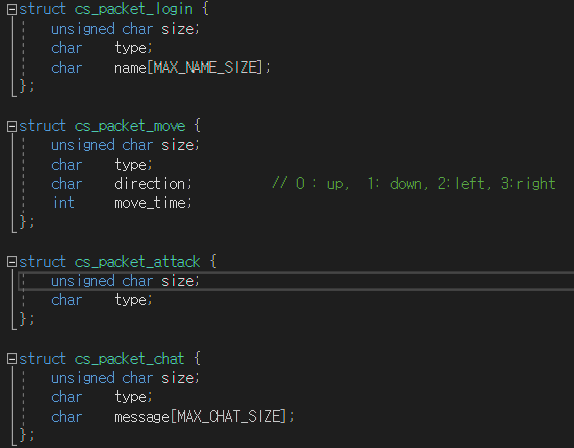
**게임설명서**

설명 사항

* 패킷 구조
* 클래스및 구조체 구조
* 맵
* 몬스터
* Concurrency::concurrent\_priority\_queue를 이용한 이벤트 처리
* 섹터 분할
* Lua
* DB
* 동접자테스트

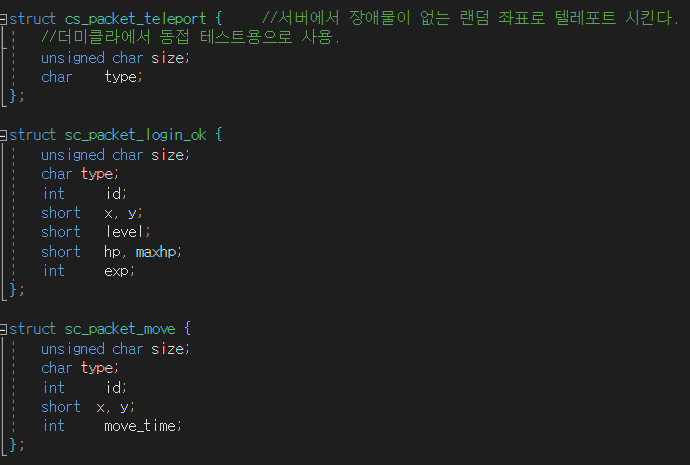
패킷 구조

닉네임을 받아 로그인을 하는 패킷

이동방향을 받는 패킷 move\_time은 더미클라 측정을 위함.

공격을 알리는 패킷

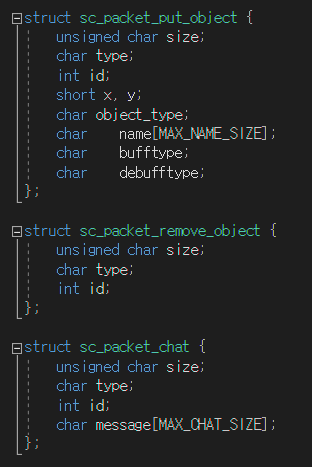
채팅메시지를 보내는 패킷

랜덤위치로 텔레포트 하게 만드는 트리거패킷

----- 서버----

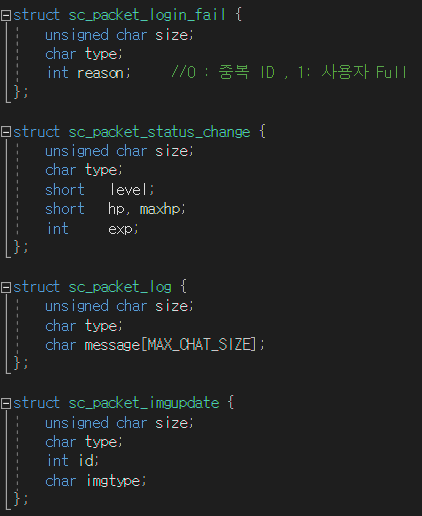
이름으로 db에 접근하여 정보를 꺼내서, player의 정보를 초기화 하게끔 하는 패킷

움직였을 때 누가, 어느좌표로 이동했는지 알려주는 패킷

어떤 오브젝트가 새롭게 나에게 왔는지, 이때 오브젝트들의 정보는 어떤 것이 있는지 알려주는 패킷

어떤 오브젝트가 나에게서 사라졌는지 알려주는 패킷

어떤 플레이어가 채팅을 쳤는지 알려주는 패킷.



로그인 실패했을 경우, 이유를 알려주는 패킷

플레이어의 상태가 변했을 경우 보내주는 패킷, 버프관련 정보도 여기에 넣는다면, 버프만 업데이트 해야하는데 불필요한 정보도 같이 가기 때문에 따로 분리

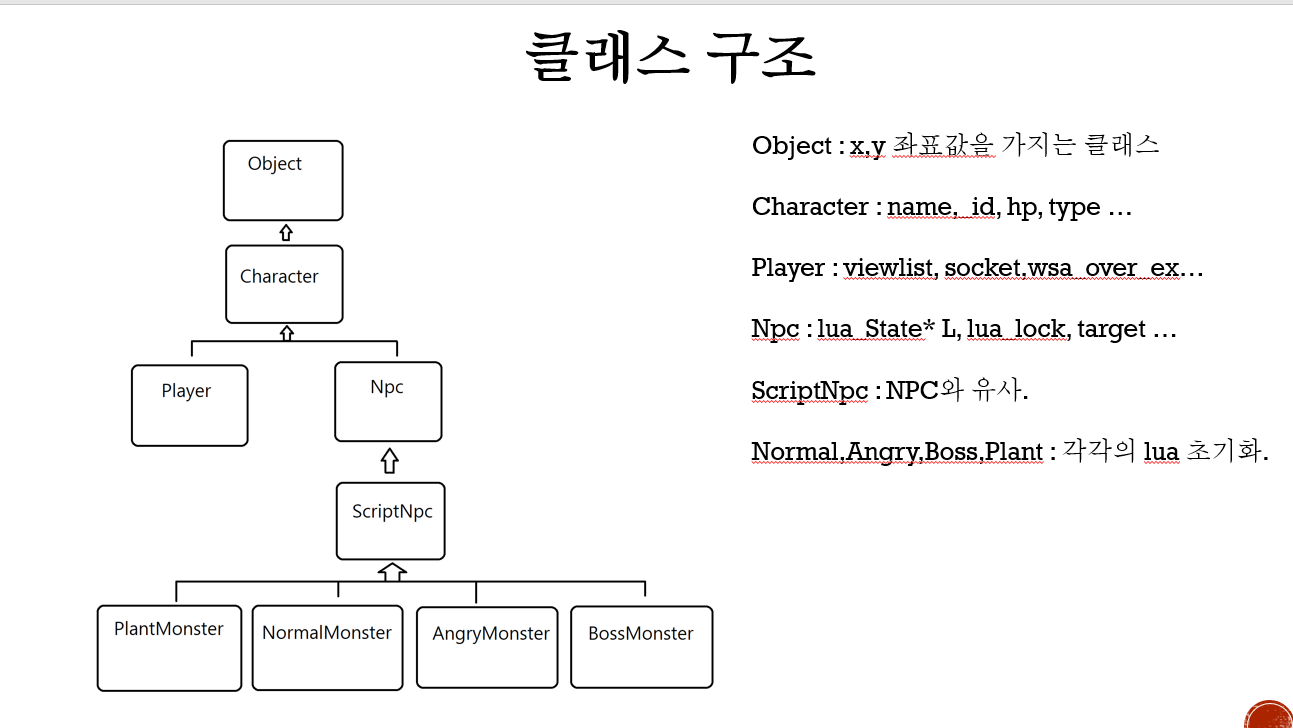
자기자신이 전투를 하여 생기는 로그를 보내주는 패킷

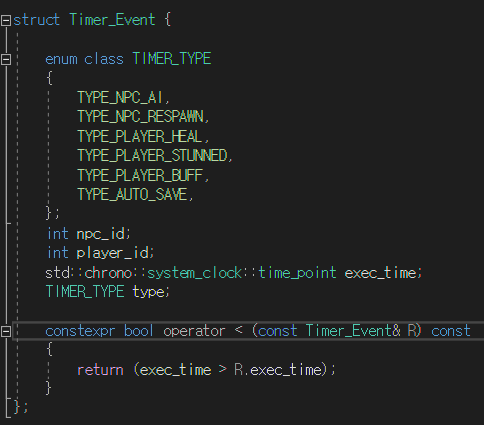
몬스터의 상태(타입)가 바뀌어 이미지를 바꿔주어야할 때, 그 타입을 알려주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

버프를 누가, 어떤버프또는 디버프를 가졌는지 업데이트 해주는 패킷



그 외 구조체 구조

타이머 스레드에 들어갈 자료구조

NPC\_AI : NPC ai를 돌려주기 위함.

Respawn : NPC 부활용 이벤트

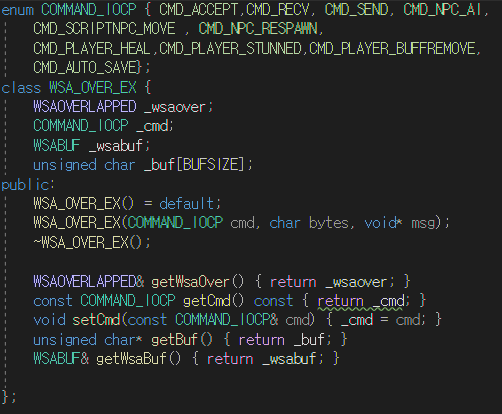
Heal : 플레이어 hp회복 이벤트

Stunned : 플레이어 속박지속시간 이벤트

Buff : 플레이어 공격력버프 지속시간 이벤트

Auto\_Save : 5~10분사이에 자동저장되는 이벤트.

npc\_id와 player\_id로 구분하여 넣는다.

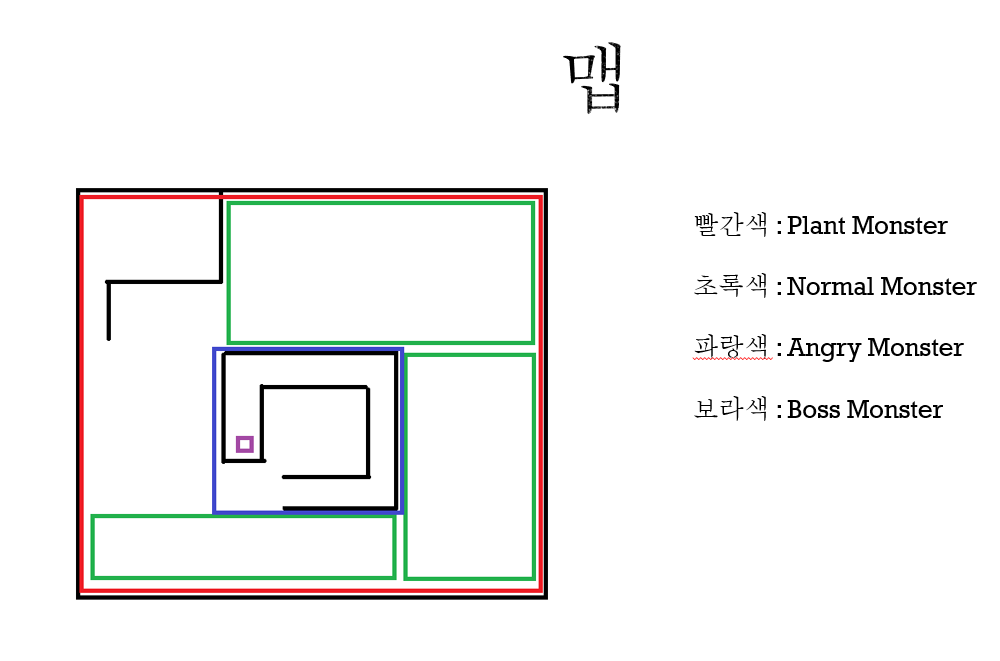
GQCS에서 발생하는 이벤트들. ACCEPT,RECV,SEND가 아닌경우는 타이머큐에서 PQCS로 보내주는 이벤트들이다.

그 외에 확장 오버랩드 구조체의 자료형을 private으로 만들고, getter setter를 만들어주었다.

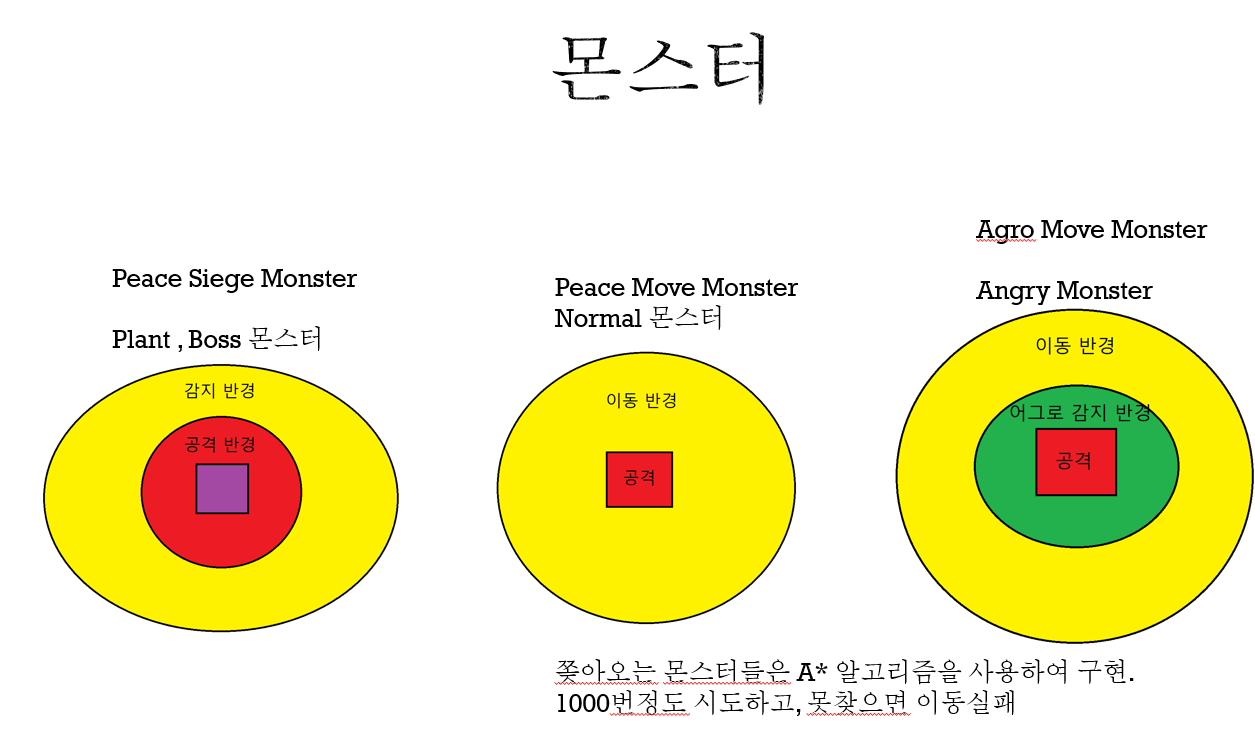
캐릭터,NPC : array<Character\*, MAX\_USER+MAX\_NPC> characters

맵 : Bool mMap[2000][2000] map.txt에서 읽어서 1,0으로 바꿔준다.

타이머관련 : Concurrent\_priority\_queue<Timer\_Event> timer\_queue;

**게임 흐름**

시작마을 좌상단에서 태어나, 빨간,초록구역에서 몬스터를 잡아 강해진 다음, 파란색 구역으로 넘어가서 레벨업을 하다가, 좁은 통로를 이동하여 최종 보스를 잡는 것이 게임의 흐름입니다.



Peace Siege 몬스터, Peace Move 몬스터, Agro Move 몬스터가 존재합니다.

타게팅 시, 어그로를 푸는 방법은 노랑색 구역을 벗어나는 방법밖엔 없습니다. 노란색 공간일 때는 monster의 ai가 계속 돌아가게 됩니다. 빨간색 공간은 몬스터가 공격을 하는 구간입니다.

어그로 몬스터

일정 HP 아래가 되면 대사를 외치며 색이 바뀝니다.

공격력 또한 강해집니다.

보스몬스터

보스몬스터텍스트, 건물, 우리이(가) 표시된 사진

자동 생성된 설명(유령)은 여러가지 공격을 합니다.

1. 일반공격
2. 어그로 대상에게 텔레포트
3. 속박 디버프(사진 검정색 이펙트)

보스몬스터를텍스트이(가) 표시된 사진

자동 생성된 설명 잡으면 근처에 있던 플레이어는 공격력 버프 (빨간화살표)를 몇초동안 얻게 됩니다.

텍스트이(가) 표시된 사진

자동 생성된 설명**타이머 스레드를 통한 이벤트 처리**

exec\_event에서 이벤트를 빼서 처리할 시간이 되었다면 바로 처리해준다.

처리할 시간이 안됐다면 2가지 경우로 나뉜다. 1초(최소 타이머 단위) 이상인가? 이하인가?

1초 이상이라면 이것보다 급한 이벤트가 들어올 가능성이 있으므로 어쩔 수 없이, 큐에 다시 넣어주고 1초를 쉰다. 1초를 쉬는 이유는 최소단위가 1초이고 지금 1초보다 아래가 없다는 건 1초뒤에 밀리는 이벤트 또한 없다는 것.

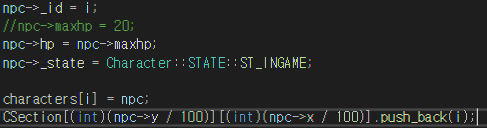
1초 이하라면, 이미 큐에서 빼 버린 이벤트를 다시 넣지 않고 실행시키기 위해 is\_already라는 타이머 이벤트에 값을 복사하고, 1초 이하만큼 쉬어준다. 이때 1초 이하만큼 쉬어주는 이유는 타이머 최소단위가 1초이기 때문에 이것보다 급한 이벤트는 절대 들어올 수 없기 때문에 그만큼 쉬어도 되는 것이다.

**섹터 분할**

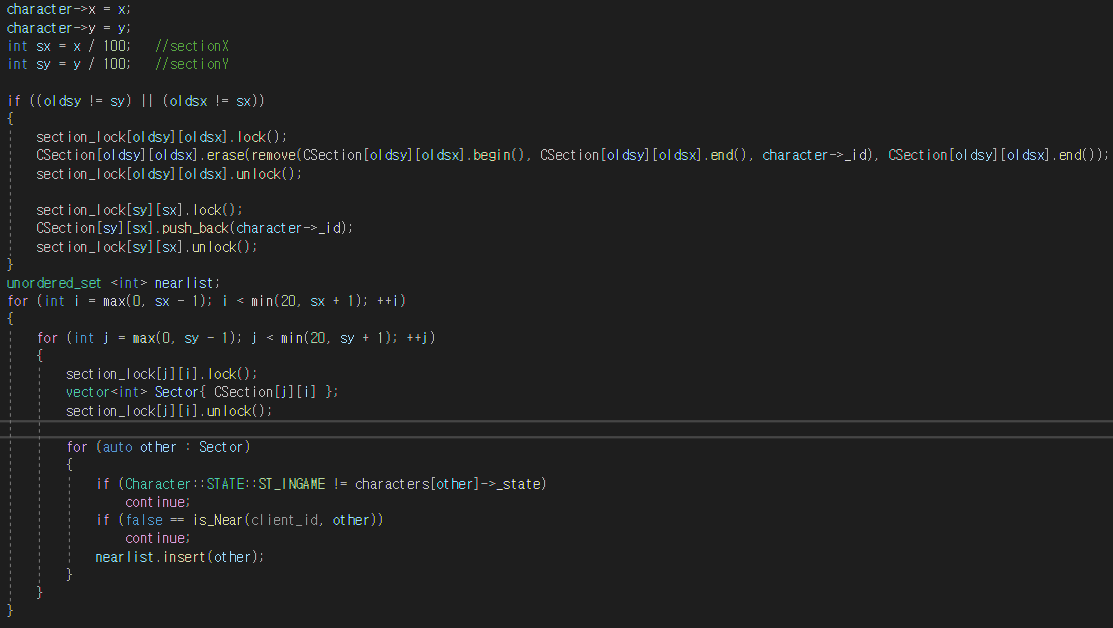
동접자를 늘리기 위한 방법으로 시야처리를 해주었지만, 시야 리스트에 넣기 위해 모든 오브젝트를 검사하는 것 또한 비효율적이기에 맵을 크게 20x20으로 나누어 분류했습니다.



섹터 분할하기 위한 자료구조와, 멀티스레드 환경에서 사용하기 위해 각 구역마다 lock을 하나씩 두었습니다.



Npc 또는 플레이어를 생성할 때, 좌표를 기준으로 각 섹터에 넣어주는 모습입니다.

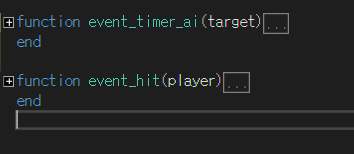


캐릭터가 움직일 때, 이전에 있었던 섹터와, 현재 바뀐 위치의 섹터의 위치가 서로 다르다면, 이전에 있던 섹터에서 지우고, 현재 섹터로 옮겨줍니다.

그리고 자신이 위치한 섹터에서 -1~1 만큼의 범위의 섹터를 검사하여 시야처리를 해줍니다.

이렇게 하면 전체 오브젝트를 검사하지 않고 근처 오브젝트들만 검사할 수 있기에 오버헤드를 줄일 수 있습니다.

**LUA 구조**



기본적으로 lua로 돌아가는 몬스터들은 event\_timer\_ai로 매초마다 자신의 상태를 점검하거나, 공격할 상대가 있다면 때리는 식의 알고리즘을 수행한다.

텍스트이(가) 표시된 사진

자동 생성된 설명또한 공격을 받았을 때, hp를 lua에서 깎는 게 아니라 깎인 hp만 확인하고, 이 hp에 대한 이벤트를 만들어주는 역할을 한다.

예시인 PlantMonster의 루아 구조이다.

event\_timer\_ai에는 몬스터의 타겟을 가져오는데, 타겟이 없다면 INVALID\_TARGET(-1)을 인자로 받아오게 된다. plantmonster의 경우 타겟이 없다면 아무것도 해주면 안 되기 때문에 그냥 나가는 경우이고, 타겟이 없더라도 움직여야 하면 다른 api를 호출하면 된다. 타겟이 있다면 그 타겟에게 공격을 시도하는 api를 호출한다.

event\_hit에서도 누가 때렸는지 인자로 들어오게 되며, 이로 인해 때린 사람의 정보 또한 얻을 수 있다. 맞았을 경우 hp를 감소시키는 건 c++코드 내부적으로 하며, 여기서는 깎인 hp를 보고 어떠한 api를 호출하는지 만 정한다. 이로 인해 api만 가지고 c++를 몰라도 간단하게 스크립트로 몬스터들을 찍어낼 수 있는 구조를 만들었다. 필요한 함수가 있다면, 그때 그때 C++로 구현하고 API화 시키면 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명SendMessage : 1명에게 메시지 보냄

AutoMoveNPC : NPC의 이동 로직 API

MonsterDie : NPC 사망 로직 (잡은 사람에게 경험치를 주거나 그런 로직이 돌아간다.)

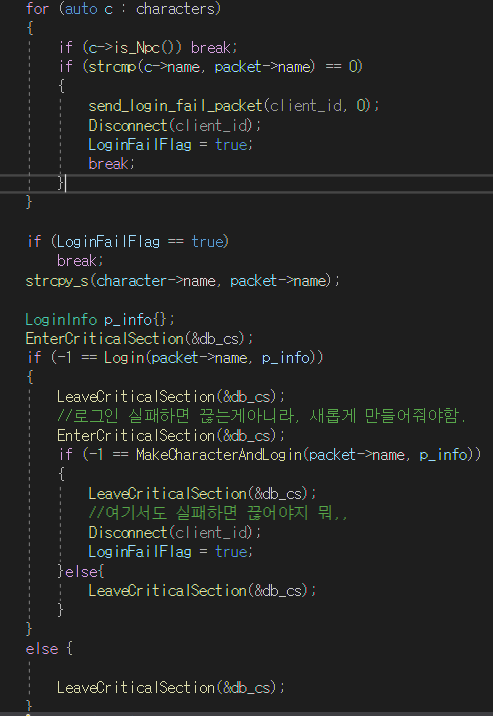
SendMessageMySight : 몬스터를 보고 있는 사람 모두에게 메시지 보냄

ChaseTarget : A\*알고리즘을 c++코드로 돌려서, 타겟을 쫓게함.

TelePortTarget : 보스전용 API로, 주어진 타겟에게 텔레포트.

API\_BossDeBuff(myid,target,디버프 타입, 디버프 지속시간) : 보스전용 API로 타겟에게 특정 상태이상(C++코드에서는 타이머큐에 속박지속시간 해제 이벤트를 넣는다)

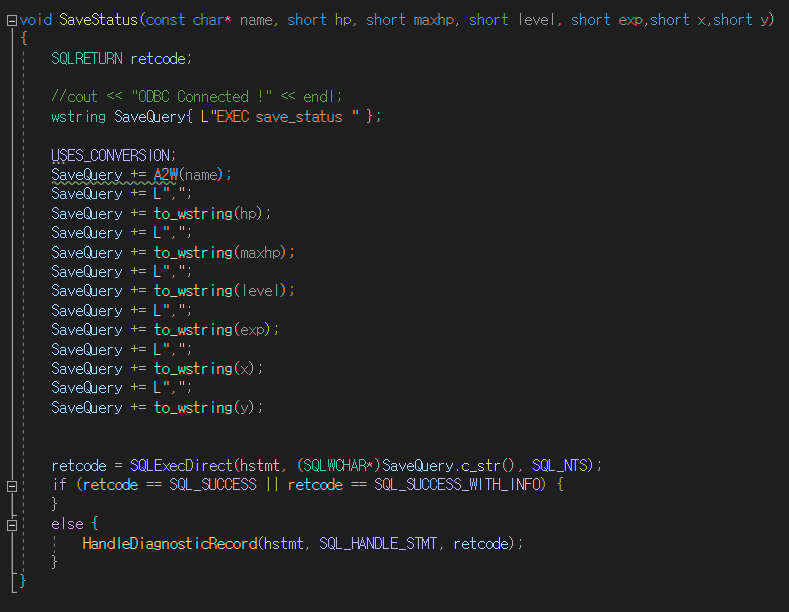
API\_BossBuffMySight(myid,버프 타입,버프 지속시간) : 보스전용 API, 보스를 보고 있는 모든 사람에게 인자 값으로 받은 타입의 버프를 인자 값으로 받은 시간만큼 준다.

**DB**

로그인 패킷 부분입니다. 모든 플레이어(최대1만)을 비교하여, 이름이 같은 플레이어가 존재한다면, 동시접속오류를 보내줍니다.

그게 아니라면 로그인을 시도하는데, 로그인이 실패하면 그 아이디로 계정을 새로 만들어서, 정보를 가져온다.

로그인정보는 p\_info에 쌓이게 되며, p\_info를 가지고 플레이어를 초기화 한 후, login\_ok 패킷을 보내게 된다.

로그인과 autoSave기능을 저장프로시저로 구현하였습니다.

DB내에서는 간단하게 쓰고 인자를 넣어주기만 하면 되고,

텍스트이(가) 표시된 사진

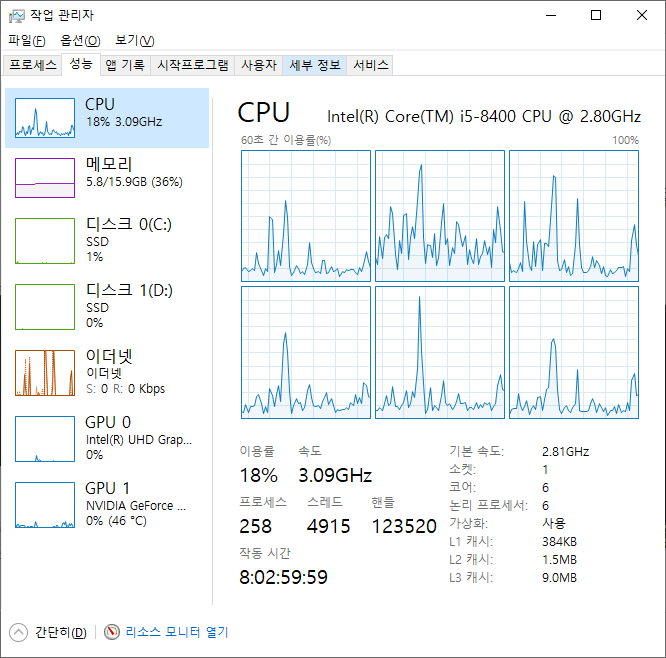
자동 생성된 설명

mssql의 저장프로시저 기능을 이용하여 db내에서 데이터들을 저장하도록 구현했습니다.

동접자 테스트

기준

1. 컴퓨터 사양



1. 몬스터 200’000마리
2. 월드 맵 크기 2000x2000

텍스트이(가) 표시된 사진

자동 생성된 설명Sector 분할 전 Sector 분할 후