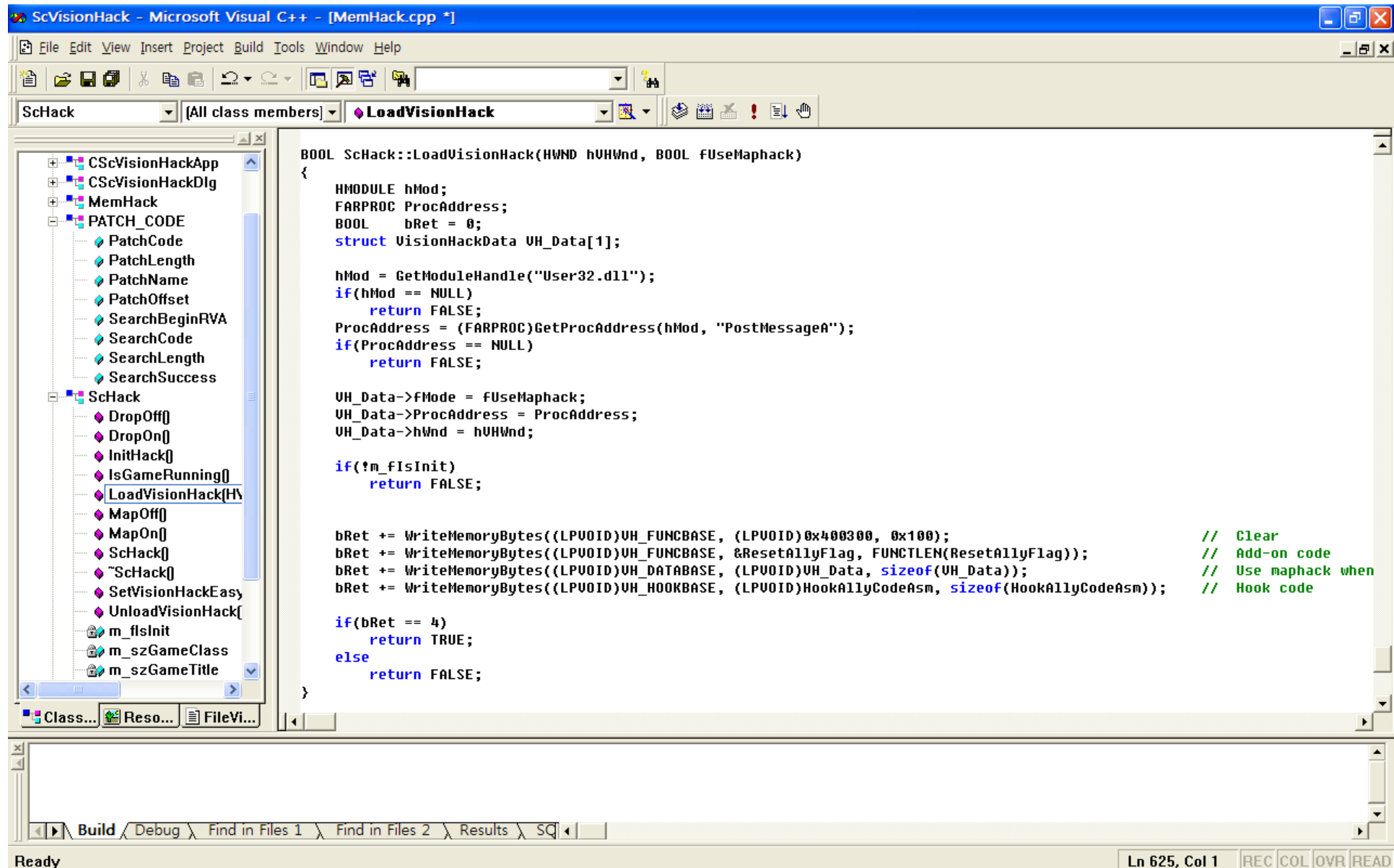
Starcraft.exe hacked memory map

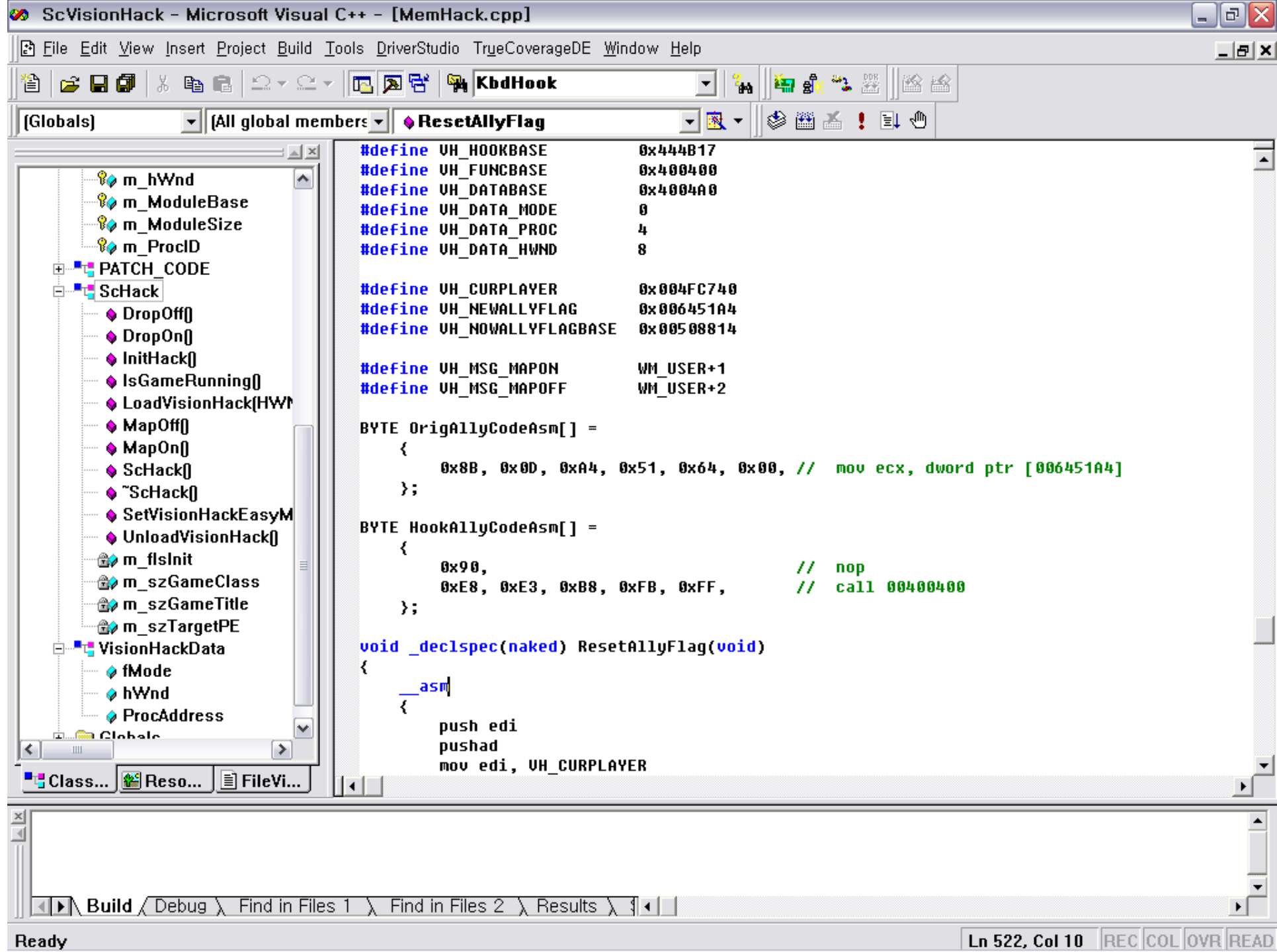


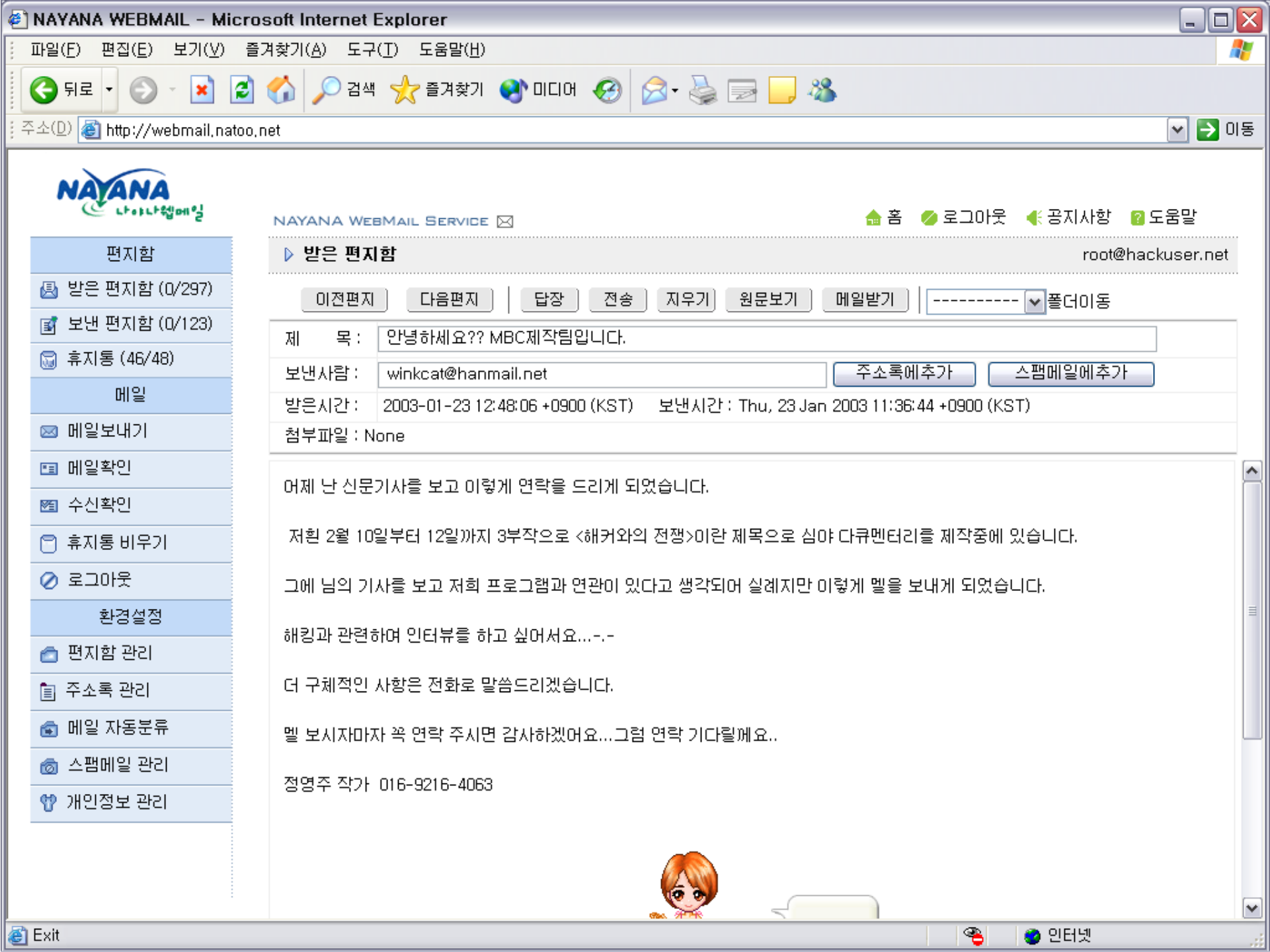
Starcraft.exe hacked memory map



ScHack::LoadVisionHack







STAR CRAFT™

Welcome to
battle.net™

Channel

Create

Join

Ladder

Quit

<Nc> 음
<Nc> ㅋㅋ
<Afx> 자자 다시
<Afx> Clan Afx 마니들 놀러오세여 ㄷ.ㄷ
Caln Afx 부마스터 김진호ㅍ
Clan Afx 乃>

<Lx4.5> ㅋㅋ
<Lx4.5> 하미~~ 방가여... ^__^ 쟁진미예옴.
<Nc> 저 나 슷
<Afx> 글들적어봐여 --;
<Afx> 이상한글 나오면 보기 안좋자나여 ㅋㅋ
<Nc> Clan Afx And Clan Play Forever~♡>
<Afx> Clan Afx 마니들 놀러오세여 ㄷ.ㄷ
Caln Afx 부마스터 김진호ㅍ
Clan Afx 乃>

<Lx4.5> --;
<Lx4.5> 하미~~ 방가여... ^__^ 쟁진미예옴.
<Nc> 항상 행복합시다~~☆☆☆☆☆☆☆☆

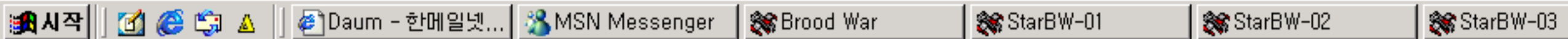
Clan Afx (21)

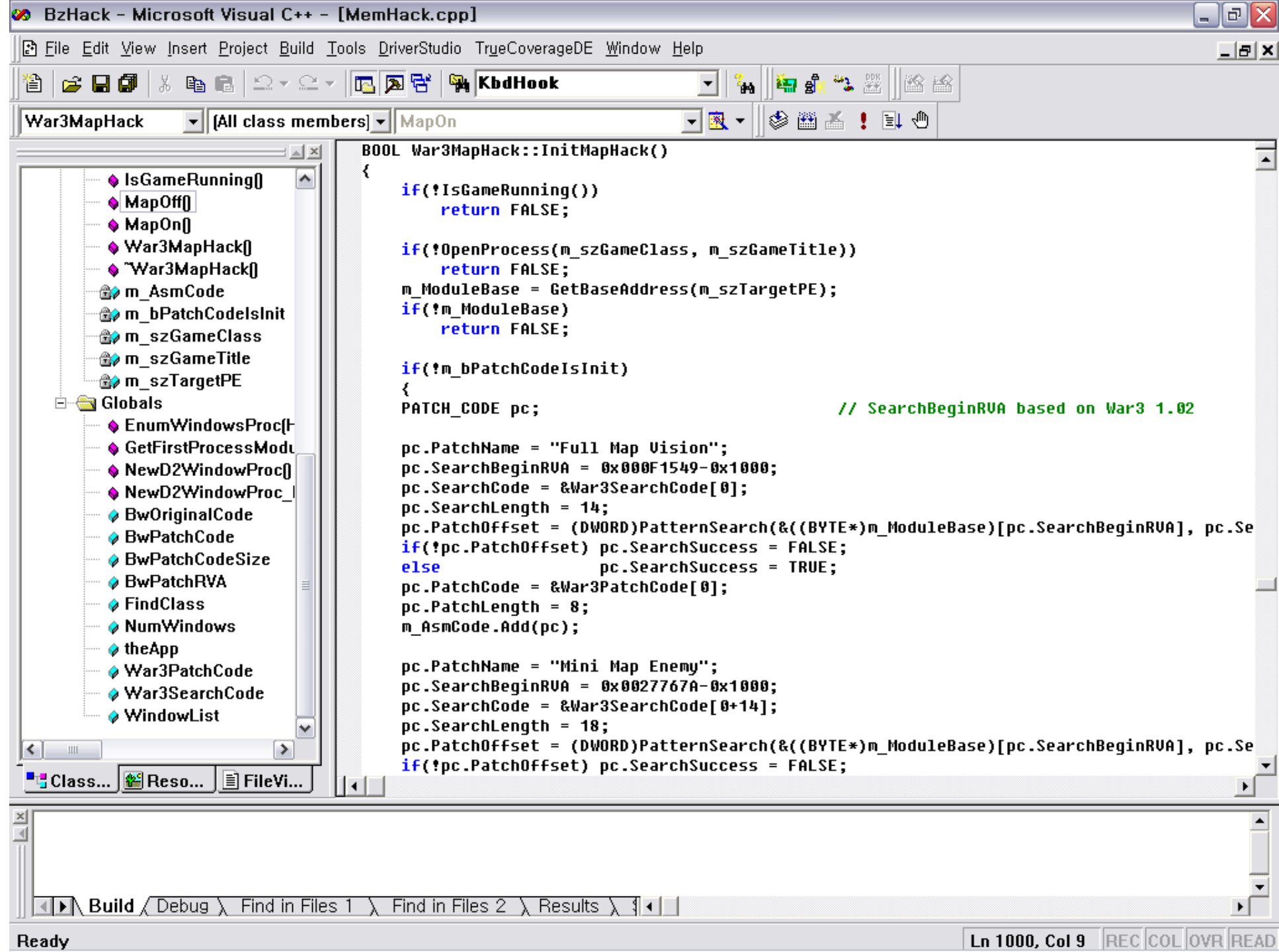
 Afx
9999 Software
9999 hackuser.net
9999 ZOLRA_GOSU
9999 TheSailorMoon |
7195 TheGoddess |
7192 CSocket |
7181 Orgling |
7183 CDialog |
7006 Lx4.5 |
3760 Nc |
3584 ToYa |
3333 JuGulRae |

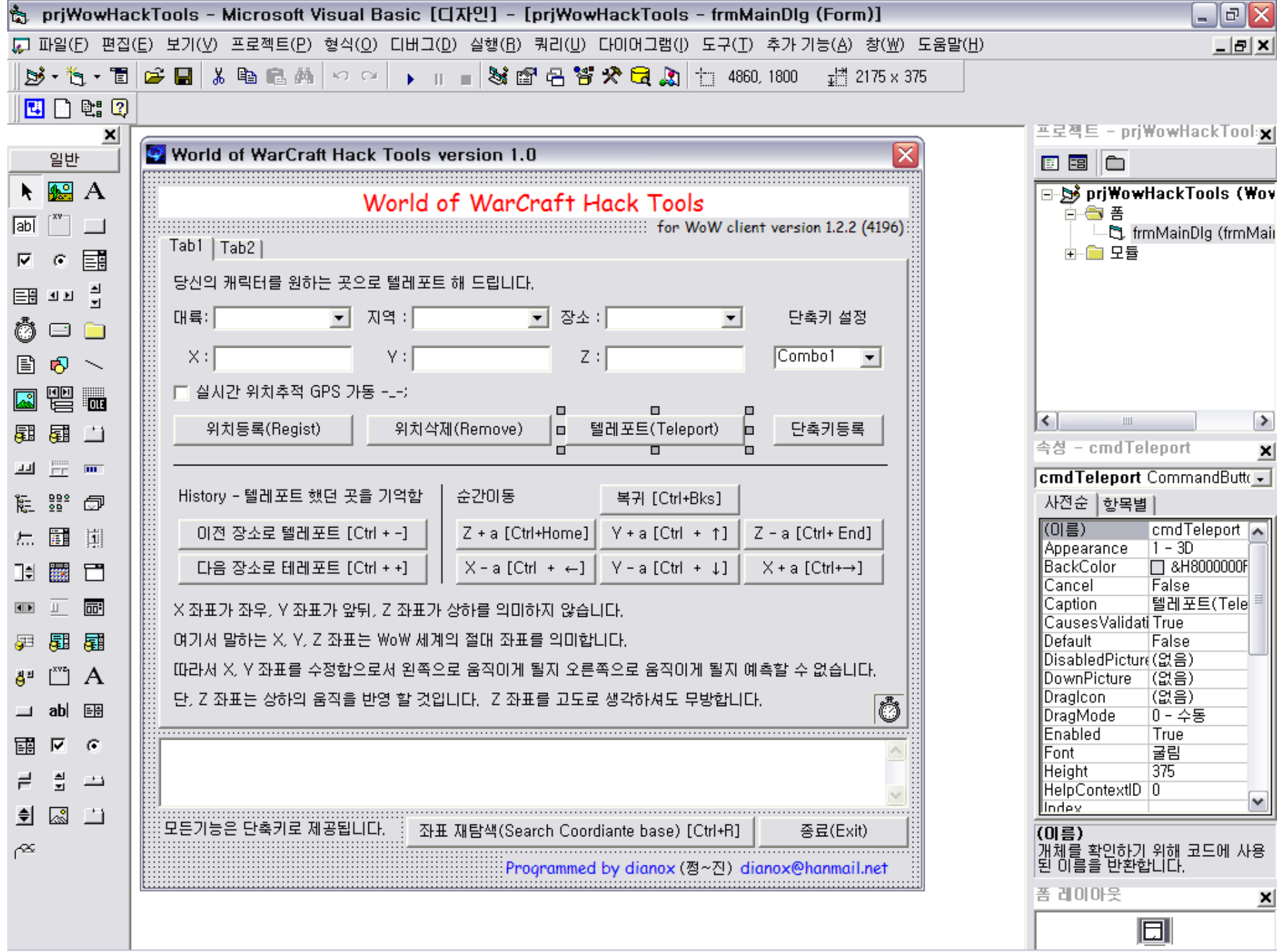
Send

Whisper










World of WarCraft Hack Tools version 1.2
✕

World of WarCraft Hack Tools

for WoW client version 1.2.4 (4222)

텔레포터 | 위치저장소 | 무빙부스터 | About...

당신의 캐릭터를 원하는 곳으로 텔레포트 해 드립니다.

대륙: 마제로스 |
 지역: 가시덤불골짜기 |
 장소: 가시덤불 봉우리 |
 단축키 설정

X: -13467,68 |
 Y: 322,1115 |
 Z: 32,26065 |
 없음

☐ 실시간 위치추적 GPS 가동 ~-~

위치등록(Register)
위치삭제(Remove)
텔레포트(Teleport)
단축키등록

History - 텔레포트 했던 곳을 기억함

이전 장소로 텔레포트 [Ctrl + -]
다음 장소로 테레포트 [Ctrl + +]

순간이동

복귀 [Ctrl+Bks]

Z + a [Ctrl+Home]
Y + a [Ctrl + ↑]
Z - a [Ctrl+ End]

X - a [Ctrl + ←]
Y - a [Ctrl + ↓]
X + a [Ctrl+→]

X 좌표가 좌우, Y 좌표가 앞뒤, Z 좌표가 상하를 의미하지 않습니다.

여기서 말하는 X, Y, Z 좌표는 WoW 세계의 절대 좌표를 의미합니다.

따라서 X, Y 좌표를 수정함으로써 왼쪽으로 움직이게 될지 오른쪽으로 움직이게 될지 예측할 수 없습니다.

단, Z 좌표는 상하의 움직임을 반영 할 것입니다. Z 좌표를 고도로 생각하셔도 무방합니다.

WoW 프로세스를 찾았습니다(WoW process detected...OK.)

Coordinate base 탐색을 완료하였습니다.

모든기능은 단축키로 제공됩니다.

좌표 재탐색(Search Coordiante base) [Ctrl+R]
종료(Exit)

Programmed by dianox (짱~진) dianox@hanmail.net

유영귀

38

HP

MP



이영귀를 제압했습니다. (승려인 유영귀의 종격)

[1. 공개] [무공] : 노노

[1. 공개] [무공] : 양씨에 가지마요

이영귀를 제압했습니다. (승려인 유영귀의 종격)

[1. 공개] [무공] : 공채결성했어

[1. 공개] [양요는도적] : <장편사는> 할호군! 하여도 무덤에서 가지않서 좋대요

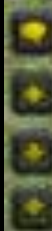
[1. 공개]





유령귀

38



1. [중간] 유령귀가 있는 데까지 성령 못 구했어다...
 2. [중간] 유령귀가 있는 데까지 성령 못 구했어... 무릎 꿇거나 폭력만 하는 자
 3. [중간] 유령귀가 있는 데까지 성령 못 구했어... 무릎 꿇거나 폭력만 하는 자
 4. [중간] 유령귀가 있는 데까지 성령 못 구했어... 무릎 꿇거나 폭력만 하는 자

신비한 화살 4 개월
 마나 320 소요 거리 30미터
 정신 집중
 5초에 걸쳐 적에게 마법 화살을 날려 해 소하
 다. 각각 85~86의 비전 피해를 입힙니다.





유령귀

38



[1. 공개] [성기사장]: 악마성별카실론 오십니다. 허접사 성기사. 뭐가중~~~

[1. 공개] [아마르헤]: [49] 타락후종의 전설. 크리스천적자사문. 타지종. 형주

생명

[1. 공개] [아마르헤]: 도착합니다.

[1. 공개] [남은도희]: 눈장투라는... 동호군]공대. 장강현상문---

신비한 화실

마나 520

정신 집중

5초에 걸쳐 적에게 마법 화실을 날려 제 소파

다. 각각 85~86의 적인 피해를 입힙니다.



37 앞집여자

해프고비온후

오늘처음이다

듀마

Honest



[뮤아티스]님에게 귓속말: 저는 상담을 요청하지 않았습니다.
 <GM>[뮤아티스]님의 귓속말: 현재 [돌주먹일족 소굴]에서 [스트롱 가드 요새]
 로 어떠한 방법으로
 [1. 공개] [죽음의제우스]: [39] 트렐레인의 방어 장치 하실분?
 <GM>[뮤아티스]님의 귓속말: 이동하셨습니다?
 자신의 몫으로 34 코퍼를 분배받았습니다.



37 앞집여자

해프고비온후

오늘처음이다

듀마

Honest

일반 전투

[파티] [듀마]: ㅎㅎㅎ
 <GM>[뮤아티스]님의 귓속말: 만약 해당 사항에 대해 정확한 답변이 해주시지 않으실 경우
 <GM>[뮤아티스]님의 귓속말: 불법 프로그램 사용으로 간주하여
 [파티] [오늘처음이다]: 동개 뒤질번
 <GM>[뮤아티스]님의 귓속말: 해당 계정은 저희측 운영정책에 의거하여 영구 계정조치가 시행됨을 알려드립니다.


37
앞집여자

스트롱가드 요새


해프고비온후
오늘처음이다
듀마
Honest

Honest님이 당신을 부활시키려 합니다.
수락 거절

[1. 공개] [데이빗투]: <못죽여도공격대장>존
<GM>[뮤아티스]님의 귓속말: 모든 계정이 영구 정지 됨을 알려드립니다.
비밀결사대 사제가 겁을 먹고 도망치려고 합니다!
<GM>[뮤아티스]님의 귓속말: 이후에 사항에 대해서는 상담이 어려움을 알려
드리며, 해당 사항에 대해서는 고객센터등을 이용해
[파티] [오늘처음이다]: 여자님 계세요?
<GM>[뮤아티스]님의 귓속말: 상담을 진행하시겠는지요?

1 2 3 4 5 6 7 8 9 0
8
?



15

입력하신 계정은 사용이 중지된 계정입니다. 자세한 정보를
참고하시려면 계정에 등록된 이메일을 확인해 보시기 바랍니다.
게다, 이름

확인

비밀번호

접속

계정 관리

홈페이지

동영상

제작진

사용 약관

아즈사라

종료





modelabc



asdam



Shureck



NetMarble



NetMarble



hun8770



na7610



neosof

2. 큰그림 스크롤 -> 마우스 휠, 스페이스바
☆ <neosof>님 입장 ☆

112

6	00
1	00
2	00

NO ITEM

5



☆ <jinjja1982>님 입장 ☆

neosof> hi

넷마블 틀린그림찾기 게임핵 만들기

개발도구 설치

❖ Visual Studio 2022

- <https://visualstudio.microsoft.com/downloads/>
- Community 버전 선택

Downloads



Visual Studio 2022 | Windows

The most comprehensive IDE for .NET and C++ developers on Windows for building web, cloud, desktop, mobile apps, services and games.

Community

Powerful IDE, free for students, open-source contributors, and individuals

Free download

Professional

Professional IDE best suited to small teams

Free trial

Enterprise

Scalable, end-to-end solution for teams of any size

Free trial

Preview

Get early access to latest features not yet in the main release

[Release notes →](#)

[Compare Editions →](#)

[How to install offline →](#)



ASP.NET 웹 앱 개발

Docker 지원이 포함된 ASP.NET Core, ASP.NET, HTML/JavaScript 및 컨테이너를 사용하여 웹 애플리케이션을 빌...



Python 개발

Python에 대한 편집, 디버깅, 대화형 개발 및 소스 제어입니...



Azure 개발

.NET 및 .NET Framework를 사용하여 클라우드 앱을 개발하...



Node.js 개발

비동기 이벤트 구동 JavaScript 런타임인 Node.js를 사용하...



데스크톱 및 모바일 (5)



.NET Multi-Platform App UI 개발

.NET MAUI와 함께 C#을 사용하여 단일 코드베이스에서...



.NET 데스크톱 개발

.NET 및 .NET Framework와 함께 C#, Visual Basic 및 F#을...



C++를 사용한 데스크톱 개발

MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용...



유니버설 Windows 플랫폼 개발

C#, VB 또는 C++(선택 사항)를 사용하여 유니버설...



설치 세부 정보

- ☒ IntelliCode
- ☒ Just-In-Time 디버거
- ☒ Live Share
- ☒ ML.NET Model Builder
- ☒ Blend for Visual Studio
- ☐ F# 데스크톱 언어 지원
- ☐ PreEmptive Protection - Dotfuscator
- ☐ .NET Framework 4.6.2-4.7.1 개발 도구
- ☐ .NET 이식 가능한 라이브러리 타기팅 팩
- ☐ Windows Communication Foundation
- ☐ SQL Server Express 2019 LocalDB
- ☒ MSIX Packaging Tools
- ☐ JavaScript 진단
- ☐ Windows 앱 SDK C# 템플릿
- ☐ .NET Framework 4.8.1 개발 도구

개별 구성 요소

- ☒ C++ 핵심 기능

위치

C:\Program Files\Microsoft Visual Studio\2022\Professional

지원되지 않는 구성 요소 제거(R)

계속하면 선택한 Visual Studio 버전에 대한 [라이선스](#)에 동의하게 됩니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 [타사 고지 사항](#) 또는 해당 라이선스에 명시된 것처럼 별도로 라이선스가 허용됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

필요한 전체 공간 0B

다운로드하는 동안 설치

닫기(C)

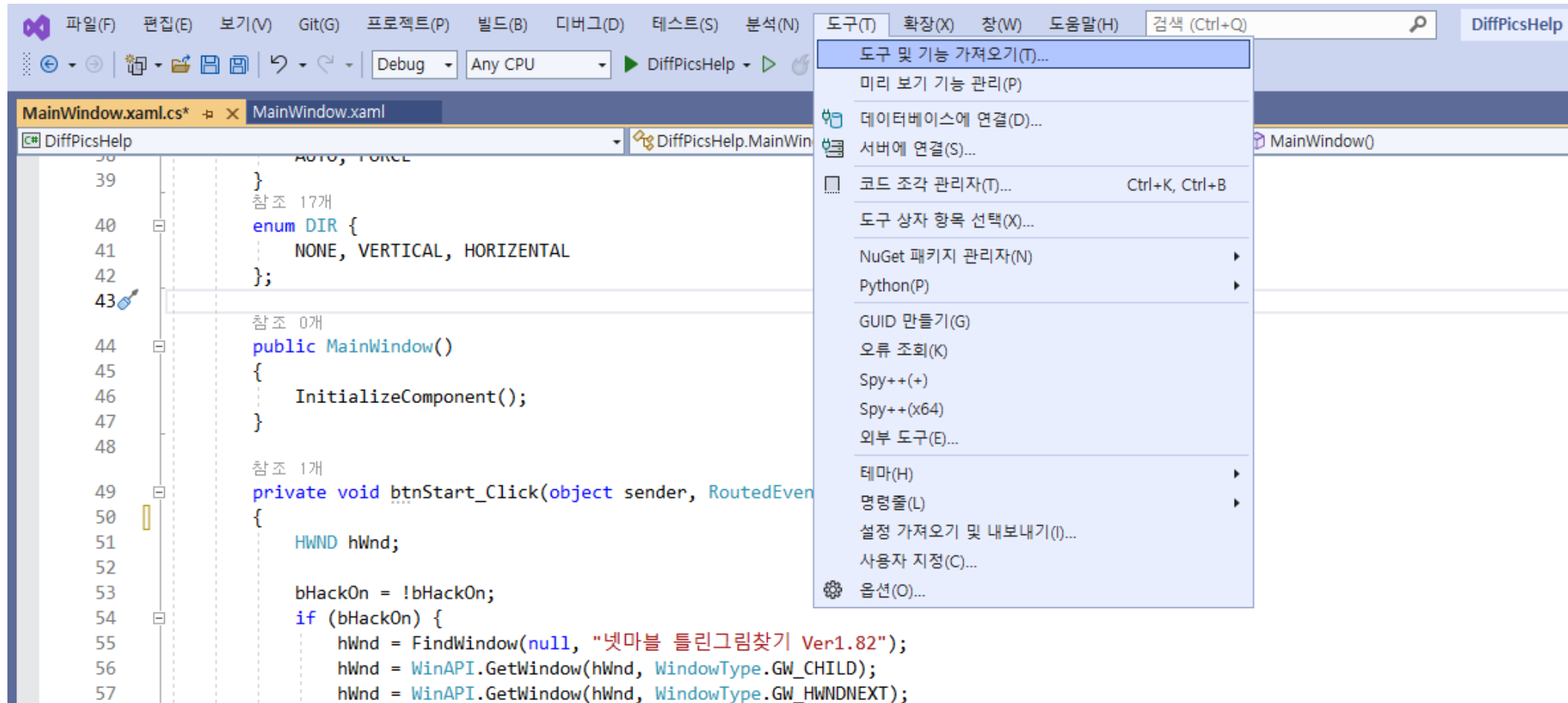
Spy++

❖ Spy++은?

- Spy ++ (SPYXX.EXE)는 시스템 프로세스, 스레드, 창 및 창 메시지를 그래픽으로 볼 수 있는 유틸리티입니다.
- Visual Studio의 Community Edition (오픈 소스 개발, 개인 및 5 명 이하의 팀에게 무료) 및 유료 에디션이 모두 제공됩니다.여기에서 다운로드 할 수도 있습니다.

Spy++

❖ 설치



수정 — Visual Studio Professional 2022 — 17.5.5

워크로드 개별 구성 요소 언어 팩 설치 위치



Docker 지원이 포함된 ASP.NET Core, ASP.NET, HTML/JavaScript 및 컨테이너를 사용하여 웹 애플리케이션을 빌드...



Python 개발

Python에 대한 편집, 디버깅, 대화형 개발 및 소스 제어입니다.



.NET 및 .NET Framework를 사용하여 클라우드 앱을 개발하고 리소스를 만들기 위한 Azure SDK, 도구 및 프로젝트입...



Node.js 개발

비동기 이벤트 구동 JavaScript 런타임인 Node.js를 사용하여 확장 가능한 네트워크 애플리케이션을 빌드합니다.



.NET Multi-Platform App UI 개발

.NET MAUI와 함께 C#을 사용하여 단일 코드베이스에서 Android, iOS, Windows 및 Mac용 앱을 빌드합니다.



C++를 사용한 데스크톱 개발

MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.



.NET 데스크톱 개발

.NET 및 .NET Framework와 함께 C#, Visual Basic 및 F#을 사용하여 WPF, Windows Forms 및 콘솔 애플리케이션을...



유니버설 Windows 플랫폼 개발

C#, VB 또는 C++(선택 사항)를 사용하여 유니버설 Windows 플랫폼용 애플리케이션을 만듭니다.

설치 세부 정보

- ☒ IntelliCode
- ☒ Just-In-Time 디버거
- ☒ Live Share
- ☒ ML.NET Model Builder
- ☒ Blend for Visual Studio
- ☐ F# 데스크톱 언어 지원
- ☐ PreEmptive Protection - Dotfuscator
- ☐ .NET Framework 4.6.2-4.7.1 개발 도구
- ☐ .NET 이식 가능한 라이브러리 타기팅 팩
- ☐ Windows Communication Foundation
- ☐ SQL Server Express 2019 LocalDB
- ☐ MSIX Packaging Tools
- ☐ JavaScript 진단
- ☐ Windows 앱 SDK C# 템플릿
- ☐ .NET Framework 4.8.1 개발 도구

개별 구성 요소

- ☒ C++ 핵심 기능

위치

C:\Program Files\Microsoft Visual Studio\2022\Professional

지원되지 않는 구성 요소 제거(R)

계속하면 선택한 Visual Studio 버전에 대한 [라이선스](#)에 동의하게 됩니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 [타사 고지 사항](#) 또는 해당 라이선스에 명시된 것처럼 별도로 라이선스가 허용됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

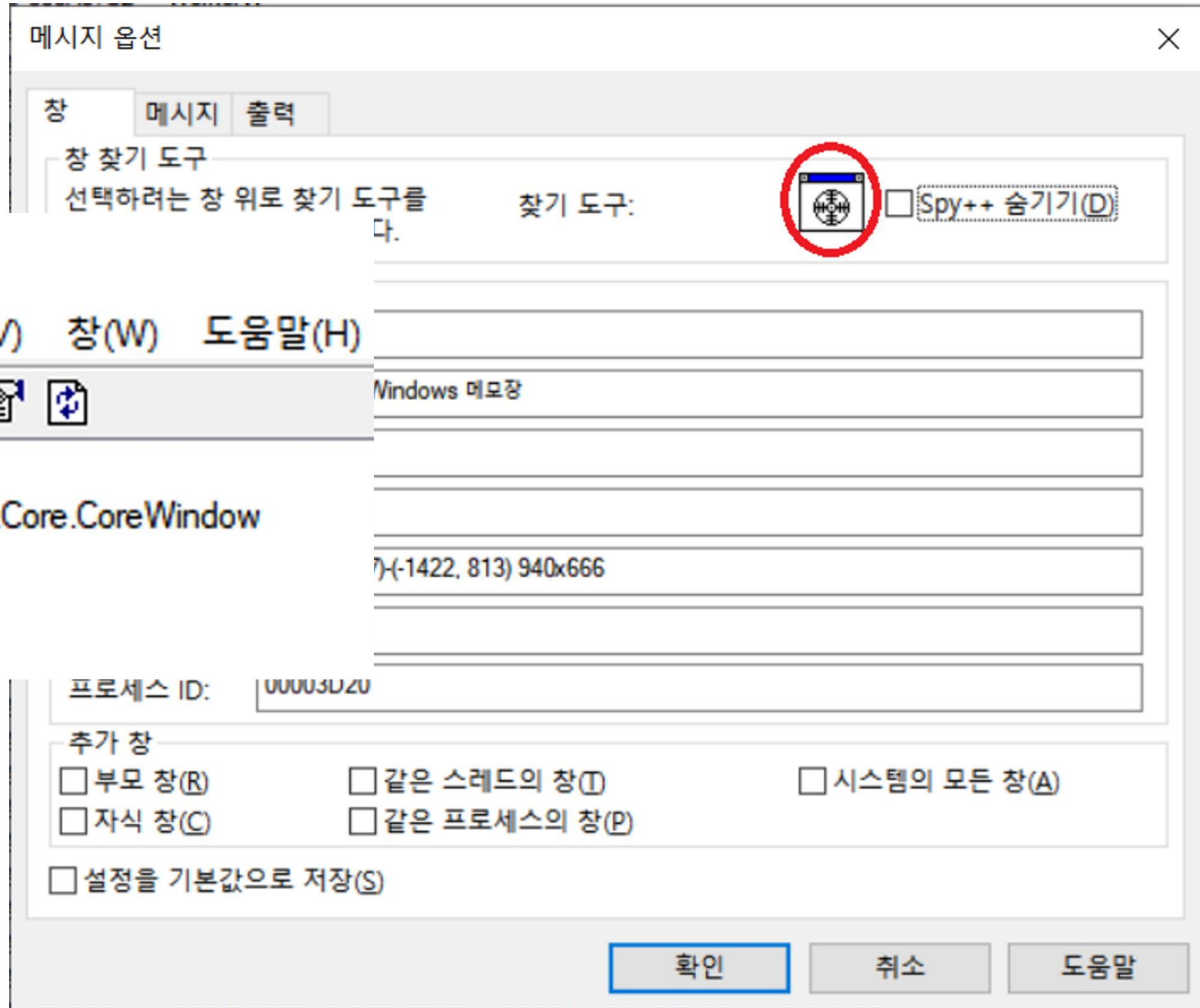
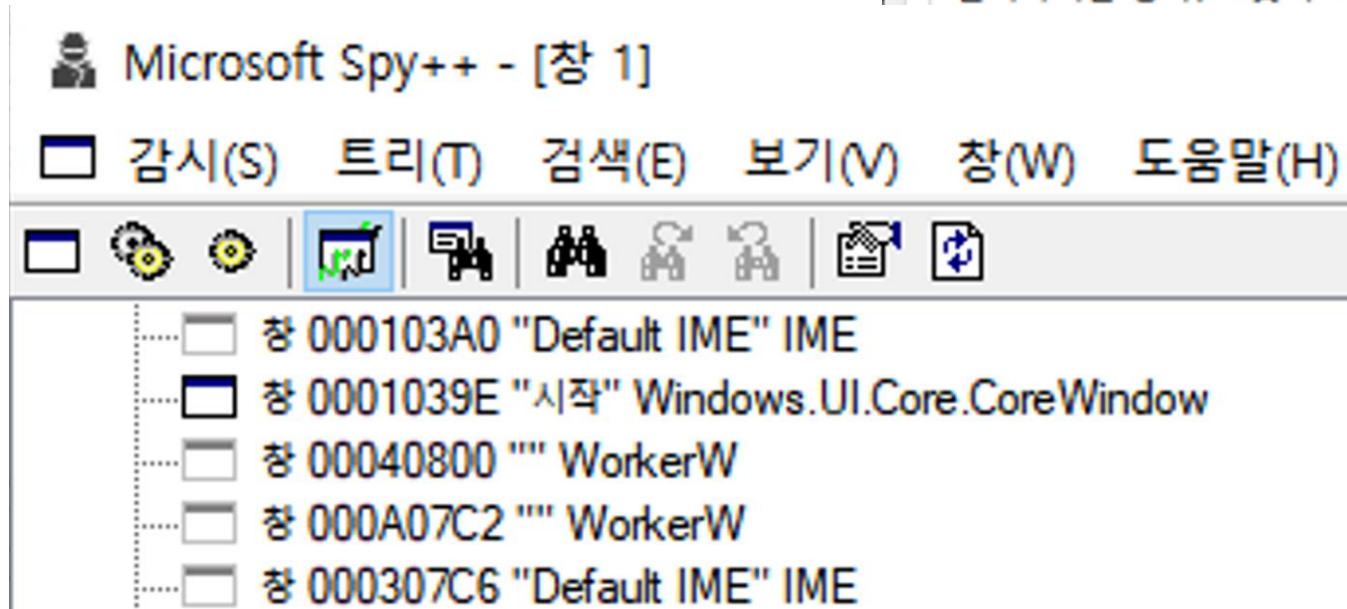
필요한 전체 공간 -2.04GB

다운로드하는 동안 설치

수정(M)

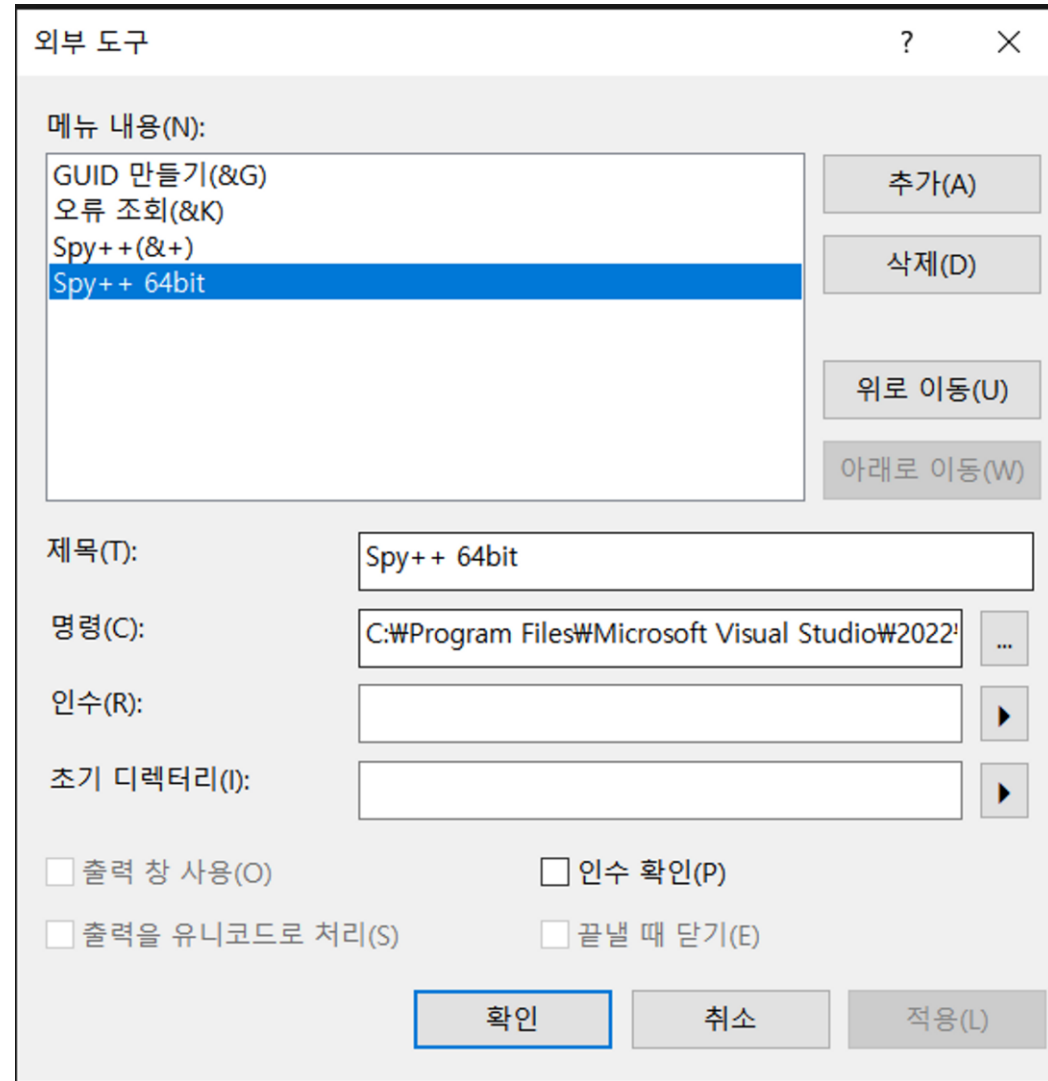
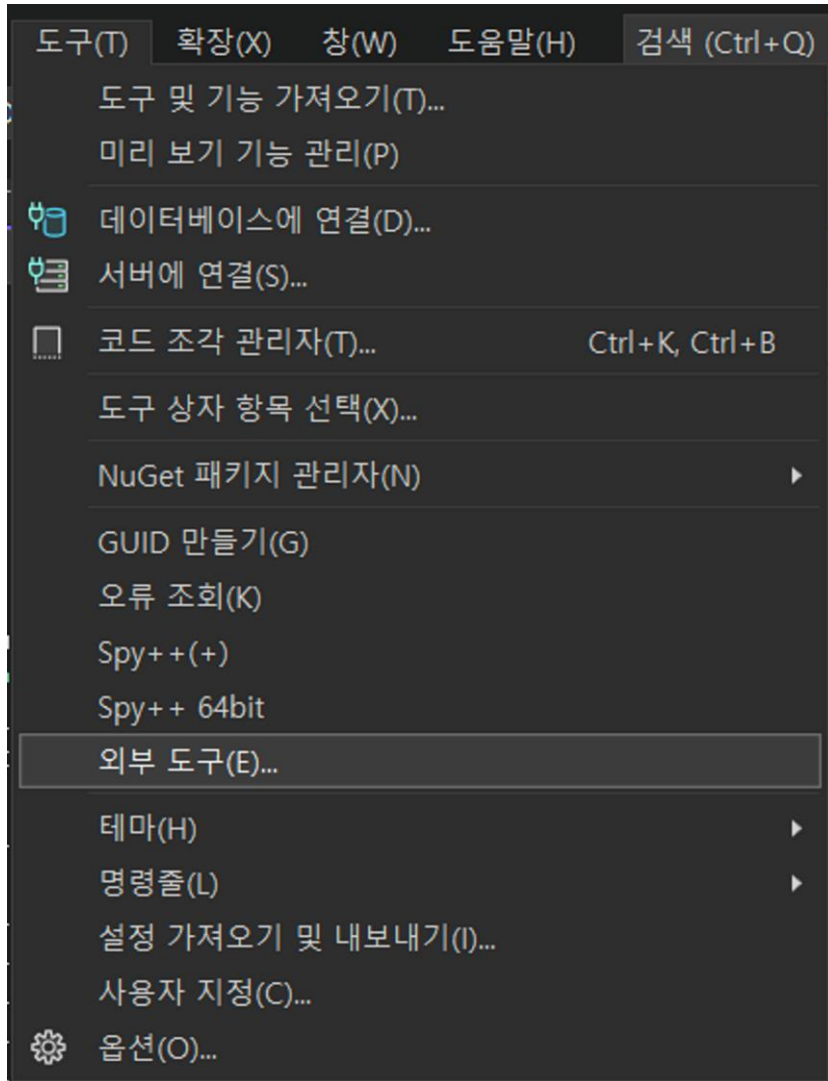
Spy++

❖ 사용법



Spy++

❖ 64bit 버전



Contents

❖ WINAPI

❖ 윈도우 프로그램 구조

- Window, Handle
- DC, Paint

❖ WINAPI

- FindWindow, GetDC, ReleaseDC,
- GetPixel, SetPixel, Rectangle
- CreateCompatibleBitmap, BitBlt
- SelectObject, DeleteObject

개념 정의

❖ Windows API

- Application Programming Interface
- 프로그램에서 직접 운영 체제와 상호 작용할 수 있도록 제공되는 기능들의 집합

❖ HANDLE

- 핸들이란 운영체제가 실행되고 있는 프로그램들을 구별하기 위한 정수 값

❖ DC

- 윈도우가 디스플레이와 프린터에 대한 액세스를 관리하는데 사용하는 도구

❖ BitMap

- 메모리에 저장되는 이미지 맵

WINAPI - BitBlt

❖ 원형

- `BOOL BitBlt(HDC hDC,int X,int Y,int nWidth,int nHeight, HDC hSrcDC,int XSrc,int ySrc,DWORD dwROP);`

❖ 인수

- `hDC`: 비트맵이 복사될 DC
- `X,Y,nWidth,nHeight`: 비트맵이 복사될 위치의 좌상단 좌표와 폭, 높이, 이 값은 논리적 좌표값이다.
- `hSrcDC`: 복사될 비트맵을 가지고 있는 DC. 만약 `dwROP`가 소스가 필요없는 값이라면 `NULL`이 될 수도 있다.
- `XSrc,YSrc`: 복사될 비트맵의 좌측 상단 좌표이며 복사처와 마찬가지로 논리적 좌표값이다. 이 값이 0일 경우 전체 비트맵을 복사한다. 복사원의 폭과 높이는 복사처의 `nWidth, nHeight`가 적용된다.
- `dwROP`: ROP코드, 즉 브러쉬와 복사원, 복사처의 비트맵 색상이 논리 연산될 방법을 지정한다. 모두 256개의 가능한 값이 있지만 실제로 의미를 갖는 것은 다음 15개이며 매크로 상수가 정의되어 있다. 이외의 ROP코드가 필요할 경우에는 상수를 직접 사용해야 한다. 아래 표에서 S는 복사원 (Source), D는 복사처(Destination), P는 패턴(Pattern, 즉 브러쉬)를 의미한다.

❖ 리턴값

- 이상 없이 수행했으면 0이 아닌 값을 리턴하고 에러가 발생했을 경우 0을 리턴한다 .

새 프로젝트 만들기

최근 프로젝트 템플릿(R)

WPF 애플리케이션

C#

템플릿 검색(Alt+S)(S)



모두 지우기(C)

모든 언어(L)

Windows

모든 프로젝트 형식(T)



WPF 애플리케이션

.NET WPF 애플리케이션 만들기 프로젝트

Visual Basic

Windows

데스크톱



WPF 클래스 라이브러리

.NET Core WPF 애플리케이션을 대상으로 하는 클래스 라이브러리 만들기 프로젝트

C#

Windows

데스크톱

라이브러리



WPF 클래스 라이브러리

.NET Core WPF 애플리케이션을 대상으로 하는 클래스 라이브러리 만들기 프로젝트

Visual Basic

Windows

데스크톱

라이브러리



WPF 앱(.NET Framework)

Windows Presentation Foundation 클라이언트 애플리케이션

C#

XAML

Windows

데스크톱



WPF 앱(.NET Framework)

Windows Presentation Foundation 클라이언트 애플리케이션

Visual Basic

XAML

Windows

데스크톱



NUnit 테스트 프로젝트

Windows, Linux 및 MacOS의 .NET에서 실행할 수 있는 NUnit 테스트를 포함하는 프로젝트

뒤로(B)

다음(N)

새 프로젝트 구성

WPF 앱(.NET Framework)

C#

XAML

Windows

데스크톱

프로젝트 이름(I)

DifferPicHeler

위치(L)

C:\Temp\

솔루션 이름(M) ⓘ

DifferPicHeler

☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

프레임워크(F)

.NET Framework 4.7.2

"C:\Temp\DifferPicHeler\DifferPicHeler"에 프로젝트이(가) 만들어집니다.

뒤로(B)

만들기(C)

C# 에서 Window API 정의 추가

❖ API 정의 파일 다운로드

▪ <https://algo.datahub.pe.kr/src/WinAPI.cs>

❖ 프로젝트에 WinAPI.cs 파일 추가

❖ 네임스페이스 변경

지정된 윈도우에 그리기 테스트

❖ Window_Loaded 이벤트 핸들러 코딩

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    HWND hWnd = FindWindow(null, "*제목 없음 - Windows 메모장");
    HDC hDC = GetDC(hWnd);
    PAINTSTRUCT ps = new PAINTSTRUCT();
    RECT rect = new RECT();

    GetClientRect(hWnd, out rect);
    //박스 그리기
    Rectangle(hDC, rect.Left + 10, rect.Top + 10, rect.Right - 10, rect.Bottom - 10);
    //라운드 박스 그리기
    RoundRect(hDC, rect.Left + 20, rect.Top + 20, rect.Right - 20, rect.Bottom - 20, 20, 20);
    //원 그리기
    Ellipse(hDC, rect.Left + 30, rect.Top + 30, rect.Right - 30, rect.Bottom - 30);
    LineTo(hDC, rect.Right - 10, rect.Bottom - 10);
    EndPaint(hWnd, out ps);
}
```

WINAPI - FindWindow

❖ 원형

- HWND FindWindow(LPCTSTR lpClassName, LPCTSTR lpWindowName);

❖ 인수

- lpClassName : 검색에 사용할 윈도우 클래스 이름, 또는 아톰값. NULL이면 모든 윈도우 클래스를 검색한다.
- lpWindowName : 검색에 사용할 윈도우의 캡션 문자열. NULL이면 모든 캡션의 윈도우를 검색한다.

❖ 리턴 값

- 리턴 성공하면 검색한 윈도우의 핸들을 리턴하며 실패했을 경우 NULL을 리턴

❖ 설명

- 윈도우 클래스와 캡션으로 윈도우를 검색하여 핸들을 얻는다. 윈도우 클래스 또는 윈도우 캡션 둘 중 하나의 조건으로 검색할 수 있다. 차일드 윈도우는 검색할 수 없으며 탑 레벨 윈도우만 검색 대상이 된다. 윈도우 클래스와 캡션 문자열은 대소문자를 가리지는 않지만 부분 문자열을 검색할 수는 없다. 따라서 캡션이 가변적으로 변하는 윈도우를 검색하는 목적으로는 이 함수를 사용할 수 없으며 우연히 같은 캡션을 가지는 윈도우가 있을 경우 잘못된 검색을 할 위험도 있다.

WINAPI - GetDC

❖ 원형

- HDC GetDC(HWND hWnd);

❖ 인수

- hWnd: DC를 구하고자 하는 윈도우 핸들. NULL일 경우 전체 화면에 대한 DC가 구해진다 .

❖ 리턴값

- 성공하면 DC핸들을 리턴해 준다. 실패 시 NULL을 리턴한다

❖ 설명

- 디바이스 컨텍스트는 디스플레이 또는 프린터와 같은 디바이스의 그리기 특성에 대한 정보를 포함하는 Windows 데이터 구조입니다.
- 윈도우 환경에서 화면이나 프린터로 출력을 하려면 DC를 먼저 구해야 한다.
- DC를 구하는 일반적인 방법은 두 가지가 있는데 WM_PAINT 메시지내에서 DC를 얻을 때는 BeginPaint, EndPaint 함수쌍을 사용하며 이 메시지 이외의 코드에서 DC를 얻을 때는 GetDC, ReleaseDC 함수쌍을 사용한다.
- GetDC의 인수로 DC를 구하고자 하는 핸들을 주면 이 윈도우의 DC를 구해준다. 윈도우의 클래스 스타일에 따라 윈도우 DC와 클래스 DC가 구해지는 경우도 있으나 대부분의 경우 작업 영역에 그리기를 할 수 있는 커먼 DC가 리턴된다.

WINAPI – GetPixel

❖ 원형

- COLORREF GetPixel(HDC hdc, int nXPos, int nYPos);

❖ 인수

- hdc : DC의 핸들
- (nXPos ,nYPos) : 조사할 점의 좌표이며 논리 단위이다.

❖ 리턴값

- 조사된 점의 색상을 리턴한다.

❖ 설명

- 설명 hdc의 한 점에 출력되어 있는 색상을 조사한다. 이 함수는 자주 사용되지는 않지만 복잡한 영역에 대한 좌표 점검을 할 때 색상으로 특정 좌표의 영역을 조사하는 용도로 종종 사용된다. 화면이나 메모리 DC등은 이 함수를 지원하지만 프린터나 플로터 등은 이 함수를 지원하지 못한다.

WINAPI - SelectObject

❖ 원형

- `HGDIOBJ SelectObject( HDC hdc, HGDIOBJ hgdibj);`

❖ 인수

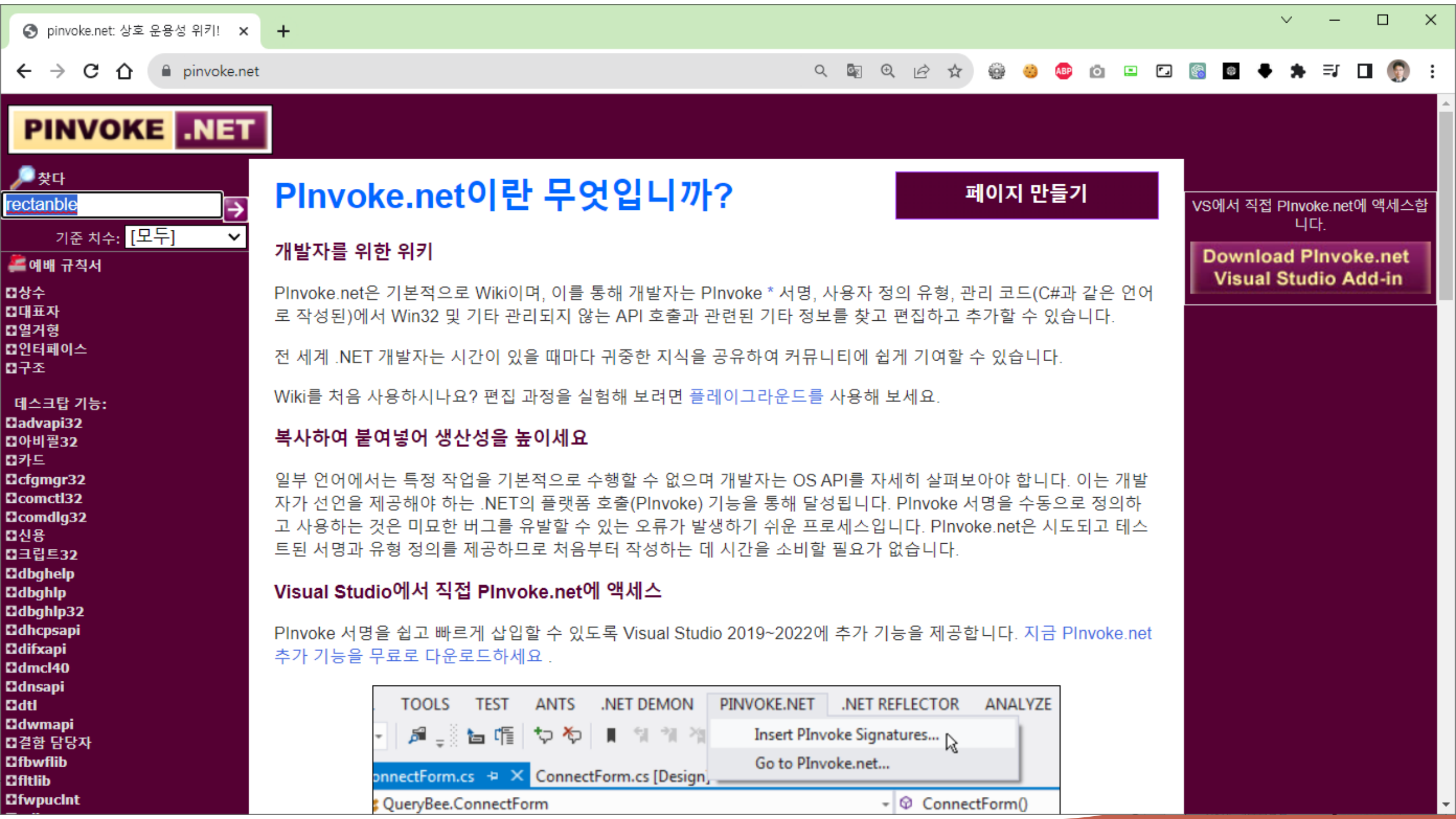
- `hdc`: DC 핸들
- `hgdibj`: DC에 선택하고자 하는 GDI 오브젝트의 핸들. 이 오브젝트는 `CreatePen`, `CreateSolidBrush`, `CreateFont` 등의 함수로 생성한 오브젝트이거나 `GetStockObject`로 구한 스톡 오브젝트여야 한다.

❖ 리턴값

- 이전에 선택되어 있던 같은 종류의 GDI 오브젝트의 핸들이 리턴된다.

❖ 설명

- GDI는 그래픽 출력에 사용되는 도구이며 펜, 브러시, 비트맵, 리전, 패스, 팔레트, 폰트 등등이 있다. 그리기에 사용할 GDI 오브젝트를 변경하고자 할 때 이 오브젝트를 만든 후 이 함수로 DC에 선택해 주어야 한다.



윈도우 기본 디자인

```
<Window x:Class="DifferPicHeler.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:DifferPicHeler"
        mc:Ignorable="d"
        Title="넷마블 틀린그림찾기 도우미" Height="420" Width="280" ResizeMode="NoResize" Loaded="Window_Loaded">
    <StackPanel Orientation="Vertical">
        <Label Content="넷마블 틀린그림찾기 도우미 v1.0" HorizontalAlignment="Center" Margin="0,10,0,0"
              Width="230" FontWeight="Bold" HorizontalContentAlignment="Center" FontSize="14" Foreground="Black"
              BorderBrush="Black" Background="Yellow"/>
        <GroupBox x:Name="gbMode" Header=" 모드 " Width="230" Height="60" Margin="0,10,0,0">
            <StackPanel Orientation="Horizontal" VerticalAlignment="Center" HorizontalAlignment="Center">
                <TextBlock Text="모드" Width="100" Height="40" Background="White" Border="1" BorderColor="Gray" Margin="5" />
                <TextBlock Text="방향" Width="100" Height="40" Background="White" Border="1" BorderColor="Gray" Margin="5" />
                <TextBlock Text="지연" Width="100" Height="40" Background="White" Border="1" BorderColor="Gray" Margin="5" />
            </StackPanel>
        </GroupBox>
        <GroupBox x:Name="gbDir" Header=" 방향 " Width="230" Height="60" Margin="0,10,0,0">
            <StackPanel Orientation="Horizontal" VerticalAlignment="Center" HorizontalAlignment="Center">
                <TextBlock Text="방향" Width="100" Height="40" Background="White" Border="1" BorderColor="Gray" Margin="5" />
                <TextBlock Text="지연" Width="100" Height="40" Background="White" Border="1" BorderColor="Gray" Margin="5" />
            </StackPanel>
        </GroupBox>
        <GroupBox Header=" 지연 " Width="230" Height="65" Margin="0,10,0,0">
            <TextBlock Text="지연" Width="100" Height="40" Background="White" Border="1" BorderColor="Gray" Margin="5" />
        </GroupBox>
    </StackPanel>
</Window>
```

넷마블 틀린그림찾기 도우미

넷마블 틀린그림찾기 도우미 v1.0

모드

방향

지연

윈도우 기본 디자인

넷마블 틀린그림찾기 도우미

넷마블 틀린그림찾기 도우미 v1.0

모드

☒ 자동감지 ☐ 강제동작

radModeAuto radModeForce

방향

☐ 세로 ☐ 가로

radDirVert radDirVHori

지연

sldBlitting

txtLog

btnStart

시작

윈도우 기본 디자인

```
<StackPanel Orientation="Vertical">
    <Label Content="넷마블 틀린그림찾기 도우미 v1.0" HorizontalAlignment="Center" Margin="0,10,0,0" VerticalAlignment="Top" Width="230"
        FontWeight="Bold" HorizontalContentAlignment="Center" FontSize="14" Foreground="Blue" BorderBrush="Black" Background="Yellow"/>
    <GroupBox x:Name="gbMode" Header=" 모드 " Width="230" Height="60" Margin="0,10,0,0">
        <StackPanel Orientation="Horizontal" VerticalAlignment="Center" HorizontalAlignment="Center">
            <RadioButton x:Name="radModeAuto" Content="자동감지" HorizontalAlignment="Left" Width="80" Height="15" Margin="10,0,0,0"
                IsChecked="True" Checked="radModeAuto_Checked"/>
            <RadioButton x:Name="radModeForce" Content="강제 동작" HorizontalAlignment="Left" Width="80" Height="15" Margin="10,0,0,0"
                Checked="radModeForce_Checked"/>
        </StackPanel>
    </GroupBox>
    <GroupBox x:Name="gbDir" Header=" 방향 " Width="230" Height="60" Margin="0,10,0,0">
        <StackPanel Orientation="Horizontal" VerticalAlignment="Center" HorizontalAlignment="Center">
            <RadioButton x:Name="radDirVert" Content="세로" HorizontalAlignment="Left" Width="80" Height="15" Margin="10,0,0,0"
                Checked="radDirVert_Checked"/>
            <RadioButton x:Name="radDirHori" Content="가로" HorizontalAlignment="Left" Width="80" Height="15" Margin="10,0,0,0"
                Checked="radDirHori_Checked"/>
        </StackPanel>
    </GroupBox>
    <GroupBox Header=" 지연 " Width="230" Height="65" Margin="0,10,0,0">
        <Slider x:Name="sldBlitting" HorizontalAlignment="Center" VerticalAlignment="Center" Width="194" Value="300" Minimum="100"
            Maximum="800" TickPlacement="Both" IsSnapToTickEnabled="True" TickFrequency="100" ValueChanged="sldBlitting_ValueChanged" />
    </GroupBox>
    <TextBox x:Name="txtLog" HorizontalAlignment="Center" Margin="0,10,0,0" Text="" VerticalAlignment="Top" Width="230" Height="60"
        VerticalScrollBarVisibility="Auto" />
    <Button x:Name="btnStart" Content="시작" Height="26" Width="230" Margin="0,10,0,0" Click="btnStart_Click"/>
</StackPanel>
```

멤버 변수 추가

```
using System.Timers;
using HANDLE = System.IntPtr;
using HWND = System.IntPtr;
using HDC = System.IntPtr;
using static DiffPicsHelper.WinAPI;

:

public partial class MainWindow : Window
{
    enum MODE {
        AUTO, FORCE
    }
    enum DIR {
        NONE, VERTICAL, HORIZONTAL
    };

    bool bHackOn = false;
    HWND hGameWnd;
    int nBlitting = 200;    // 블리팅 횟수 지정
    MODE mMode = MODE.AUTO; // 동작모드
    DIR mDir = DIR.NONE;    // 그림 방향
    Timer timer = new System.Timers.Timer();
```

라디오버튼 등 이벤트 핸들러 추가

```
private void radModeAuto_Checked(object sender, RoutedEventArgs e)
{
    mMode = MODE.AUTO;
}

private void radModeForce_Checked(object sender, RoutedEventArgs e)
{
    mMode = MODE.FORCE;
}

private void radDirVert_Checked(object sender, RoutedEventArgs e)
{
    mDir = DIR.VERTICAL;
}

private void radDirHori_Checked(object sender, RoutedEventArgs e)
{
    mDir = DIR.HORIZONTAL;
}

private void sldBlitting_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
{
    nBlitting = (int)sldBlitting.Value;
}
```

bool IsContinuePixel(**int** x, **int** y, DIR dir, **int** count)

```
// (x, y)좌표부터 dir방향으로 같은 색깔의 픽셀이 count개 이상 존재하는가?
bool IsContinuePixel(int x, int y, DIR dir, int count)
{
    uint Color, Color0;
    int i = 0;

    HDC hDC = GetDC(hGameWnd);           // 게임화면 캔버스 획득
    Color0 = GetPixel(hDC, x, y);         // (x, y)좌표의 색상 획득

    do
    {
        i++;
        if (dir == DIR.VERTICAL)
            Color = GetPixel(hDC, x, y + i);
        else
            Color = GetPixel(hDC, x + i, y);
    }
    while (Color0 == Color && i < 1024); // 같은색이면 계속 반복

    ReleaseDC(hGameWnd, hDC);

    if (i >= count)
        return true;
    else
        return false;
}
```

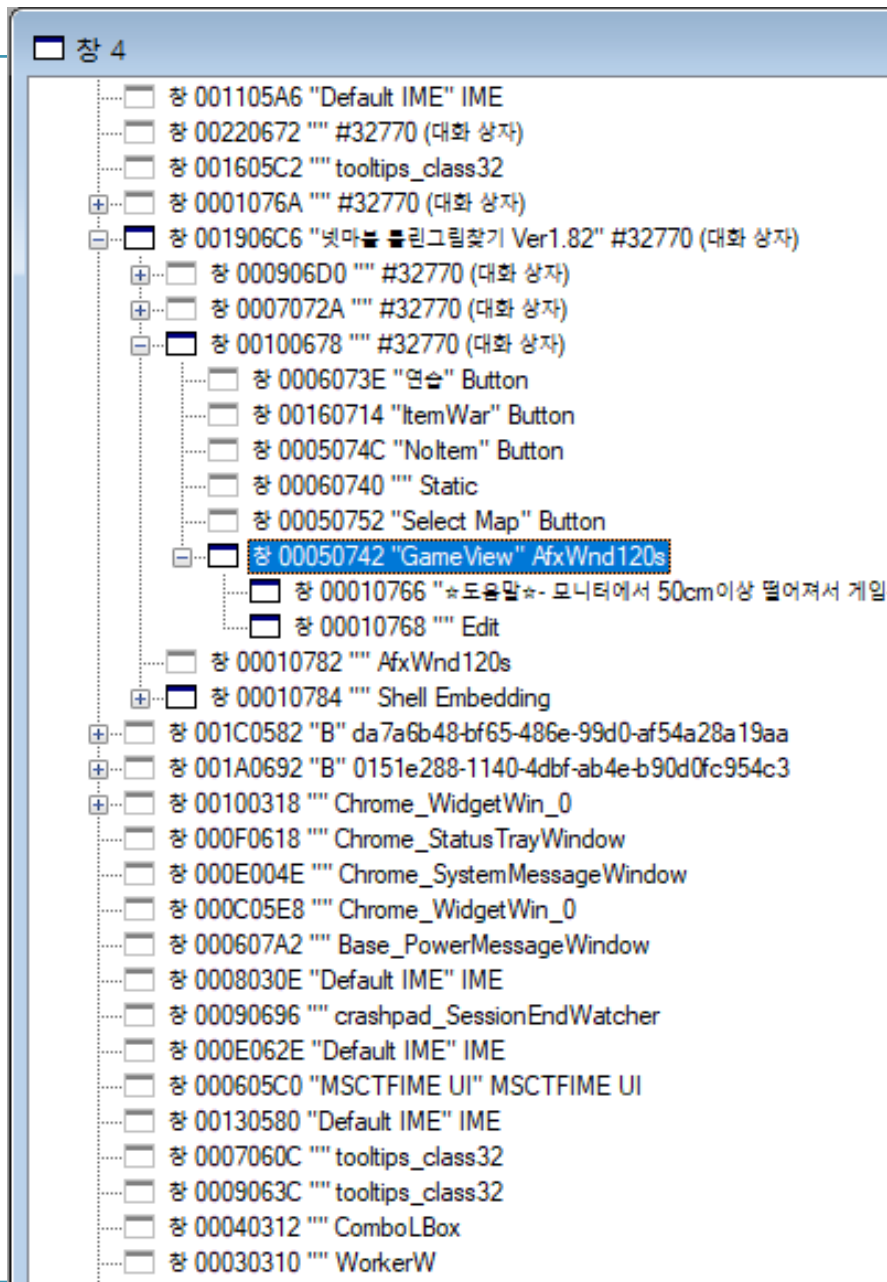
btnStart_Click 이벤트 핸들러 작성

```
private void btnStart_Click(object sender, RoutedEventArgs e)
{
    HWND hWnd;

    bHackOn = !bHackOn;
    if (bHackOn) {
        hWnd = FindWindow(null, "넷마블 틀린그림찾기 Ver1.82");
        hWnd = WinAPI.GetWindow(hWnd, WindowType.GW_CHILD);
        hWnd = WinAPI.GetWindow(hWnd, WindowType.GW_HWNDNEXT);
        hWnd = WinAPI.GetWindow(hWnd, WindowType.GW_HWNDNEXT);
        hGameWnd = WinAPI.FindWindowEx(hWnd, IntPtr.Zero, null, "GameView");

        if (hGameWnd == NULL) {
            MessageBox.Show("게임창을 찾을 수 없습니다.", "에러");
            return;
        }

        txtLog.AppendText("Hack Started.\n");
        btnStart.Content = "Stop";
    }
    else
    {
        txtLog.AppendText("Hack Stopped.\n");
        btnStart.Content = "Start";
    }
}
```




```

void timer_Elapsed(object sender, ElapsedEventArgs e)
{
    if (!bHackOn) return;

    HWND hWnd;
    hWnd = FindWindow(null, "넷마블 틀린그림찾기 Ver1.82");
    hWnd = WinAPI.GetWindow(hWnd, WindowType.GW_CHILD);
    hWnd = WinAPI.GetWindow(hWnd, WindowType.GW_HWNDNEXT);
    hWnd = WinAPI.GetWindow(hWnd, WindowType.GW_HWNDNEXT);
    hGameWnd = WinAPI.FindWindowEx(hWnd, IntPtr.Zero, null, "GameView");

    if (hGameWnd == NULL) return;

    if(mMode == MODE.AUTO)
    {
        // 자동감지모드인 경우
        if(IsContinuePixel(265, 531, DIR.HORIZONTAL, 460)) // 게임 시작 전 화면
            ;
        else if(IsContinuePixel(220, 543, DIR.HORIZONTAL, 400)) // 게임 결과 화면
            ;
        else if(IsContinuePixel(150, 600, DIR.HORIZONTAL, 600) && // 게임 가로 그림
                IsContinuePixel(570, 316, DIR.HORIZONTAL, 100)) // Clear 마크 나오는 자리
            DrawDiffPic(DIR.HORIZONTAL);
        else if(IsContinuePixel(511, 265, DIR.VERTICAL, 100) && // 게임 세로 그림
                IsContinuePixel(889, 46, DIR.VERTICAL, 500)) // Clear 마크 나오는 자리
            DrawDiffPic(DIR.VERTICAL);
        else // 모두 아닌 경우
            Beep(700, 50);
    }
    else if(mMode == MODE.FORCE)
    {
        // 강제동작모드인 경우
        DrawDiffPic(mDir);
    }
}

```

1 Lv.50
eho*****



Ready!

3 Lv.22
ne*****



Ready!

5

Scant

2 Lv.42
카라3131



Ready!

4

Scant

6

Scant

Tip
매직게이지가 모이면 매직아이버튼 사용할수 있어요!
그림페이지를 마음대로 넘길수 있습니다!

6장 8장 10장

가로 세로 랜덤

3분 5분 7분 10분

노템 아이템

일반 경쟁

SOLO & TEAM

R G B

START



(265, 531)

아이템 슬롯

TIME

NEW
매직게이지와
매직아이버튼(단축키 S 사용)

MAGIC EYE

No ITEM

V 단서수집
카라3131> 오늘도 사람이 없네요
eho*****> 맞아요...
eho*****> ㅏㅏㅏㅏㅏ



1
2
3
4
5
6

NEW
그림페이지버튼(단축키 A,D 사용)
현재그림 표시창

< >

1 Lv.37
qkd*****



21

3 Lv.38
저라구요



27

5

Scents



2 Lv.39
59*****



18

4 Lv.21
ne****



29

6

Scents

No ITEM

V 단서수집

☆도움말☆
- 모니터에서 50cm이상 떨어져서 게임을 하세요.
☆ <ne****>님 입장 ☆

112



MAGIC EYE

1 2 3 4 5 6

40 50



< >

7/10

1 Lv.37
qkd*****

35

3 Lv.38
저라구요

40

5
Scent




2 Lv.39
59*****

32

4 Lv.21
ne****

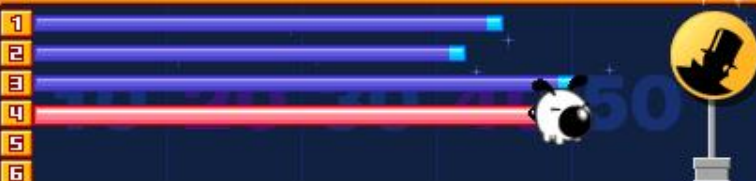
39

6
Scent


No ITEM

V 단서수집
☆도움말☆
- 모니터에서 50cm이상 떨어져서 게임을 하세요.
☆ <ne****>님 입장 ☆
112



MAGIC EYE


Navigation buttons: < >
9/10

```

static void DrawDiffPic(DIR dir) {
    uint i;
    HDC hDC, hMemDC;          // 메모리의 화가
    IntPtr hBitMap = NULL;    // 캔버스
    IntPtr hOldObj = NULL;    // 그림도구
    bool bRes;

    hDC = GetDC(hGameWnd);      // 게임윈도의 화가를 불러온다.
    hMemDC = CreateCompatibleDC(hDC); // 메모리에 위의 화가와 동일한 화가를 만든다.

    if (hDC == NULL || hMemDC == NULL) return;

    if (dir == DIR.VERTICAL)
    {
        hBitMap = CreateCompatibleBitmap(hDC, 366, 558);
        hOldObj = SelectObject(hMemDC, hBitMap);
        bRes = BitBlt(hMemDC, 0, 0, 366, 558, hDC, 515, 38, TernaryRasterOperations.SRCCOPY);
        // hDC (515, 38)좌표에서 366 x 558 크기만큼 hMemDC (0, 0)좌표에 붙여 넣는다.

        if(bRes)
            for (i = 0; i < nBlitting; i++)
                BitBlt(hDC, 139, 38, 366, 558, hMemDC, 0, 0, TernaryRasterOperations.SRCCOPY);
    }
    else if (dir == DIR.HORIZONTAL)
    {
        hBitMap = CreateCompatibleBitmap(hDC, 742, 274);
        hOldObj = SelectObject(hMemDC, hBitMap);
        bRes = BitBlt(hMemDC, 0, 0, 742, 274, hDC, 139, 322, TernaryRasterOperations.SRCCOPY);

        if (bRes)
            for (i = 0; i < nBlitting; i++)
                BitBlt(hDC, 139, 38, 742, 274, hMemDC, 0, 0, TernaryRasterOperations.SRCCOPY);
    }
}

```

```

    SelectObject(hMemDC, hOldObj);
    DeleteObject(hBitMap);
    DeleteObject(hMemDC);
    ReleaseDC(hGameWnd, hMemDC);
    ReleaseDC(hGameWnd, hDC);
}

```


1 Lv.37
qkd*****

35

3 Lv.38
저라구요

40

5
Scents



2 Lv.39
59*****

32

4 Lv.21
ne****

39

6
Scents

No ITEM

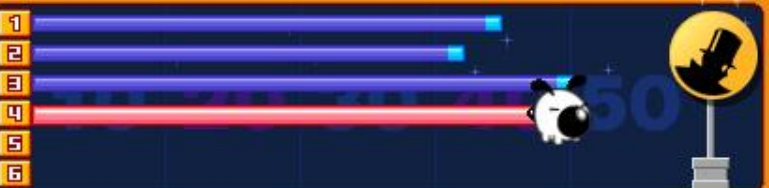
V 단서수집

☆도움말☆
- 모니터에서 50cm이상 떨어져서 게임을 하세요.
☆ <ne****>님 입장 ☆

112



- 1
- 2
- 3
- 4
- 5
- 6



MAGIC EYE

9/10

1 Lv.37
qkd*****



21

3 Lv.38
저라구요



27

5

Scents



2 Lv.39
59*****



18

4 Lv.21
ne****



29

6

Scents

No ITEM

V 단서수집

☆도움말☆
- 모니터에서 50cm이상 떨어져서 게임을 하세요.
☆ <ne****>님 입장 ☆

112



MAGIC EYE

1 2 3 4 5 6

40 50

< >

7/10

프로그램 테스트

넷마블 틀린그림찾기 도우미

×

넷마블 틀린그림찾기 도우미 v1.0

모드

☒ 자동감지 ☐ 강제동작

방향

☐ 세로 ☐ 가로

지연

시작

게임핵 강좌

- 치트엔진 따라 만들기 -

다운로드

❖ 디버깅 툴

- [x64dbg](#)

❖ CheatOmatic

- [완성본](#)

❖ 게임들

- [피카츄 배구](#)
- [Mame32 \(메탈슬러그1,3 포함\)](#)
- [겜보이 포켓몬 파이어레드](#)
- [삼국지10PK](#)
- [플래시 게임들](#)
- [던전크롤 스톤수프](#)

Content

❖ 메탈슬러그에 게임핵 사용 예시

❖ Cheat O matic 제작

- Visual Studio C#(wpf) 프로젝트 코딩

❖ 게임핵 테스트

- 피카츄 배구로 실험
- 삼국지10 PK에 게임핵 적용해보기
- 게임보이 포켓몬 파이어레드에 게임핵 적용해 보기
- 플래시 게임 – 타원디펜스에 게임핵 적용해보기
- 플래시 게임 – 검강화하기에 게임핵 적용해보기

설명	이름	유형	트랙볼	슬
 Final Fight (World)	ffight	래스터	아니오	0
 Final Fight (US 900613)	ffightub	래스터	아니오	0
 Metal Slug - Super Vehicle-001	mslug	래스터	아니오	0
 Metal Slug 3	mslug3	래스터	아니오	0



제작년도 : 1996.04

개발사 : 나즈카, 공급사 : SNK

장르 : 액션

컨트롤러 : 8 방향 레버 + 3 버튼

시스템기판 : SNK 네오지오 멀티 비디오 시스템 (NEO-GEO MVS)

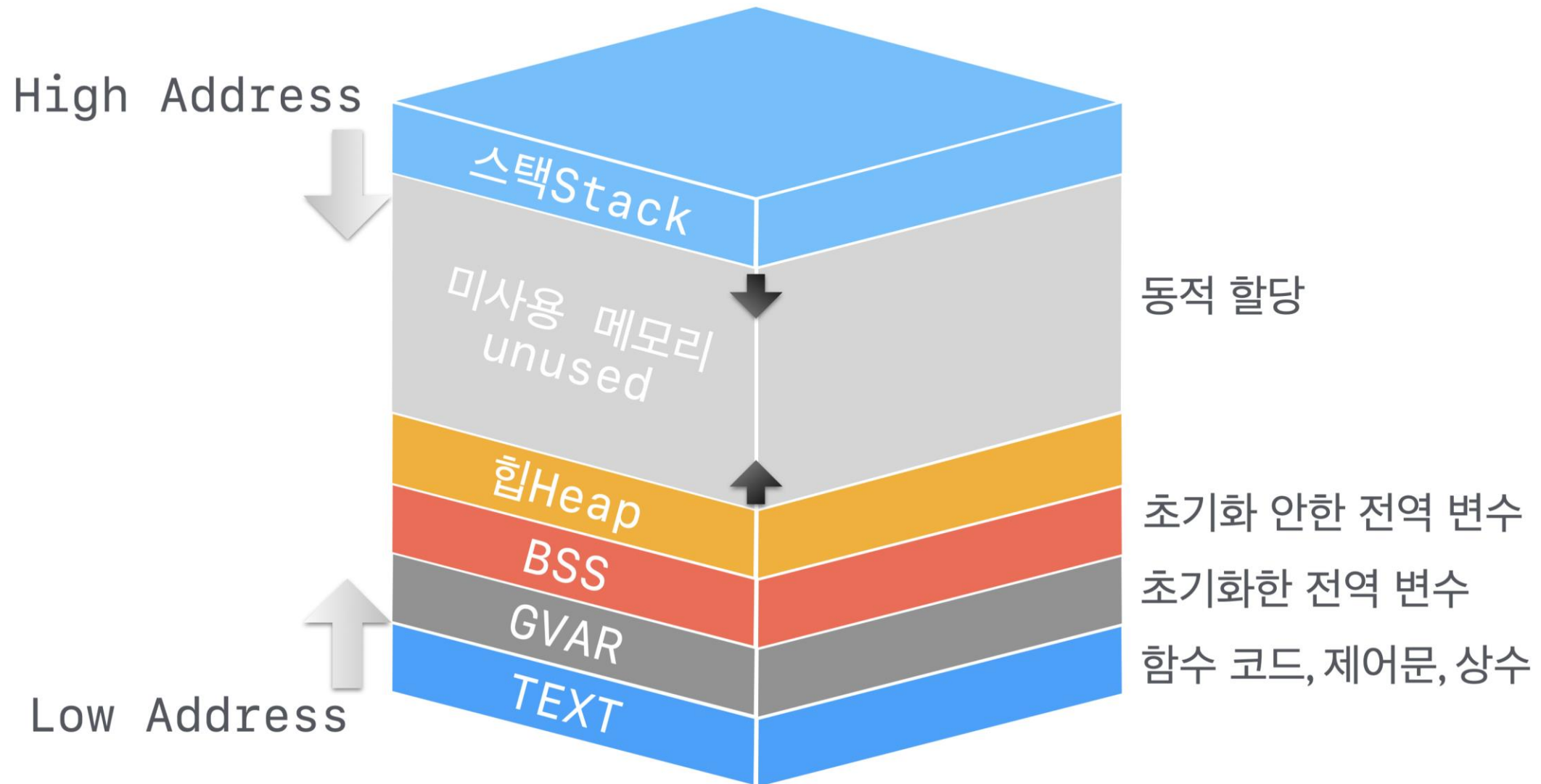
CPU : 68000, Z80

사운드칩 : YM2610

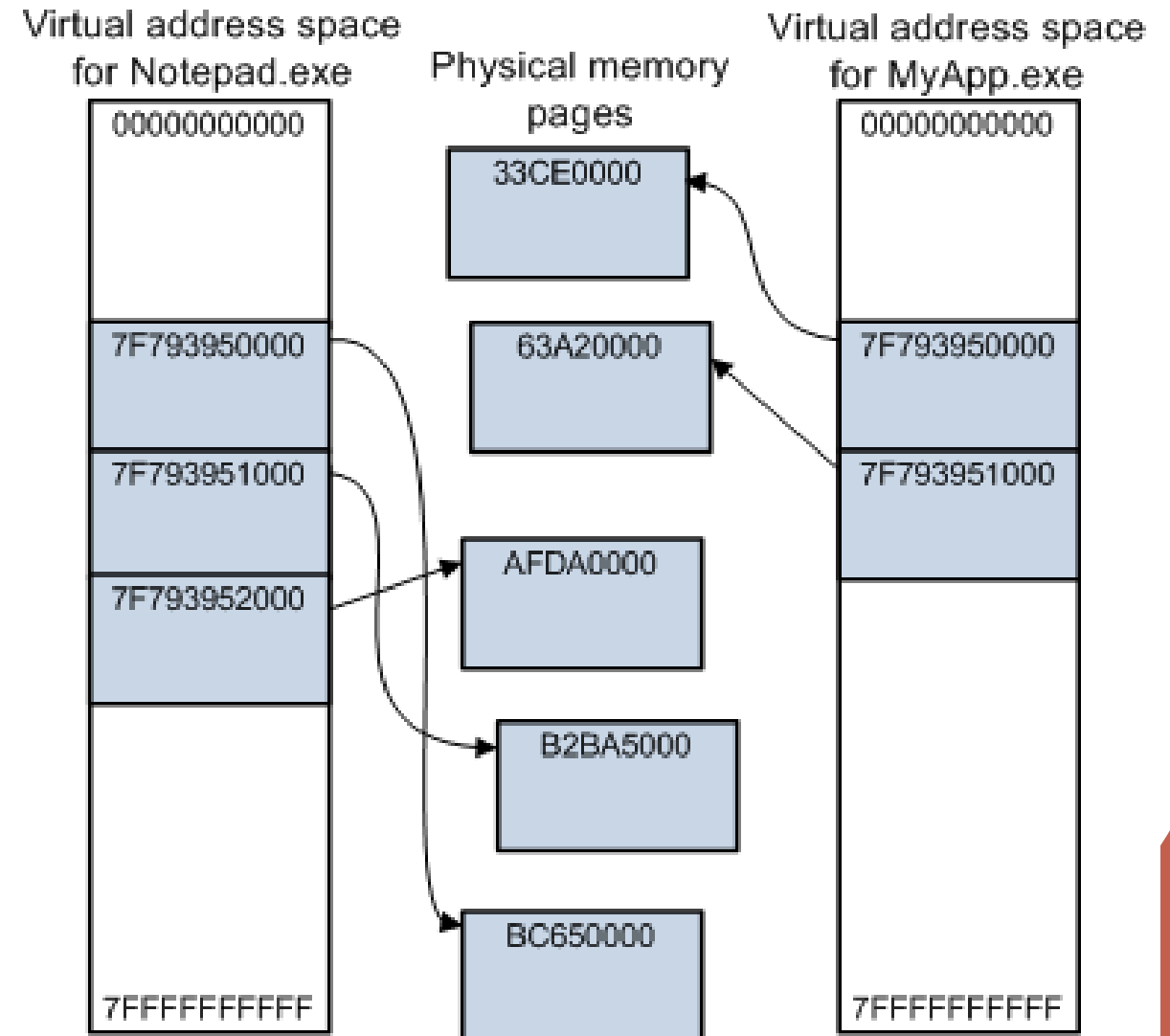
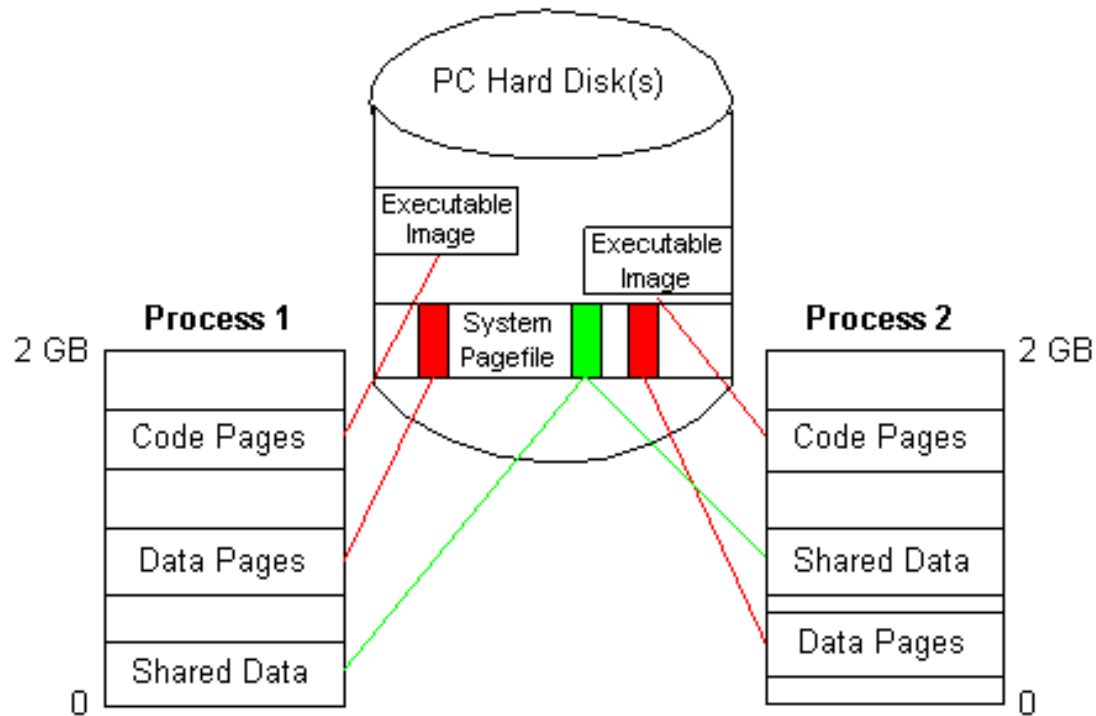
게임의 주된 목적은, 세계를 정복하려는 야심을 지닌 악의 수장 모엔 원수가 이끄는 반란군을 제압해 나가는 것이다. 처음에는 무기가 기종과 수류탄밖에 없지만, 스테이지를 진행하면서 인질을 수감하거나 주변 오브젝트를 파괴하면 다양한 무기를 구할 수 있으며, 중간중간에 이 게임 제1무기인도 한 메탈슬러그를 더 얻어 가지 슬러그에 탄슬러그에 알집을 가로막는 적들을 날려버리는 경파함! 이 시리즈의 재미였으나, 메탈슬러그 3부터는 난이도가 대폭 높아져서 가볍게 플레이할 수 있는 물건은 못 된다. 특히 이후의 6이나 7(XX) 등은 게임성은 물론퍼 치더라도 난이도가 너무 높아져서 완전히 대니야를 위한 물건으로 불변해 버렸을 정도.

메탈슬러그는 과거 아이렘에 소속되어 있었던 제작자들이 설립한 '나즈카'라는 회사에서 제작한 작품이기 때문에 아이렘에서 만트 게인의 해저대장전 지오스톤(해위문 거프스 II)과 상당한 그래픽 노력이 비슷하다.

Memory Model



WIN32 Process Memory Map



64-bit Address Spaces

- ▶ 64-bits = 2^{64} = 17,179,869,184 GB
 - ▶ x64 today supports 48 bits virtual = 262,144 GB = 256 TB
 - ▶ IA-64 today support 50 bits virtual = 1,048,576 GB = 1024 TB
 - ▶ 64-bit Windows supports 44 bits = 16,384 GB = 16 TB

x64



32-bit
process
on x64



CheatOmatic 만들기

❖ Visual Studio 프로젝트 생성

- WPF 애플리케이션

❖ 프로젝트에 파일 추가

- [WinAPI.cs 다운로드](#)
- WinAPI.cs 파일 추가
- 아이콘 파일 추가

❖ using 문 추가

- `using System.Timers;`
- `using static CheatOmatic.WinAPI;`

CheatOmatic UI 디자인

CheatOmatic1*

프로세스 목록:

자료 검색: 초기탐색 재탐색 전체삭제 ☒ 자동갱신

고정	주소(hex)	크기	현재값(설정값)	이전값	설명	얼리기
☐						☐

CheatOmatic UI 디자인

```
<Window x:Class="CheatOmatic.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CheatOmatic"
        mc:Ignorable="d"
        Title="CheatOmatic" Height="500" Width="630" ResizeMode="CanMinimize">
    <!--Window.DataContext>
        <local:ViewModelDI/>
    </Window.DataContext-->
    <Grid Margin="0,0,0,0">
        <Grid.RowDefinitions>
            <RowDefinition Height="36"/>
            <RowDefinition Height="36"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="26"/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="1*/>
        </Grid.ColumnDefinitions>

        <StatusBar Grid.Row="3" FontSize="11">
            <TextBlock x:Name="txtMessage" Margin="10,0,10,0" />
        </StatusBar>
    </Grid>
</Window>
```

CheatOmatic UI 디자인

```
</Grid.ColumnDefinitions>
```

```
<Label Content="프로세스 목록: " HorizontalAlignment="Left" Margin="7,0,0,0" VerticalAlignment="Center" FontWeight="Bold"/>  
<ComboBox x:Name="cmbProcess" HorizontalAlignment="Left" Margin="101,0,0,0" VerticalAlignment="Center" Width="504" Height="22"/>
```

```
<DataGrid>
```

```
</DataGrid>
```

```
<ComboBox x:Name="cmbDataType" HorizontalAlignment="Left" Margin="101,0,0,0" Grid.Row="1" VerticalAlignment="Center"  
    Width="105" SelectedIndex="0">  
    <ComboBoxItem Content="4Byte"/>  
    <ComboBoxItem Content="2Byte"/>  
    <ComboBoxItem Content="1Byte"/>
```

```
</ComboBox>
```

```
<TextBox x:Name="txtSearchValue" HorizontalAlignment="Left" Margin="211,0,0,0" Grid.Row="1" TextWrapping="Wrap" Text=""  
    VerticalAlignment="Center" Width="130" Height="22"/>
```

```
<Label Content="자료 검색:" HorizontalAlignment="Left" Margin="33,0,0,0" Grid.Row="1" VerticalAlignment="Center" FontWeight="Bold"/>
```

```
<Button x:Name="btnNewSearch" Content="초기탐색" HorizontalAlignment="Left" Margin="346,0,0,0" Grid.Row="1" VerticalAlignment="Center"  
    Height="22" Width="58" IsEnabled="True"/>
```

```
<Button x:Name="btnFilterSearch" Content="재탐색" HorizontalAlignment="Left" VerticalAlignment="Center" Width="58" Margin="409,0,0,0"  
    Grid.Row="1" Height="22" IsEnabled="False"/>
```

```
<CheckBox x:Name="chkAutoUpdate" Content="자동갱신" HorizontalAlignment="Left" Margin="537,0,0,0" Grid.Row="1"  
    VerticalAlignment="Center" IsChecked="True"/>
```

```
<Button x:Name="btnClear" Content="전체삭제" HorizontalAlignment="Left" Margin="472,0,0,0" Grid.Row="1" VerticalAlignment="Center"  
    Width="60" Height="22"/>
```

```
<StatusBar Grid.Row="3" FontSize="11">
```

```
<DataGrid x:Name="dgDataItem" Grid.Row="2" Margin="10,0,10,10" ScrollViewer.HorizontalScrollBarVisibility="Auto"
    ScrollViewer.VerticalScrollBarVisibility="Auto" AutoGenerateColumns="False"
    ItemsSource="{Binding Path=ocDataItem, Mode=TwoWay,
    NotifyOnSourceUpdated=True, NotifyOnTargetUpdated=True, UpdateSourceTrigger=PropertyChanged}">
    <DataGrid.Columns>
        <DataGridCheckBoxColumn Header="고정" Binding="{Binding Fix, Mode=TwoWay}" Width="35">
        <DataGridCheckBoxColumn.CellStyle>
            <Style>
                <!--EventSetter Event="CheckBox.Checked"/>
                <EventSetter Event="CheckBox.Unchecked"/-->
            </Style>
        </DataGridCheckBoxColumn.CellStyle>
        </DataGridCheckBoxColumn>
        <DataGridTextColumn Header="주소(hex)" Binding="{Binding HexAddress, Mode=TwoWay}" Width="2*" />
        <DataGridTextColumn Header="크기" Binding="{Binding Size}" Width="0.5*" />
        <DataGridTextColumn Header="현재값(설정값)" Binding="{Binding NowValue, Mode=TwoWay}" Width="1.5*">
            <DataGridTextColumn.CellStyle>
                <Style TargetType="DataGridCell">
                    <Setter Property="Foreground" Value="{Binding NowValueColor, Mode=TwoWay}"/>
                </Style>
            </DataGridTextColumn.CellStyle>
        </DataGridTextColumn>
        <DataGridTextColumn Header="이전값" Binding="{Binding PrevValue}" Width="1.5*" IsReadOnly="True" />
        <DataGridTextColumn Header="설명" Binding="{Binding Desc, Mode=TwoWay}" Width="1.5*" />
        <DataGridCheckBoxColumn Header="얼리기" Binding="{Binding Freeze, Mode=TwoWay}" Width="50">
            <DataGridCheckBoxColumn.CellStyle>
                <Style>
                    <!--EventSetter Event="CheckBox.Checked"/>
                    <EventSetter Event="CheckBox.Unchecked"/-->
                </Style>
            </DataGridCheckBoxColumn.CellStyle>
        </DataGridCheckBoxColumn>
    </DataGrid.Columns>
</DataGrid>
```

```

public class DataItem : INotifyPropertyChanged
{
    public ulong Address { get; set; }
    public string? hexAddress;
    public string? HexAddress
    {
        get { return hexAddress; }
        set
        {
            if (value == null || value.Equals(""))
                return;
            hexAddress = value;
            this.Address = (ulong)Convert.ToInt64(value, 16);
            OnPropertyChanged(nameof(HexAddress));
        }
    }
    public int nowValue;
    public int NowValue {
        get { return nowValue; }
        set
        {
            if(this.nowValue != value) {
                this.nowValue = value;
                OnPropertyChanged(nameof(NowValue));
            }

            if(this.PrevValue != value)
            {
                this.NowValueColor = Brushes.Red;
                OnPropertyChanged(nameof(NowValueColor));
            }
            else
                this.NowValueColor = Brushes.Black;
        }
    }
}

```

```

    public int PrevValue { get; set; }
    public ushort size;
    public ushort Size {
        get { return size; }
        set
        {
            size = value;
            OnPropertyChanged(nameof(Size));
        }
    }
    public string? Desc { get; set; }
    public bool Fix { get; set; }
    public bool Freeze { get; set; }

    public SolidColorBrush? NowValueColor { get; set; }

    public event PropertyChangedEventHandler PropertyChanged;

    protected virtual void OnPropertyChanged(string propertyName)
    {
        if (PropertyChanged != null)
        {
            PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
        }
    }
}

```

DataGrid에 사용할 뷰 모델 정의

```
public class ViewModelDI
{
    public ViewModelDI()
    {
    }

    public static ObservableCollection<DataItem>? _ocDataItem = null;
    public static ObservableCollection<DataItem> ocDataItem
    {
        get
        {
            if (_ocDataItem == null)
            {
                _ocDataItem = new ObservableCollection<DataItem>();
            }
            return _ocDataItem;
        }
        set
        {
            _ocDataItem = value;
        }
    }
}
```

// xaml 파일 뷰모델 주석해제

멤버 변수 정의

```
public partial class MainWindow : Window
{
    private int    selectedProcId;
    private ushort selectedSize = 4;
    private bool   is32bitProcess = true;
    private bool   autoUpdateEnabled = true;
    private List<MEMORY_BASIC_INFORMATION64> lstMemoryBlock;
    Timer tmrUpdater, tmrFreezer;

    public MainWindow()
    {
        InitializeComponent();
        lstMemoryBlock = new List<MEMORY_BASIC_INFORMATION64>();

        tmrUpdater = new Timer();
        tmrUpdater.Interval = 1000;
        tmrUpdater.Elapsed += new System.Timers.ElapsedEventHandler(tmrUpdater_Tick);
        tmrUpdater.Start();

        tmrFreezer = new Timer();
        tmrFreezer.Interval = 500;
        tmrFreezer.Elapsed += new System.Timers.ElapsedEventHandler(tmrFreezer_Tick);
        tmrFreezer.Start();
    }
}
```

타이머 함수 정의

```
public void tmrUpdater_Tick(object sender, System.Timers.ElapsedEventArgs e)
{
    Application.Current.Dispatcher.Invoke(new Action(() =>
    {
        try
        {
            if (chkAutoUpdate.IsChecked==false)
                return;

            //UpdateViewModelDI();
        }
        catch (Exception) { }
    }));
}

public void tmrFreezer_Tick(object sender, System.Timers.ElapsedEventArgs e)
{
    Application.Current.Dispatcher.Invoke(new Action(() =>
    {
        try
        {
            //FreezeViewModelDI();
        }
        catch (Exception) { }
    }));
}
```

public void UpdateProcessList()

```
public void UpdateProcessList()
{
    Process[] processes = Process.GetProcesses();
    foreach (Process p in processes)
    {
        if (p.Id == 0)
            continue;

        if (p.ProcessName.Equals("svchost") || p.ProcessName.Equals("csrss") || p.ProcessName.Equals("Registry")
            || p.ProcessName.Equals("wininit") || p.ProcessName.Equals("ctfmon")
            || p.ProcessName.Equals("fontdrvhost") || p.ProcessName.Equals("winlogon") || p.ProcessName.Equals("LogonUI"))
            continue;

        IntPtr hProcess = WinAPI.OpenProcess(PROCESS_ALL_ACCESS, false, p.Id);
        WinAPI.IsWow64Process(hProcess, out is32bitProcess);
        WinAPI.CloseHandle(hProcess);

        string text = "";
        if (is32bitProcess)
            text = "[" + p.Id + "] (x32) " + p.ProcessName;
        else
            text = "[" + p.Id + "] (x64) " + p.ProcessName;

        if (p.MainWindowTitle == "")
            cmbProcess.Items.Add(text);
        else
            cmbProcess.Items.Add(text + " - " + p.MainWindowTitle.Substring(0, Math.Min(p.MainWindowTitle.Length, 40)));
    }
}
```

```

private void cmbProcess_DropDownOpened(object sender, EventArgs e)
{
    UpdateProcessList();
}

private void cmbProcess_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (cmbProcess.SelectedIndex >= 0)
    {
        int procId = 0;
        if (cmbProcess.SelectedItem is not null) {
            string strPid = cmbProcess.SelectedItem.ToString().Split(' ')[0];
            procId = int.Parse(strPid.Substring(1));
        }

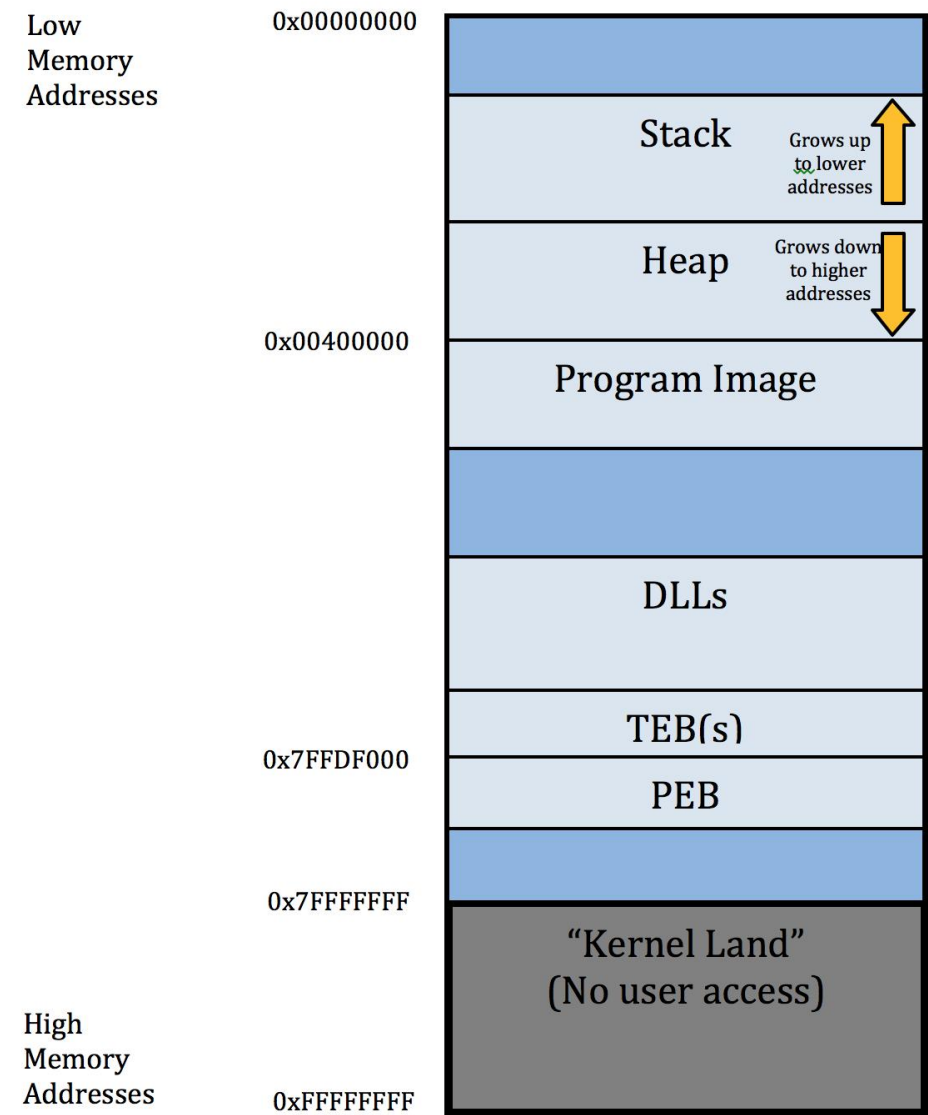
        if (this.selectedProcId != procId)
        {
            this.selectedProcId = procId;
            MakeProcessMemoryBlockInfo();
            txtMessage.Text = lstMemoryBlock.Count().ToString() + " 블록을 찾았습니다.";
        }

        if (lstMemoryBlock.Count > 0)
        {
            btnNewSearch.IsEnabled = true;
            btnFilterSearch.IsEnabled = true;
        }
        else
        {
            MessageBox.Show("해당 프로세스의 메모리 접근이 차단되었습니다.", "에러");
            btnNewSearch.IsEnabled = false;
            btnFilterSearch.IsEnabled = false;
        }
    }
    ViewModelDI.ocDataItem.Clear();
}

```

WIN32 메모리 구조

Win32 Memory Map
(simplified)



Immunity Examples (using Windows Media Player)

Address	Size	Owner	Section	Contains
00010000	00001000			
00020000	00001000			
00030000	00001000			
00070000	00001000			
00071000	0000F000			stack of main thread

00A0F000	### Immunity's Heapdump ###
000A0000	Heap: 0x000a0000
001A0000	Heap: 0x001a0000
001B0000	Heap: 0x001b0000
003C0000	Heap: 0x003c0000
003D0000	Heap: 0x003d0000
003F0000	Heap: 0x003f0000

Address	Size	Owner	Section	Contains
01000000	00001000	wmplayer		PE header
01001000	00002000	wmplayer	.text	code, imports
01003000	00001000	wmplayer	.data	data
01004000	0000E000	wmplayer	.rsrc	resources

Address	Size	Owner	Section	Contains
7E1E0000	00001000	urlmon		PE header
7E1E1000	00071000	urlmon	.text	code, imports, exports
7E252000	00001000	urlmon	.orpc	code
7E253000	0000F000	urlmon	.data	data
7E262000	0001B000	urlmon	.rsrc	resources
7E27D000	00005000	urlmon	.reloc	relocations
7E410000	00001000	USER32		PE header
7E411000	00060000	USER32	.text	code, imports, exports
7E471000	00002000	USER32	.data	data
7E473000	0002B000	USER32	.rsrc	resources
7E49E000	00003000	USER32	.reloc	relocations
7E720000	00001000	SXS		PE header
7E721000	00099000	SXS	.text	code, imports, exports
7E7BA000	00003000	SXS	.data	data
7E7BD000	0000A000	SXS	.rsrc	resources
7E7C7000	00009000	SXS	.reloc	relocations
7F6F0000	00007000			
7FFB0000	00024000			
7FFDA000	00001000			
7FFDB000	00001000			
7FFDC000	00001000			
7FFDE000	00001000			
7FFDF000	00001000			
7FFE0000	00001000			

TEB

PEB

DLLs

data block of thread 0000FF8

data block of thread 0000B80

data block of thread 00001384

data block of main thread



CPU 로그 메모 중단점 메모리 맵 호출 스택 SEH 스크립트 기호 <> 소스 참조 스레드 핸들 Trace

주소	크기	파티	정보	내용	유형	보호	초기
00010000	00010000	사용자			MAP	-RW--	-RW--
00020000	00001000	사용자			MAP	-R---	-R---
00030000	00001000	사용자			MAP	-R---	-R---
00040000	0001D000	사용자			MAP	-R---	-R---
00060000	00035000	사용자	예약됨		PRV		-RW--
00095000	00008000	사용자			PRV	-RW-G	-RW--
000A0000	000F8000	사용자	예약됨		PRV		-RW--
00198000	00008000	사용자	Stack (13112)		PRV	-RW-G	-RW--
001A0000	00004000	사용자			MAP	-R---	-R---
00180000	00002000	사용자			PRV	-RW--	-RW--
001C0000	00001000	사용자			MAP	-R---	-R---
001D0000	00002000	사용자			PRV	-RW--	-RW--
001D2000	00006000	사용자	예약됨 (001D0000)		PRV		-RW--
001E0000	00004000	사용자			MAP	-R---	-R---
001E4000	00004000	사용자	예약됨 (001E0000)		MAP		-R---
001F0000	00001000	사용자			PRV	-RW--	-RW--
001F1000	00007000	사용자	예약됨 (001F0000)		PRV		-RW--
00200000	000D2000	사용자	예약됨		PRV		-RW--
002D2000	00005000	사용자	PEB, TEB (13112), Wow64 TEB (13112)		PRV	-RW--	-RW--
002D7000	0000C000	사용자	예약됨 (002D0000)		PRV		-RW--
002E3000	00003000	사용자	TEB (10048), Wow64 TEB (10048)		PRV	-RW--	-RW--
002E6000	00003000	사용자	예약됨 (002D0000)		PRV		-RW--
002E9000	00006000	사용자	TEB (12820), Wow64 TEB (12820),		PRV	-RW--	-RW--
002EF000	000A5000	사용자	예약됨 (002D0000)		PRV		-RW--
00394000	0000C000	사용자	TEB (15748), Wow64 TEB (15748),		PRV	-RW--	-RW--
003A0000	00060000	사용자	예약됨 (002D0000)		PRV		-RW--
00400000	00001000	사용자	pika.exe		IMG	-R---	ERWC-
00401000	0000E000	사용자	".text"	실행 가능한 코드	IMG	ER---	ERWC-
0040F000	00001000	사용자	".rdata"	읽기 전용 초기화 데이터	IMG	-R---	ERWC-
00410000	00003000	사용자	".data"	초기화된 데이터	IMG	-RW--	ERWC-
00413000	00001000	사용자	".idata"	데이터 가져오기	IMG	-RW--	ERWC-
00414000	0006A000	사용자	".rsrc"	리소스	IMG	-R---	ERWC-
0047E000	00002000	사용자	".reloc"	기준 재배치	IMG	-R---	ERWC-
00480000	00035000	사용자	예약됨		PRV		-RW--
00485000	00008000	사용자			PRV	-RW-G	-RW--
004C0000	00035000	사용자	예약됨		PRV		-RW--
004F5000	00008000	사용자			PRV	-RW-G	-RW--
00500000	00001000	사용자			MAP	-RW--	-RW--
00510000	00001000	사용자			PRV	-RW--	-RW--
00520000	00001000	사용자			PRV	-RW--	-RW--
00530000	00008000	사용자			PRV	-RW--	-RW--
00538000	00008000	사용자	예약됨 (00530000)		PRV		-RW--
00550000	00005000	사용자			MAP	-RW--	-RW--
00580000	00004000	사용자			MAP	-R---	-R---
00590000	00081000	사용자	Heap (ID 0)		PRV	-RW--	-RW--
00611000	0007F000	사용자	예약됨 (00590000)		PRV		-RW--
00690000	000C9000	사용자	\Device\HarddiskVolume3\windows\		MAP	-R---	-R---
00760000	000FD000	사용자	예약됨		PRV		-RW--
0085D000	00003000	사용자	Stack (15748)		PRV	-RW-G	-RW--
00860000	000FD000	사용자	예약됨		PRV		-RW--
0095D000	00003000	사용자	Stack (14708)		PRV	-RW-G	-RW--
00960000	00035000	사용자	예약됨		PRV		-RW--
00995000	00008000	사용자			PRV	-RW-G	-RW--
009A0000	00035000	사용자	예약됨		PRV		-RW--
009D5000	00008000	사용자			PRV	-RW-G	-RW--
00A60000	00011000	사용자			MAP	-R---	-R---
00A71000	001EF000	사용자	예약됨 (00A60000)		MAP		-R---
00C60000	00181000	사용자			MAP	-R---	-R---
00DF0000	000A0000	사용자			MAP	-R---	-R---
00E90000	01361000	사용자	예약됨 (00DF0000)		MAP		-R---
02240000	0000F000	사용자	Heap (ID 3)		PRV	-RW--	-RW--
0224F000	00001000	사용자	예약됨 (02240000)		PRV		-RW--
02250000	000FD000	사용자	예약됨		PRV		-RW--

65728000	00001000	문	시	스	스	".data"		IMG	-RW--	ERWC-
65729000	00002000	문	시	스	스	".idata"	초기화 된 데이터	IMG	-R---	ERWC-
65728000	00001000	문	시	스	스	".didat"	데이터 가져오기	IMG	-R---	ERWC-
6572C000	00001000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
6572D000	00006000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
65740000	00001000	문	시	스	스	imkrapi.dll		IMG	-R---	ERWC-
65741000	00027000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65768000	00001000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
65769000	00002000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
65768000	00004000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
6576F000	00003000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
65780000	00001000	문	시	스	스	dui70.dll		IMG	-R---	ERWC-
65781000	0013A000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65888000	00003000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
6588E000	00003000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
658C1000	00001000	문	시	스	스	".didat"		IMG	-R---	ERWC-
658C2000	0000E000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
658D0000	0000F000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
658E0000	00001000	문	시	스	스	imetip.dll		IMG	-R---	ERWC-
658E1000	000E7000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
659C8000	00001000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
659C9000	00005000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
659CE000	00001000	문	시	스	스	".didat"		IMG	-R---	ERWC-
659CF000	0000C000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
659D8000	0000F000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
659F0000	00001000	문	시	스	스	imkrtip.dll		IMG	-R---	ERWC-
659F1000	00033000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65A24000	00001000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
65A25000	00002000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
65A27000	00001000	문	시	스	스	".didat"		IMG	-R---	ERWC-
65A28000	0007D000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
65AA5000	00004000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
65AB0000	00001000	문	시	스	스	audioses.dll		IMG	-R---	ERWC-
65AB1000	0010A000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65BB8000	00001000	문	시	스	스	"RT_CODE"		IMG	ER---	ERWC-
65BBC000	00001000	문	시	스	스	"RT_BSS"		IMG	-RW--	ERWC-
65BBD000	00002000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
65BBF000	00004000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
65BC3000	00001000	문	시	스	스	".didat"		IMG	-R---	ERWC-
65BC4000	00001000	문	시	스	스	"RT_CONST"		IMG	-R---	ERWC-
65BC5000	00001000	문	시	스	스	"RT_DATA"		IMG	-RWC-	ERWC-
65BC6000	00010000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
65BD6000	00010000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
65BF0000	00001000	문	시	스	스	rdpendp.dll		IMG	-R---	ERWC-
65BF1000	00037000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65C28000	00001000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
65C29000	00002000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
65C28000	00001000	문	시	스	스	".didat"		IMG	-R---	ERWC-
65C2C000	00002000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
65C2E000	00004000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
65C40000	00001000	문	시	스	스	mmdevapi.dll		IMG	-R---	ERWC-
65C41000	00058000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65C9C000	00001000	문	시	스	스	"RT_BSS"		IMG	-RW--	ERWC-
65C9D000	00002000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
65C9F000	00003000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
65CA2000	00001000	문	시	스	스	".didat"		IMG	-R---	ERWC-
65CA3000	00001000	문	시	스	스	"RT_DATA"		IMG	-RWC-	ERWC-
65CA4000	00001000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
65CA5000	00006000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
65CB0000	00001000	문	시	스	스	inputhost.dll		IMG	-R---	ERWC-
65CB1000	000D4000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65D85000	00003000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-
65D88000	00003000	문	시	스	스	".idata"	데이터 가져오기	IMG	-R---	ERWC-
65D88000	00001000	문	시	스	스	".didat"		IMG	-R---	ERWC-
65D8C000	00001000	문	시	스	스	".rsrc"	리소스	IMG	-R---	ERWC-
65D8D000	00010000	문	시	스	스	".reloc"	기준 재배치	IMG	-R---	ERWC-
65DA0000	00001000	문	시	스	스	dinput.dll		IMG	-R---	ERWC-
65DA1000	0001C000	문	시	스	스	".text"	실행 가능한 코드	IMG	ER---	ERWC-
65DBD000	00003000	문	시	스	스	".data"	초기화 된 데이터	IMG	-RW--	ERWC-

명령: Commands are comma separated (like assembly instructions): mov eax, ebx

기본값

실행 중 DLL 로드: 65670000 C:\Windows\SysWOW64\IME\SHARED\IMJKAPI.DLL

디버깅에 사용한 시간: 0:00:10:48

private void MakeProcessMemoryBlockInfo()

```
private void MakeProcessMemoryBlockInfo()
{
    IntPtr hProcess = OpenProcess(PROCESS_ALL_ACCESS, false, selectedProcId);
    IsWow64Process(hProcess, out is32bitProcess);

    ulong lastAddress;
    if (is32bitProcess) lastAddress = 0x7fffffff;
    else lastAddress = 0x00007fffffffffff;

    lstMemoryBlock.Clear();
    for (ulong address = 0x0000; address < lastAddress;)
    {
        MEMORY_BASIC_INFORMATION64 mbi = new MEMORY_BASIC_INFORMATION64();

        int len = VirtualQueryEx(hProcess, (IntPtr)address, out mbi, Marshal.SizeOf(mbi));
        if (len > 0)
        {
            ulong beginAddr = mbi.BaseAddress;
            ulong endAddr = mbi.BaseAddress + mbi.RegionSize-1;
            if (mbi.State != (uint)StateEnum.MEM_FREE && mbi.State != (uint)StateEnum.MEM_RESERVE && mbi.Type != (uint)TypeEnum.MEM_MAPPED)
            {
                _ = string.Format($"{beginAddr:X}:{endAddr:X}({mbi.RegionSize:X}) State:{mbi.State:X}, Type:{mbi.Type:X}\n");
                lstMemoryBlock.Add(mbi);
            }
        }
        if (mbi.RegionSize != 0)
            address = mbi.BaseAddress + mbi.RegionSize;
        else
            address += 0x1000000;
    }
    WinAPI.CloseHandle(hProcess);
}
```

int GetMemoryBlockIndex(**ulong** address)

```
private int GetMemoryBlockIndex(ulong address)
{
    for (int i = 0; i < lstMemoryBlock.Count; i++)
    {
        ulong beginAddress = lstMemoryBlock[i].BaseAddress;
        ulong endAddress = lstMemoryBlock[i].BaseAddress + lstMemoryBlock[i].RegionSize - 1;

        if (beginAddress <= address && address <= endAddress)
            return i;
    }
    return -1;
}

private void cmbDataType_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (cmbDataType.SelectedIndex == 0) {
        this.selectedSize = 4;
    }
    else if (cmbDataType.SelectedIndex == 1) {
        this.selectedSize = 2;
    }
    else if (cmbDataType.SelectedIndex == 2)
    {
        this.selectedSize = 1;
    }
}
```

```
private int? ParseInt(string? text)
{
    if (text == null) return null;

    int inval = 0;
    try { inval = int.Parse(text); }
    catch (Exception) { return null; }
    return inval;
}

private int? GetSearchValue()
{
    if (txtSearchValue.Text == "")
    {
        MessageBox.Show("찾을 값을 입력하세요.", "에러");
        txtSearchValue.Focus();
        return null;
    }
    return this.ParseInt(txtSearchValue.Text);
}

public int ByteArrayToInt(byte[] byteArray, ushort len)
{
    int result;

    if (len == 8)
        result = BitConverter.ToInt32(byteArray, 4) << 16 + BitConverter.ToInt32(byteArray, 0);
    else if (len == 4)
        result = BitConverter.ToInt32(byteArray, 0);
    else if (len == 2)
        result = BitConverter.ToInt16(byteArray, 0);
    else result = byteArray[0];

    return result;
}
```


프로세스 메모리 읽기/쓰기 함수 정의

```
// Windows Debugging API - ReadProcessMemory
private bool ReadProcessMemory(IntPtr hProcess, ulong address, byte[] buffer, uint len, out uint bytesRead)
{
    bool bRet;
    uint oldProtect;

    //WinAPI.VirtualProtectEx(hProcess, (IntPtr)address, len, PAGE_READWRITE, out oldProtect);
    bRet = WinAPI.ReadProcessMemory(hProcess, (IntPtr)address, buffer, len, out bytesRead);
    //WinAPI.VirtualProtectEx(hProcess, (IntPtr)address, len, oldProtect, out _);
    return bRet;
}

private bool WriteProcessMemory(IntPtr hProcess, ulong address, byte[] buffer, ushort len)
{
    bool bRet;
    uint oldProtect;
    uint bytesWritten = 0;

    WinAPI.VirtualProtectEx(hProcess, (IntPtr)address, len, PAGE_READWRITE, out oldProtect);
    bRet = WinAPI.WriteProcessMemory(hProcess, (IntPtr)address, buffer, len, out bytesWritten);
    WinAPI.VirtualProtectEx(hProcess, (IntPtr)address, len, oldProtect, out _);
    return bRet;
}
```

```
private void btnNewSearch_Click(object sender, RoutedEventArgs e)
{
    tmrUpdater.Stop();
    MakeProcessMemoryBlockInfo();

    IntPtr hProcess = OpenProcess(PROCESS_ALL_ACCESS, false, selectedProcId);
    if (hProcess == NULL)
    {
        MessageBox.Show("프로세스에 접근할 수 없습니다.", "에러");
        return;
    }

    int? searchVal = GetSearchValue();
    if (searchVal == null) return;

    ushort size = this.selectedSize;
    this.ClearViewModelDI();
    uint cnt = 0;

    for (int i = 0; i < lstMemoryBlock.Count; i++)
    {
        ulong beginAddress = lstMemoryBlock[i].BaseAddress;
        ulong endAddress = lstMemoryBlock[i].BaseAddress + lstMemoryBlock[i].RegionSize;
        uint regionSize = (uint)lstMemoryBlock[i].RegionSize;

        byte[] block = new byte[lstMemoryBlock[i].RegionSize];
        bool bRet;
        uint bytesRead = 0;

        bRet = this.ReadProcessMemory(hProcess, beginAddress, block, regionSize, out bytesRead);
        if (bRet != true) continue;
    }
}
```

```

for (uint pos = 0; pos < regionSize; pos += size)
{
    byte[] buffer = new byte[size];
    for (int a = 0; a < size; a++)
        buffer[a] = block[pos + a];

    int readVal = this.ByteArrayToInt(buffer, size);
    if (readVal == searchVal)
    {
        ulong address = beginAddress + pos;
        DataItem de = new DataItem();
        if (is32bitProcess)
            de.HexAddress = address.ToString("X8");
        else
            de.HexAddress = address.ToString("X12");
        de.PrevValue = readVal;
        de.NowValue = readVal;
        de.Size = size;
        ViewModelDI.ocDataItem.Add(de);
        cnt++;
    }
}
txtMessage.Text = cnt + "개를 찾았습니다.";

if(ViewModelDI.ocDataItem.Count <= 0)
    btnFilterSearch.IsEnabled = false;
else
    btnFilterSearch.IsEnabled = true;

tmrUpdater.Start();
}

```

private void ClearViewModelDI()

```
private void ClearViewModelDI()
{
    for (int i = 0; i < ViewModelDI.ocDataItem.Count; i++)
    {
        DataItem di = ViewModelDI.ocDataItem[i];
        if (!di.Fix && !di.Freeze)
        {
            ViewModelDI.ocDataItem.RemoveAt(i);
            i--;
        }
    }
}

private void btnClear_Click(object sender, RoutedEventArgs e)
{
    this.ClearViewModelDI();
}
```

```

private void btnFilterSearch_Click(object sender, RoutedEventArgs e)
{
    tmrUpdater.Stop();
    IntPtr hProcess = OpenProcess(PROCESS_ALL_ACCESS, false, selectedProcId);
    if (hProcess == NULL)
    {
        MessageBox.Show("프로세스에 접근할 수 없습니다.", "에러");
        return;
    }

    int? searchVal = GetSearchValue();
    if (searchVal == null)
        return;

    int cnt = 0;
    bool bRet = false;
    uint bytesRead = 0;
    byte[] memBlock = new byte[100];

    ObservableCollection<DataItem> colNewViewModelDI = new ObservableCollection<DataItem>();
    int preBlockIndex = -1, curBlockIndex = -1;
    for (int i=0; i<ViewModelDI.ocDataItem.Count; i++)
    {
        DataItem di = ViewModelDI.ocDataItem[i];
        ushort size = di.Size;

        curBlockIndex = GetMemoryBlockIndex(di.Address);
        if (preBlockIndex != curBlockIndex)
        {
            MEMORY_BASIC_INFORMATION64 mbi = lstMemoryBlock[curBlockIndex];
            memBlock = new byte[mbi.RegionSize];
            bRet = this.ReadProcessMemory(hProcess, mbi.BaseAddress, memBlock, (uint)mbi.RegionSize, out bytesRead);
            preBlockIndex = curBlockIndex;
        }
    }
}

```



```

if (bRet)
{
    byte[] buffer = new byte[size];
    ulong beginAddr = di.Address - 1stMemoryBlock[curBlockIndex].BaseAddress;

    for (uint a=0; a<size; a++)
        buffer[a] = memBlock[beginAddr+a];

    int readVal = this.ByteArrayToInt(buffer, size);
    if (readVal != searchVal)
    {
        if(di.Fix || di.Freeze)
        {
            colNewViewModelDI.Add(di);
        }
    }
    else
    {
        colNewViewModelDI.Add(di);
        int lastIdx = colNewViewModelDI.Count - 1;
        colNewViewModelDI[lastIdx].PrevValue = colNewViewModelDI[lastIdx].NowValue;
        colNewViewModelDI[lastIdx].NowValue = readVal;
        cnt++;
    }
}

dgDataItem.ItemsSource = null;
ViewModelDI.ocDataItem = colNewViewModelDI;
dgDataItem.ItemsSource = ViewModelDI.ocDataItem;
WinAPI.CloseHandle(hProcess);

txtMessage.Text = "재탐색하여 " + cnt + "개를 찾았습니다.";
tmrUpdater.Start();
}

```

private void dgDataItem_CellEditEnding(...)

```
private void dgDataItem_CellEditEnding(object sender, DataGridViewCellEditEndingEventArgs e)
{
    // 편집된 셀 index 얻는법
    int index = dgDataItem.Items.IndexOf(dgDataItem.CurrentItem);

    if (e.EditAction == DataGridViewEditAction.Commit)
    {
        var column = e.Column as DataGridViewBoundColumn;
        if (column != null)
        {
            var bindingPath = (column.Binding as Binding).Path.Path;
            if (bindingPath == "NowValue")
            {
                int rowIndex = e.Row.GetIndex();
                var el = e.EditingElement as TextBox;

                int? intval = this.ParseInt(el?.Text);
                if (intval == null) return;

                DataItem di = ViewModelDI.ocDataItem[rowIndex];
                byte[] buffer = BitConverter.GetBytes((int)intval);

                IntPtr hProcess = WinAPI.OpenProcess(PROCESS_ALL_ACCESS, false, selectedProcId);
                this.WriteProcessMemory(hProcess, di.Address, buffer, selectedSize);
                WinAPI.CloseHandle(hProcess);
            }
        }
    }
}
```

private void dgDataItem_BeginningEdit(...)

```
        else if(bindingPath == "HexAddress")
        {
            int rowIndex = e.Row.GetIndex();
            var el = e.EditingElement as TextBox;
        }
    }
}

ObservableCollection<DataItem> oc = dgDataItem.ItemsSource.Cast<DataItem>() as ObservableCollection<DataItem>;
index = dgDataItem.Items.IndexOf(dgDataItem.CurrentItem);
try
{
    if (0 <= index && index < dgDataItem.Items.Count && oc[index].Size <= 0)
    {
        oc[index].Size = selectedSize;
    }
}
catch (Exception) { };
this.autoUpdateEnabled = true;
}

private void dgDataItem_BeginningEdit(object sender, DataGridBeginningEventArgs e)
{
    autoUpdateEnabled = false;
}
```

public void UpdateViewModelDI()

```
public void UpdateViewModelDI()
{
    tmrUpdater.Stop();
    IntPtr hProcess = OpenProcess(PROCESS_ALL_ACCESS, false, selectedProcId);
    if (hProcess == NULL)
        return;

    int? searchVal = GetSearchValue();
    if (searchVal == null)
        return;

    bool bRet = false;
    uint bytesRead = 0;
    byte[] memBlock = new byte[100];

    int preBlockIndex = -1;
    for (int i = 0; i < ViewModelDI.ocDataItem.Count; i++)
    {
        DataItem di = ViewModelDI.ocDataItem[i];
        ushort size = di.Size;
```

```

if ((bool)chkAutoUpdate.IsChecked && autoUpdateEnabled)
{
    if (di.Freeze != true)
    {
        int curBlockIndex = GetMemoryBlockIndex(di.Address);
        if (preBlockIndex != curBlockIndex)
        {
            MEMORY_BASIC_INFORMATION64 mbi = lstMemoryBlock[curBlockIndex];
            memBlock = new byte[mbi.RegionSize];
            bRet = this.ReadProcessMemory(hProcess, mbi.BaseAddress, memBlock, (uint)mbi.RegionSize, out bytesRead);
            preBlockIndex = curBlockIndex;
        }

        if (bRet)
        {
            byte[] buffer = new byte[size];
            ulong beginAddr = di.Address - lstMemoryBlock[curBlockIndex].BaseAddress;

            for (uint a = 0; a < size; a++)
                buffer[a] = memBlock[beginAddr + a];

            int readVal = ByteArrayToInt(buffer, size);
            ViewModelDI.ocDataItem[i].NowValue = readVal;
        }
    }
}

WinAPI.CloseHandle(hProcess);
tmrUpdater.Start();
}

```


public void FreezeViewModelDI()

```
public void FreezeViewModelDI()
{
    IntPtr hProcess = WinAPI.OpenProcess(PROCESS_ALL_ACCESS, false, selectedProcId);
    if (hProcess == NULL || cmbProcess.SelectedIndex <= 0)
        return;

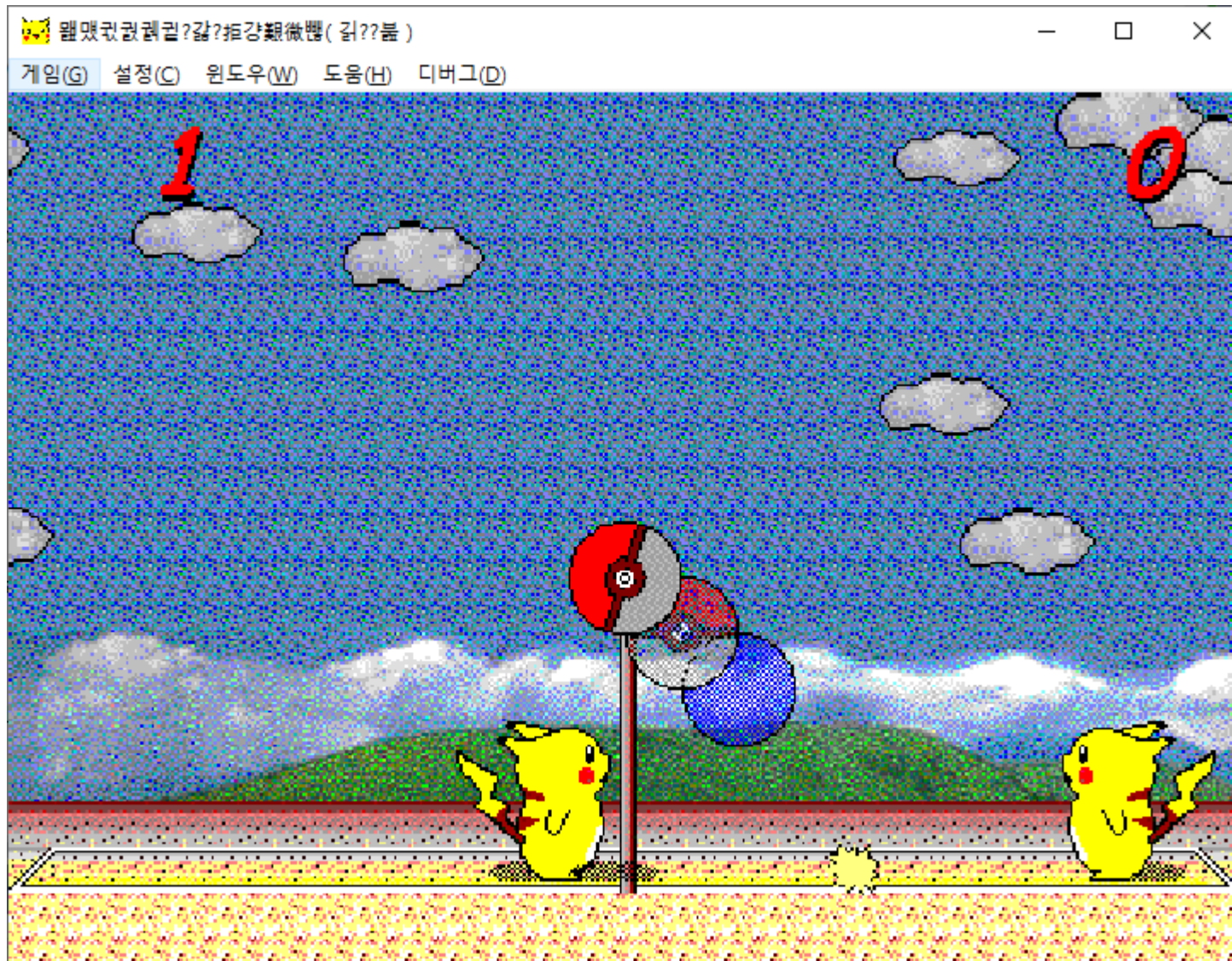
    foreach (DataItem di in ViewModelDI.ocDataItem)
    {
        if (di.Freeze == true)
        {
            byte[] buffer = BitConverter.GetBytes(di.NowValue);
            this.WriteProcessMemory(hProcess, di.Address, buffer, di.Size);
        }
    }
    WinAPI.CloseHandle(hProcess);
}
```

private void OnDataItemChecked(...)

```
<DataGridCheckBoxColumn.CellStyle>  
    <Style>  
        <EventSetter Event="CheckBox.Checked" Handler="OnDataItemChecked"/>  
        <EventSetter Event="CheckBox.Unchecked" Handler="OnDataItemChecked"/>  
    </Style>  
</DataGridCheckBoxColumn.CellStyle>
```

```
private void OnDataItemChecked(object sender, RoutedEventArgs e)  
{  
    chkAutoUpdate.Focus();  
}
```

피카츄 배구



[젼보이] 포켓몬 Fire Red



삼국지 10 PK



플래시 게임 - 검 강화하기

Adobe Flash Player 30

검 강화하기

판매하다

강화비용:35,000원
판매가격:350,000원



+12 화염의 쌍검

성공률 70%

방지권:0
돈: 19980000 원

by NBS

CheatOmatic

프로세스 목록: [15172] (x32) FlashPlayer 30 - Adobe Flash Player 30

자료 검색: 4Byte 15912800 초기탐색 재탐색 전체삭제 ☒ 자동갱신

고정	주소(hex)	크기	현재값(설정값)	이전값	설명	얼리기
☐	0408203C	4	8000000	7988000		☐
☐	0408209C	4	8000000	7960000		☐
☐	040822AC	4	8000000	7976000		☐
☐	040822CC	4	8000000	7984000		☐
☐	0408233C	4	8000000	7862400		☐
☐	040829FC	4	8000000	7997600		☐
☐	04082AAC	4	16000000	7960000		☐
☐	04082AEC	4	8000000	7993600		☐
☐	04082B3C	4	8000000	7997600		☐
☐	04082B6C	4	8000000	7976000		☐
☐	04082BDC	4	8000000	7996000		☐
☐	04082C1C	4	160000000	15912800		☐
☐	04082C2C	4	8000000	7953600		☐
☐	04082E0C	4	8000000	7984000		☐
☐						☐

1개를 찾았습니다.

플래시 게임 - 검 강화하기

Adobe Flash Player 30

보관하기 ● **검 강화하기** ● 판매하다 ●

강화비용:왕푸야샤 1자루
판매가격:101,000,000원

TIP
다음 강화재료는
템페스트 골드 2자루입니다



+21 템페스트 골드

왕푸야샤:0
방지권:19999926 성공률 40%
돈: 18500000 원

by NBS

CheatOmatic

프로세스 목록: [15172] (x32) FlashPlayer 30 - Adobe Flash Player 30

자료 검색: 4Byte 10637600 초기탐색 재탐색 전체삭제 ☒ 자동갱신

고정	주소(hex)	크기	현재값(설정값)	이전값	설명	얼리기
☐	00AFAE64	4	160000000	10637600		☒
☐	0408203C	4	160000000	7988000		☒
☐	0408209C	4	160000000	7960000		☒
☐	040822AC	4	160000000	7976000		☒
☐	040822CC	4	160000000	7984000		☒
☐	0408233C	4	160000000	7862400		☒
☐	040829FC	4	160000000	7997600		☒
☐	04082AAC	4	160000000	7960000		☒
☐	04082AEC	4	160000000	7993600		☒
☐	04082B3C	4	160000000	7997600		☒
☐	04082B6C	4	160000000	7976000		☒
☐	04082BDC	4	160000000	7996000		☒
☐	04082C1C	4	160000000	15912800		☒
☐	04082C2C	4	160000000	7953600		☒
☐	04082E0C	4	160000000	7984000		☒
☐	04082E8C	4	160000000	10637600		☒
☐						☐

2개를 찾았습니다.

플래시 게임 – ravin 타워디펜스

Adobe Flash Player 30

파일(F) 보기(V) 컨트롤(C) 도움말(H)



CheatOmatic

프로세스 목록: [10776] (x32) FlashPlayer 30 - Adobe Flash Player 30

자료 검색: 4Byte 38888 초기탐색 재탐색 전체삭제 ☒ 자동갱신

고정	주소(hex)	크기	현재값(설정값)	이전값	설명	얼리기
☒	00537828	4	36416	4552		☐
☒	00DFF77C	4	36416	4552		☐
☒	0414603C	4	67410126	4552		☐
☐	0036C114	4	38888	38888		☐
☐	0414684C	4	800000	38888		☐
☐	04191C3C	4	800000	38888		☐
☐						☐

3개를 찾았습니다.



107

삼국지 10 PK

일 모음

SBA파일들



```

Umeisot the Poker
Human of Trog *.....
체력: 18/18
마력: 1/1
AC: 2             힘 : 17
회피: 12          지능: 7
방패: 0           민첩: 12
XL: 1 Next: 0%   위치: Dungeon:1
소음:             시간: 0.0 (0.0)
a) +0 spear
-) Nothing quivered

```



```
Lua error:
D:/Games/dcss0.24_korean_60_99_1228/dat/des/arrival/twisted.des:925: unknown monster: "인간" x25
Welcome, Umeisot the Human Berserker.
당신은 깊은 곳으로부터, 전설 속의 조트의 오브를
되찾아올 유일한 자가 될 것인가?
트로그가 말했다: 모두 죽여라!
?를 누르면 명령 일람 및 기타 정보를 확인할 수
있습니다.
--다음장--
```

