

whyask37's blog

Project SMD

7. Let's print the terrain again.



why do you ask
2013. 9. 19. 3:48

add neighbor

This document is a course-style log of developing a StarCraft viewer .

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.

I started with ..., but I don't think there's any need to even make a map editor, and I'll have to use the unprotected technology someday anyway,

I'm just trying to make an upgraded version of TriggerViewer2. Then the course will start again.

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

This time we will print the terrain. yet again.

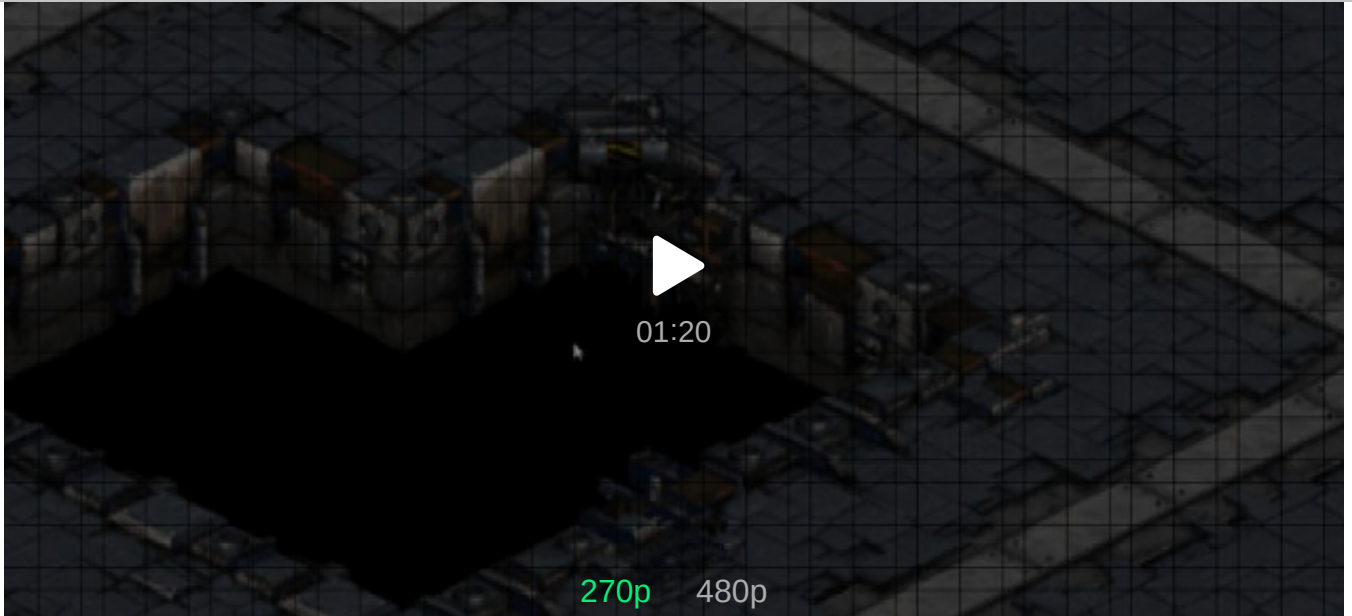
Yes, it's that simple, right? Except for a few problems.

quadruped)

quadruped)

The video below is an example I made while writing the 6th course. I closed it because there was a bug.

whyask37's blog



The bug might be something like this:

Bug List

1. Very slow. explode inside The rendering speed itself, not the scroll speed, is so slow that it can't even maintain 30fps.
2. There is a crack in the middle of the map and it explodes. (This is a problem with SDL_RenderCopy. It is said that you can use SDL_RenderCopyEx, but it is annoying)

예제는 간단하게 모든 미니타일들을 다 묶어가지고 하나의 텍스처로 만들어서 각 미니타일마다 하나하나씩 SDL_RenderCopy한겁니다. 단순해요. 그런데 그냥 이렇게 단순무식하게 하면 망한다는거죠.

문제점들

1. 만약에 모든 미니타일들을 출력하려 한다면
미니타일은 이론적으로 최대 65536개 존재 * 8*8 * 4byte(RGBA) = 2^26byte = 64MB : 이정도면 적당하다 보기는 어렵지만 넘어가자.
하지만 모든 타일이 한번에 보이도록 최소로 스크롤을 하면 한번에 미니타일 256 * 256 * 16 = 1048576개를 출력해야한다.
한마디로 멘붕이다. 이건 아니다.
2. 좋아. 맵을 통째로 렌더링하자.
텍스처 크기는 8192 * 8192 (256*32가 각 맵의 가로세로길이)... 이거 몇메야... 2^30 = 1GB 그래픽 메모리를 잡아먹는다. 나라도 이따구인 맵에디터 안쓴다. 버려.
3. 좋아. 메가타일 단위로 렌더링하자.
필요한 메모리수는 4 * 32 * 32 * 32768 = 2^33 = ... 뭔가 아니다. 버리자.
음 생각해보니까 제일 타일 많이 쓰는 타일셋도 사실 8192개의 메가타일만 쓰는것같다. 이러면 총 32MB정도밖에 안쓰는군. 좋아.

whyask37's blog

지 괜찮아보인다. 실제로 SCMDraft2가 이런 방식으로 연산하는것으로 알려져 있기도 하고.

5. 그러면 어떻게 해야하지? 뭔가 메모리가 엄청나게 낭비되고있는것 같다. 스타에서 실제 타일셋 데이터는 1MB도 안되는데 1GB라니 32MB라니 뭔가 좀 이상하다.

제가 원하는 것은

1. 자유로운 Zooming
2. 자유로운 Scrolling

2가지입니다. 이 2가지만 하면 됩니다. 그런데 이 2가지를 잡기가 꽤 힘들것같아요.

그런데 이거 2마리 토끼를 제대로 잡으려면 꽤나 엄청나게 많은 노력이 필요하다고 합니다. (처음 생각보다 훨씬 많은 노력이 필요해요)

저는 귀찮으므로 일단은 대충 때우도록 합시다. 메모리가 얼마나 많이 들든 속도가 얼마나 느리든 일단 되게만 해놓고

나중에 다시 와서 고칩시다.

스타크래프트 자체가 좀 변태적으로 설계되서 픽셀 렌더링을 하는것 같습니다.

만 스타크래프트의 해상도는 $640 * 480$ 이고요 저는 $1366 * 768$ 의 해상도를 다루고 있습니다. (노트북) 아무래도 매 프레임마다 $1366 * 768$ 개의 픽셀을 일일이 CPU로 계산해서 채우는건 좀 무리가 있는것 같습니다.

그러므로 저는 $256 * 256$ 짜리 블록들 각각을 픽셀 렌더링을 하고, 블록 단위로 GPU에서 렌더링을 하도록 할겁니다.

일단 기본적으로 다음 알고리즘을 깔고 시작하죠.

타일 렌더링 방법 (가설)

1. 각 타일맵 스프라이트는 $256px * 256px$ 크기를 가진다.
2. 타일맵 스프라이트는 총 128개의 있다. 총 32MB정도의 메모리를 차지한다. 그래픽카드 메모리가 요즘 256MB는 넘는다고 들었습니다. 제 컴퓨터에 달린거는 Intel꺼 Nvidia꺼 둘다 1024MB 이상이네요. 따라서 32MB정도는 그냥 써도 될것 같습니다.
3. 타일맵 스프라이트는 2가지의 원소로 이루어진다.
 1. 시작 (x,y)
 2. 배율.
4. 각 배율의 스프라이트는 픽셀 단위로 렌더링이 가능하다.

대충 이런 방식으로 렌더링을 합니다.

whyask37's blog

너다. 네이머에 적잖은것같아요.

스타크래프트에는 여러 MPQ들을 하나로 생각하기때문에 MpqRead 클래스를 약간 변형해서 여러개의 MPQ 파일을 동시에 읽을 수 있게 해놓았습니다. 첨부파일에서 한번 봐보세요.

오리지널 타일셋은 StarDat.mpq,
브루드워 타일셋은 BrooDat.mpq
에 있습니다. patch_rt.mpq는 여러 유닛들의 정보들을 담고 있고요.
맵에디터를 만들면서 우리는 이 3가지 mpq를 사용할거기 때문에 이 mpq들을 모두 읽어야할것같아요.

레지스트리에서 스타가 어디 깔려있는지 확인하는 작업도 있겠지만 일단 당장은 테스트용으로
그냥 .exe랑 같은 폴더에 저 3개의 mpq가 있다 가정하고 시작하겠습니다.

이제 mpq를 로딩하고, (scdata/scmpq.cpp) 타일셋을 로딩하면 (scdata/tileset.cpp) 됩니다.
타일셋 포맷은 3강을 참조하세요. 3강과 거의 동일한 방법으로 타일셋을 로딩할겁니다.

[scdata/tileset.h]

Colored By **Color Scripter™**

```

1  #ifndef SCDATA_TERRAIN_HEADER
2  #define SCDATA_TERRAIN_HEADER
3
4  #include "../sdl/SDL.h"
5  #include "../typedef.h"
6  #include "../mpqrw/mpqfent.h"
7  #include "../chunk.h"
8
9  #include <string>
10
11 namespace scdata {
12     //tileset thing
13     namespace terrain {
14         typedef struct { //megatile group
15             struct CV5Data {
16                 uint8 Something[20];
17                 uint16 MegatileIndex[16];
18             } group[4096];
19         } CV5;
20
21         typedef struct {
22             struct VF4Data {
23                 enum flags {
24                     WAI K A R I E = 0x0001

```

whyask37's blog

```

29     };
30     uint16 flag[16];
31     } tileflags[65536];
32 }VF4;
33
34 typedef struct { //megatile graphics
35     struct VX4Data {
36         uint16 VR4Index[16];
37     } data[65536];
38 }VX4;
39
40 typedef struct {
41     struct VR4Data {
42         uint8 color[64];
43     } image[32768];
44 }VR4;
45
46 typedef struct {
47     struct WPEData {
48         uint8 r;
49         uint8 g;
50         uint8 b;
51         uint8 padding;
52     } data[256];
53 }WPE;
54 }
55
56 class Tileset {
57 public:
58     enum {
59         TILE_LOW = 0,
60         TILE_MID,
61         TILE_HIGH
62     };
63
64     Tileset();
65     ~Tileset();
66
67     int load(int era);
68
69     int getERA() const;
70
71     SDL_Color getTilePixel(uint16 mtxm, int x, int y) const; //x,y : by pixel. each 0~31
72     SDL_Color getMinitilePixel(int minitileindex, bool hflipped, int x, int y) const;
73
74     int getTileHeight(uint16 mtxm, int x, int y) const; //x,y : by pixel. each 0~31
75     uint32 getTileFlag(uint16 mtxm, int x, int y) const; //x,y : by pixel. each 0~31. Use VF4::VF4Data::flags

```

whyask37's blog

```

81 private:
82     int _era;
83     terrain::CV5 _cv5;
84     terrain::VF4 _vf4;
85     terrain::VX4 _vx4;
86     terrain::VR4 _vr4;
87     terrain::WPE _wpe;
88 };
89
90 const Tileset* getTileset(int era);
91 }
92
93 #endif

```

[sdata/tileset.cpp]

Colored By **Color Scripter™**

```

1  #include "tileset.h"
2  #include <stdlib.h>
3  #include <assert.h>
4  #include "scmpq.h"
5
6  namespace sdata {
7      Tileset::Tileset() {
8          memset(&_amp;_cv5, 0, sizeof(terrain::CV5));
9          memset(&_amp;_vf4, 0, sizeof(terrain::VF4));
10         memset(&_amp;_vx4, 0, sizeof(terrain::VX4));
11         memset(&_amp;_vr4, 0, sizeof(terrain::VR4));
12         memset(&_amp;_wpe, 0, sizeof(terrain::WPE));
13     }
14
15     Tileset::~Tileset() { }
16
17
18     int Tileset::load(int era) {
19         static const std::string tileset_array[8] = {
20             "badlands",
21             "platform",
22             "install",
23             "ashworld",
24             "jungle",
25             "desert",
26             "ice",
27             "twilight"
28         };
29
30         std::string path("tileset\\");
31         const std::string& tileset_name = tileset_array[era & 0x7];

```

whyask37's blog

```

36     std::string vx4path = path + tileset_name + ".vx4";
37     std::string vr4path = path + tileset_name + ".vr4";
38     std::string wpepath = path + tileset_name + ".wpe";
39
40     Chunk cv5, vf4, vx4, vr4, wpe;
41     if(!mr->extract(cv5path.c_str(), &cv5)) return -1;
42     if(!mr->extract(vf4path.c_str(), &vf4)) return -1;
43     if(!mr->extract(vr4path.c_str(), &vx4)) return -1;
44     if(!mr->extract(vx4path.c_str(), &vr4)) return -1;
45     if(!mr->extract(wpepath.c_str(), &wpe)) return -1;
46
47     memcpy(&_cv5, cv5.data(), min(cv5.size(), sizeof(terrain::CV5)));
48     memcpy(&_vf4, vf4.data(), min(vf4.size(), sizeof(terrain::VF4)));
49     memcpy(&_vx4, vx4.data(), min(vx4.size(), sizeof(terrain::VX4)));
50     memcpy(&_vr4, vr4.data(), min(vr4.size(), sizeof(terrain::VR4)));
51     memcpy(&_wpe, wpe.data(), min(wpe.size(), sizeof(terrain::WPE)));
52     _era = era & 0x7;
53
54     return 0;
55 }
56
57
58 int Tileset::getERA() const {
59     return _era;
60 }
61
62
63 //WPE
64
65
66 SDL_Color Tileset::getPaletteColor(uint8 palcol) const {
67     const terrain::WPE::WPEData &wpedata = _wpe.data[palcol];
68     SDL_Color col = {wpedata.r, wpedata.g, wpedata.b, 255};
69     return col;
70 }
71
72
73 //Tile pixel
74
75
76 SDL_Color Tileset::getTilePixel(uint16 mtxm, int x, int y) const {
77     assert(!(x < 0 || y < 0 || x >= 32 || y >= 32));
78
79     int minitileindex;
80     bool hflipped;
81     getMinitileIndex(mtxm, x, y, &minitileindex, &hflipped);
82

```

whyask37's blog

```

88  SDL_Color Tileset::getMinitilePixel(int minitileindex, bool hflipped, int subx, int suby) const {
89      assert(!(subx < 0 || suby < 0 || subx >= 8 || suby >= 8));
90      if(hflipped) subx = 7 - subx;
91
92      const terrain::VR4::VR4Data &vr4data = _vr4.image[minitileindex];
93      uint8 pal = vr4data.color[(suby << 3) + subx];
94      return getPaletteColor(pal);
95  }
96
97
98  //Tile height
99
100
101  int Tileset::getTileHeight(uint16 mtxm, int x, int y) const {
102      assert(!(x < 0 || y < 0 || x >= 32 || y >= 32));
103
104      //int minitileindex;
105      //getMinitileIndex(mtxm, x, y, &minitileindex, nullptr);
106      uint32 flag = getTileFlag(mtxm, x, y);
107      if(flag & terrain::VF4::VF4Data::MID) return 1;
108      else if(flag & terrain::VF4::VF4Data::HIGH) return 2;
109      else return 0;
110  }
111
112
113  //flag
114
115
116  uint32 Tileset::getTileFlag(uint16 mtxm, int x, int y) const {
117      int group = mtxm >> 4;
118      int index = mtxm & 0xf;
119      int megatile = _cv5.group[group].MegatileIndex[index];
120
121      int row = y >> 3;
122      int col = x >> 3;
123      return _vf4.tileflags[megatile].flag[(row << 2) | col];
124  }
125
126
127  //Base
128
129
130  int Tileset::getMinitileIndex(int mtxm, int x, int y, int *minitileindex, bool *hflipped) const {
131      assert(!(x < 0 || y < 0 || x >= 32 || y >= 32));
132
133      // 1. get megatile
134      int group = mtxm >> 4;

```


whyask37's blog

```

140     int col = x >> 3;
141     if(minitileindex) *minitileindex = _vx4.data[megatile].VR4Index[(row << 2) | col] >> 1; //15 MSB : minitile
142     if(hflipped) *hflipped = (_vx4.data[megatile].VR4Index[(row << 2) | col] & 1) == 1; //LSB : horizontal flip
143
144     return 0;
145 }
146
147
148 const Tileset* getTileset(int era) {
149     era &= 0x7;
150
151     static Tileset* tileset[8];
152     if(tileset[era] = nullptr) {
153         tileset[era] = new Tileset;
154         tileset[era]->load(era);
155     }
156
157     return tileset[era];
158 }
159 }
160

```

코드가 참 깔끔하고 간단하죠? (인정해줘요.)

마지막으로 맵 파일을 읽고, 맵 파일을 렌더링해봅시다.

벌써부터 GetOpenFileName같은거 쓰기는 좀 귀찮으니까, .exe 폴더에 맵 하나 놓고 파일명으로 읽어들이는걸로 합시다.

저는 제가 예전에 만든 자작맵인 saibi.scx를 테스트용으로 쓸게요.

타일맵 렌더링은 MainProcessUnit에서 부르는 RenderFunc의 형식으로 같겁니다.

MainProcessUnit은 이렇게 구현합시다.

Colored By **Color Scripter™**

```

1  #include "mainproc.h"
2  #include "mainproc_render/rendertile.h"
3
4
5  Camera::Camera() : _camx(0), _camy(0), _zoom(1), _scrx(0), _scry(0), _scrw(0), _scrh(0) {}
6  Camera::~Camera() {}
7
8  int Camera::getPosition(double* camx, double* camy, double* zoom) const {
9      if(camx) *camx = _camx;
10     if(camy) *camy = _camy;

```

whyask37's blog

```

16  _camx = camx; _camy = camy; _zoom = zoom;
17  return 0;
18 }
19
20 int Camera::setScreen(double scrx, double scry, double scrw, double scrh) {
21  _scrx = scrx;
22  _scry = scry;
23  _scrw = scrw;
24  _scrh = scrh;
25  return 0;
26 }
27
28 int Camera::Scr2Coord(double *x, double *y) const {
29  *x = (*x - _scrx - _scrw / 2) / _zoom + _camx;
30  *y = (*y - _scry - _scrh / 2) / _zoom + _camy;
31  return 0;
32 }
33
34 int Camera::Coord2Scr(double *x, double *y) const {
35  *x = (*x - _camx) * _zoom + _scrx + _scrw / 2;
36  *y = (*y - _camy) * _zoom + _scry + _scrh / 2;
37  return 0;
38 }
39
40
41
42
43 MainProcessUnit::MainProcessUnit() {}
44 MainProcessUnit::~MainProcessUnit() {}
45
46
47 int MainProcessUnit::Init() {
48  SDL_Window *window = SDL_CreateWindow("MapCanvas", SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOS_CENTERED,
49  800, 600, SDL_WINDOW_RESIZABLE);
50  SDL_Renderer *renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC);
51  SetWindow(window, renderer); //이 ProcessUnit에서는 Window를 이걸로 씁니다.
52
53  RegisterRenderFunc(rendertile); //rendertile을 렌더러로 삼습니다.
54
55  // 임시로 대충 scenario.chk를 얻는 로직을 만들었습니다.
56  MpqRead mr;
57  mr.open("saibi.scx");
58  Chunk chk;
59  mr.extract("staredit\\scenario.chk", &chk);
60  _map.load(chk);
61  mr.closeall();
62

```

whyask37's blog

```

68 while(GetEvent(&ev)) {
69     switch(ev.type) {
70         case SDL_WINDOWEVENT:
71             switch(ev.window.event) {
72                 case SDL_WINDOWEVENT_CLOSE:
73                     return -1; //quit.
74             }
75         }
76     }
77
78     // 전체 해상도를 얻어서 카메라가 전체 화면에 에디터 내 좌표를 투영시킬 수 있도록 재조정합니다.
79     {
80         int scrw, scrh;
81         getWindowDimension(&scrw, &scrh);
82         _cam.setScreen(0, 0, scrw, scrh);
83     }
84
85     // 키 입력에 따라서 카메라를 움직입니다.
86     // Q : 확대, W : 축소, 방향키 : 이동
87     {
88         double x, y, zoom;
89         int mapw, maph;
90         mapw = _map.getTerrain()->getWidth();
91         maph = _map.getTerrain()->getHeight();
92
93         _cam.getPosition(&x, &y, &zoom);
94
95         if(GetAsyncKeyState(VK_LEFT)) x -= 8 / zoom;
96         if(GetAsyncKeyState(VK_RIGHT)) x += 8 / zoom;
97         if(GetAsyncKeyState(VK_UP)) y -= 8 / zoom;
98         if(GetAsyncKeyState(VK_DOWN)) y += 8 / zoom;
99         if(GetAsyncKeyState('Q')) zoom *= 1.03;
100        if(GetAsyncKeyState('W')) zoom /= 1.03;
101
102        // 클리핑
103        if(x < 0) x = 0;
104        if(y < 0) y = 0;
105        if(x > 32*mapw) x = 32*mapw;
106        if(y > 32*maph) y = 32*maph;
107        if(zoom < 0.01) zoom = 0.01;
108        if(zoom > 30) zoom = 30;
109
110        _cam.setPosition(x, y, zoom);
111    }
112
113    // 각 RenderFunction들을 불러서 실제로 렌더링을 시킵니다.
114    SDL_RenderClear(getRenderer());

```

whyask37's blog

```

120   SDL_RenderPresent(getRenderer());
121
122   return 0;
123 }
124
125
126 int MainProcessUnit::Fin() {
127     for(uint32 i = 0 ; i < _rf.size() ; i++) {
128         _rf[i](nullptr, nullptr, nullptr, _ud[i]);
129     }
130
131     return 0;
132 }
133
134
135
136 int MainProcessUnit::RegisterRenderFunc(RenderFunc f) {
137     _rf.push_back(f);
138     _ud.push_back(nullptr);
139     return 0;
140 }
141
142
143 mapdata::Map* MainProcessUnit::getMap() { return &_amp;_map; }
144 const mapdata::Map* MainProcessUnit::getMap() const { return &_amp;_map; }
145 const Camera* MainProcessUnit::getCamera() const { return &_amp;_cam; }
146

```

이렇게 구현했구요.

그러면 이제 대강 rendertile 함수를 구현해보겠습니다. 이 함수는 타일을 렌더링합니다. 렌더링 알고리즘은 위에 서술한대로 256*256 텍스처를 업데이트하는 식입니다.

음 그리고 앞으로 주석 스타일은 영어로 통일하겠습니다.

주석 한글로 달려고 했는데 영어로 다는게 너무 익숙해져서 올라오는 코드 대부분에 제가 이미 영어 주석을 달아놓아서 그냥 영어 주석으로 갈게요.

Colored By **Color Scripter**™

```

1  #include "rendertile.h"
2  #include <math.h>
3  #include "../lodepng.h"
4
5  struct TileSprite {

```

whyask37's blog

```

11
12 struct TileSpriteMap {
13     TileSprite tile[128];
14 };
15
16 static void CreateTileSprite(MainProcessUnit* mp, TileSprite &renderrect) {
17     const mapdata::Map &map = *(mp->getMap());
18     int x, y, i, j, dsize;
19     const scdata::Tilesset *tileset = map.getTerrain()->getTilesset();
20     dsize = renderrect.size;
21
22     SDL_Color *colormap = new SDL_Color[256*256];
23     array2d<SDL_Color> color(colormap, 256);
24
25     //render
26
27     int minitileindex;
28     bool hflipped;
29     uint32 prevtilehash = 0xffffffff;
30
31     y = renderrect.y;
32     for(j = 0 ; j < 256 ; j++) {
33         x = renderrect.x;
34         for(i = 0 ; i < 256 ; i++) {
35             int ix, iy, fx, fy;
36             ix = x >> 5;   iy = y >> 5;
37             fx = x & 0x1f; fy = y & 0x1f;
38
39             uint16 tile = map.getTerrain()->getTile(ix, iy);
40             uint32 tilehash = (tile << 4) | (fx >> 3 << 2) | (fy >> 3);
41             if(prevtilehash != tilehash) {
42                 prevtilehash = tilehash;
43                 tileset->getMinitileIndex(tile, fx, fy, &minitileindex, &hflipped);
44             }
45
46             color[j][i] = tileset->getMinitilePixel(minitileindex, hflipped, fx & 0x7, fy & 0x7);
47
48             x += dsize;
49         }
50         y += dsize;
51     }
52
53     //update texture data
54     SDL_Texture *tex = renderrect.tex;
55     SDL_UpdateTexture(tex, nullptr, colormap, 256 * 4);
56
57     //done.

```

whyask37's blog

```
63 int tileindex = -1;
64 for(int i = 0 ; i < 128 ; i++) {
65     if(map.tile[i].x == x && map.tile[i].y == y && map.tile[i].size == size) {
66         tileindex = i;
67         break;
68     }
69 }
70
71 if(tileindex == -1) {
72     SDL_Texture* tex = map.tile[127].tex;
73     memmove(&map.tile[1], &map.tile[0], sizeof(TileSprite) * 127);
74     map.tile[0].x = x;
75     map.tile[0].y = y;
76     map.tile[0].size = size;
77     map.tile[0].tex = tex;
78     CreateTileSprite(mp, map.tile[0]);
79 }
80
81 else {
82     if(tileindex) {
83         TileSprite ts = map.tile[tileindex];
84         memmove(&map.tile[1], &map.tile[0], sizeof(TileSprite) * tileindex);
85         map.tile[0] = ts;
86     }
87 }
88
89 return map.tile[0].tex;
90 }
91
92
93 void rendertile(MainProcessUnit* mp, SDL_Window *win, SDL_Renderer *ren, void* &_userdata) {
94     //free things
95     if(mp == nullptr && win == nullptr && ren == nullptr) {
96         if(_userdata) free(_userdata);
97         return;
98     }
99
100     if(_userdata == nullptr) {
101         //init userdata
102         _userdata = malloc(sizeof(TileSpriteMap));
103
104         TileSpriteMap *tmpmap = (TileSpriteMap*)_userdata;
105         for(int i = 0 ; i < 128 ; i++) {
106             tmpmap->tile[i].x = -1;
107             tmpmap->tile[i].y = -1;
108             tmpmap->tile[i].size = -1;
109             tmpmap->tile[i].tex = SDL_CreateTexture(mp->getRenderer(), SDL_PIXELFORMAT_ABGR8888,
```

whyask37's blog

```
115 TileSpriteMap *map = (TileSpriteMap*)_userdata;
116
117 //render!
118 int w, h;
119 mp->getWindowDimension(&w, &h);
120
121 double xcoord[2] = {0, w};
122 double ycoord[2] = {0, h};
123 const Camera* cam = mp->getCamera();
124 cam->Scr2Coord(&xcoord[0], &ycoord[0]);
125 cam->Scr2Coord(&xcoord[1], &ycoord[1]);
126
127 double zoom;
128 cam->getPosition(nullptr, nullptr, &zoom);
129
130 int size = 1;
131 while(size < 1/zoom) size *= 2; //get appropriate texture size
132
133 int ixstart, iystart;
134 int ixend, iyend;
135 ixstart = (int)xcoord[0] & ~(size - 1);
136 iystart = (int)ycoord[0] & ~(size - 1);
137 ixend = (int)xcoord[1] + 1;
138 iyend = (int)ycoord[1] + 1;
139
140 for(int iy = iystart ; iy < iyend ; iy += size * 256) {
141     for(int ix = ixstart ; ix < ixend ; ix += size * 256) {
142         SDL_Texture *tex = GetTileSprite(mp, *map, ix, iy, size);
143         double scrx[2] = {ix, ix + size*256}, scry[2] = {iy, iy + size*256};
144         cam->Coord2Scr(&scrx[0], &scry[0]);
145         cam->Coord2Scr(&scrx[1], &scry[1]);
146
147         SDL_Rect dstrect = {
148             (int)floor(scrx[0]),
149             (int)floor(scry[0]),
150             (int)floor(scrx[1]) - (int)floor(scrx[0]),
151             (int)floor(scry[1]) - (int)floor(scry[0])
152         };
153
154         SDL_RenderCopyEx(mp->getRenderer(), tex, nullptr, &dstrect, 0, nullptr, SDL_FLIP_NONE);
155     }
156 }
157
158 //drawing finished.
159 }
160
```

whyask37's blog

7. 지형을 다시 한번 출력해보자.



여기까지의 프로젝트와 saibi.scx는 첨부파일로 올렸습니다.
(saibi.scx는 원래 오픈소스예요. 괜찮아요)

원래 계획과 좀 많이 틀어졌긴 했지만, 이제 다시 GRP로 갈 수 있게 됐어요.
GRP로 가보죠.

SDL로 갈지 아니면 OpenGL로 갈아탈지가 계속 고민되네요.
조만간 결정해야겠어요.

#IT·컴퓨터

첨부파일

MapCanvas.zip

saibi.scx

whyask37's blog



왜물어

whyask37님의 블로그입니다.

이웃추가

이 블로그 **Project SMD** 카테고리 글

9. 쉬어가기 - 텍스트 트리거 구상

2013. 9. 26.

0

8. GRP 파일을 읽고 출력해보자. (1)

2013. 9. 25.

6

7. 지형을 다시 한번 출력해보자.

2013. 9. 19.

6

6. 기초 작업

2013. 9. 12.

0

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

2013. 9. 11.

1



이 블로그 인기글

6

whyask37's blog

11

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

2013. 9. 11.

1

[빨강의] 13. 트리거 프로그래밍 - TRIG-MRGN 루프

2014. 2. 24.

0

4. scenario.chk

2013. 9. 10.

0

[빨강의] 2. 데스 사이의 대입, 더하기

2014. 1. 19.

1



back to top

blog market
농민후계자가 추천하는 메뚜기쌀



View in PC version