mud lecture

[Lesson Lecture] 8. Trigger Programming - Basics

why do you ask
2014. 2. 11. 12:11

Changing the tile set doesn't seem like a big deal, so I'll update the 7th or later to fill it up.
First, let's start with the 8th.
It's not that big of a problem, but there's a problem that's annoying to solve; vf4 value cached? I t's become similar.

Required Pre-Course
[Rock lecture] 4. TRG file format - http://blog.naver.com/whyask37/140205150587

Today, we will only give a taste of what trigger programming is.
Let's just change the sound of a marine dying.

The basics of trigger programming are:

trigger's prev, next (a doubly linked list)
SetDeaths(0x01234567, Add, 1234, 5678);

here

trigger's prev, next (a doubly linked list)
SetDeaths (0x01234567 , Add , 1234 , 5678);

In that way, it means changing the trigger type, the value inside, the linked state of the doubly li nked list, etc. through EUD.

In fact, the concept is simple and the principle is simple, but it is difficult to use;

If you use that concept well, you can implement looping statements and conditional statements.
The trigger is also a doubly linked list, so you can handle the prev and next pointers appropriate

whyask37's blog

Roughly, the course will proceed as follows.

1: foundation
- A taste of trigger programming
2: loop
- This is a basic loop. Let's do the structure offset initialization in a loop.
3: Connect MRGN and TRIG
- This is a process to overcome this because the MRGN capacity is small.
4: STR section design, tool making
- It is the process of drawing a map for what we want and what we want to achieve.
5: Relocation table
- Inspired by PE format. Learn about relocation tables.
6: Create a Trigger Loader
- Implement Trigger Loader in trigger corresponding to PE Loader.
7: Implement adler32 hash function
- Program a trigger to output adler32 for chat contents.
8 : [For Programmers] Implementing eudasm
- Implements a utility eudasm (eud assembler) that makes it easier to create triggers.
Part 9: Implementing a calculator using eudasm
- Create a calculator that gives you 2 when you hit 1+1 in chat. Let's support arithmetic operations and parentheses.

I wish it was possible. if not, don't

The reason why the courses are so subdivided and large
1. Even though this part is quite difficult (the difficulty of the structure rather than the difficulty of the concept.)
(Probably as difficult as sequential logic circuits? Personally, flip-flops are too difficult...)
2. As far as I know, this is my first time trying this technique. (No) Anyway, there is no reference material, so there is quite a lot to do.

When this technique is settled later (although it is doubtful whether this technique will be settled at the time of transition to S2)
I think it might be possible to shorten the course further... I think.

Here's a simple way to change the sound of a marine dying.

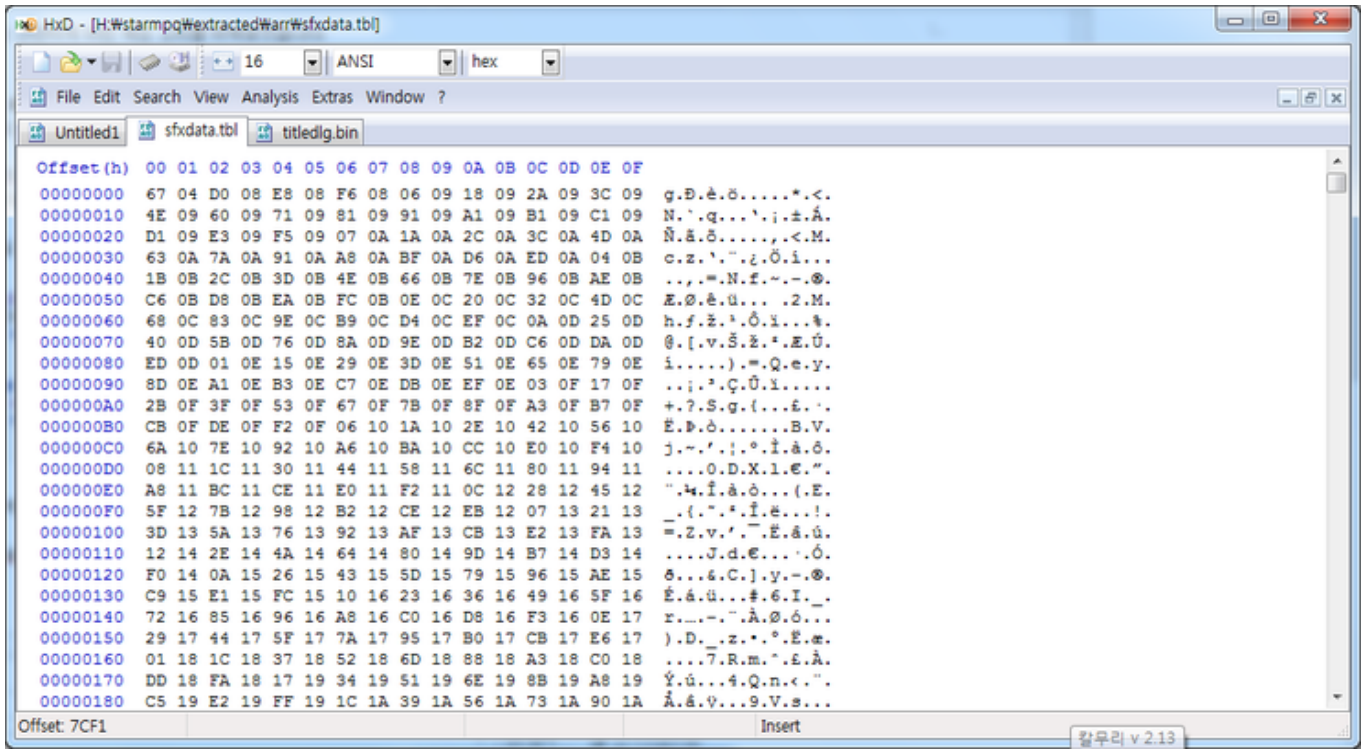
1. TBL file format - we'll cover this too.
You may have mentioned it at first glance as the STR section before. It follows the TBL file for

Star uses the TBL file format to manage string tables. TBL is a structure that efficiently manage s an array of strings (=strings).
The TBL file format can be organized into one function.

```
const char * GetString ( const char * TBLFile , int index ) { if ( index == 0 ) { return NULL ; } else { unsigned short stroffset = (( unsigned short * ) TBLFILE + index ); return TBLFile + stroffset ; } }
```

Let's take an example.

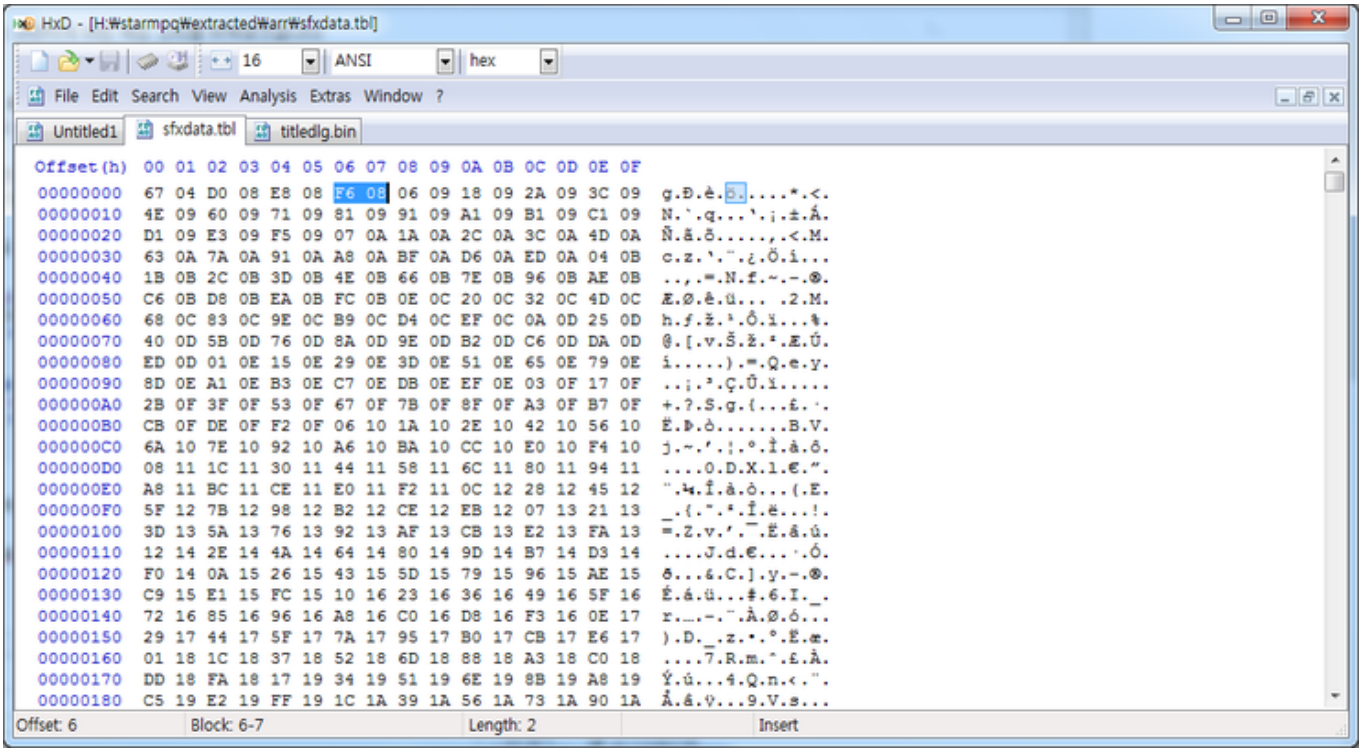
Let's open arr\sfxdata.tbl (in the star mpq files. See lesson 6 for how to extract it).



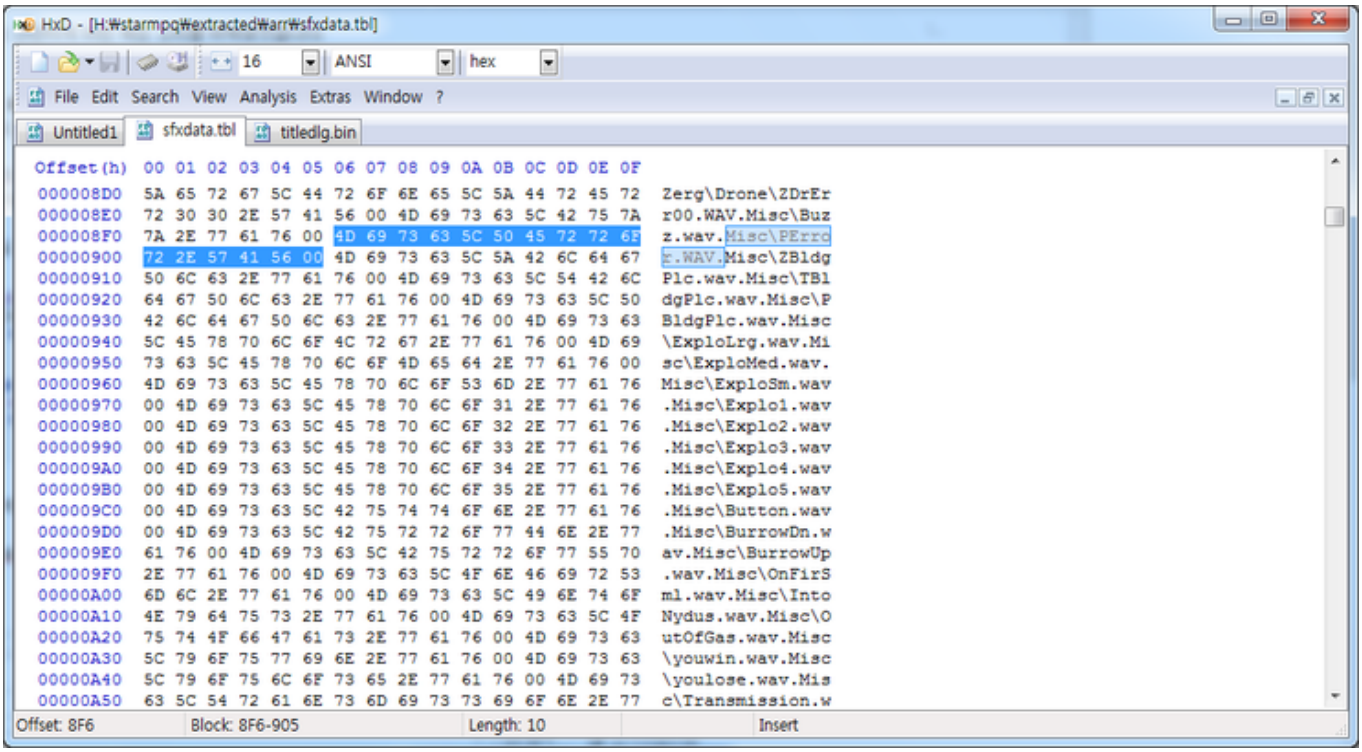
The first 0467 is the total number of strings in the TBL file. You just have to grab it a bit big. (Yo u can also grab it as FFFF)

To get string number **3 from** here, 2* **3** = 2 bytes from byte 6 should be read. (string 0 is just N ULL)

whyask37's blog



08F6 is the index of string #3. Let's go.



It says Misc\PError.WAV is the 3rd string .

the sound of marin dying

String #265 Terran\MARINE\TMaDth00.WAV. The offset relative to sfxdata.tbl for this string is 2 1B6

String #266 Terran\MARINE\TMaDth01.WAV. The offset relative to sfxdata.tbl for this string is 2

whyask37's blog

is stored in I'll have to find out more about why it's making this sound. (does not appear in DatE dit)
For once, it doesn't matter. Let's do something fun.

If you look for memory
arr\sfxdata.tbl will be in dynamically allocated memory when star first starts up
The address pointing to that memory becomes 5999B0.

Perhaps you can change the value of 5999B0 to 'STR + something' as before.

> If you try, you'll probably get a star if Marine dies after this map is over and you play another map; Dynamicall y allocated memory limit
> This limitation is somewhat difficult to overcome with simple pointer manipulation.

Try it yourself. I haven't tested it either

Let's start with a plan.
We overwrite Terran\MARINE\TMaDth00.WAV as appropriate, so that custom\00.WAV
맵의 sound\custom\00.WAV에다가 우리가 원하는 웨이브 파일을 넣을겁니다.
TMaDth01.WAV도 마찬가지로 적당히 돌릴거고요.

일단 이러기 위해서는 다음과 같은 트리거를 만들면 될것같습니다.

SetMemory(Terran\MARINE\TMaDth00.WAV 가 저장된 주소 + 0, SetTo, "cust");
SetMemory(Terran\MARINE\TMaDth00.WAV 가 저장된 주소 + 4, SetTo, "om\0");
SetMemory(Terran\MARINE\TMaDth00.WAV 가 저장된 주소 + 8, SetTo, "0.wa");
SetMemory(Terran\MARINE\TMaDth00.WAV 가 저장된 주소 + 8, SetTo, "v\0\0\0");

(대략요. 21B6이 4의 배수가 아니라서 실제로는 약간 조절해야하지만 넘어가요)

이 때 Terran\MARINE\TMaDth00.WAV가 저장되있는 주소는 (sfxdata.tbl 이 로딩된 주소) + 21B6으로 구하면 됩니다.
이 때 (sfxdata.tbl 이 로딩된 주소) 는 005999B0에 저장되어있고요.

005999B0에 저장된 값은 스타가 실행될때마다 달라집니다.
그러므로 어쨌든 우리는 저 값을 맵 만들 때 몰라요.

그래서 우리는

SetMemory(**Terran\MARINE\TMaDth00.WAV 가 저장된 주소** + **0**, SetTo, "cust");

SetMemory(**Terran\MARINE\TMaDth00.WAV 가 저장된 주소** + **8**, SetTo, "v\0\0\0");

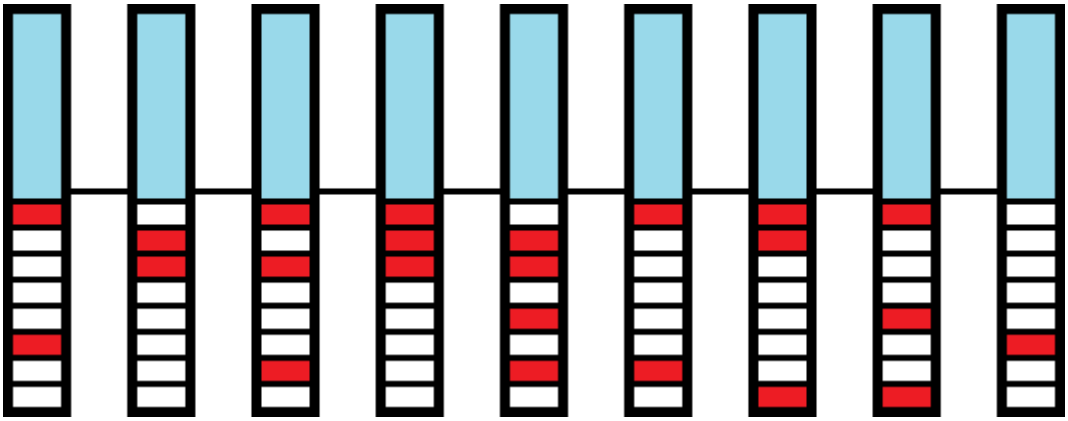
저걸 실시간으로 계산해야합니다.

사실 실시간으로 계산하려고 마음먹으면 쉬워요.
- 005999B0의 값을 4로 나눈 값을 저곳에다가 복사하고
- EPD 공식에 맞도록 적당히 상수 더하고

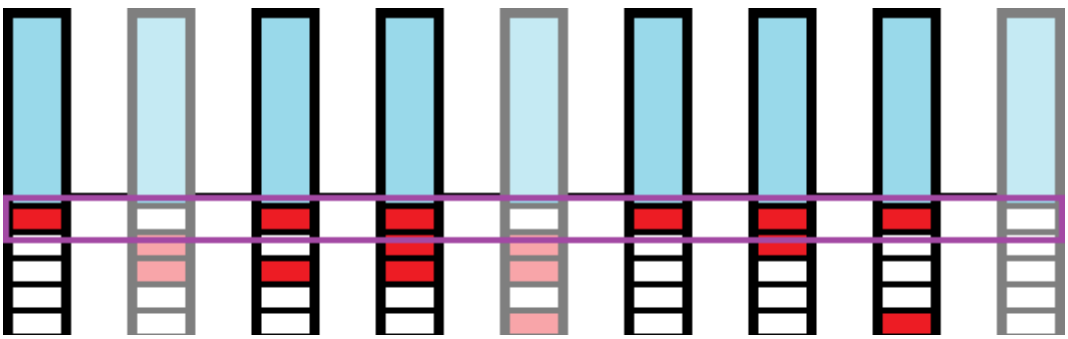
하면 저 값들 자체는 트리거 실행 도중에 어떻게든 만들 수 있는데
그래서 저렇게 만든 SetMemory (실제로는 SetDeaths겠죠) 를 어떻게 하면 실행시킬것인가
이게 관건입니다. 이를 위해서는 스타가 트리거를 어떻게 다루는지에 대해 조금 더 알아볼 필요가 있습니다.

스타가 트리거를 어떻게 다루는지에 대해서 좀 더 알아볼 필요가 있습니다.
스타는 트리거도 이중 연결 리스트로 관리합니다.

맵에 트리거가 다음과 같이 있다고 합시다.

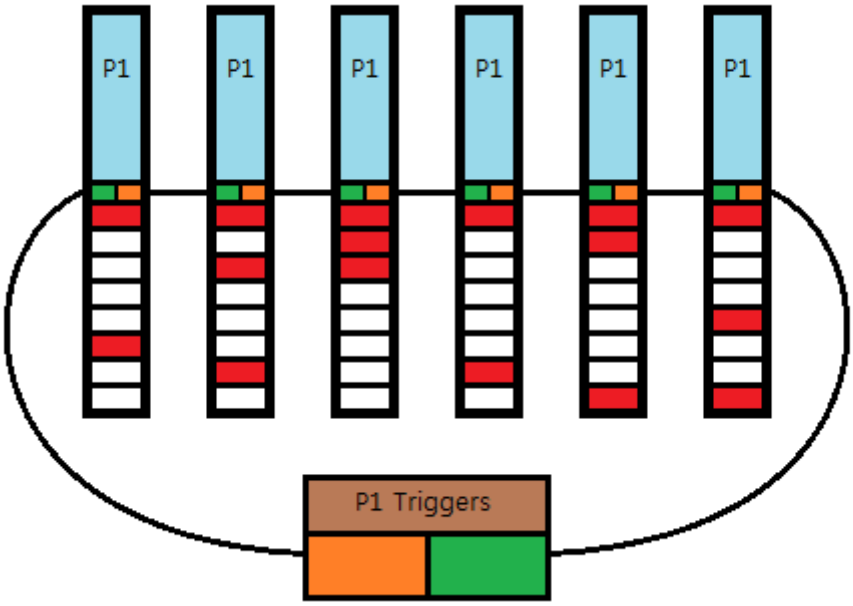


그러면 그 트리거중에서 Player 1이 실행시키는것만 다음과 같이 뽑아내서 따로 **복사**해서
(그러니까 같은 트리거라도 플레이어마다 실제로 메모리에 들어가는건 각각이 원본에 대한 복사본이니까 다 주소가 달라요.)



whyask37's blog

아래 그림처럼 연결 리스트를 만듭니다. (prev, next 등등도 포함)



(next는 주황색, prev는 녹색.)

이렇게 P1에 적용되는 트리거 리스트를 얻어낸 다음에 '**트리거가 적용되는 플레이어**'**는 더이상 의미를 가지지 않게 됩니다.**
(그러니까 연결 리스트 변경해서 P2 Triggers를 P1 Triggers에 연결시켜도 P1 실행될 때 P2것까지 잘 실행되요)

트리거의 prev, next는 사실 트리거 맨 앞부분에 있습니다. 그림이니까 봐줘요
밑에 P1 Triggers라고 적혀있는 것은 트리거는 아니지만 하여튼 이중 연결 리스트에 포함은 되어있습니다.
앞으로 P1의 **PlayerTriggerStruct**라고 부르겠습니다.

prev는 사실 별 역할이 없고, next만 다루면 됩니다.

사실 그림까지 그려가며 설명할만큼 난이도 있는 내용은 아닌데, 말로 하면 너무 괴상한 용어가 많이 나오게 됩니다;

각각 트리거 복사본들은 동적 할당된 메모리에 있기 때문에
실시간으로 계산한 주소들을 트리거의 저 위치에 바로 SetDeaths로 대입할 수는 없습니다.

whyask37's blog

- SetDeaths를 이용해서는 정해진 위치를 정해진 값 만큼을 바꿀 수 있다.
인데,
- 트리거의 주소 등등은 모두 랜덤이다.
입니다.

이것을 해결하기 위해서는
- 트리거를 정해진 위치에 생성시켜놓고 (Location Table에 5100byte만큼 공간이 있네요. 여기 만듭시다)
- 연결 리스트를 조작해서 새로 만든 트리거를 실행되도록 해놓고
- 그 정해진 위치의 트리거를 잘 조작해서 어떻게든 하면 될것같습니다.

예제는 간단해야하니까, 그냥 휴먼 플레이어 하나와 컴퓨터 플레이어 있는 맵에서
(P1이 휴먼, P2가 컴퓨터)
마린의 죽는 소리를 sound\Boom.wav 로 바꿔보도록 하겠습니다. (Boom.wav는 미사일피하기에서 일꾼 죽을 때 폭발음입니다. 맵 파일 안에 있습니다.)

트리거 박스는 2400byte입니다. (TRG 포맷 참조)
맵에서 사용하는 트리거 박스는 2400 + 포인터 2개 = 2408byte입니다.

TriggerBox 구조
==========
TriggerBox* prev;
TriggerBox* next;
TriggerCondition conditions[16];
TriggerAction actions[64];
DWORD internal1;
BYTE effplayer[27];
BYTE executing_action;
총 크기 2408byte

(굵게 표시한 것이 스타에서 더 사용하는 8byte)

그러니까 MRGN에서는 5200byte가 있으니까 일단은 2개의 트리거 박스를 넣을 수 있겠네요.
우리 경우에는 트리거 박스 하나만 넣어도 충분하겠지만요.

MRGN에는 다음과 같은 트리거를 넣을겁니다.
Player : 상관 없음. 어차피 연결 리스트 직접 조작하기 때문에 그냥 아무것도 안넣어도 되요.
Condition : 없음 (Always랑 동일)
Action :
// Terran\MARINE\TMaDth00.WAV<0> 의 앞에부분을 boom.wav<0>로 변경하기

whyask37's blog

```
    // -1450092는 오프셋 0x21b4 에 대한 EPD 플레이어값
  SetDeaths(-1450091, SetTo, 1999531375, 0); // om.w
  SetDeaths(-1450090, SetTo, 30305, 0); //av..

  // Terran\MARINE\TMaDth01.WAV<0> 의 앞에부분을 boom.wav<0>로 변경하기
  SetDeaths(-1450085, SetTo, 1869570560, 0); // sfxdata.tbl + 0x21d0 값을 .boo로 변경
  SetDeaths(-1450084, SetTo, 1635200621, 0); //m.wa
  SetDeaths(-1450083, SetTo, 118, 0); //v...
```

그 다음에 굵게 칠한 부분을 EUD를 통해서 (sfxdata.tbl의 주소) / 4 만큼 더해줄겁니다.
epd의 원리 생각해보시면 이해되실겁니다.

이 트리거를 어떻게 실행시키느냐?가 문제인데

아무래도 어디 있는지도 모르는 트리거의 prev랑 next를 다루는건 좀 아닌것같아요.
PlayerTriggerStruct 의 위치는 정해져있으니까 이걸 다뤄야 하는데
P1 -> P2 순서로 트리거가 실행될 때

P1 트리거가 실행중일 때에는 이미 P1의 PlayerTriggerStruct에 있는 next (그러니까, 최초의 트리거) 내용물이 이미 사용된 뒤입니다.
뭔가 좀 그래요. 다음 트리거루프가 올 때서야 P1의 트리거가 실행되겠군요.

그러니까 우리는 다음과 같은 코드를 상상해봅니다.

1. 원본 맵에서 P1에만 트리거를 넣고 P2에는 트리거를 아얘 넣지 않는다. (이 제한조건은 사실 별 의미가 없습니다. 해결 가능)
2. P1 트리거에서 P2의 PlayerTriggerStruct의 next를 MRGN 영역으로 돌린다.
3. P1 트리거에서 MRGN 영역 트리거를 적당히 조작한다.
4. P1 트리거가 끝난 뒤에 P2의 트리거가 실행되어야 하는데, 이 때 P2의 PlayerTriggerStruct가 사용된다. 2) 과정에서 P2의 PlayerTriggerStruct를 변경해놓았으므로 우리가 원하던 대로 MRGN 영역의 트리거부터 P2를 Current Player로 한 채 트리거가 실행된다. (Current Player 또한 변경이 가능하다)
5. 그렇게 해서 MRGN 영역의 트리거가 실행된다. 3)에서 적당히 이 트리거의 Set Deaths들의 player 값들을 조작해놓았을것이므로 로딩된 sfxdata.tbl 자체의 값이 변경될것이다.
6. MRGN 영역의 추가적으로 P2의 PlayerTriggerStruct를 원래 상태로 복원하고 난 다음에 P2의 트리거 루프를 끝낸다. 이제 마린의 죽는 소리가 boom.wav로 바뀌어 있다.

좀 복잡합니다. 차근차근 읽어보세요. 트리거 프로그래밍 개념에서 제일 어려운 파트입니다.

구조체 하나를 설명을 안했군요.

whyask37's blog

==========

DWORD unknown1; //더 분석해봐야겠습니다만 딱히 쓰이는 용도는 없는것같아요.
TriggerBox* prev_last;
TriggerBox* next_first;
총 크기 12byte

0051A20 오프셋부터 총 8개의 PlayerTriggerStruct가 존재합니다.

트리거의 실행 순서는 다음과 같습니다. **(중요)**

```
player의 트리거를 실행시킬때 :

PlayerTriggerStruct* current_pstruct = (PlayerTriggerStruct*)(0x0051A20 + 12 * player);
 // 먼저 0x0051A20 + 12 * player 번지에 있는 PlayerTriggerStruct를 읽어들입니다.
if((int)current_pstruct->prev_last < 0) return;
 // prev_last < 0이면 ( 또는 0x80000000 이상이면 ) 이 플레이어가 실행시키는 트리거가 없다고 판단하고 바로 끝.

for(TriggerBox* box = current_pstruct->next_first ; (int)box >= 0 ; box = box->next) {
 // 해당 PlayerTriggerStruct의 next_first가 가르키는 트리거부터 시작해서 이중 연결 리스트를 돌아다니면
  ExecuteTrigger(box);
  // 트리거를 실행.
}

// 스타에서 포인터 값이 0x80000000 이상이 될 일은 없습니다. 그래서 포인터가 지니는 주소값이 0x80000000 이상인 것을 '트리거 끝'으로 판단하는겁니다.
```

그래서 코딩해봤습니다.
일단 MRGN 영역에 넣을 코드부터 넣어야죠.
TRG 포맷과 거의 유사하니까 뭐 괜찮을것같아요.
코드는 늘 하던듯이 이렇게.

<pre style="margin-top: 0.5em; margin-bottom: 0.5em; padding: 0px; line-height: 1.1em;"> from trgformat import * triggers = [] triggers . append (Trigger (players = [], conditions = [], actions = [SetMemory (0x21b4 , SetTo , 0x6F620056), SetMemory (0x21b8 , SetTo , 0x772E6D6F), SetMemory (0x21bc , SetTo) 0x00007661), SetMemory(0x21d0, SetTo, 0x6F6F6200), SetMemory(0x21d4, SetTo, 0x61772E6D), SetMemory(0x21d8, SetTo, 0x00000076), SetMemory(0x51A28C + 4, SetTo, 0x51A28C), SetMemory(0x51A28C + 8, SetTo, 0xFFAE5D73),])) WriteTrg('payload.trg' , triggers) </pre>

Explanation :
The 1st to 6th SetMemory statements set the wav file path
7th and 8th memories recover PlayerTriggerStruct.

You can create a TRG file like this and edit only prev and next with hex.

whyask37's blog

If you transform out.trg like this, it becomes **payload.trg in** the attached file . Do it yourself.

Put this as the first 2408 bytes of MRGN
Creates a TRIG paragraph.

The TRIG short initializes the MRGN short and operates the TriggerPlayerStruct of P2. See comments.

```
from trgformat import * triggers = [] # 5999B0의 값 / 4 를 MRGN 단락쪽으로 복사를 해줍니다. # Location table의 시작은 0x0058DC60입니다. # 나누기 4 하는 이유는 EPD니까요. for i in range(29, -1, -1): triggers.append(Trigger( players = [Player1], conditions = [ Deaths(Player1, Exactly, 0, 0), Memory(0x005999B0, AtLeast, 2**(i+2)) ], actions = [ SetMemory(0x005999B0, Subtract, 2**(i+2)), SetMemory(0x0058DC60 + 8 + 320 + 32 * 0 + 16, Add, 2**i), # MRGN 단락에 있는 트리거의 0번째 SetDeaths 액션의 player 부분 SetMemory(0x0058DC60 + 8 + 320 + 32 * 1 + 16, Add, 2**i), # MRGN 단락에 있는 트리거의 1번째 SetDeaths 액션의 player 부분 SetMemory(0x0058DC60 + 8 + 320 + 32 * 2 + 16, Add, 2**i), # MRGN 단락에 있는 트리거의 2번째 SetDeaths 액션의 player 부분 SetMemory(0x0058DC60 + 8 + 320 + 32 * 3 + 16, Add, 2**i), # MRGN 단락에 있는 트리거의 3번째 SetDeaths 액션의 player 부분 SetMemory(0x0058DC60 + 8 + 320 + 32 * 4 + 16, Add, 2**i), # MRGN 단락에 있는 트리거의 4번째 SetDeaths 액션의 player 부분 SetMemory(0x0058DC60 + 8 + 320 + 32 * 5 + 16, Add, 2**i), # MRGN 단락에 있는 트리거의 5번째 SetDeaths 액션의 player 부분 SetDeaths(Player1, Add, 2**i, 1) ] )) for i in range(29, -1, -1): triggers.append(Trigger( players = [Player1], conditions = [ Deaths(Player1, Exactly, 0, 0), Deaths(Player1, AtLeast, 2**i, 1) ], actions = [ SetMemory(0x005999B0, Add, 2**(i+2)), SetDeaths(Player1, Subtract, 2**i, 1) ] )) triggers.append(Trigger( players = [Player1], conditions = [ Deaths(Player1, Exactly, 0, 0) ], actions = [ SetMemory(0x51A28C + 4, SetTo, 0x0058DC60), SetMemory(0x51A28C + 8, SetTo, 0x0058DC60) ] )) WriteTrg('trig.trg', triggers)
```

(trig.trg)
실행시키면 다음과 같습니다.

whyask37's blog

[Lesson Lecture] 8. Trigger Programming - Basics
674

예제맵 올렸고 (trigexec1.scx) TRIG 단락과 MRGN 단락만 보시면 됩니다.
원래 boom.wav가 저렇게 잡음이 많지 않았는데 어디선가 에러가 생겼나봅니다만 관심 없어요. 예제인데 뭐.

사족) 버그 없이 한번에 예제가 성공했어요 만세

사족) 저 부분 메모리는 초기화가 안되기 때문에 다음과 같은 일이 벌어집니다

[Lesson Lecture] 8. Trigger Programming - Basics
443

whyask37's blog

sound\boom.wav가 스타 안에는 없으니까요. 우리가 만든 EUD 유즈맵 안에만 있지.
나름 **민폐왕** 유즈맵입니다.

앞으로 남은 내용은 좀 더 어렵습니다. 이번 강좌는 꼭 확실하게 이해해주세요.

앞으로의 강좌에서는 좀 강좌를 짧고 간결하게 쓰기 위해서 프로그래밍 문법을 조금 쓸겁니다.
프로그래밍을 한번도 안 해본 사람이라도 쉽게 배울 수 있는 문법을 상한선으로 잡고 있습니다. (제 나름대로)
좀 어렵겠지만 일단 파이썬 언어의 기초적 문법만 안다면 강좌가 꽤나 할만해질겁니다.
(C언어의 포인터 문법은 일단은 모르셔도 됩니다)

연습문제

Q1) 고스트가 말하는 소리를 바꿔보자.
Q2) [중급] rez\stat_txt.tbl을 변형해보자. Terran Marine 문자열을 "테란 마린"으로 패치해보자. 제대로 했다면 다른 맵을 할 때에도 Terran Marine이 "테란 마린"으로 그대로 남아있을것이다. (한스타 패치.scx?)
Q3) [중급] TriggerBox의 next를 자기 자신을 가르키도록 해서 트리거 무한루프를 만들자. **(창모드로 실험하세요)**

Q4) [고급] 우리가 만든 유즈맵에서 민폐를 없애라.
(sfxdata.tbl이 할당된 메모리 뒷부분에 이상한 다른 내용이 있고 그 다음에 빈 공간이 있다.
TBL 파일 포맷을 직접 다뤄서 이 빈 공간 안에 새 문자열을 넣은 다음,
테란 마린의 죽는 소리 번호를 이쪽으로 바꿔보라.

Hint) Shadow Protector로 프로텍트된 맵의 STR 단락을 해석해보면 아이디어를 얻을 수 있다.

제 생각에 w런쳐가 저쪽 메모리를 초기화시킬 이유가 없어요. 저거부터 시작해서 메모리 초기화할 곳이 순식간에 스타 메모리 전체로 불어나기 때문에. EUD 쓴 기록을 w런이 유지하는것같진 않고요. 그냥 EUD로 수정한 데 원래대로 돌려주면 안되나...

#IT·컴퓨터

whyask37's blog

payload.trg
trigexec1.scx
trig.trg

1

왜물어

whyask37님의 블로그입니다.

이웃추가

이 블로그 뻘강좌 카테고리 글

[뻘강의] 10. Relocation table

2014. 2. 18.

0

[뻘강의] 9. 트리거 프로그래밍 - 반복문

2014. 2. 12.

0

[뻘강의] 8. 트리거 프로그래밍 - 기초

2014. 2. 11.

1

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

2014. 2. 5.

3

whyask37's blog

1

이 블로그 인기글

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

whyask37's blog

View in PC version

One