

whyask37's blog

Project SMD

6. Foundation work



why do you ask
2013. 9. 12. 20:28

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>

Our map editor will use SDL as its rendering engine. libsdl.org (SDL official website) says that SDL 2.0 is awesome. Rumor has it that SDL 2.0 will offer its own hardware-accelerated graphics engine. very good.

The map editor's graphics we're going to make are... Well, the map editor isn't finished yet. Unfortunately, there are no graphics in the map editor.
Let's start by following something similar to SCMDraft2.

Let's print the terrain first. I'm also learning SDL 2.0 right now, so the codes here are probably at the level of compiling examples;

like. I want to print the terrain. But how do I print it?

Let's implement three simple things first. Actually, these three are all I want.

- 1. Terrain Output**
- 2. Scroll**
- 3. Free Zoom In/Zoom Out**

yes we will go

Actually, I don't think it's that simple at all. I need to do some shoveling.

whyask37's blog

You can download the SDL by following the link below.

SDL 2.0: <http://libsdl.org/download-2.0.php>

SDL_image : http://www.libsdl.org/projects/SDL_image/ : Makes an image into a surface

SDL_ttf : http://www.libsdl.org/projects/SDL_ttf/ : Font rendering

Yes, it seems like this would be it. You do the SDL installation yourself, (I'll deal with it properly later when I write a real course.)

Earlier we learned how to print a terrain and put it in an array. Today, let's print the terrain on the screen.

Let's write the basic framework of SDL 2.0 as follows.

Colored By **Color Scripter**™

```

One #include "sdl/SDL.h"
2
3 int SDL_main( int argc, char * argv[]) {
4     // 0. Initialize SDL
5     SDL_Init(SDL_INIT_VIDEO);
6
7     // 1. Open a window
8     SDL_Window *window = SDL_CreateWindow("test window", SDL_WINDOWPOS_CENTERED, SDL_WINDOW
9         800, 600, SDL_WINDOW_RESIZABLE | SDL_WINDOW_MAXIMIZED);
10
11     // 2. 렌더러를 만든다.
12     SDL_Renderer *renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED | SDL_RENDER
13
14
15     int color = 0;
16     int dc = 1;
17
18     // 3. 루프
19     SDL_Event ev;
20     bool active = true;
21     while(active) {
22         while(SDL_PollEvent(&ev)) { //처리할 이벤트가 있으면
23             switch(ev.type) { //적당히 처리
24                 case SDL_QUIT: //X버튼을 눌렀으면 (Alt+F4 는 아직 안되네요)
25                     active = false; //active = false. 이러면 while(active)를 나오겠죠.
26             }
27         }
28
29         // 여기에 프로그램의 메인 루프를 넣으면 됩니다.
30
31         // color가 0~255~0~255 이렇게 왔다갔다 하게 하는 코드

```

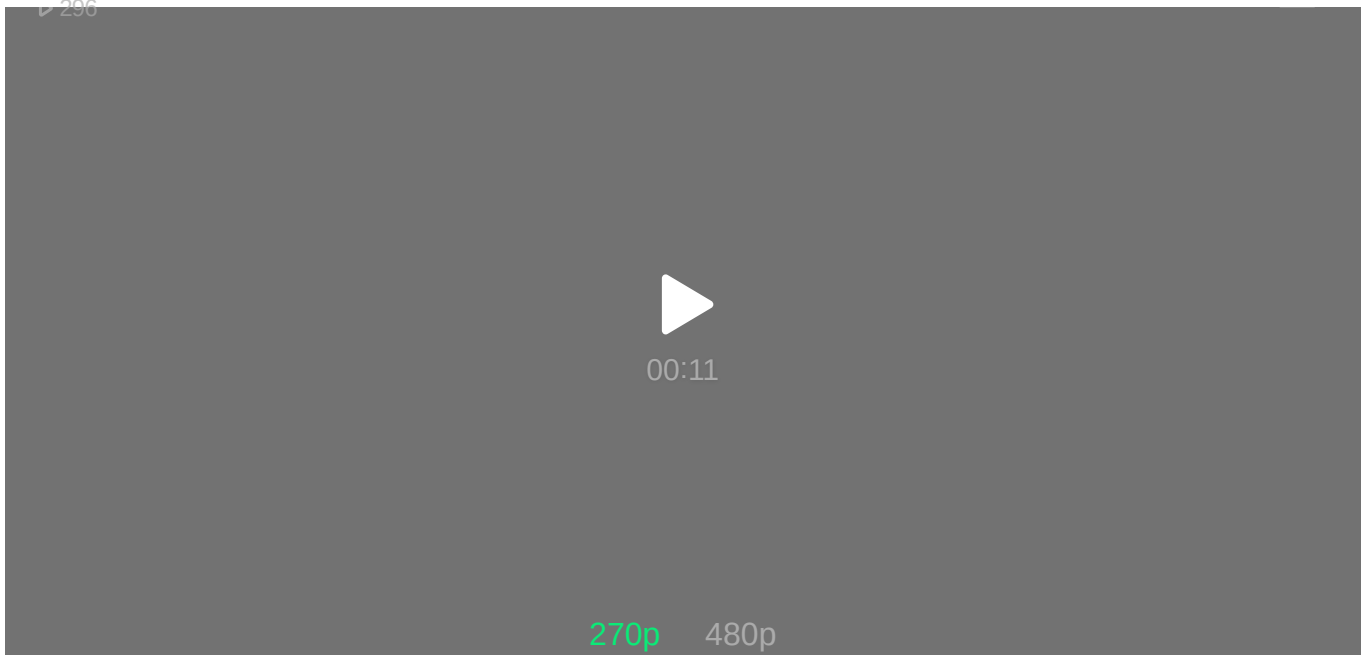
whyask37's blog

```
36
37     SDL_SetRenderDrawColor(renderer, color, color, color, 255); //로 FillRect에서 쓸 색상 지정
38     SDL_RenderFillRect(renderer, NULL); //해서 화면 전체(NULL)에 렌더링
39
40     SDL_RenderPresent(renderer); //하고난 다음에 지금까지 그린거를 모니터에 띄운다. (더블버퍼링의 Flip()) 같은것같
41 }
42
43 // 4. 다 마치고 종료
44 SDL_DestroyRenderer(renderer);
45 SDL_DestroyWindow(window);
46 SDL_Quit();
47 return 0;
48 }
49
```

실행하면 아래 동영상과 같이 나옵니다.

6. 기초 작업

296



우왕 저는 DirectX같은거 쓴 적도 없는데 저절로 DirectX를 골라서 하드웨어 렌더링해주네요 우왕 굳굳

좋아요. 일단 이렇게 해두죠.

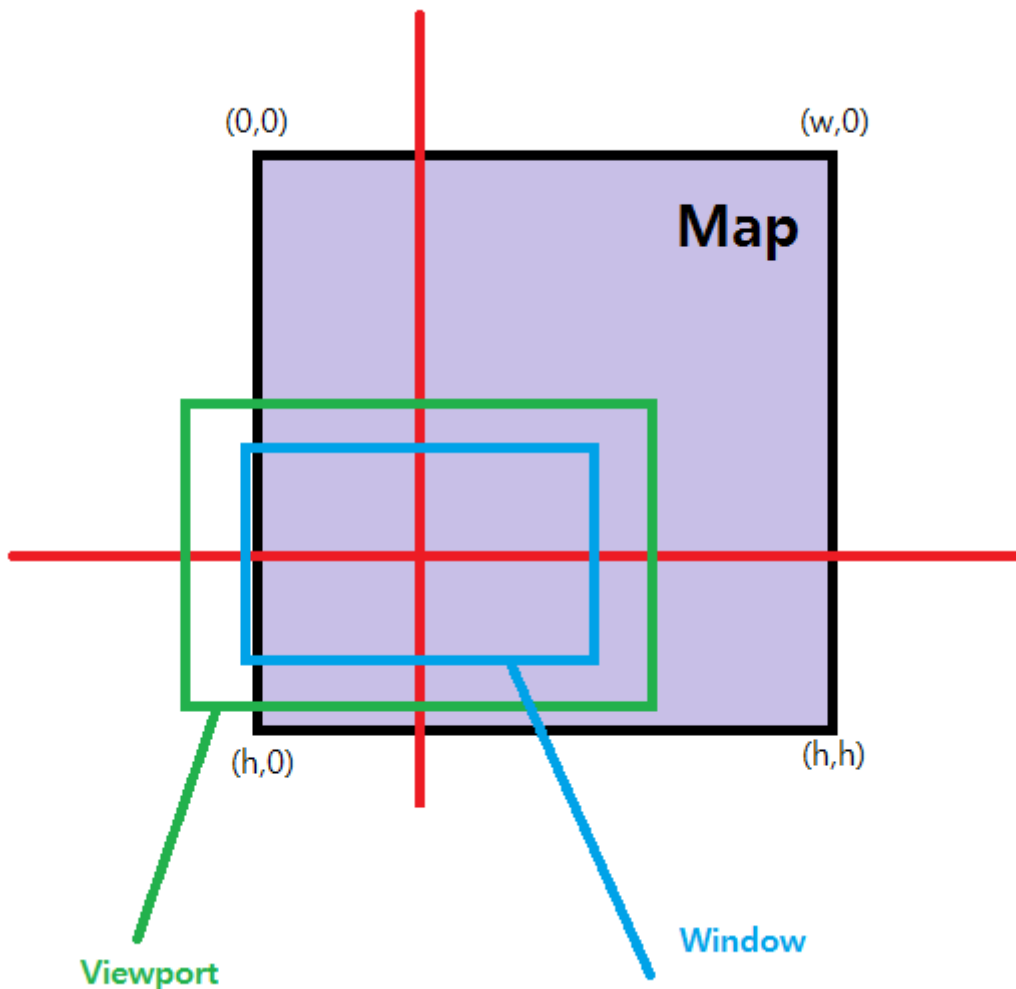
whyask37's blog

우왕 좋아요. 기쁜 마음을 살려가지고 재미있게 타일맵까지 그려봅시다.

일단 카메라부터 만들죠. 대강 구조는 이렇게 하면 될 듯 합니다.

1. 지형 출력
2. 스크롤
3. 자유로운 줌인 / 줌아웃

을 지원하기 위해서 다음과 같이 구상하였습니다.



Map이라는게 있고, Map 밖에는 아무것도 없습니다. Map은 좌표의 기준을 정해줍니다.

whyask37's blog

할 수 있도록 하는게 낫습니다.

이 때 resize 기준은... 아무래도 맵의 확대 비율은 조절하지 않는게 좋을것같아요.

2. Viewport의 중심은 Map 전체를 돌아다닐 수 있도록 하면 좋을것같아요.

3. Window는 실제로는 맵 위에 있는게 아니라 그냥 Viewport 내 내용물을 출력하는 역할만 하는게 좋을것같습니다.

4. 일단은 Window와 실제 띄워진 윈도우의 크기가 일치하지 않는다 가정해야겠습니다. 윈도우에 툴바를 붙일수도 있잖아요.

이런 대강의 결과들을 보니까 Viewport와 Map 구조체를 어떻게 만들어야할지 대강 느낌이 왔네요.

1. Viewport는 Viewport의 중심점의 좌표와 Map의 확대 비율만을 가집니다. Viewport의 가로 세로 길이는 맵이 렌더링될 때 실시간으로 계산됩니다.

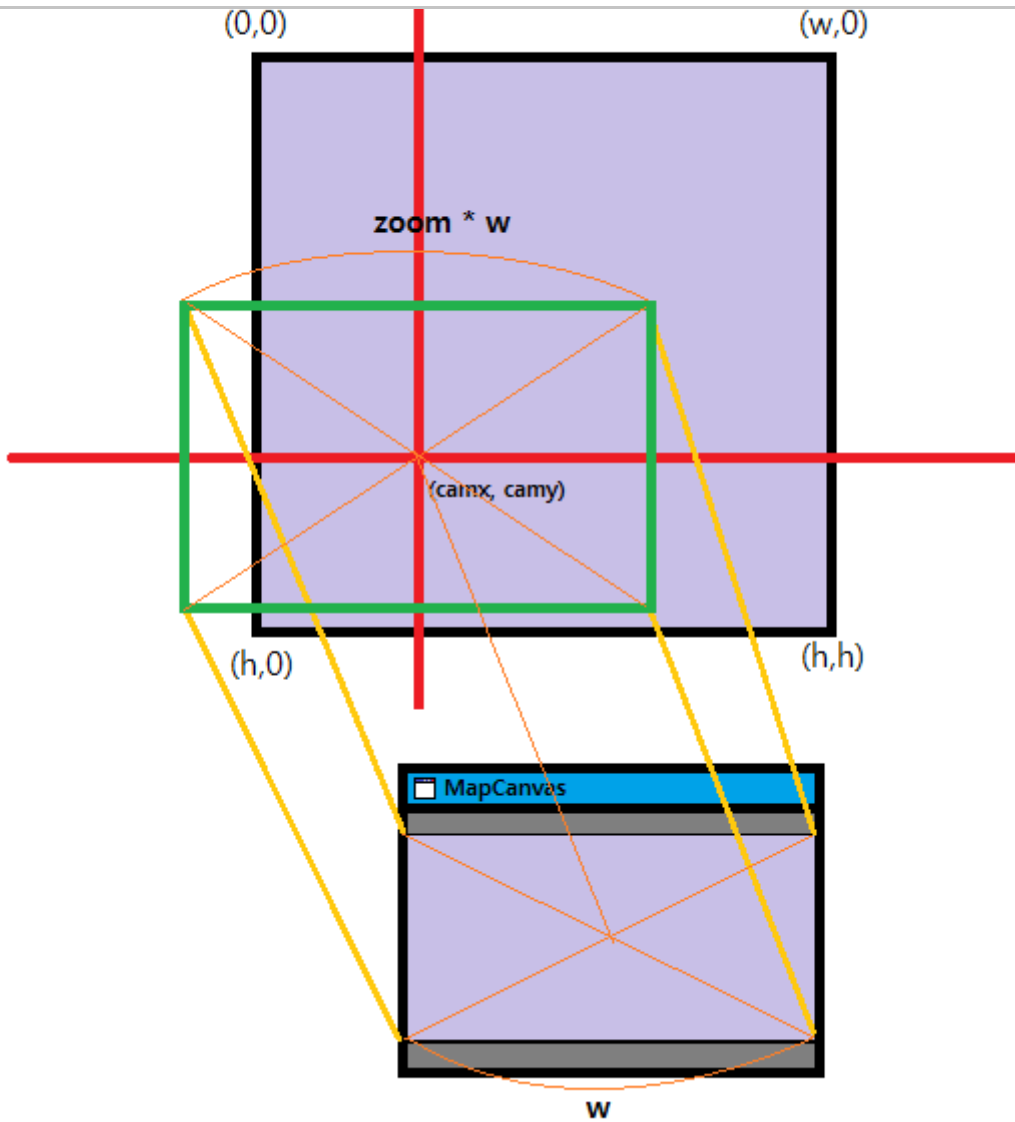
2. 위 그림에서 Window라고 한 것이 사실 Viewport가 될 것 같습니다... 뭔가 용어가 꼬였어요.

이러고보니 다음과 같이 용어를 재정의하는게 나을것같네요.

Viewport -> Camera, Window -> Viewport 로 바꿉시다.

그럼 이렇게 되겠네요.

whyask37's blog



그림에 오타가 있네요... $\text{zoom} * w$ 가 아니라 w / zoom 이겠쥬.

맵을 렌더링하는 거는 왠지 특별한것같아요. 뭔가 배경화면을 렌더링하는것같거든요.

생각해보니 배경화면에 맵만 들어갈 수 있는건 아닌것같아요. 로케이션 표시, 유닛 표시 등도

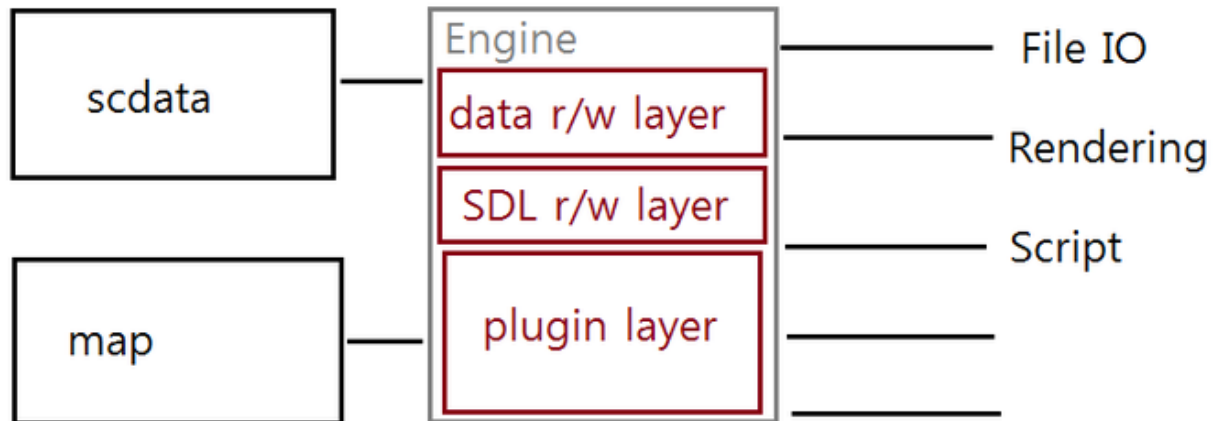
그러면 왠지 이런 배경화면을 그리는 클래스와 맵을 그리는 클래스를 분리해놓아야할것같다는 생각이 문득 드네요

그리고 맵 자체를 다루는 클래스를 분리해야할것같네요.

일단 기초공사부터 해놓고 시작합시다.

whyask37's blog

간제적인 프로그램을 이렇게 팔것인가, 가 문제일텐데요
저는 이렇게 팔거예요.



별거 아니예요. 그냥 일부러 큼직큼직하게 영어로 써본거예요.

간단하게 말해서 데이터와 데이터 처리부, 그리고 프로그래밍 로직을 완전히 분리해버리겠다는 이야기입니다.

... 너무 당연한가요?

우리가 사용해야할 데이터가 어떤건지부터 설정해야죠. 먼저 우리는

- 읽을 맵 파일이 있어야 합니다.
- 읽을 StarDat.mpq 파일 등이 있어야 합니다.

그러니까 읽을 파일은 스타크래프트 데이터와 맵 파일, 2가지로 분류가 되겠죠. 각각 sdata, map 이라고 합시다. 이것들을 raw data예요.

이런 데이터들과 engine이 서로 상호작용할 수 있도록 하는데 data i/o layer입니다.

데이터를 읽고 쓰는 하나의 레이어죠. 쉽게 말해서 데이터 파서예요.

그런데 이런건 raw data와 뭐가 다른건지 저도 이해는 잘 되지 않지만 일단 설계를 더 해봐야할것같아요.

맵 데이터들을 전부 서로 연관이 없도록 분리시킨다...정도로 수정하도록 하죠. 서로 참조할 필요가 없게.

그리고 우리는 SDL을 쓸건데 아무래도 이 SDL 라이브러리 자체를 로직에서 다루게 하면 곤란한것같아요. 래핑을 시켜야겠죠.

SDL이랑 관련되서 우리가 해야 할 일들을 클래스화한 곳이 SDL layer입니다. 그런데 제가 왜 r/w를 넣

whyask37's blog

노석할줄 도루 들디그인이나 증명해요. 네. 들디그인이요.

지형을 출력하는 로직도 플러그인으로 취급할거예요. **플러그인은 서로 독립된 스레드처럼 동작할 겁니다.**

plugin layer는 외부 플러그인과 데이터가 상호작용할 수 있도록 해주는 layer예요. (외부 플러그인은 어떤 경우에서도 절대 raw data와 상호작용하지 못해요. 대신 data i/o layer는 외부 플러그인이 부족하지 않을 만큼 충분히 많은 정보를 raw data에서 읽어야겠죠.)

raw data와 직접 상호작용해야하는것은 다음으로 제한하고, 이런것들은 data i/o layer에 들어갑니다.

1. 스타크래프트의 경로명, 유닛명 등등에 의존하는 함수.
2. 맵의 파일명, 맵의 구체적인 경로, 맵 파일의 크기, 맵 파일에 들어있는 wav 파일 각각의 용량 등 물리적인 파일과 상호작용하는 함수.

아 하여튼 말이 좀 엉키네요. 좀 더 생각해보고 이부분은 확장하거나 수정할게요.

생각해보니까 data i/o layer 이런거 다 뻔짓인것같고
간단하게 모두 프로세스라고 하죠.

그리고 한가지 제한점을 두죠.

우리는 에디터를 만드는게 아니라 뷰어를 만드는것으로 갑니다.

메뉴 이름도 수정하고요. Starcraft Map Debugger 이라고 이름붙이면 되겠네요. (SMD?)

```
#ifndef WINSYS_HEADER #define WINSYS_HEADER #include "sdl/SDL.h" #include "sdl/SDL_image.h" #include "sdl/SDL_ttf.h" #include "typedef.h" #include <string> #include <map> #include <queue> #include <functional> namespace mainsys { class ProcessUnit { public: ProcessUnit(); virtual ~ProcessUnit(); virtual int Init() = 0; virtual int Process() = 0; virtual int Fin() = 0; int GetWindowDimension(int &w, int &h) const; int PostEvent(const SDL_Event &ev); public: SDL_Window* getWindow(); SDL_Renderer* getRenderer(); protected: int SetWindow(SDL_Window *win, SDL_Renderer *renderer); int GetEvent(SDL_Event *ev); private: int _winw, _winh; SDL_Window* _window; SDL_Renderer *_renderer; std::queue<SDL_Event> _evq; }; int RegisterProcessUnit(const std::string &pluginname, ProcessUnit *process); ProcessUnit* GetProcessUnit(const std::string &pluginname); } class MainProcessUnit : public mainsys::ProcessUnit { public: typedef std::function<int(SDL_Window*, SDL_Renderer*)> RenderFunc; typedef std::multimap<int, RenderFunc>::iterator RenderPtr; virtual int Init(); virtual int Process(); virtual int Fin(); int RegisterRenderFunc(int layer, RenderFunc f, RenderPtr *ptr); int UnregisterRenderFunc(RenderPtr ptr); private: std::multimap<int, RenderFunc> _rf; }; #endif
```


whyask37's blog

상 모듈과 프로그래밍이예요. 각 모듈은 윈도우 하나씩 띄울 수 있고

이러한 모듈 중에 data i/o layer도 있는거예요. plugin layer나 sdl layer같은건 없습니다. 각 모듈이 알아서 처리.

MainProcessUnit은 지형 등의 기본적인 화면, 메인화면을 그리는 프로세스고, 여러개의 렌더러를 넘겨 받을 수 있어요.

지형 렌더러나 유닛 렌더러 등등의 각각의 RenderFunc들이 MainProcessUnit에 의해 실행되는 형태죠.

그 결과는 다음과 같아요. (<http://codepad.org/LtJeSIE6>)

```
#include "winsys.h" #include <Windows.h> #include <map> namespace mainsys { static std::map<std::string, ProcessUnit*> processes; static std::map<int, ProcessUnit*> winid_map; int InitMainSystem() { SDL_Init(SDL_INIT_VIDEO); TTF_Init(); IMG_Init(IMG_INIT_PNG); return 0; } int FinMainSystem() { IMG_Quit(); TTF_Quit(); SDL_Quit(); return 0; } ProcessUnit::ProcessUnit() : _winw(0), _winh(0), _window(nullptr), _renderer(nullptr) {} ProcessUnit::~ProcessUnit() { if(_renderer) SDL_DestroyRenderer(_renderer); if(_window) SDL_DestroyWindow(_window); } int ProcessUnit::GetWindowDimension(int &w, int &h) const { w = _winw; h = _winh; return 0; } int ProcessUnit::PostEvent(const SDL_Event &ev) { _evq.push(ev); return 0; } SDL_Window* ProcessUnit::getWindow() { return _window; } SDL_Renderer* ProcessUnit::getRenderer() { return _renderer; } int ProcessUnit::GetEvent(SDL_Event *ev) { if(_evq.empty()) return 0; *ev = _evq.front(); _evq.pop(); return 1; } int ProcessUnit::SetWindow(SDL_Window *window, SDL_Renderer *renderer) { if(_window) { winid_map.erase(winid_map.find(SDL_GetWindowID(_window))); } _window = window; _renderer = renderer; SDL_GetWindowSize(window, &_winw, &_winh); winid_map[SDL_GetWindowID(window)] = this; return 0; } int RegisterProcessUnit(const std::string &pluginname, ProcessUnit *process) { if(process->Init() != 0) return -1; processes.insert(std::make_pair(pluginname, process)); return 0; } ProcessUnit* GetProcessUnit(const std::string &pluginname) { auto it = processes.find(pluginname); if(it != processes.end()) return it->second; else return nullptr; } int TransferEvent(const SDL_Event& ev, int windowID) { auto it = winid_map.find(windowID); if(it == winid_map.end()) return 0; ProcessUnit *win = it->second; win->PostEvent(ev); return 0; } int Mainloop(int argc, char *argv[]) { InitMainSystem(); RegisterProcessUnit("init", new MainProcessUnit); while(!processes.empty()) { //약 30fps 유지 const int desiredtm = 1000 / 30; int starttm = SDL_GetTicks(); //SDL 메시지를 분배해줍니다. SDL_Event ev; while(SDL_PollEvent(&ev)) { switch(ev.type) { case SDL_WINDOWEVENT: TransferEvent(ev, ev.window.windowID); break; case SDL_KEYDOWN: case SDL_KEYUP: TransferEvent(ev, ev.key.windowID); break; case SDL_MOUSEMOTION: TransferEvent(ev, ev.motion.windowID); break; case SDL_MOUSEBUTTONDOWN: case SDL_MOUSEBUTTONUP: TransferEvent(ev, ev.button.windowID); break; case SDL_MOUSEWHEEL: TransferEvent(ev, ev.wheel.windowID); break; case SDL_TEXTINPUT: TransferEvent(ev, ev.text.windowID); break; case SDL_TEXTEDITING: TransferEvent(ev, ev.edit.windowID); break; case SDL_USEREVENT: TransferEvent(ev, ev.user.windowID); break; } } for(auto it = processes.begin(); it != processes.end(); ) { int res = it->second->Process(); if(res == 0) it = processes.erase(it); else it++; } int elapsed = SDL_GetTicks() - starttm; if(elapsed < desiredtm) SDL_Delay(desiredtm - elapsed); } FinMainSystem(); return 0; } int SD
```

whyask37's blog

```
SDL_CreateWindow( "MapCanvas", SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOS_CENTERED, 800, 600, SDL_WINDOW_RESIZABLE | SDL_WINDOW_MAXIMIZED); SDL_Renderer *renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_TARGETTEXTURE); SetWindow(window, renderer); RegisterRenderFunc(-1, BaseRenderer, nullptr); return 0; } int MainProcessUnit::Process() { SDL_Event ev; while(GetEvent(&ev)) { switch(ev.type) { case SDL_WINDOWEVENT: switch(ev.window.event) { case SDL_WINDOWEVENT_CLOSE: return 0; //quit. } } } for(auto it = _rf.begin(); it != _rf.end(); it++) { (it->second)(getWindow(), getRenderer()); } SDL_RenderPresent(getRenderer()); return 1; //no process yet; } int MainProcessUnit::Fin() { return 0; } int MainProcessUnit::RegisterRenderFunc(int layer, RenderFunc f, RenderPtr *ptr) { RenderPtr it = _rf.insert(std::make_pair(layer, f)); if(ptr) *ptr = it; return 0; } int MainProcessUnit::UnregisterRenderFunc(RenderPtr ptr) { _rf.erase(ptr); return 0; }
```

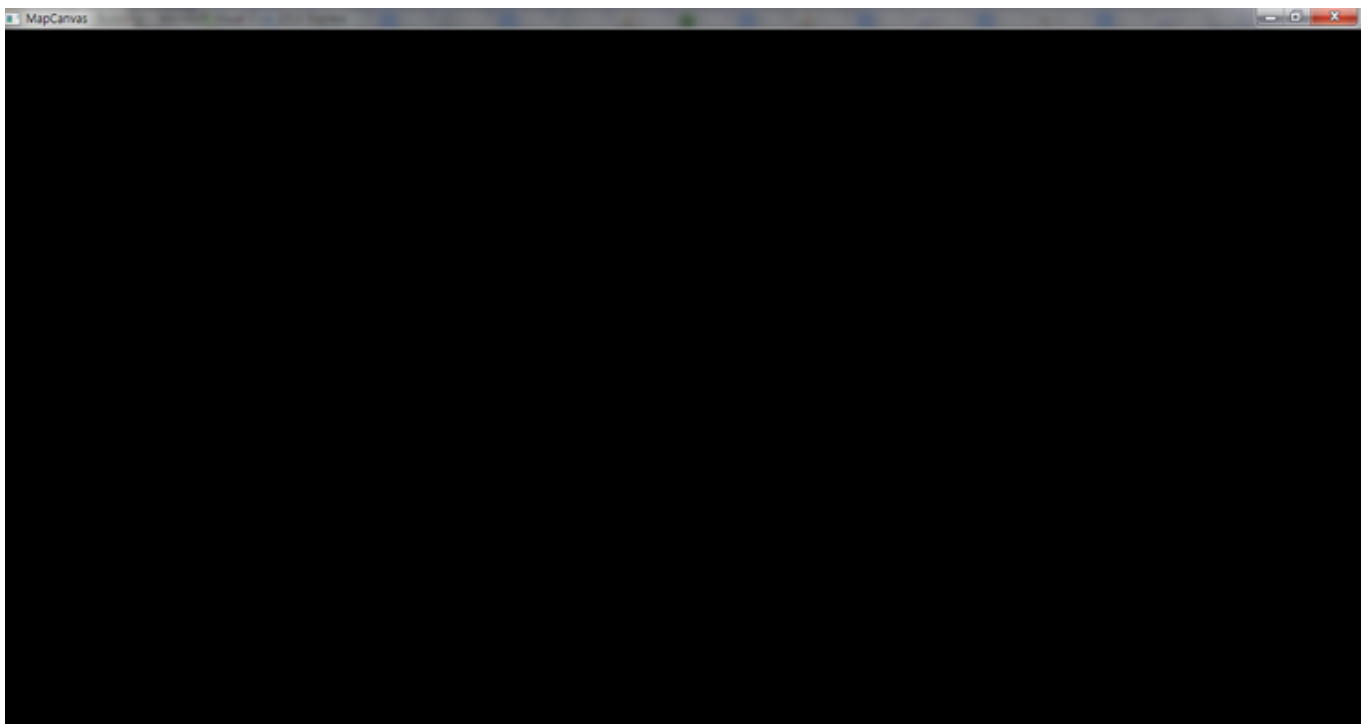
p.s) 왜 std::multimap으로 RenderFunc들을 구현하나요?

저는 그냥 항상 layer로 정렬된 list를 원했을 뿐인데 이렇게 나와버렸네요.

네. 지형이 유닛보다 위에 나와있으면 좀 그렇잖아요? 검은 배경은 layer = -1, 지형은 layer = 0, 유닛은 layer = 1, 스프라이트는 layer = 2 이런 식으로 구성하고싶었어요.

layer들의 리스트를 관리하는건 어렵죠. 나중에 레이어 목록도 다 enum으로 해야할것같아요. 아무래도.

대충 이렇게 해서 실행시키면 이렇게 됩니다.

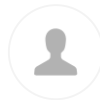


whyask37's blog

이제 여기에 지형을 출력하는 RenderFunc를 먹여보겠습니다. MainProcessUnit에는 카메라도 달아줘야겠죠. 차차 해나갑시다.

음.... 그런데 이거 하나 만들다 날새겠어요. 다음강으로 나머지를 넘기죠.

0



왜물어

whyask37님의 블로그입니다.

이웃추가

이 블로그 Project SMD 카테고리 글

8. GRP 파일을 읽고 출력해보자. (1)

2013. 9. 25.

6

7. 지형을 다시 한번 출력해보자.

2013. 9. 19.

6

6. 기초 작업

0

whyask37's blog

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

2013. 9. 11.

1

4. scenario.chk

2013. 9. 10.

0



이 블로그 인기글

MPQ 가지고 놀기 (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

0

whyask37's blog

back to top



View in PC version