mud lecture

[Rude Lecture] 2. Substitution between death, plus

why do you ask
2014. 1. 19. 21:55

There is no ultimate goal for the course.
I'm just going to write it down.

How difficult is the course?

I'll explain how to use EUD in a fun way.
I was just trying to keep the EPD simple.
Let's try some interesting applications.

We're going to do a simple addition.
As an example, simply all death values will be played only in Player 1.
In practice, you'll have to work a little harder.

goal

1. When Marine's death is a and Fabet's death is b, Medic's death can be made a+b.

It's basic stuff, so I tried to skim through it.
Because it is the basis of the very basics of things to play with EUD in the future.
I'm just going to go into detail.

The course is intended for those who have gone from C to arrays. (No pointer needed)
Most implementations will be done in Python, so it doesn't matter.

Since we have to deal with death, we have to use Deaths and Set Deaths conditions/actions. L et's start with what you can do with them.
Deaths allow comparisons with constants.
Set Deaths allows you to add and subtract constants or assign constants

What Deaths can do is very simple.
Is Marine's death greater than 1000?
Is the Medic's death rate less than 165342?
Is Pabet's Deathga 6132785?

I can't do anything but this.

Death Death is annoying.
If you look at the same expression that Marine contains 1
Please read Marine's death is 1.

Now let's get to the problem.

There is some number between 1 and 10000 in the marine.
There is a number between 1 and 10000 in the pabet.
Now we're going to put (Marin's Death) + (Fabet's Death) into the Medic.
What should I do?

Here is the code we will create:
marine, firebat, and medic are the deaths of marines, favets, and medics.
In C, it looks like this:

```
// It's a bit difficult to do + right away.
// Although it is possible to add 1000 to Medic death with the Set Deaths command, there is no function to ad
d as much as 'Marine's Death'.
// In general, implementation of + is done through +=.
// medic = marine + firebat is calculated as follows.
medic = marine //Substitute marine's death into medic's death. (Actually, this is also difficult)
medic += firedat //Add the death of the medic to the death of the medic .
```

Now let's implement the above code as a star trigger.

1. medic = marine
You have to substitute Marine for the Medic's death, as I said before.
Substituting a marine death into a medic's death is outside the scope of Set Deaths.
medic = marine should also be split. star is annoying
Even a simple task like this has to be cut and split and split again.

In fact, it is difficult to enter. First, let's move the Marine's value to the Medic.

Probably

whyask37's blog

```
Subtract 1 from Marine
add 1 to medic
```

If you do, when Marine = 3 at first, Medic and Marine's death values will change as follows, so it will be fine.

Marine	Medic
3	0
2	One
One	2
0	3
End	

As the value of the marine decreases one by one, the medic will increase one by one.

If Marine's death value is 123456789
It seems that the loop above has to run 123456789 times.
If we turn the loop above 123456789
The Medic's death value must also increase by 1 123456789 times.
Marine's death cost 123456789 times must be lowered by 1
The comparison between the marine death value and 0 must also be done 123456789 times.
I don't think this is something.

The above is called the **base 1 system, but** it's not quite. Yes.

So we write in binary.

123456789 = 111010110111100110100010101(2) .

.. It's a bit too big to use as an example. Let's go to 17, 8, 51.
17 = 10001(2)
51 = 110011(2)
8 = 100(2)
If you all fit in the 6th place
110011 = 51
010001 = 17
000100 = 8
This will be

Instead of subtracting one by one, I think you should just pick and subtract one digit.
I think it should be expressed like this in Python.

```
medic = 0
```

whyask37's blog

```
  medic += 32

if marine >= 16: # if more than 10000(2)
  marine -= 16
  medic += 16

if marine >= 8: # if more than 1000(2)
  marine -= 8
  medic += 8

if marine >= 4: # if greater than 100(2)
  marine -= 4
  medic += 4

if marine >= 2: # if more than 10(2)
  marine -= 2
  medic += 2

if marine >= 1: # 1(2) or greater
  marine -= 1
  medic -= 1
```

This will reduce the death value of Marines and Medics as follows.

Marine	Medic	Marine	Medic	Marine	Medic
110011	000000	010001	000000	000100	000000
0 10011	**1** 00000	**0** 10001	**0** 00000	**0** 00100	**0** 00000
0 **0** 0011	1 **1** 0000	0 **0** 0001	0 **1** 0000	0 **0** 0100	0 **0** 0000
00 **0** 011	11 **0** 000	00 **0** 001	01 **0** 000	00 **0** 100	00 **0** 000
000 **0** 11	110 **0** 00	000 **0** 01	010 **0** 00	000 **0** 00	000 **1** 00
0000 **0** 1	1100 **1** 0	0000 **0** 1	0100 **0** 0	0 000 **0** 0	0001 **0** 0
00000 **0**	11001 **1**	00000 **0**	01000 **1**	00000 **0**	00010 **0**

Marine's value has been transferred to the Medic. Something seems to be going well.
To make this even simpler:

```
medic = 0
for i in range(5, -1, -1): # For i from 5 to 0.
  # range(5, -1, -1) means 'starting from 5 and adding -1 to i until -1',
  # Literally, the loop below will run for i = 5, 4, 3, 2, 1, 0.
  # range(5, -1, -2) will run the loop below for i = 5, 3, 1.

  if marine >= 2 ** i: # If marine's death value is greater than 2^i: In python, write a to the power of b as a**b.
    marine -= 2 ** i
    medic += 2 ** i
```

In order to properly explain range, I have to bring the concept of iterator and do it, but I won't. I don't need that.

whyask37's blog

good like this

1. 123456789 = 111010110111100110100010101

To do this, we only need to look up to 2^26, so we only need to run the loop above 26 times.

However, I moved the value of the marine to the medic, but the original value of the marine is lo st. Marine's value is now 0.
In this case, not medic = marine, but medic = marine; marine = 0; it becomes this

Actually, it's not a big deal. It's just a matter of introducing a temporary variable.

```
medic = 0
ghost = 0 # temporary variable
# Change the value of marine to medic, ghost
for i in range(5, -1, -1):
  if marine >= 2 ** i:
    marine -= 2 ** i
    medic += 2 ** i
    ghost += 2 ** i

# Move the ghost back to marine.
for i in range(5, -1, -1):
  if ghost >= 2 ** i:
    ghost -= 2 ** i
    marine += 2 ** i

# The ghost becomes 0 again, and the original value is returned to marine.
```

That's it.

Come to think of it, if there was no medic = 0 in the first place
It seems that the death value of the marine was added to the value the medic had at first.
Now, the value of medic contains the value of marine.
without resetting medic back to zero
If you 'substitute' the value of firebat into medic as above (actually, it is added)
marine + firebat 계산이 될것같아요. 코드는 다음과 같습니다.

```
medic = 0
ghost = 0 # 임시 변수
# marine의 값을 medic, ghost로 옮긴다
for i in range(5, -1, -1):
  if marine >= 2 ** i:
    marine -= 2 ** i
    medic += 2 ** i
    ghost += 2 ** i
```

whyask37's blog

```
for i in range(5, -1, -1):
  if ghost >= 2 ** i:
    ghost -= 2 ** i
    marine += 2 ** i

# ghost의 값은 다시 marine으로 옮겼으니까 지금 ghost = 0이다.
# 따라서 다시 ghost = 0을 해줄 필요는 없다.

# firebat의 값을 medic, ghost에 더한다.
for i in range(5, -1, -1):
  if firebat > 2 ** i:
    firebat -= 2 ** i
    medic += 2 ** i
    ghost += 2 ** i

# ghost를 다시 firebat으로 옮긴다.
for i in range(6):
  if ghost >= 2 ** i:
    ghost -= 2 ** i
    firebat += 2 ** i
```

이러면 되요. 이제 이것을 스타 트리거로 구현해봅시다.

이번 강좌 구현은 php으로 하겠습니다.
php 예제 : http://cafe.naver.com/edac/26416

뭐 여러 언어를 쓰고 있기는 한데
다 C로부터 파생되었기 때문에
프로그래밍 언어들이 모두 다 비슷비슷하게 생겨먹었습니다.
뭐 괜찮겠지요.
다음 강좌부터는 C + python으로만 진행합니다.

우리 원래 목표가 1~10000까지 marine, firebat이 있는거였죠?
10000 = 10011100010000 니까 잘은 모르겠고
2^13 까지만 확인하면 될것같습니다. 코드는 아래와 같습니다.

코드 : http://codepad.org/sdjRkjTo

주의) 맵에 보면 Switch 1을 맨 처음에 세팅하고 중간에 계산을 하고 Switch 1을 해제하는 식으로 되어 있을겁니다.
위에 계산이 제멋대로 아무때나 일어나도록 놔두면 다음과 같은 일이 벌어집니다.

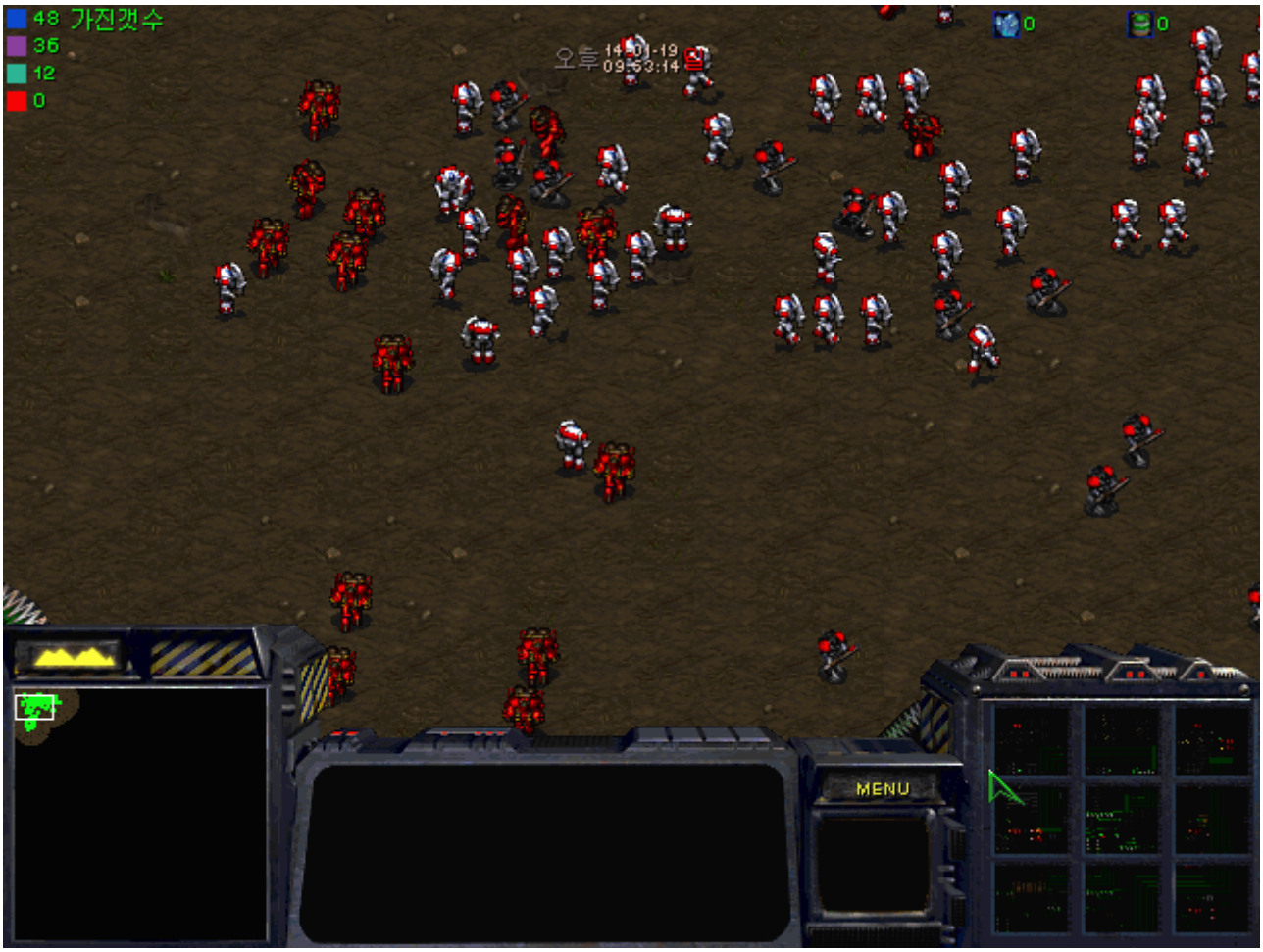
마린의 데스가 12였을 때
12 -> 4 (-8) -> 0 ->(-4) -> 메딕에 8+4 = 12 고스트에 의해 12가 마린에 다시 채워짐

-> 메딕의 값이 15가 됩니다.

-> **마린과 메딕의 데스를 내가 원하는 때에 계산해서 꺼내다 쓰기 어렵게 됩니다.**

맵을 실행하면 다음과 같이 뜰겁니다. 성공.
Leader Board에
파랑이 메딕 갯수
보라가 파벳 갯수
연두가 마린 갯수

마지막에 메딕, 마린, 파벳 데스만큼 각각 메딕, 마린, 파벳 생성하도록 했습니다.



맵 여시면 첫번째 트리거에 마린과 파벳 데스 설정하는게 있을텐데,
0~10000 사이에서 아무 값이나 골라서 돌려보세요.

연습문제) 다음을 스타에서 구현하라.
1) medic = marine * 173
2) medic = marine + firebat + ghost
3) [HARD] 1번 강좌에서 나온 EPD를 엮어서 0x0058D740 번지의 값을 marine만큼 더해보라.

whyask37's blog

정리합시다.
1장과 연결지어보면 Deaths랑 Set Deaths를 통해서 다음을 할 수 있음을 알았네요.

1. 임의의 정해진 메모리 주소를 읽을 수 있다. (EPD 사용)
2. 메모리 주소에 들어있는 값 + 상수 를 계산할 수 있다. (EPD 사용)
3. 메모리 주소에 들어있는 값 끼리의 덧셈, 뺄셈이 된다. (뺄셈은 덧셈과 원리 똑같게)
4. 메모리 주소에 들어있는 값 * 상수 를 계산할 수 있다. (*는 곱하기를 뜻합니다)

다음 강좌에서는 포인터에 대해 다뤄보겠습니다.
스타는 C++로 짜여졌고
C++에서 중요한 부분은 전부 포인터로 짜여져 있습니다.
스타를 갖고 놀라면 포인터를 써야겠죠.

이번 뻘짓의 난이도는 쓸데없이 높네요.
원래 엄청 쉬운건데 왜이리 설명하기 힘들지;

#IT·컴퓨터

첨부파일

tttt.scm

1

왜물어

whyask37님의 블로그입니다.

이웃추가

이 블로그 뻘강좌 카테고리 글

[뻘강의] 5. 포인터 예제 - 유닛 제한 줄이기

2014. 1. 23.

0

[뻘강의] 4. TRG 파일 포맷

2014. 1. 21.

3

[뻘강의] 3. 포인터

2014. 1. 21.

0

[뻘강의] 2. 데스 사이의 대입, 더하기

2014. 1. 19.

1

[뻘강의] 1. EPD 간단요약

2014. 1. 19.

0

이 블로그 인기글

whyask37's blog

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] 5. 포인터 예제 - 유닛 제한 줄이기

2014. 1. 23.

0

back to top

View in PC version

One