

## whyask37's blog

### Project SMD

## 3. Terrain Output



why do you ask  
2013. 9. 10. 17:41

add neighbor

This document is a course-style log of developing the StarCraft editor .  
Since I am also in the position of learning, what is the point of calling this a 'course';  
This is a project to make a map editor, realizing that making a map editor involves understanding the overall structure of the game.

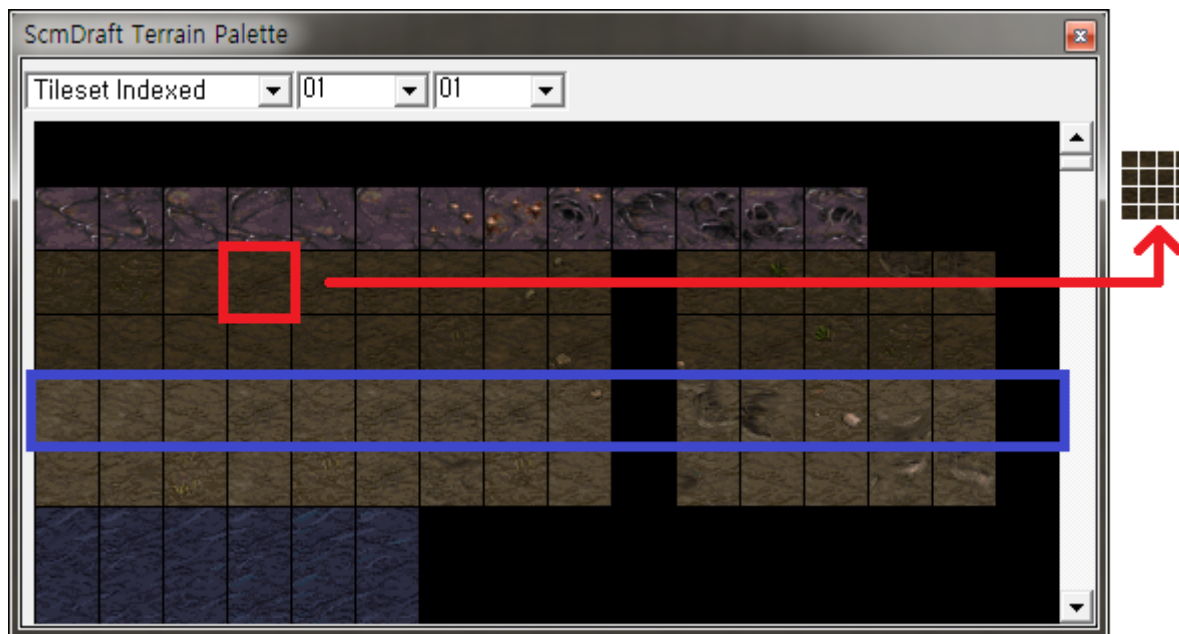
<http://blog.naver.com/whyask37>

I think you should start by printing out the terrain first.

Main source: [http://www.modcrafters.com/wiki/index.php?title=Terrain\\_Format](http://www.modcrafters.com/wiki/index.php?title=Terrain_Format)

When I read the article in the link above, it says Minitile Minitile Megatile Megatile, but I was a little angry that there was no explanation about what Minitile was anywhere.

1. Each tile is 32\*32 size. (pixels) Each of these tiles is called a Megatile.
2. Each tile consists of 8 tiles of size 8 \* 8. Each of the 16 tiles that make up a Megatile is called a Minitile.



3. Each tile on the map has a 2 byte number, so let's go to the details below.

4. In the picture above, the blue one (one line) is called a group, and there are 16 megatiles in a

## whyask37's blog

like. Now let's see how to read these megatiles and minitiles.

-----

For each tileset you will need five files with extensions CV5, VF4, VR4, VX4 and WPE. each rol  
e

1. CV5 : Define the above-mentioned Groups. Each group has 16 megatiles, right? There are 4096 groups.
2. VX4: Defines how to render the 16 mini-tiles of the megatiles.
3. VF4: Stores the characteristics of each mini-tile (High, Medium or Low, whether ground units can pass through it, whether it blocks the view, etc.).  
(Since these attributes are determined in units of minitiles, sometimes there are tiles that only a part of the se units can go to.



4. VR4: Designate the graphics of the mini tiles. Did you know that a star has 256 colors? For each Minitile, we set the tile's graphics to  $8 \times 8 = 256$ byte. (You can think of it as similar to 8-bit BMP)
5. WPE: This is a palette of 256 colors required in VR4. RGBP RGBP RGBP ... They are stored in order, so a total of  $4 \times 256 = 1024$  bytes. (P is the padding that tries to break the color by 4 bytes. There is no such thing as an alpha value)

**맵에 지형 정보 (MTXM) -> 타일셋(CV5) -> Megatile(VX4) -> Minitile(VF4)와 그 그래픽(VR4) -> 팔레트(WPE)**

이런거죠. 이런 식으로 렌더링하면 되요. MTXM이 뭔지는 밑에 곧 나올거고요.

그런데 MTXM이든 CV5든 VX4든 VF4든 VR4든 그게 뭔지 스타크래프트 폴더에는 아무리 찾아봐도 없어요. 어디 있을까요?

-----

스타크래프트 맵 파일은 MPQ 파일입니다. MPQ 파일은 블리자드 게임이 쓰는 .zip같은거라 보시면 편해요. MPQ 포맷은 좀 무지 복잡해요. 지금은 안다룰꺼고 나중에 다루죠. 일단 Ladik's MPQ Editor를 써보도록 합시다. (궁금한 사람은 글 맨 밑에 링크 참조)

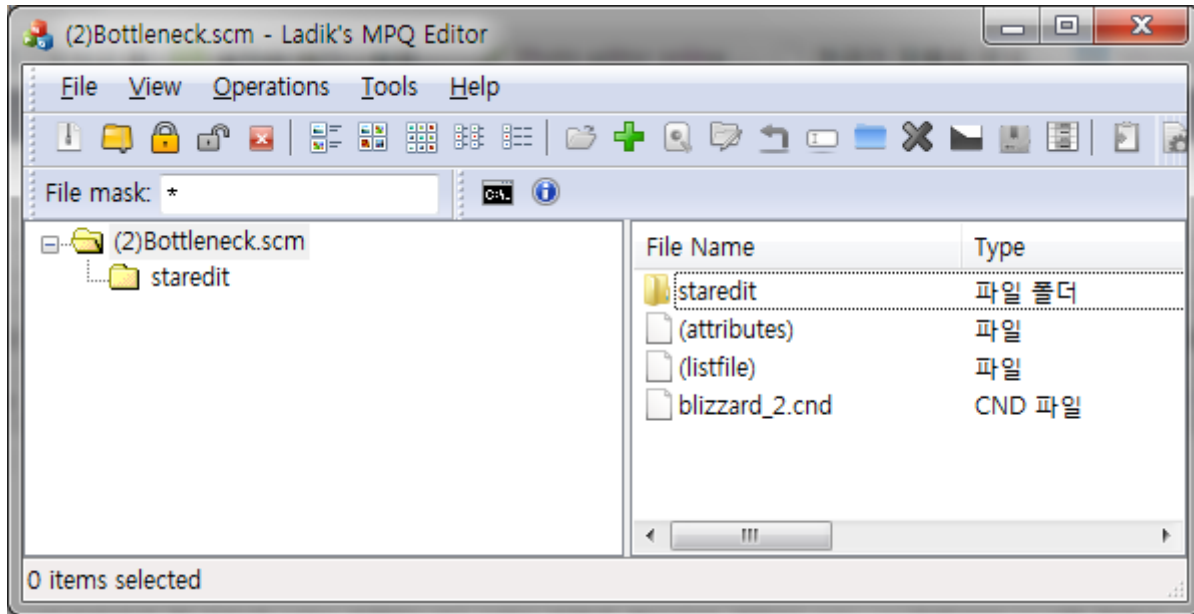
Ladik's MPQ Editor : <http://www.zezula.net/download/mpqediten32.zip>

그 외에도 이 파일이 필요할겁니다. 이게 왜 필요한지는 적당한 시점에 설명하도록 하죠. 리스트파일이  
라 합니다.

Listfiles : <http://www.zezula.net/download/listfiles.zip> - 미리 압축풀어둡시다.

## whyask37's blog

합니다.



(이 글이 언프로텍터를 만드는 강좌는 아니므로 일단은 MPQ 프로젝트 뚫는건 생략합니다. MPQ플렉 걸린맵은 안열려요.)

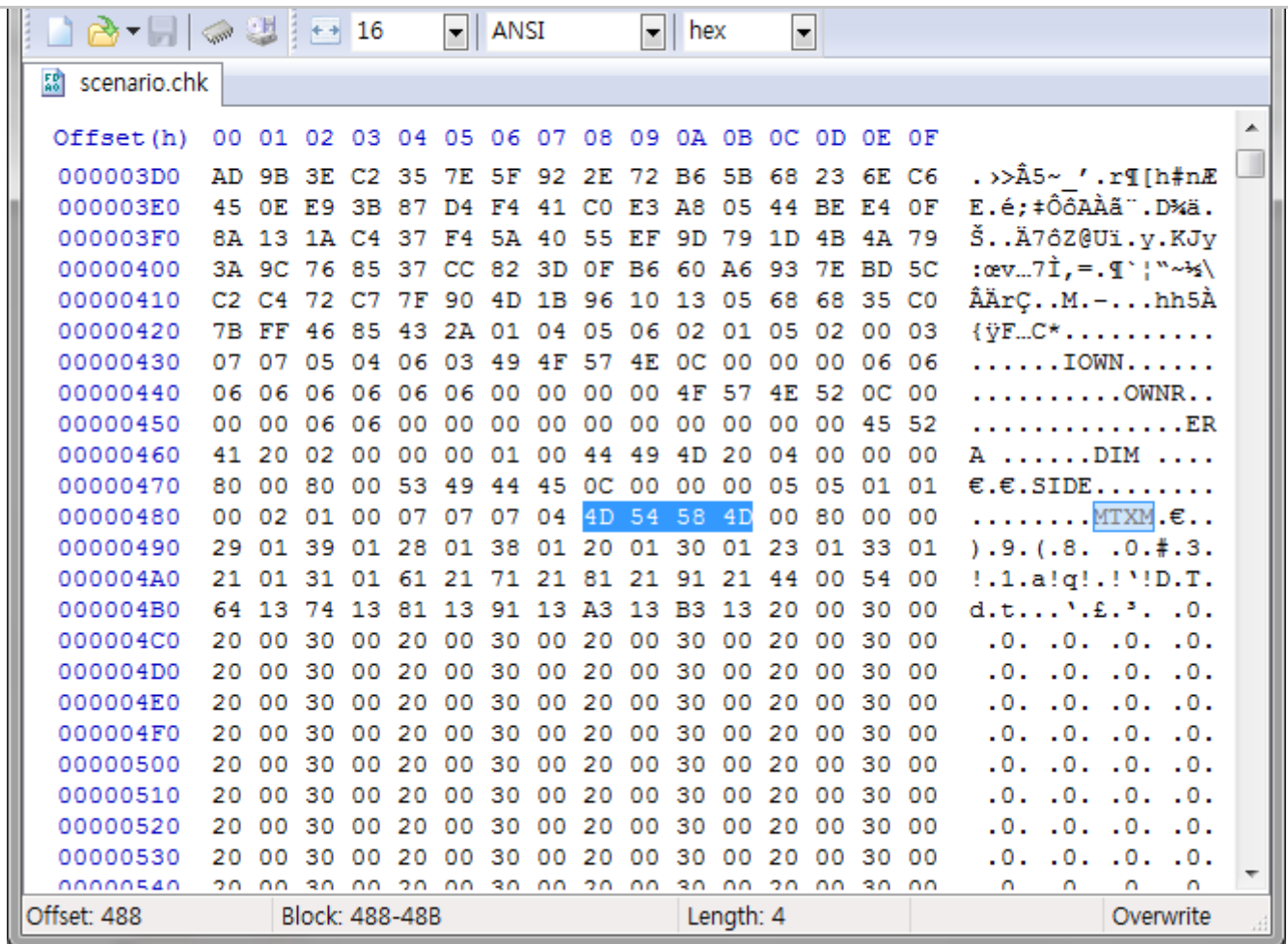
staredit로 들어가서 scenario.chk를 드래그 & 바탕화면에 드롭 해서 staredit.chk를 추출해줍니다. staredit 폴더가 생기고 거기에 scenario.chk가 생길겁니다.

네 좋아요. 이렇게 chk를 추출했습니다.

이제 chk 파일을 열어봐야죠. 헥스에디터가 필요합니다. 여러가지가 있지만 저는 HxD를 씁니다.  
<http://mh-nexus.de/en/hxd/> 여기서 적당히 찾아서 다운로드받으시면 됩니다.

그렇게 해서 HxD를 켜고 이걸 열면 뭔가가 뜹니다. scenario.chk를 열어줍니다.  
 이제 이 창에다가 Ctrl + F로 MTXM을 찾아줍니다. 문자열로요.

## whyask37's blog



뭔가가 나왔어요. 그리고 MTXM 뒤에는 뭔가 규칙적인 문자열이 반복되고있군요. 이제 MTXM 바로 뒤에 4byte인 00 80 00 00을 리틀 엔디안으로 읽어줍니다. 그러니까 0x00008000을요. 32768라고 나오죠? (프로그래머라면 다들 이해하실겁니다)

(사족 :  $32768 = 2 * 128 * 128 = (\text{맵 가로}) * (\text{맵 세로}) * (\text{short의 크기})$ )

#### [Little endian]

(뭔가 설명을 안해두면 안될것같아요)

인텔에서 CPU를 만들때 회로설계하기 어렵다고 little endian이라는걸 만들었어요.

예를 들어서 MTXM은 위에 화면에 보이듯이 4D 54 58 4D죠?

그 다음 len (section의 길이)는 00 80 00 00 입니다.

00 80 00 00 -> 00 00 80 00 -> 0x00008000 을 16진수로 읽으면 됩니다. 32768이죠? byte 단위로 뒤집어요.

ex) 12 34 56 78 -> 78 56 34 12 -> 0x78563412 = 2018915346

scenario.chk 포맷의 각 단락들은 다음과 같은 포맷으로 되어있습니다.

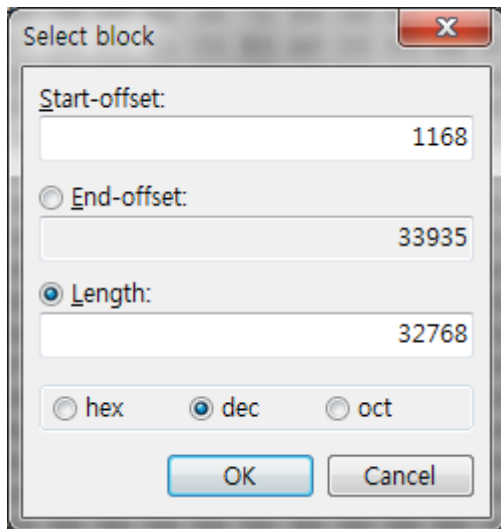
```
struct Chunk {
    uchar type[4];
    ulong len;
```

## whyask37's blog

(uchar, ulong, ushort는 각각 unsigned char, unsigned short, unsigned long의 줄임말이다. 합시다. [len]같은것도 된다 합시다)

우리가 찾은 MTXM은 type에 해당하고, 32768은 len에 해당합니다.

이제 data를 읽어봐야겠습니다. len 뒤로 32768byte를 적당히 끊어서 찾아봅시다. Ctrl + E를 쓰면 편합니다.



dec(10진법)을 선택했고, length를 선택해줬습니다.

네 이렇게 선택했으면 복사해서 새 파일 열고 붙여넣어줍시다. 그리고 MTXM이라는 파일명으로 적당한 데에 저장해줍시다.

첨부파일같이 나왔어요. 여러분도 같게 나왔는지요?

-----

좋아요. 이제 이 MTXM의 포맷을 파헤쳐봅시다.

출처 : <http://www.stareidit.net/starcraft/CHK#MTXM>

MTXM은 2byte들이 가로 \* 세로 형식으로 되어있습니다. ushort의 2차원 배열이라고 생각하시면 되요.

## whyask37's blog

---

다음과 같이 프로그램을 짜면 됩니다.

Colored By **Color Scripter™**

```
1 #include <stdio.h>
2
3 //동적 2d array를 쉽게 다루기 위한 클래스.
4 template <typename T>
5 class array2d {
6 public:
7     array2d(T* ptr, int pitch) : _ptr(ptr), _pitch(pitch) {}
8     T* operator[](int idx) { return _ptr + (_pitch * idx); }
9
10 private:
11     T* const _ptr;
12     const int _pitch;
13 };
14
15
16
17 typedef unsigned int uint32;
18 typedef unsigned short uint16;
19 typedef unsigned char uchar8;
20
21 int main() {
22     FILE *fp;
23     const int w = 128;
24     const int h = 128;
25     int x, y;
26
27     //파일 열기
28     fp = fopen("MTXM", "rb");
29     if(fp == NULL) return -1;
30
31     //동적 메모리 할당, array2d wrapper에 포인터를 먹임.
32     uint16 *mtxmdata = new uint16[w * h];
33     array2d<uint16> mtxm(mtxmdata, w);
34
35     //파일에서 읽어온다
36     fread(mtxmdata, 2, w * h, fp);
37     fclose(fp);
38
39     //파일을 쓰자.
40     fp = fopen("MTXM.txt", "w");
41     for(y = 0 ; y < h ; y++) {
42         for(x = 0 ; x < w ; x++) {
```

## whyask37's blog

```

48     }
49     fclose(fp);
50
51     delete[] mtxmdata;
52     return 0;
53 }
```

일단 간단하게 만들었습니다. 코드를 보면 대강 MTXM의 2byte에서 group와 index(한 그룹 안에서 몇 번째 타일인가)를 얻어내는 방법을 아실겁니다. 이제 이 정보와 타일셋 파일을 통해서 렌더링을 해보도록 하죠.

-----  
**강좌가 왜이리 생소한 단어가 많냐고요? 강좌가 아니라 개발일기니까요. 봐주세요.**  
**최대한 강좌처럼 쓰고 있는데 어렵네요 ㄹ**  
 -----







SCMDraft2로 확인해본 결과, Bottleneck 맵의 타일셋은 Space Platform입니다. CV5 등을 추출해보죠.

MPQ 에디터로 스타크래프트 폴더의 StarDat.mpq를 열어봅시다. 이 때 무슨 이상한 창이 하나 뜰텐데,

잘 찾아보면 Additional Listfile라고 오른쪽 위 썸에 있습니다. 여기에서 Listfiles (글 위에 다운받으라고 한거) 압축폰거 안의 ListFile.txt를 지정해줍니다. 그리고 OK를 누릅니다.

(지정을 안해주면 FileXXXXXXXXX 등으로 막 나옵니다. 왜 이렇게 나오는지 설명이 많이 필요하므로 일단 생략)

terrain/ 에서 Platform.cv5 vf4 vr4 vx4 wpe 를 적당히 추출해줍니다.

 platform.cv5	2013-09-10 오후...	CV5 파일	77KB
 platform.vf4	2013-09-10 오후...	VF4 파일	96KB
 platform.vr4	2013-09-10 오후...	VR4 파일	1,175KB
 platform.vx4	2013-09-10 오후...	VX4 파일	96KB
 platform.wpe	2013-09-10 오후...	WPE 파일	1KB
 MTXM	2013-09-10 오후...	파일	32KB

## whyask37's blog

출처 : [http://www.modcrafters.com/wiki/index.php?title=Terrain\\_Format](http://www.modcrafters.com/wiki/index.php?title=Terrain_Format)

Offset	Type	Name	Description
+0x00	SHORT	<i>Index?</i>	Unknwon
+0x02	SHORT	<i>Flags</i>	Various flags (usually broken up into nibbles) 0x0001 = Edge? 0x0004 = Cliff? 0x0040 = Creep 0x0080 = Unbuildable 0x0n00 = Deprecated ground height? 0x8000 = Buildable for Start Location and Beacons
+0x04	SHORT	<i>Left Edge</i>	Not fully understood. Value matches Right Edge on connecting tiles.
+0x06	SHORT	<i>Top Edge</i>	Not fully understood. Value matches Bottom Edge on connecting tiles.
+0x08	SHORT	<i>Right Edge</i>	Not fully understood. Value matches Left Edge on connecting tiles.
+0x0A	SHORT	<i>Bottom Edge</i>	Not fully understood. Value matches Top Edge on connecting tiles.
+0x0C	SHORT	<i>Unknown/Unused</i>	Unknown/Unused
+0x0E	SHORT	<i>Edge Up</i>	Not fully understood. Edge piece has rows above it? 1 = Basic edge piece. 2 = Right edge piece. 3 = Left edge piece.
+0x10	SHORT	<i>Unknown/Unused</i>	Unknown/Unused
+0x12	SHORT	<i>Edge Down</i>	Not fully understood. Edge piece has rows below it? 1 = Basic edge piece. 2 = Right edge piece. 3 = Left edge piece.
+0x14	16xSHORT	<i>MegaTile References</i>	16 references to VF4 and VX4.

대충 이러이러하다고 나와있습니다. 영어가 원지는 잘 모르겠습니다만 우리는 그냥 VX4가 목적이므로 minitile만 얻으면 됩니다.

대충 생략하고 megatile부터 읽어봅시다. *MegaTile References* 를 참조해봅시다.

그리고 이렇게 Megatile에서 Minitiles들의 그래픽을 얻어보기 위해서 VX4를

Offset	Type	Name	Description
+0x00	16xSHORT	<i>Images</i>	The first bit (bit0) is the horizontal flip flag, the remaining 15 bits (bit1-15) are the VR4 index.

좋아요. 이제 VR4에서 필요한거를 읽어봅시다.

Offset	Type	Name	Description
+0x00	64xBYTE	<i>Bitmap</i>	Each byte is an index of the WPE. Pixels are drawn left to right (unless flipped in VX4), top to bottom in an 8x8 square.



## whyask37's blog

Offset	Type	Name	Description
+0x00	BYTE	Red	Red component
+0x01	BYTE	Green	Green component
+0x02	BYTE	Blue	Blue component
+0x03	BYTE	Pad	Padding

이제 이거를 프로그램으로 구현해봅시다. 일단 Color cycling은 무시하고 해보죠. 그냥 빠르게 만들기 위해서 복잡하게 만들지는 않겠습니다. (color cycling : 팔레트를 바꿔가지고 물 지형 등의 색깔이 바뀌게 하는 것. 스타에서 물이 반짝반짝거리는 이유)

PNG 렌더링은 LodePNG를 사용했습니다. <http://lodev.org/lodepng/> lodepng.cpp와 lodepng.h는 저기서 받아주세요.

좋아요. 로직은 다음과 같습니다.

각 mtxm의 short마다  
group와 index로 분리한다.  
cv5->group[group]의 index번째가 우리가 원하는 Megatile의 인덱스다. 이것을 megatile이라 하자.

megatile의 각 minitile마다  
VR4에서 색상을 얻고, 이 색상값을 WPE를 통해 RGB값으로 변환시켜서 이미지에 출력한다.

Colored By **Color Scriptor™**

```

1  #include <stdio.h>
2  #include "lodepng.h"
3
4  //동적 2d array를 쉽게 다루기 위한 클래스.
5  template <typename T>
6  class array2d {
7  public:
8      array2d(T* ptr, int pitch) : _ptr(ptr), _pitch(pitch) {}
9      T* operator[] (int idx) { return _ptr + (_pitch * idx); }
10
11  private:
12      T* const _ptr;
13      const int _pitch;
14  };
15
16
17  typedef unsigned int uint32;
18  typedef unsigned short uint16;
19  typedef unsigned char uint8;
20

```

## whyask37's blog

---

```
25 struct CV5Data {
26     uint8 Something[20];
27     uint16 MegatileIndex[16];
28 } group[4096];
29 }CV5;
30
31 typedef struct {
32     struct VX4Data {
33         uint16 VR4Index[16];
34     } data[65536];
35 }VX4;
36
37 typedef struct {
38     struct VR4Data {
39         uint8 color[64];
40     } image[32768];
41 }VR4;
42
43 typedef struct {
44     struct WPEData {
45         uint8 r;
46         uint8 g;
47         uint8 b;
48         uint8 padding;
49     } data[256];
50 }WPE;
51
52 typedef struct {
53     uint8 r;
54     uint8 g;
55     uint8 b;
56     uint8 a;
57 }RGBAbyte;
58
59 #include <packoff.h>
60
61
62 int main() {
63     FILE *fp;
64     FILE *CV5fp, *VX4fp, *VR4fp, *WPEfp;
65     CV5 *cv5;
66     VX4 *vx4;
67     VR4 *vr4;
68     WPE *wpe;
69
70     const int w = 128;
71     const int h = 128;
```

## whyask37's blog

```

77 VR4fp = fopen("Platform.VR4", "rb");
78 WPEfp = fopen("Platform.WPE", "rb");
79 if(CV5fp == NULL || VX4fp == NULL || VR4fp == NULL || WPEfp == NULL) {
80     fcloseall();
81     return 0;
82 }
83
84 cv5 = new CV5;
85 vx4 = new VX4;
86 vr4 = new VR4;
87 wpe = new WPE;
88 memset(cv5, 0, sizeof(CV5));
89 memset(vx4, 0, sizeof(VX4));
90 memset(vr4, 0, sizeof(VR4));
91 memset(wpe, 0, sizeof(WPE));
92
93 fread(cv5, 1, sizeof(CV5), CV5fp);
94 fread(vx4, 1, sizeof(VX4), VX4fp);
95 fread(vr4, 1, sizeof(VR4), VR4fp);
96 fread(wpe, 1, sizeof(WPE), WPEfp);
97
98 fcloseall();
99
100
101 //MTXM 읽기
102 fp = fopen("MTXM", "rb");
103 if(fp == NULL) return -1;
104
105 uint16 *mtxmdata = new uint16[w * h];
106 fread(mtxmdata, 2, w * h, fp);
107 fclose(fp);
108
109 array2d<uint16> mtxm(mtxmdata, w);
110
111
112 //이미지 파일의 데이터를 준비.
113 const int imgw = 32 * w, imgh = 32 * h;
114 RGBABYTE *imgdata = new RGBABYTE[imgw * imgh];
115 array2d<RGBABYTE> img(imgdata, imgw);
116
117 //이미지 기록
118 for(y = 0 ; y < h ; y++) {
119     for(x = 0 ; x < w ; x++) {
120         int group = mtxm[y][x] >> 4;
121         int index = mtxm[y][x] & 0xf;
122         //megatile을 얻고 안의 4*4개의 minitile들을 얻어서
123         int megatile = cv5->group[group].MegatileIndex[index];

```

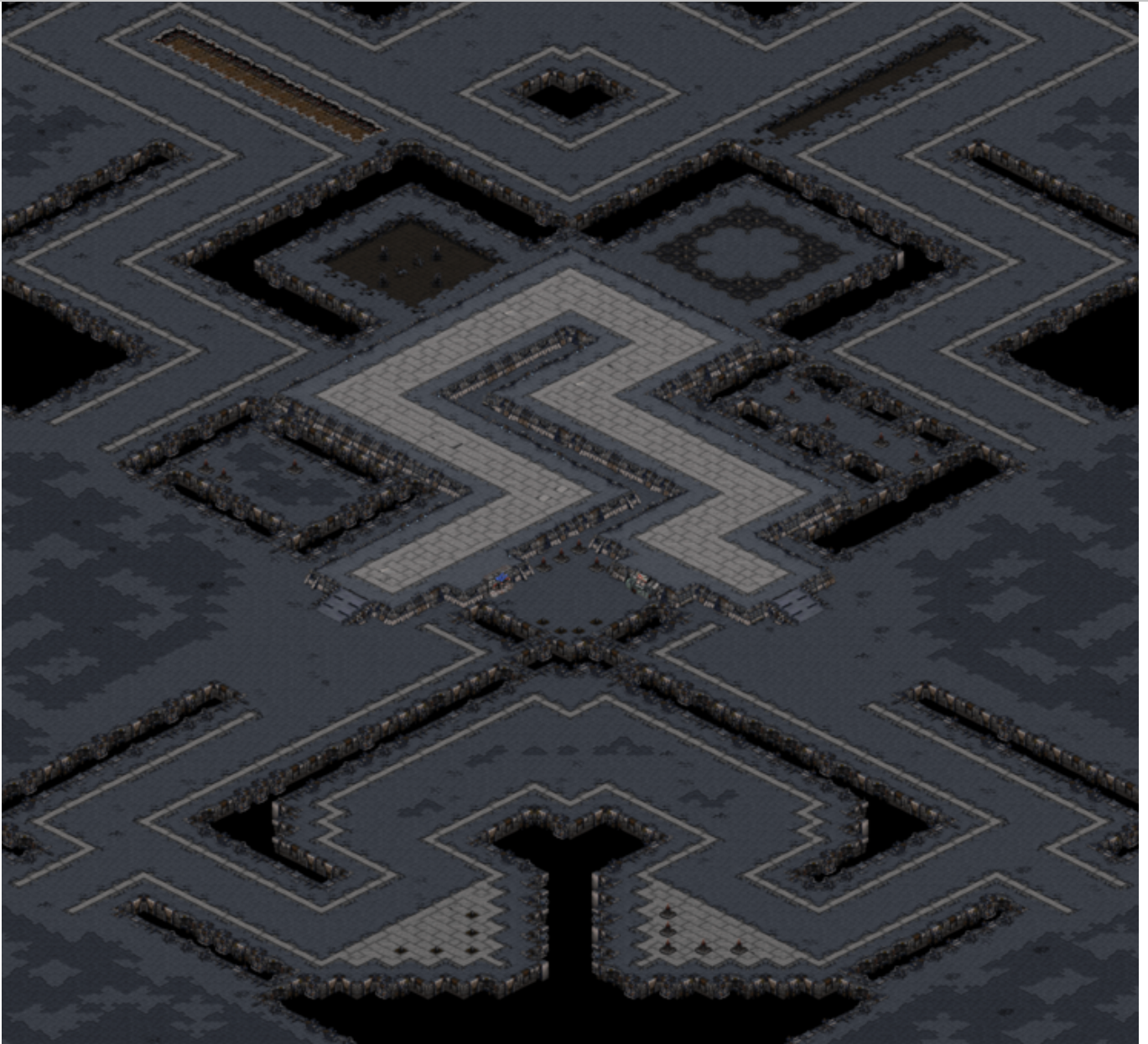
## whyask37's blog

```

129
130     const VR4::VR4Data &vr4data = vr4->image[initileindex];
131
132     const int draw_offsetx = x * 32 + subx * 8;
133     const int draw_offsety = y * 32 + suby * 8;
134
135     //미니타일의 8*8을 렌더링.
136     for(j = 0 ; j < 8 ; j++) {
137         for(i = 0 ; i < 8 ; i++) {
138             int drawx = draw_offsetx + (flipped ? 7-i : i);
139             int drawy = draw_offsety + j;
140             const WPE::WPEData &wpedata = wpe->data[vr4data.color[j*8 + i]];
141             RGBAbyte color = {wpedata.r, wpedata.g, wpedata.b, 255};
142             img[drawy][drawx] = color; //setpixel(drawx, drawy, color); 와 비슷한 역할.
143         }
144     }
145 }
146 }
147 }
148 }
149
150 //PNG 파일로 출력!
151 lodepng_encode32_file("Rendered.png", (unsigned char*)imgdata, imgw, imgh);
152
153 delete[] imgdata;
154 delete[] mtxmdata;
155
156 delete cv5;
157 delete vx4;
158 delete vr4;
159 delete wpe;
160
161 return 0;
162 }
```

결과 :

## whyask37's blog



완벽하게 렌더링됐군요.

음 다음번에는 한번 유닛을 출력해봐야겠군요.

유닛 출력은 지형 출력의 10배정도 어려울것같네요. 스타포지같은게 아니라 SCMDraft2같은 살아숨쉬는(?) 유닛이 목표라서요.

일단 유닛 출력은 5번째정도로 구현해보도록 하겠습니다. 밀리맵을 전부 출력할 수 있으면 로케이션 출력은 껌이니까.

## whyask37's blog

다음 강좌는 chk 파일 열람으로 배우겠습니다. 사실 이게 엄청나게 중요하기 때문에...

맵에디터는 chk 파일을 파싱해서 출력해주고 수정도 조금 지원해주는 프로그램에 지나지 않아요.

이번 일기는 여기까지입니다. 감사합니다.

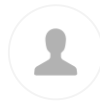
-----

MPQ 포맷 : [http://web.archive.org/web/20120222093346/http://wiki.devklog.net/index.php?title=The\\_MoPaQ\\_Archive\\_Format](http://web.archive.org/web/20120222093346/http://wiki.devklog.net/index.php?title=The_MoPaQ_Archive_Format)

### 첨부파일

MTX.

3



### 왜물어

whyask37님의 블로그입니다.

이웃추가

### this blog **Project SMD** Category article

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

2013. 9. 11.

1

#### 4. scenario.chk

2013. 9. 10.

0

3

## whyask37's blog

---

### 2. ChkDraft

2013. 9. 10.

0

---

### 1. 구상

2013. 9. 10.

0

---



---

## 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

---

### [별강의] 13. 트리거 프로그래밍 - TRIG-MRGN 루프

2014. 2. 24.

0

---

### 4. scenario.chk

2013. 9. 10.

0

---

### [별강의] 2. 데스 사이의 대입, 더하기

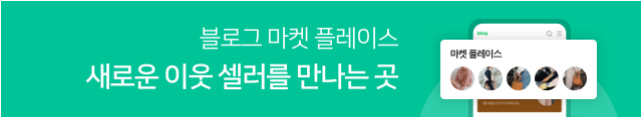
2014. 1. 19.

1

---

whyask37's blog

back to top



View in PC version