mud lecture

[Lesson] 6. Playing with tilesets (1) - Tileset format, dynamic allocation

why do you ask
2014. 1. 27. 17:37

add neighbor

In this lesson, we will change the tileset in real time.
So from Badlands to Ice or something like that.
In (3), we will expand on how to upload a custom tileset.

Roughly, the order is as follows:
1: Tileset format, dynamic allocation
2: Custom tilesets

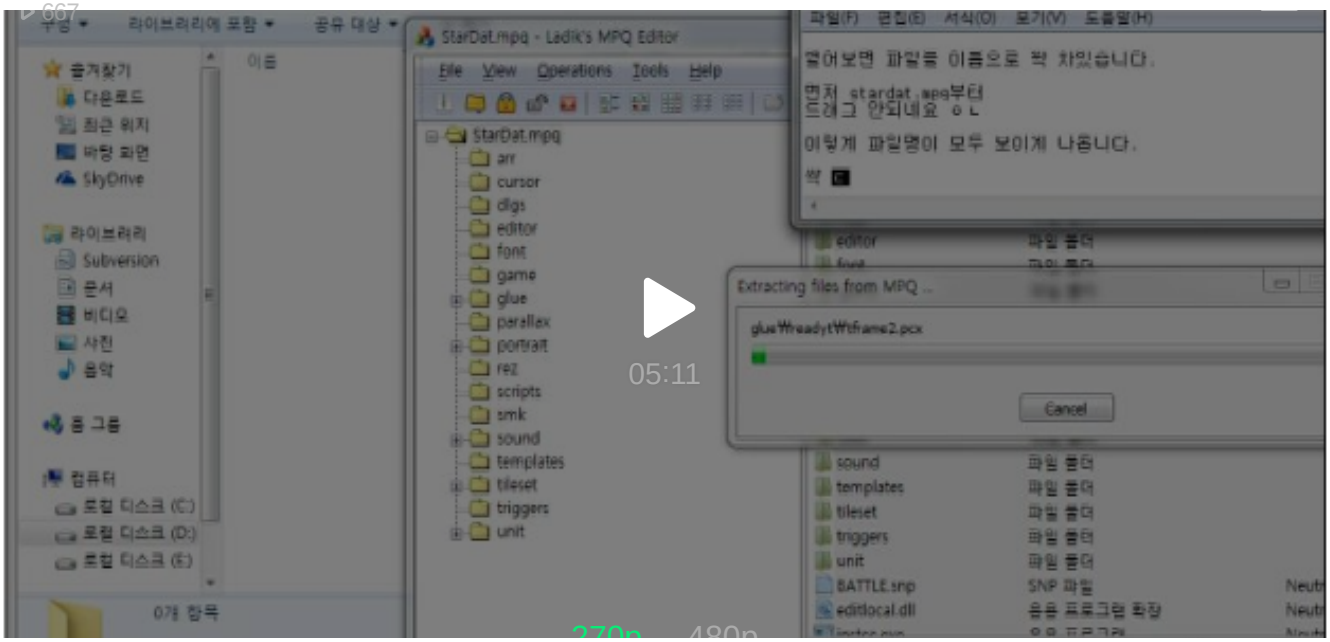
Part 1 is a bit difficult, but it's worth it.
Part 2 may seem difficult because it's programming, but it's actually easy.
(The concept is very easy, but it will seem difficult to just use Python for no other reason.)

necessary basic knowledge)

1 . You must know how to extract files from stardat.mpq, broodat.mpq, and patch_rt.mpq.

If you do not know how to extract, see the following video.

[Lesson] 6. Playing with tilesets (1) - Tileset format, dynamic allocation



whyask37's blog

Download Ladik's mpq editor: http://www.zezula.net/en/mpq/download.html
Listfile.txt is Listfiles for MPQ Archives . It seems to be a bit too big for uploading attachments.

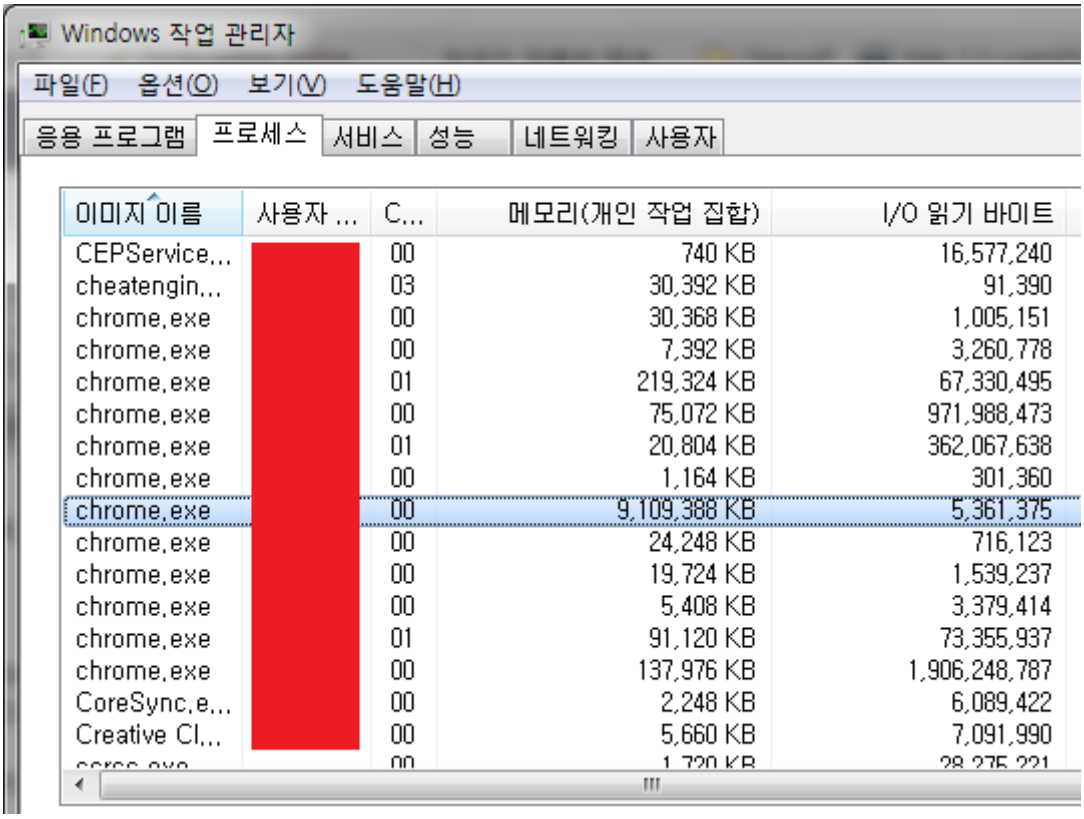
Let's start with dynamic allocation as a theory.
I tried to explain it as briefly as possible, but I don't know if I'm trying to convey the content well.

1. What is dynamic allocation?
I'm making a program
' Oh, some maps have 12345 triggers. 29776140byte (12345 * 2412) would be needed as a tri gger in this map.' (about 28 MB)
' Oh, some maps have three triggers. You'll need 7236 bytes as a trigger in this map.' (About 7 kb)

In this way, **there are times when you need a memory of** which the **capacity is not specifie d** . In this case, the star
1. Calculate the required capacity.
2. Dynamically allocates memory with the required amount of space. (malloc)
-> The operating system finds **a space somewhere** between 29776140byte or 7236byte and gives it to the star.
3. Take the memory and stir-fry it well.
4. Release the memory that has been used up. (free)
-> The operating system manages the space corresponding to the exhausted memory again. This memory is later recycled when other programs or stars dynamically allocate it.

You will go through the process. in short
Dynamic allocation: requesting and receiving the amount of memory we want from the operatin g system
Release: Giving the memory we received back to the operating system so that other programs can use it again.

This is what happens if you don't turn it off.



Warrior) , but did not want Although long synthetic paint like that and turn the program on my PC is like that anhamyeon the free memory is really well.
Warrior) yieotdamyeon real chrome anhaneun a free program like that would have lost the entir e chromium. Chrome doesn't. It just quiets a lot of memory
A case in which the dynamically allocated memory cannot be deallocated is called a 'memory le ak'.

The address of dynamically allocated memory is at the discretion of the operating system.
The dynamically allocated memory itself is not at the address 00613FF4.
With a pointer to this memory, the star handles dynamically allocated memory,
We're going **to play around with pointers to dynamically allocated memory** like this **.**

(The dynamically allocated memory is somewhere, which means that a variable with a value of 'somewhere' will be manipulated as EUD. 'Somewhere' is whatever you want it to be.)

If the address of dynamically allocated memory is 0x 01000000
SMemFree(0x 01000000); This will free the memory block at 0x 01000000.

When the star puts this address in 005993D0 (that is , the value of address 005993D0 is 0x01 000000)
SMemFree(value of 005993D0); I have code like

whyask37's blog

```
fastFileRead(0, 0, &szFileName, &wpedata_5994E0, 0, "StarcraftWWSWARWWlangWWgamedata.cpp", 210);
sprintf_s("%s%s%s", "TilesetWW", off_512998[(unsigned __int16)word_57F1DC], ".vF4");
dword_5993D0 = fastFileRead(0, (int)&hFile, &szFileName, 0, 0, "StarcraftWWSWARWWlangWWgamedata.cpp", 210);
word_5998E0 = (unsigned int)hFile >> 5;
```

If you have this (fastFileRead is SMemAlloc + a function that does miscellaneous things)

```
  SMemFree(dword_6D5EC8, "StarcraftWWSWARWWlangWWGamemap.cpp", 472, 0);
if ( dword_5993D0 )
  SMemFree(dword_5993D0, "StarcraftWWSWARWWlangWWGamemap.cpp", 473, 0);
if ( mapDataPtr )                                       // map data
  SMemFree(mapDataPtr, "StarcraftWWSWARWWlangWWGamemap.cpp", 475, 0);
```

There is also this. It's always a match.

If you change the value of 005993D0 to 0x02000000 with EUD,
SMemFree(0x02000000); It crashes when it runs.

That is, when going to SMemFree, you must unconditionally change the value of 005993D0 ba ck to 0x01000000.
But it's not that easy.

I clean it up.

1. Memory allocation (receives a pointer to a block of memory)
2. Use
3. Release memory

You should follow these three steps carefully.

(To receive a pointer to memory means to receive the address of that memory as a value)

Now let's get to the point.

I want to change the tileset! so
First of all, there is something like this in EUDDB. Let's transform this.
0057F1DC1.16.1WinTileset2OneThe current tileset ID being used.
(2) Start the game with Astral Balance.scm (Space tileset) , (This is the Broodwar base map)
I changed that address from 01 (Space) to 02 (Installation) with Cheat Engine.

whyask37's blog

How many stars are floating in space? Other than that, nothing happened.
ah it's messed up I have to give up.

Until then, there would have been no need to explain dynamic allocation like this.
아무래도 그냥 저기 변경하는것만으론 안되나봅니다.
스타에서 지형을 어떻게 출력하는지 더 알아보도록 합시다.

이번 강좌에서 다루는 것

1. STR 단락에 데이터를 넣는 법을 배운다.
2. 포인터를 어떻게 사용할 수 있는지 알아본다.
3. 동적 할당의 개념을 이해한다.

간단하게 CV5, VX4, VR4, VF4, WPE 파일에 대해 설명하고 지나가도록 하겠습니다.
는 예전에 했던 설명이 있으니까 이거 참고하세요.

whyask37's blog

코드 빼고 다 이해하시면 되요. 제가 지금 다시 쓴다 하더라도 저 설명보다 더 잘 쓰기는 힘들것같네요.

스타에서는 매 맵을 로딩할때마다 CV5 파일같은것들을 메모리에 읽어들입니다.
그 다음에 그 메모리들에 대한 포인터를 써가지고
그 포인터들을 통해서 타일맵을 렌더링하고, 충돌 체크를 하고, 등등을 합니다.
대략 오프셋들을 정리하면 다음과 같습니다.

005993D0가 .vf4가 로딩된 메모리를 가르키는 포인터. 동적 할당된 메모리를 가르킴.
006D5EC8가 .cv5가 로딩된 메모리를 가르키는 포인터. 동적 할당된 메모리를 가르킴.
00628458가 .vx4가 로딩된 메모리를 가르키는 포인터. 동적 할당된 메모리를 가르킴.
00628444가 .vr4가 로딩된 메모리를 가르키는 포인터. 동적 할당된 메모리를 가르킴.
005994E0에는 .wpe가 직접 로딩됨. (1024바이트짜리 배열)

.wpe만 로딩 방식이 좀 특이합니다.
wpe 만지기는 귀찮으니까 (만지는 방법은 좀 위험합니다. 메모장 띄우는 수준. 궁금하신분은 SDrawUpdatePalette 연구해보세요.)
나머지를 조작해봅시다.

[시나리오]
1. 어떻게든 내가 원하는 타일셋의 vf4, cv5, vx4, vr4 파일을 스타 메모리에 올린다.
2. 위에 있는 잡다한 포인터가 가르키는 대상을 전부 메모리에 올린 cv5 등으로 변경한다.

vf4, cv5, vx4, vr4 어떻게든 메모리에 올려야겠죠. 이거부터가 좀 힘듭니다.
스타크래프트는 Badlands 타일셋의 맵에서 Space 타일셋의 vf4같은걸 메모리에 올려두지 않습니다.
오직 badlands.vf4같은것만 올려둡니다. space.vf4같은거는 **우리가 만드는 맵에 포함시켜야 합니다.**

맵 용량이 wav 파일 없이 거뜬히 1MB를 넘는 괴현상을 보시게 될겁니다.
편하게 BMP 파일 하나 맵에 넣는다 생각하시면 되요. 진짜로요.

우리는 STR 단락에다가 cv5같은 것들을 넣을겁니다.
005993D41.16.1WinPointer to map "STR " section41Pointer default: first string
이걸 이용해서 STR 단락 안에 있는 데이터를 어떻게든 꺼내쓸 수 있겠죠.

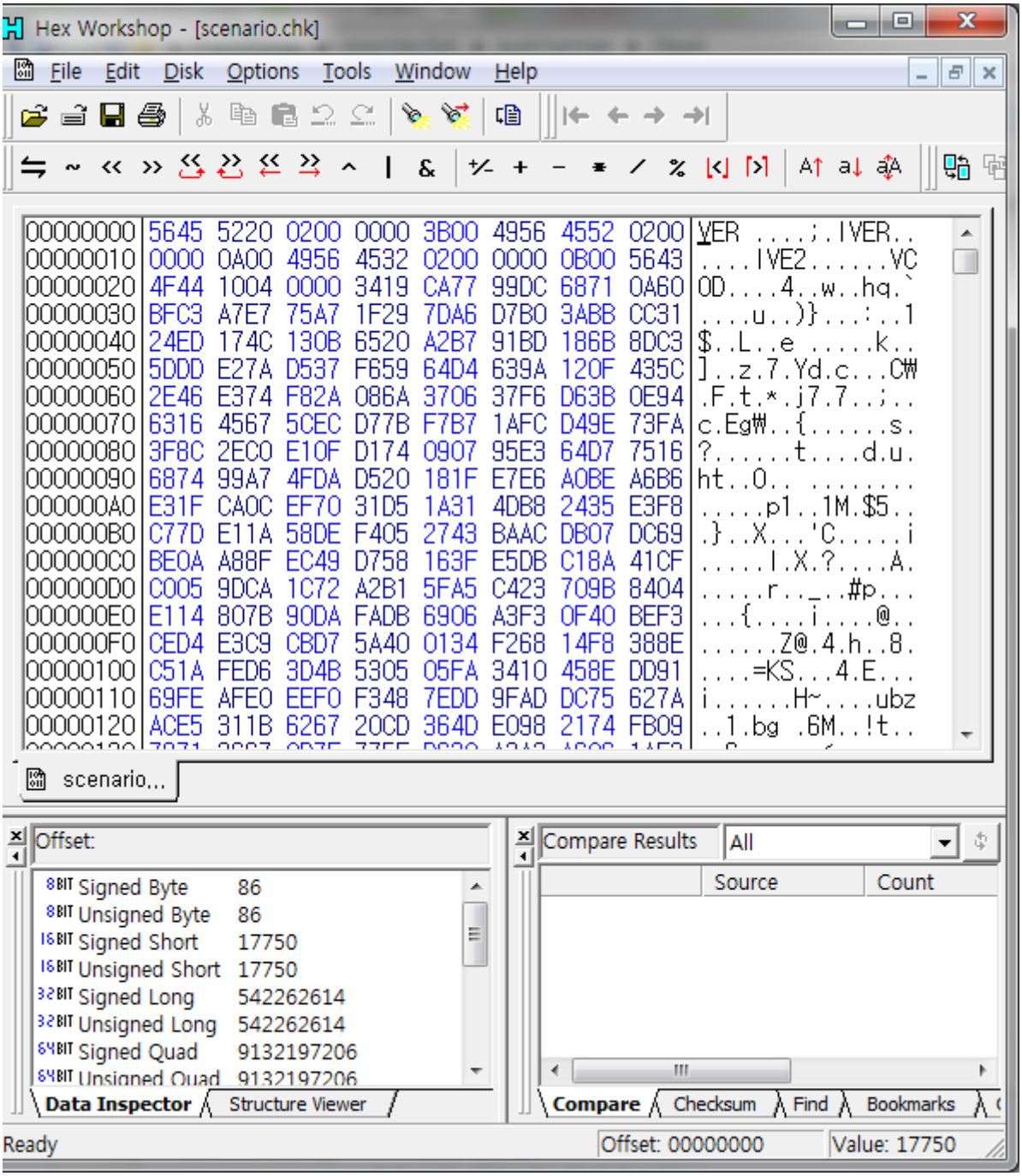
STR 단락이 어떤 것인지 알아봐야겠지요. 이를 위해서 scenario.chk의 전반적인 구조에 대해 알아볼 필요가 있습니다.
세부적인 구조는 http://www.staredit.net/starcraft/CHK 나 Hex로 배우는 scenario.chk 참조.

사족) STR 단락에 데이터를 넣는 기법은 여기에서도 쓰였습니다. http://cafe.naver.com/edac/28677

whyask37's blog

(2) Astral Balance.scm의 scenario.chk를 헥스 에디터로 열어보도록 합시다.
(scm 파일에서 scenario.chk 추출하는 방법은 알아서. WinMPQ 유틸리티 검색하면 나와요)

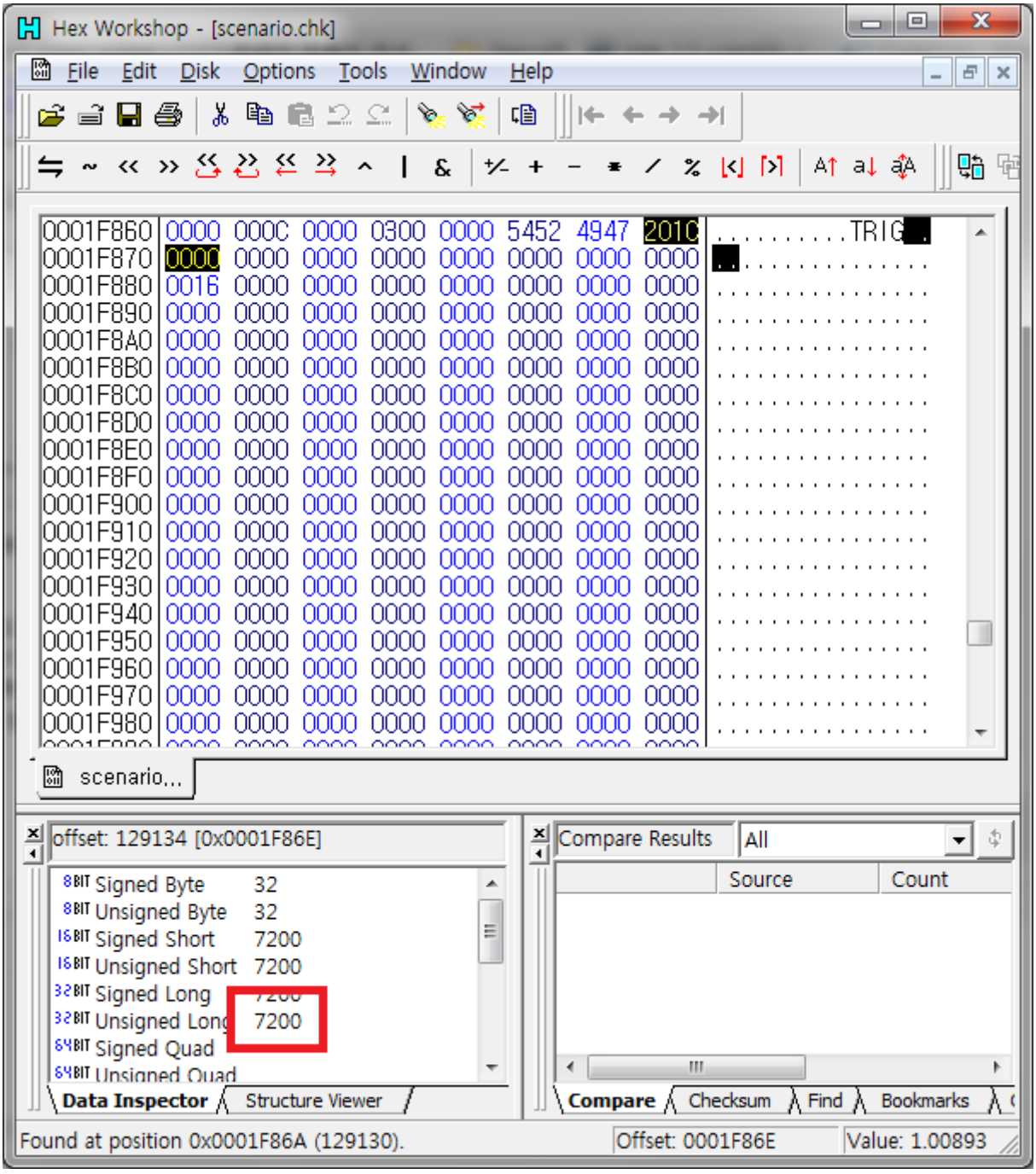
이거 배우면 언프로텍트 없이 제작자 수정만 딱 할 수 있어서 쓸까 말까 고민되긴 했는데
좀 아이러니하긴 하네요. 연구 목적으로 그냥 씁니다. 어차피 알만한 언플러들은 다 알겠지요.
프로텍트란게 어차피 결국 다 뚫리게 되있는데 프로텍트를 지키기 위해서 강좌를 안 쓰는 건
좀 맞지 않은것같습니다.



chk 파일은 여러가지의 단락으로 이루어져있습니다.
위에 보이는 VER, IVER, IVE2, VCOD같은게 단락 이름이고
단락 이름 바로 뒤에 따라오는 4byte가 해당 단락의 길이

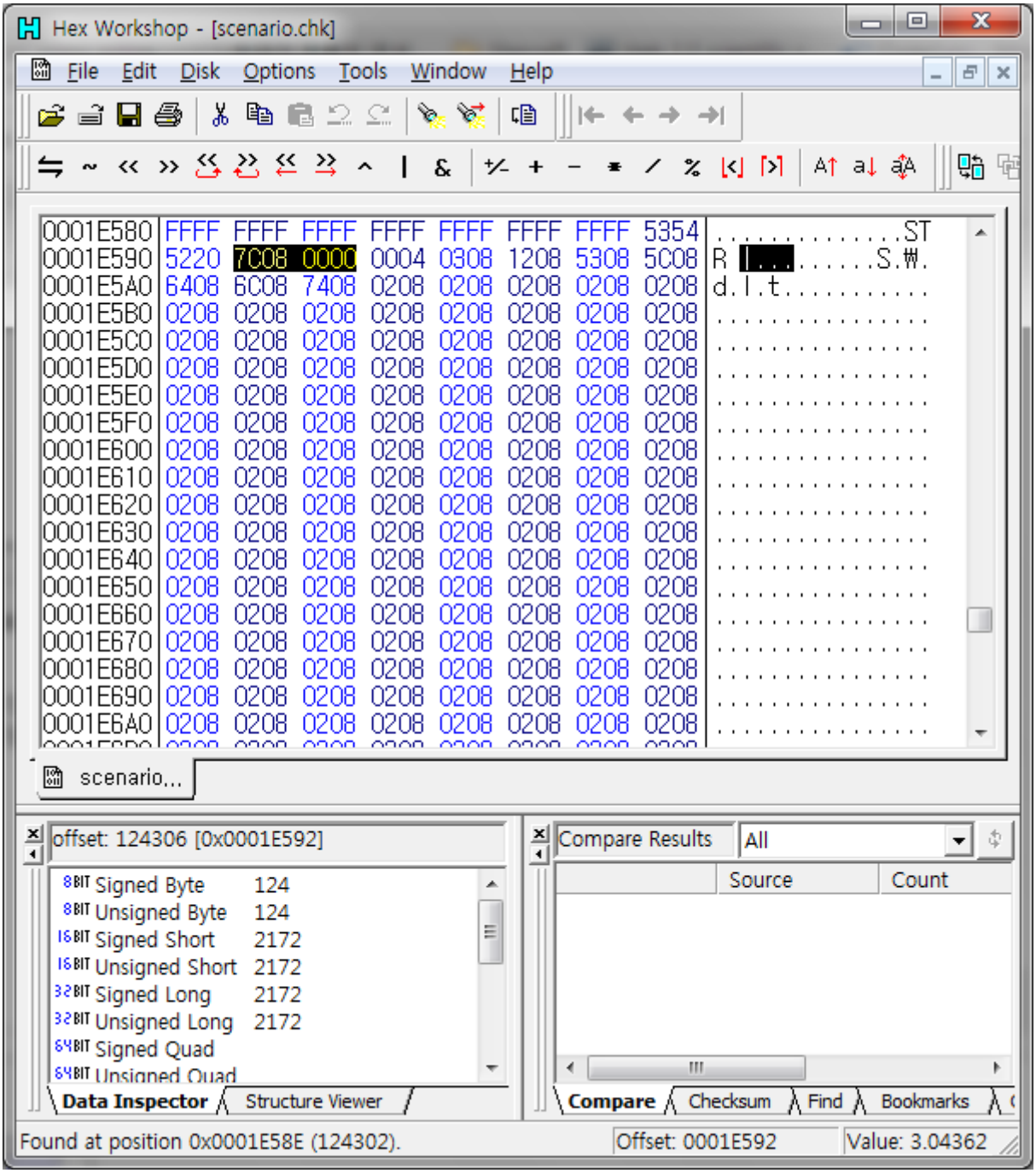
뭐 이런 식입니다.

예를 들어서 트리거는 TRIG 단락에 있는데



위 그림에 따라 TRIG 단락의 길이는 7200이고, 저 단락 길이 바로 뒤에서부터 7200byte가 TRIG 단락의 데이터를 나타냅니다.
우리가 사용할 단락은 STR입니다. 문자열 정보를 담고 있는 단락이지요.

whyask37's blog



네 바로 이 단락에다 cv5같은 것들을 넣으려고 합니다.

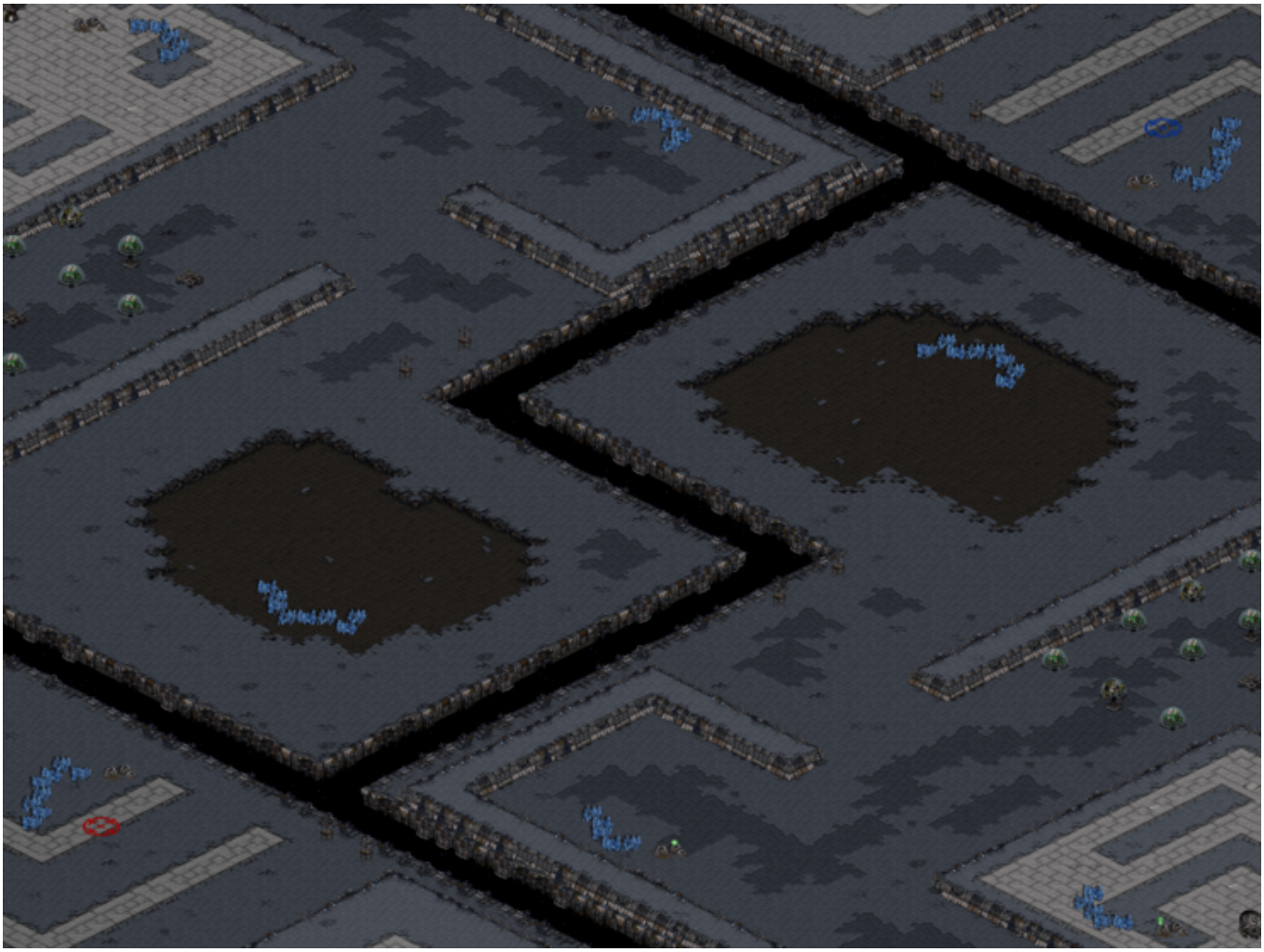
왜 하필 STR 단락을 골랐냐면,
chk 파일에 단락은 크게 다음 3종류로 나뉩니다.
- 맵 크기, 타일셋 종류, 지형, Fog of war, 로케이션 정보 등등 : 크기가 정해져 있음. (MTXM같은 단락도 정해져 있다봐야합니다)
- 유닛, 두데드, 스프라이트, 트리거, 미션 브리핑 : 무한정의 데이터를 넣을 수 있는데, 이중 연결 리스트로 관리되서 EUD로 다루기 힘듬.
- STR : 메모리가 통으로 관리되서 EUD로 다루기도 쉽고 무한정 크기의 데이터를 넣을 수 있음.

그래서 STR 단락을 쓴겁니다. 저것밖에 선택지가 없어요.

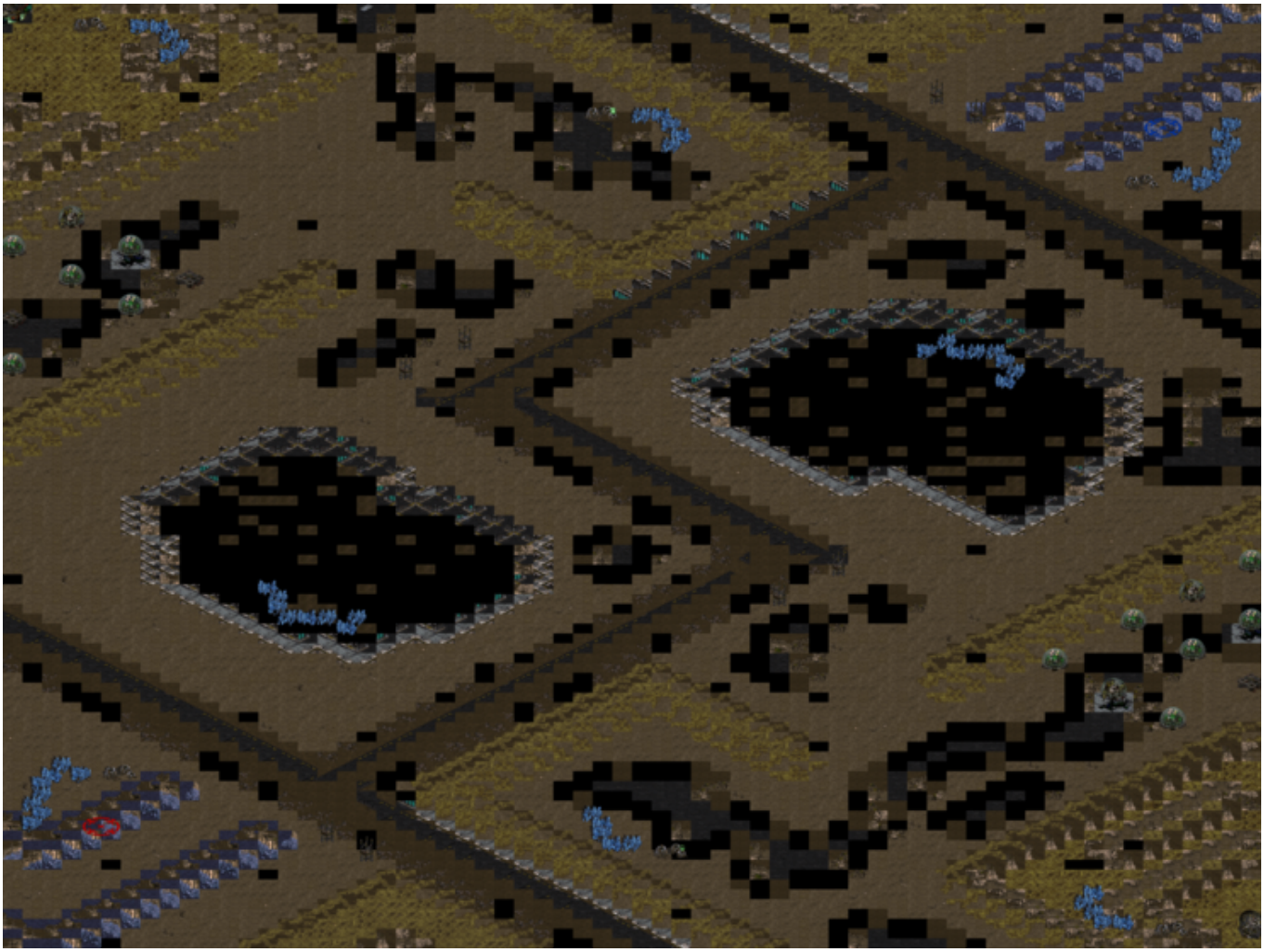
whyask37's blog

네 좋아요. 이제 이 맵에 Badlands 타일을 한번 넣어보도록 합시다.

원본:



목표 :



(SCMDraft2에서 그냥 Map Dimension에서 tileset 바꾼겁니다. 색깔은 더 깨질겁니다)

Badlands에 해당하는 타일셋 데이터를 추출하면 (StarDat.mpq 안에 tileset\ 폴더에 있습니다. WinMPQ나 Ladik's MPQ Editor로 추출)

badlands.cv5	2013-09-26 오전 11:32	CV5 파일	85KB
badlands.vf4	2013-09-26 오전 11:32	VF4 파일	152KB
badlands.vr4	2013-09-26 오전 11:32	VR4 파일	1,641KB
badlands.vx4	2013-09-26 오전 11:32	VX4 파일	152KB

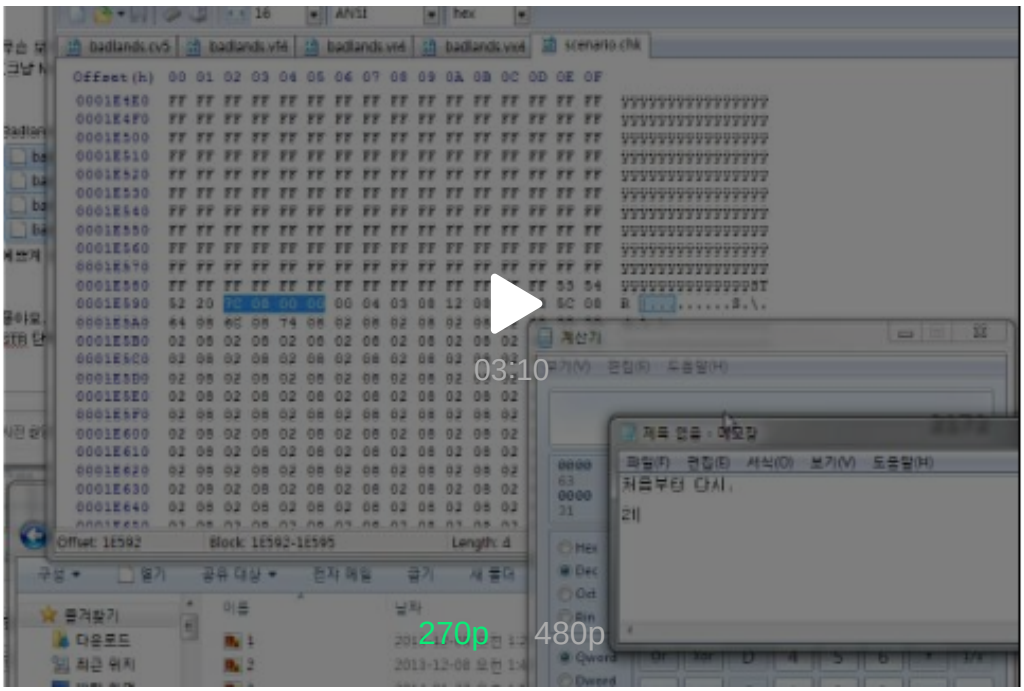
예쁘게 생겼네요.

좋아요. STR 단락을 수정해서
STR 단락 뒤에다 저 데이터들을 다 붙여넣읍시다.

whyask37's blog

[Lesson] 6. Playing with tilesets (1) - Tileset format, dynamic allocation

95



(솔직히 망해서 실수를 어딘가서 했고요 (UPRP를 실수로 70 B7 1F 00으로 덮어씌워버림) 실제 맵은 첨부파일 참조)

vf4 87C
vr4 265FC
vx4 1C07BC
cv5 1E653C
오프셋이죠.

(vr4가 265FC4라고 오타가 동영상에는 있는데 무시)

이제 이에 따라서 scenario.chk를 만들었고, 새로 맵 파일을 만들었습니다. (첨부파일의 Astral balance 2.scm)

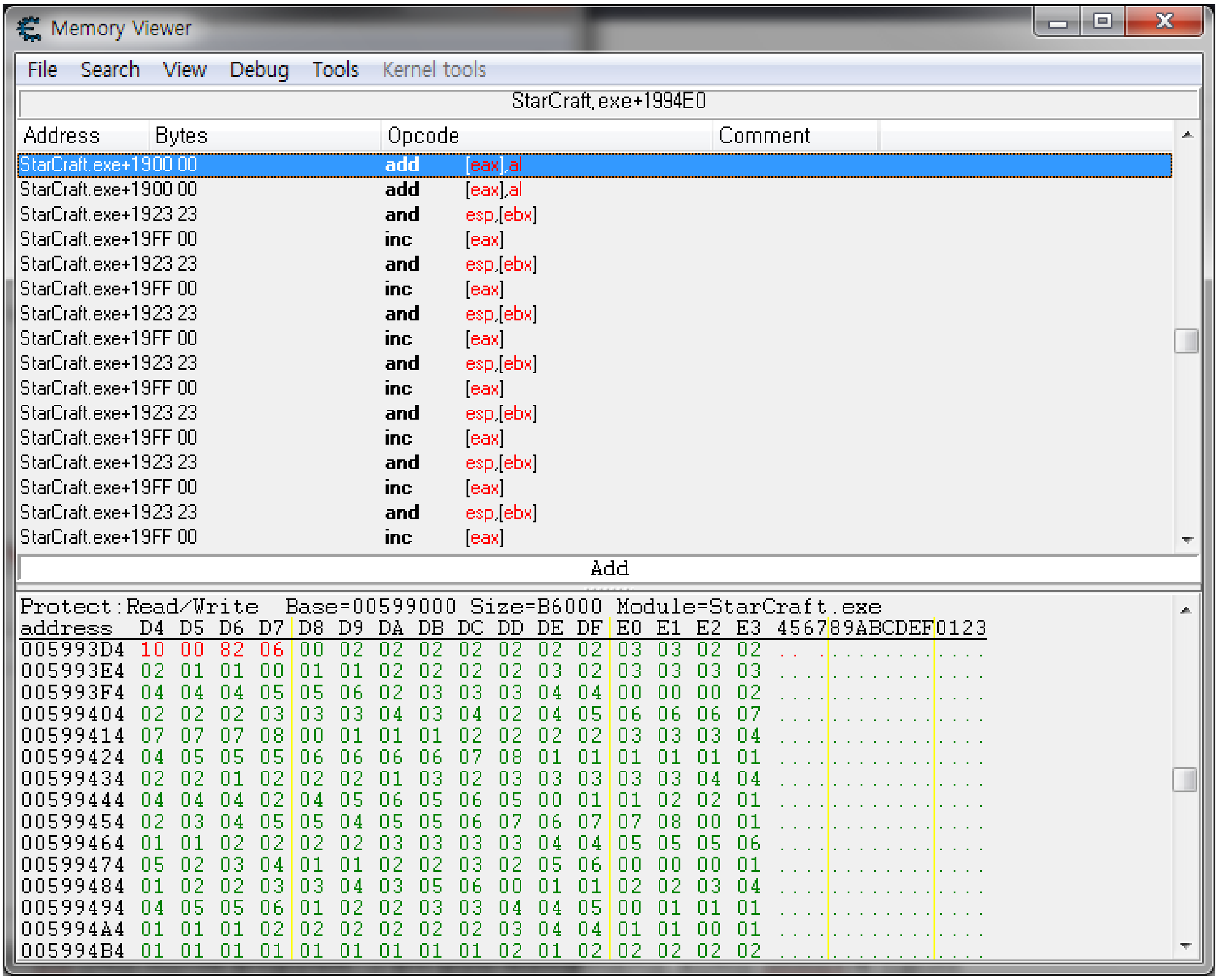
이제 이 맵을 띄웁시다.

whyask37's blog

EUD같은거 하나도 안넣었으니까 처음은 당연히 이렇죠. 테스트로 치트엔진으로 직접 메모리 조작을 해봅시다.

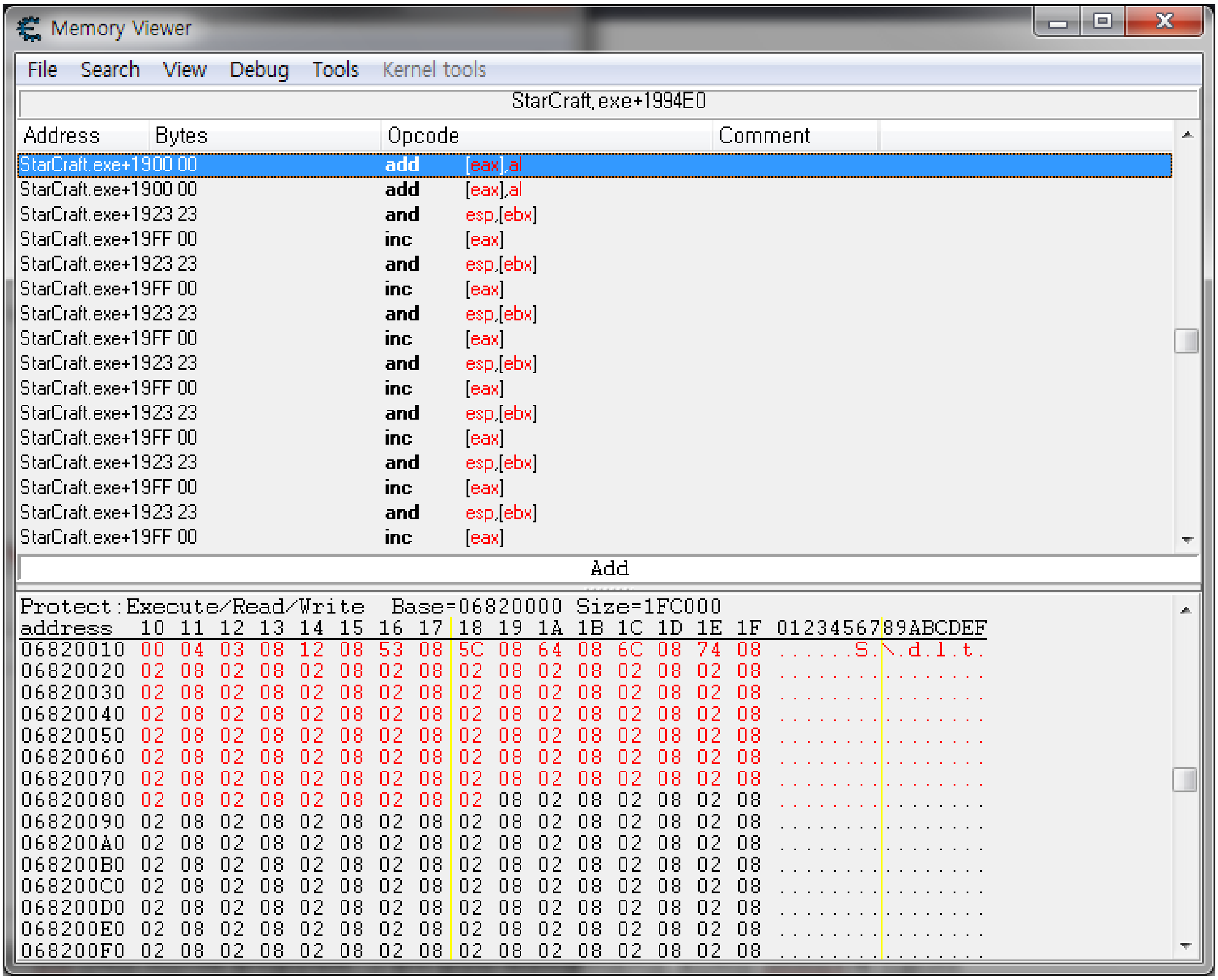
STR 단락은 어딘가에 동적할당되며, 그 동적 할당된 메모리를 가르키는 포인터는 005993D4 에 있습니다.

whyask37's blog



제 경우에는 06820010에 STR 단락의 메모리가 있다고 하네요. 가보면

whyask37's blog



STR 섹션 내용이 그대로 복붙되어있습니다.

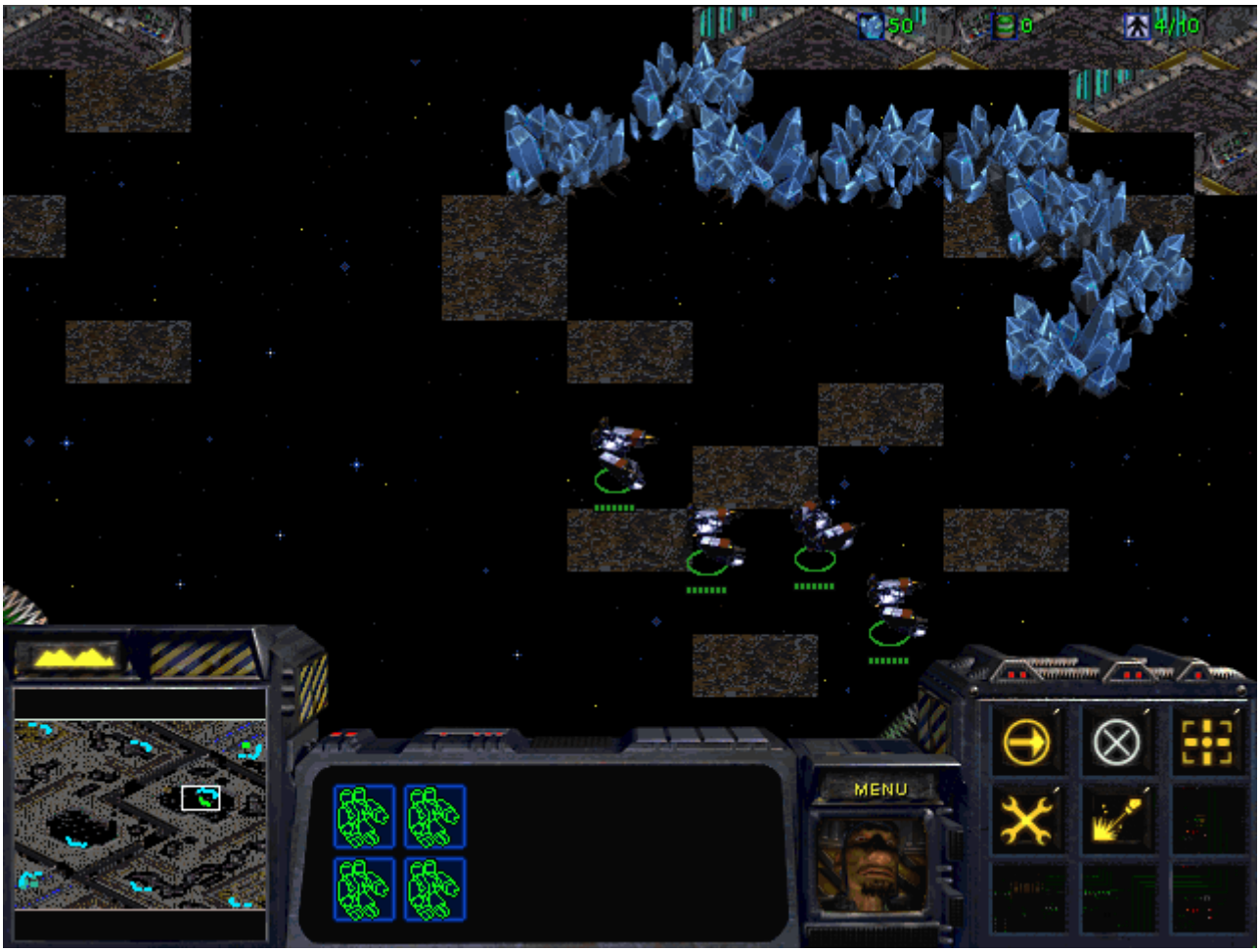
각 파일들의 오프셋이
vf4 87C
vr4 265FC
vx4 1C07BC
cv5 1E653C
이고 원래 STR 단락 시작이 06820010 니까 (여러분의 경우 다른 곳에 STR 단락이 있을겁니다.)

vf4는 682088C
vr4는 684660C
vx4는 69E07CC
cv5는 6A0654C
주소에 있겠죠.

005993D0가 .vf4가 로딩된 메모리를 가르키는 포인터. 동적 할당된 메모리를 가르킴.
006D5EC8가 .cv5가 로딩된 메모리를 가르키는 포인터. 동적 할당된 메모리를 가르킴.
00628458가 .vx4가 로딩된 메모리를 가르키는 포인터. 동적 할당된 메모리를 가르킴.

whyask37's blog

이 메모리 값들을 위 주소로 바꾸면 다음과 같이 됩니다.
사족) 바꿀 때 Ollydbg를 쓰든 뭐를 쓰든 breakpoint를 걸어서 스타 실행을 멈춰야 합니다.
치트 엔진의 경우에는 Memory viewer - Debug - Break로.



대충 된것같긴 하네요, 문제점은
1. 색깔이 깨지고 SCV가 우주를 걸어다닌다. (Region 데이터가 제대로 업데이트되지 않아서 그럼)

이정도?

whyask37's blog

저그도 해봤는데 상태가 영 안좋군요.

연습문제 (꽤 어려움)

치트엔진이나 아트머니로 메모리를 직접 수정하지 않고 EUD를 통해서 자동으로 005993D0 등등의 포인터들을 수정하는 트리거를 짜라.

Hint) STR 단락을 가르키는 포인터를 2강의 방법대로 복사한다.

뒷이야기

> 즉 SMemFree로 가는 때에는 무조건 005993D0의 값을 다시 원래대로 0x01000000 으로 바꿔두어야 합니다.
> 이게 그런데 그닥 쉬운 일이 아니죠.

포인터를 게임이 끝날 때 원래대도 되돌리는건 어렵습니다.
포인터를 그대로 방치했다가는 스타가 팅기거나, 운 좋게 안팅기더라도 메모리 리크가 생깁니다.
게임 한번 할때마다 2MB씩 메모리 리크가 생기는건 좀 그래요.

whyask37's blog

그런데 그런 방법은 없습니다. 게임이 끝날 때 트리거를 한번 더 실행시킨다는건 못들어봤어요.

원래 강좌 계획은
' 아 이러니까 스타가 팅기는군요. 이제 트리거 메타프로그래밍을 배워서 memcpy를 만들어봅시다. '
로 가려 했었는데 꼬였어요;

트리거 메타프로그래밍 강좌는 나중에 어셈블러 완성되고 할께요.

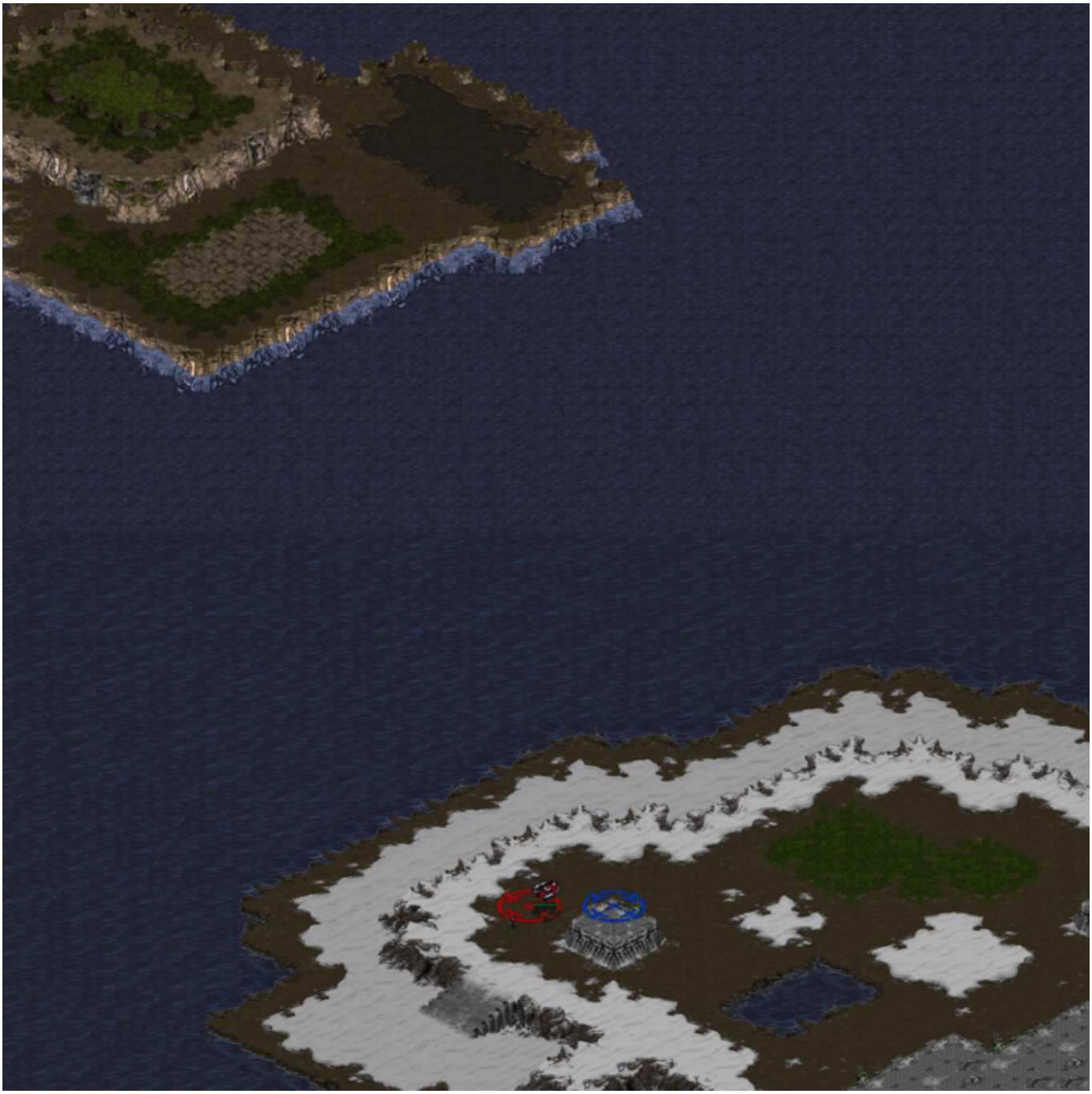
사족) vf4가 제대로 적용이 안됐네요. vf4까지 미리 다 작업해서 커스텀 타일셋 만들어야겠습니다.

이번 강좌는 제대로 망했네요 ㅇㄴ 내용도 이해 안되고 괜히 어그로만 끌듯
STR 단락에다 데이터 넣는 기법은 앞으로도 굉장히 많이 쓸 기법입니다.

(2)에서 제대로 살려볼께요.

(2) 목표 :

whyask37's blog



재미있을것같긴 하네요. 되면 좋겠습니다만...

#IT·컴퓨터

첨부파일

(2)Astral Balance 2.scm

whyask37's blog

왜물어

whyask37님의 블로그입니다.

이웃추가

이 블로그 뻘강좌 카테고리 글

[뻘강의] 8. 트리거 프로그래밍 - 기초

2014. 2. 11.

1

[뻘강의] 7. 타일셋 가지고 놀기 (2) - 커스텀 타일셋 적용 시도 1

2014. 2. 5.

3

[뻘강의] 6. 타일셋 가지고 놀기 (1) - 타일셋 포맷, 동적 할당

2014. 1. 27.

1

[뻘강의] 5. 포인터 예제 - 유닛 제한 줄이기

2014. 1. 23.

0

[뻘강의] 4. TRG 파일 포맷

2014. 1. 21.

3

MPQ 가지고놀기 (1) - 간단한 MPQ 파일 분석

2013. 10. 19.

11

5. SFmpq (ShadowFlare's MPQ Library) 와 예제

2013. 9. 11.

1

[뻘강의] 13. 트리거 프로그래밍 - TRIG-MRGN 루프

2014. 2. 24.

0

4. scenario.chk

2013. 9. 10.

0

[뻘강의] 2. 데스 사이의 대입, 더하기

2014. 1. 19.

1

back to top

View in PC version

One