게임 서버

프로그래밍

term Project

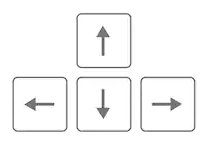
-게임 설명서-

**01 분반 (월 23 화 23)**

**게임공학과 2016182007 김동년**

**목차**

1. **조작법**
2. **구현 내용**
   1. **기본 기능 구현 완료 내용**
   2. **추가기능 구현 완료 내용**
3. **게임 구성 소개**
   1. **게임 화면**
   2. **게임 오브젝트 소개**
   3. **전투 메시지, 플레이어 회복, 채팅 메시지**
4. **최대 동시접속**
5. **프로토콜**
   1. **공용 프로토콜**
   2. **서버 프로토콜**
6. **자료구조**
   1. **Server**
   2. **Client**
7. **게임 흐름**
   1. **게임 로그인**
   2. **플레이어 소개**
   3. **몬스터 소개**
   4. **채팅**
   5. **재접속**
8. **알고리즘**
9. **조작법**

****

상, 하, 좌, 우 이동 방향키

Z: 기본 공격 X: 범위 공격

게임종료

1. **구현 내용**

기본기능 구현 완료 내용

* 이동 구현
  + 플레이어 이동
  + NPC 이동
  + A\* 길 찾기 구현
* 공격 구현
  + 플레이어 공격
  + NPC 공격
* 장애물 구현
* DataBase 구현
  + Stored Procedure 구현
* LUA Script 구현
* Chatting 기능 구현
* 전투 메시지 구현
* ID 로그인 구현, 로그인 실패 시 처리 구현
* 경험치 및 경험치에 따른 레벨 구현
* 5초마다 HP 회복(공격당할 시) 구현
* 플레이어 죽음 및 부활 구현
* HP, Level, EXP 및 채팅 메시지 UI 구현
* 몬스터 Type(Peace, Agro), 이동(고정, 이동) 구현
* 몬스터 부활 구현

추가기능 구현 완료 내용

* 스크립트를 사용한 몬스터 배치
* 즉시 소모 아이템(HP) 구현
* 범위 공격 구현

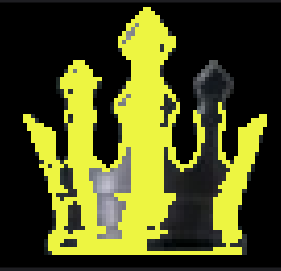
1. **게임 구성 소개**

* 게임 화면



좌측 상단에는 플레이어의 기본정보(현재 좌표, HP, 레벨, 경험치, ID)가 표시됩니다. 맵 하단에는 전투 메시지와 회복 알림, 채팅이 보여집니다. 몬스터 마다 자신의 HP와 레벨, 이름이 표시됩니다. 초록색 네모는 장애물이고 하트 아이콘은 플레이어가 먹을 수 있는 HP 회복 아이템입니다.

* 게임 오브젝트 소개

 플레이어 오브젝트

 다른 플레이어 오브젝트



고정되어 있고 플레이어에게 공격받으면 공격하는 몬스터 오브젝트



원래 위치 20\*20 공간을 자유롭게 이동하면서 플레이어에게 공격받으면 공격하는 몬스터 오브젝트

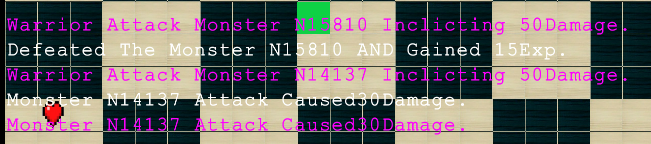
원래 위치 20\*20 공간을 자유롭게 이동하면서 플레이어가 10\*10 근처에 나타날 시 플레이어를 쫓아가 공격하는 몬스터 오브젝트

 장애물 오브젝트



아이템(HP 50 회복) 오브젝트

* 전투 메시지 및 회복 메시지 채팅



플레이어가 몬스터를 공격할 때 뜨는 메시지

“Warrior Attack Monster (몬스터 이름) Inflicting (데미지)Damage.”

몬스터가 플레이어를 공격할 때 뜨는 메시지

“Monster (몬스터 이름) Attack Caused (데미지)Damage.”

플레이어가 몬스터를 죽였을 때 뜨는 메시지

“Defeated The Monster (몬스터 이름) And Gained (획득한 EXP)Exp.”

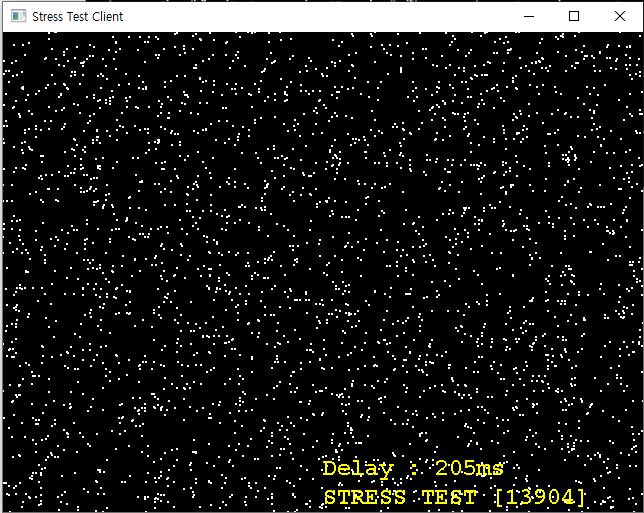
플레이어가 HP를 회복할 때 뜨는 메시지

“Warrior Recovered (회복한 HP)HP.”

플레이어 채팅 기능

“[Player: (플레이어 ID) ]=> (채팅 내용)”

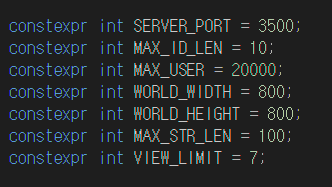
1. **최대 동시 접속**



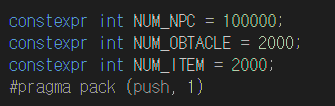
NPC 100’000개를 기준으로 위와 같은 동시 접속을 보여주고 있습니다. DB와 연동, 스크립트 구현, 모든 기본 기능 구현, 추가 기능 구현한 상태로 노트북에서 사양을 측정하였습니다. 현재 최적화 기법으로는 섹터링을 사용하였습니다. 4섹터를 나누어 플레이어와 NPC를 나누었습니다. 이전에 16섹터를 만들어 관리하였으나 4섹터와 큰 차이가 없고 현재 동시 접속에서 부하가 크지 않아 4섹터를 유지하였습니다.

1. **프로토콜**

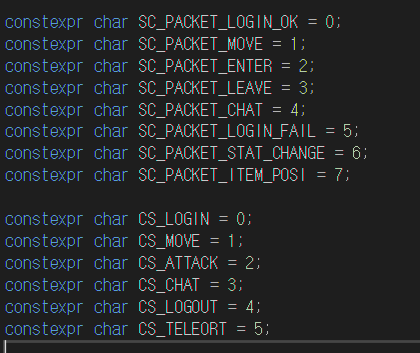
* 공용 프로토콜



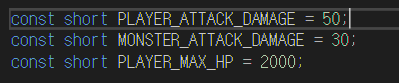
서버 포트 번호 및 최대 유저 접속 수, 맵의 크기와 시야 반지름을 정의합니다.

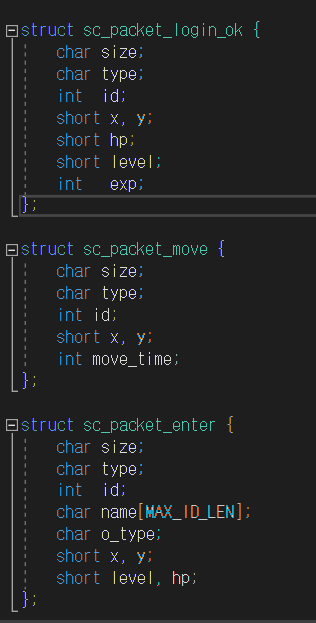
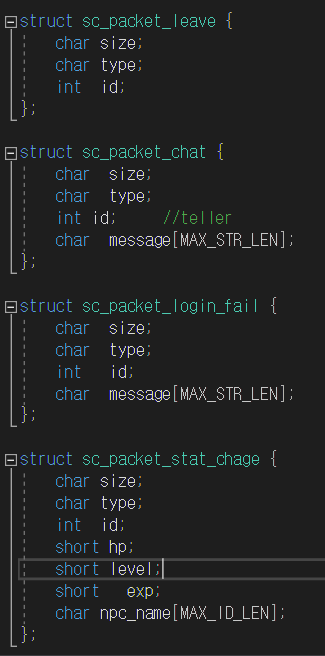


NPC의 수와 장애물 그리고 아이템의 수를 정의합니다.

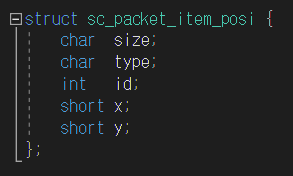


서버와 클라이언트 패킷 전송 간의 패킷 타입을 정의합니다. 기본 프로토콜 이외에 SC\_PACKET\_ITEM\_POSI는 플레이어에게 아이템을 먹은 것을 알려주고 아이템의 바뀐 좌표를 전송하는 패킷의 타입입니다.

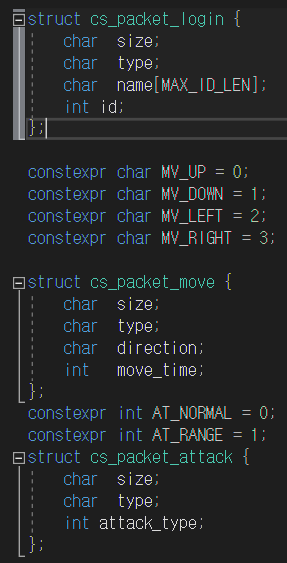


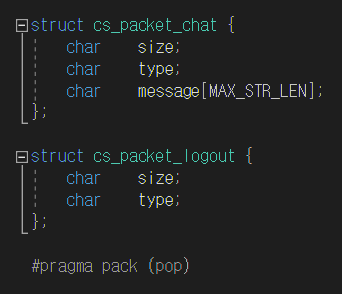
플레이어의 공격력, 몬스터의 공격력, 플레이어의 기본 HP를 정의합니다.

기본 정해진 프로토콜 패킷을 사용하였고 sc\_packet\_state\_change 구조체에서 npc\_name을 추가하였습니다. 플레이어가 공격을 받거나 몬스터를 죽였을 시 해당 몬스터의 이름을 전투 메시지로 표시하기 위함입니다.



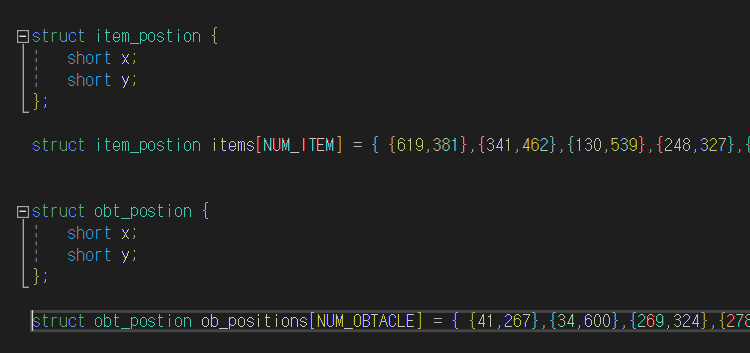
플레이어가 아이템을 먹은 것을 알리기 위해 전송하는 패킷입니다. 아이템의 바뀐 위치 정보도 함께 저장하여 보냅니다.





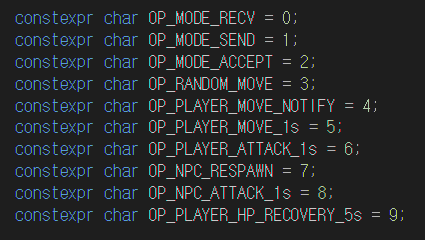
클라이언트에서 서버로 보내는 패킷들입니다. cs\_packet\_attack에는 int 형 변수로 공격 타입을 보내는 것을 추가하였습니다.

AT\_NOMAL은 기본 4방향 공격이고 AT\_RANGE는 8방향 범위 공격입니다.



장애물과 아이템의 위치 배열을 서버와 클라이언트가 공유합니다.

* Server 프로토콜



서버에서 발생하는 이벤트 ID를 정의합니다. 해당 ID를 사용하여 타이머에 이벤트를 등록하고 정해진 시간에 타이머 큐에서 꺼내어 worker\_thread로 보냅니다.

1. **자료구조**

* Server



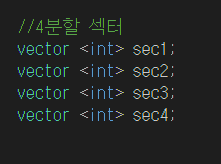
플레이어와 NPC의 정보가 저장되는 있는 배열입니다.

0부터 MAX\_USER까지 플레이어의 정보가 저장되고

MAX\_USER부터 NUM\_NPC까지 NPC(몬스터)의 정보가 저장됩니다.



타이머가 발생하는 이벤트들을 저장하는 큐입니다. 설정한 타이머가 되면 큐에서 꺼내어 이벤트 처리를 합니다.



최적화를 위하여 섹터를 나누고 각 섹터마다 벡터 자료구조를 활용하였습니다. 플레이어와 NPC가 이동하면서 해당 섹터에 들어가게 되고 충돌처리, 시야처리를 플레이어가 있는 섹터에서 실행하였습니다.





장애물과 아이템의 배열입니다.

장애물의 경우 초기 배열 그대로 게임이 끝날 때까지 유지되고

아이템의 경우 플레이어가 아이템을 먹을 경우 위치를 수정하여 다시 배열에 저장합니다.

* Client



플레이어를 제외한 모든 NPC를 저장하는 MAP 자료구조입니다.

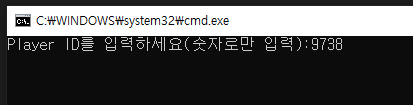
NPC의 ID를 키 값으로 저장하고 NPC의 속성 정보를 OBJECT 클래스에 저장하여 MAP으로 관리합니다.



전투 메시지와 채팅을 담는 채팅 큐입니다. 최대 5개의 메시지를 저장하고 추가되는 메시지는 push를 하고 pop을 하여 최근 5개의 메시지를 담고 있습니다.

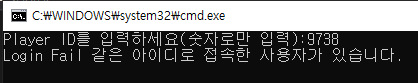
1. **게임 흐름**

* 게임 로그인



클라이언트에서 아이디(숫자)를 입력 받아 게임에 접속합니다.

동일한 아이디로 접속한 플레이어가 없을 경우 게임에 접속하고

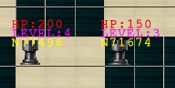


이미 같은 아이디로 접속되어 있는 경우 Login Fail을 알리고 해당 소켓을 종료시킵니다.

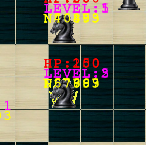


성공적으로 로그인 한 경우 게임에 접속하여 게임을 플레이 합니다.

* 몬스터 소개



몬스터는 레벨에 따라 서로 다른 HP를 가지고 있습니다. 또한 몬스터의 이동 여부와 공격 타입에 따라 플레이어 시야에 들어왔을 때 가만히 있거나 플레이어를 쫓아와 공격합니다.



플레이어를 쫓아와 공격하는 몬스터입니다. 몬스터 시야에 플레이어가 들어오면 플레이어를 쫓아가 공격하고 몬스터 초기위치에서 설정된 거리를 벗어나면 더 이상 쫓아가지 않습니다.



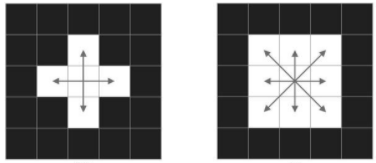
몬스터는 이동 시에 A\* 알고리즘을 이용하여 장애물을 피하면서 플레이어에게 접근합니다.

몬스터는 플레이어한테 공격받아 죽은 경우 30초 후에 초기 위치에서 부활합니다.

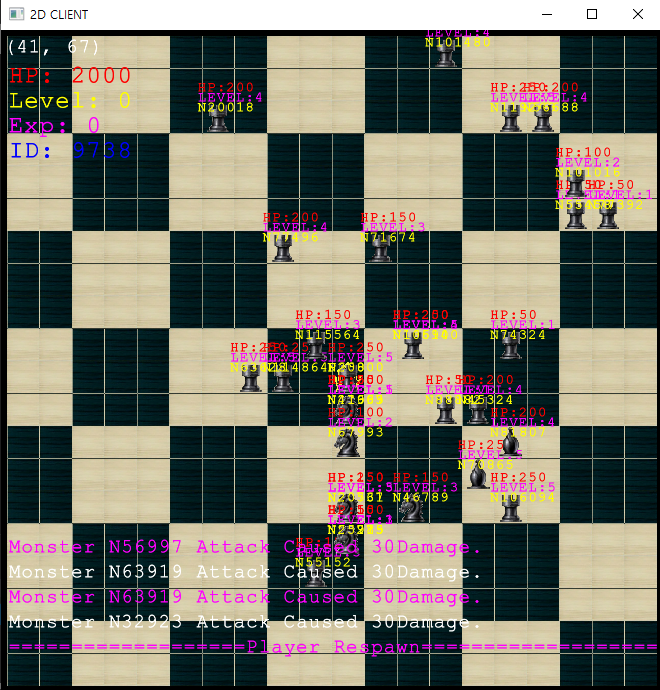
* 플레이어 소개



플레이어는 1초에 한 번 움직이고 1초에 한번 공격할 수 있습니다.



플레이어는 플레이어 현위치를 포함한 4방향 또는 대각선을 포함한 8방향 공격을 할 수 있습니다.





몬스터에게 공격을 받아 죽은 플레이어는 죽은 뒤 경험치의 절반이 깎인 사태로 초기 위치에서 부활합니다.

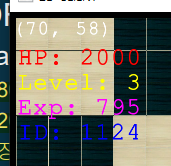


플레이어 HP가 MAX가 아닐 경우에는 5초 마다 현재 HP의 10%씩 회복합니다.

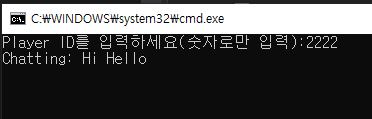
플레이어가 HP 회복 아이템을 먹은 경우 50HP를 회복합니다.

아이템을 먹은 경우 MAX HP 이상으로 HP를 얻을 수 있으며, 플레이어에게 공격 받을 시 MAX HP 이하로 떨어져야만 회복할 수 있습니다.



몬스터를 죽인 경우 몬스터의 레벨과 이동 타입, 공격 타입에 따라 다른 경험치를 얻게 됩니다. 플레이어의 레벨은 플레이어의 경험치에 따라 바뀌게 됩니다.

* 채팅



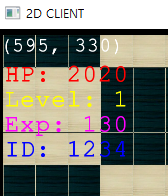
엔터를 누르면 콘솔창에서 채팅을 입력할 수 있습니다.

보낼 채팅 메시지를 입력하고 엔터를 누르면 채팅이 주변 플레이어들에게 전송됩니다.



<채팅 기능>

* 재접속

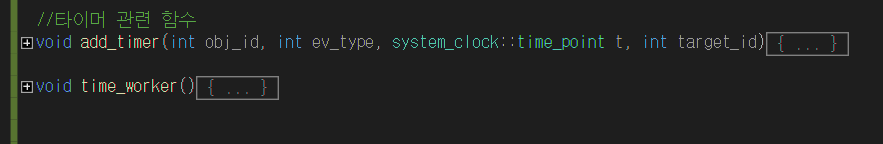


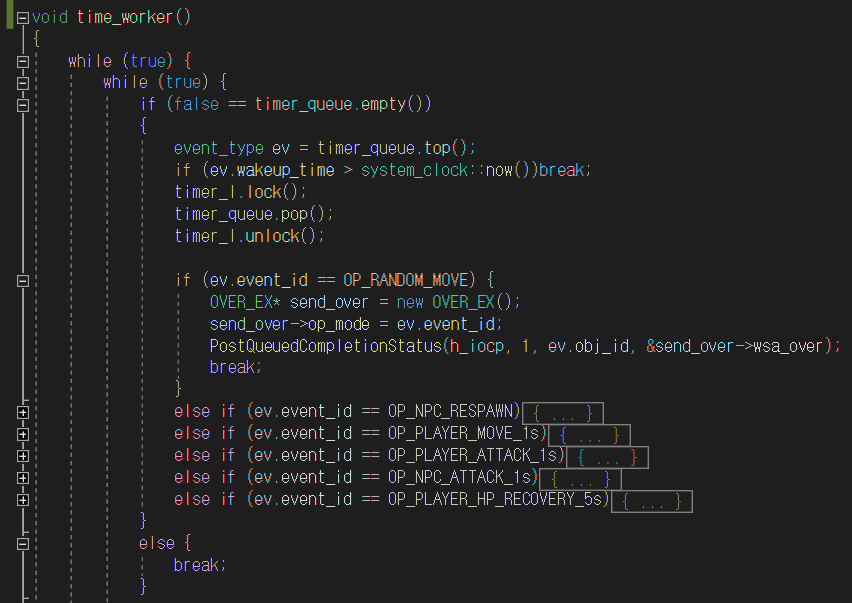
게임종료 후 해당 ID로 다시 접속을 한 경우에도 DB에 해당 플레이어의 HP, 좌표, 레벨, 경험치 등의 정보가 기록되어 이전 기록을 유지합니다.

1. **알고리즘**

서버에는 크게 8개 종류의 함수들이 있습니다.

1. 타이머 관련 함수



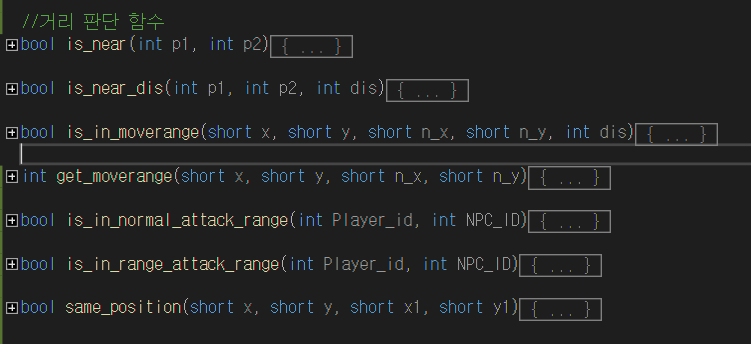


time\_worker 함수는 서버 시작과 동시에 생성되는 스레드 함수입니다.

time\_worker 함수는 타이머 큐에 등록된 이벤트가 대기 시간을 마치고 pop 되면 등록된 이벤트 ID에 따라 처리하며 PostQueuedCompletionStatus를

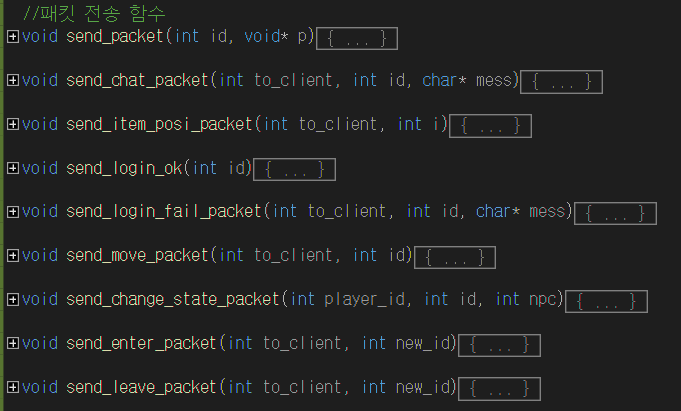
호출하여 worker\_thread로 처리를 합니다.

1. 거리 판단 함수



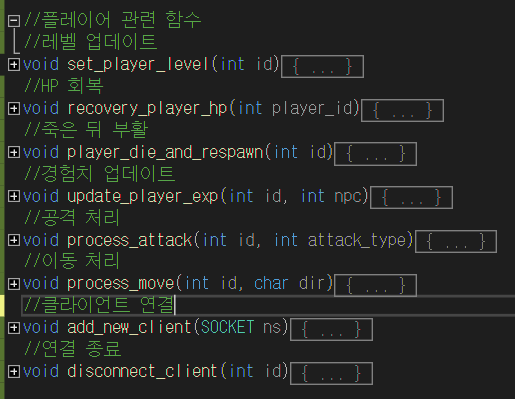
플레이어와 몬스터의 거리를 판단하여 공격 또는 이동처리를 위한 함수입니다.

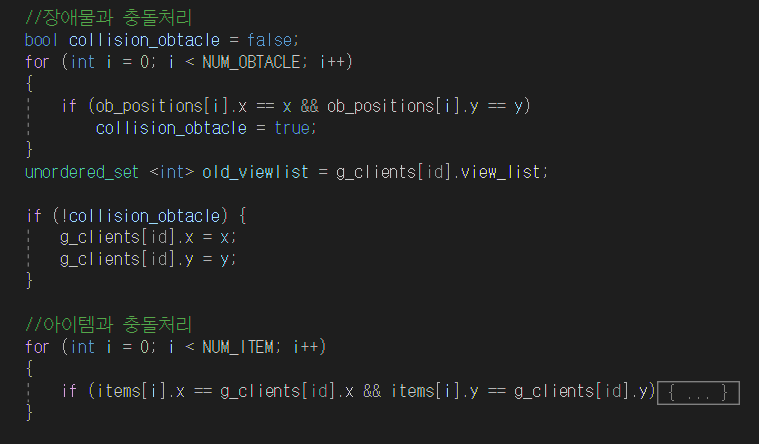
1. 패킷 전송 함수



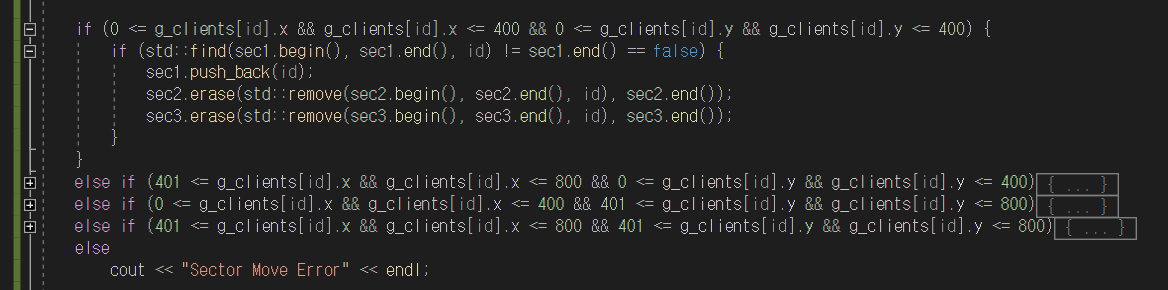
서버에서 클라이언트로 데이터를 전송하기 위한 함수들입니다.

1. 플레이어 관련 함수

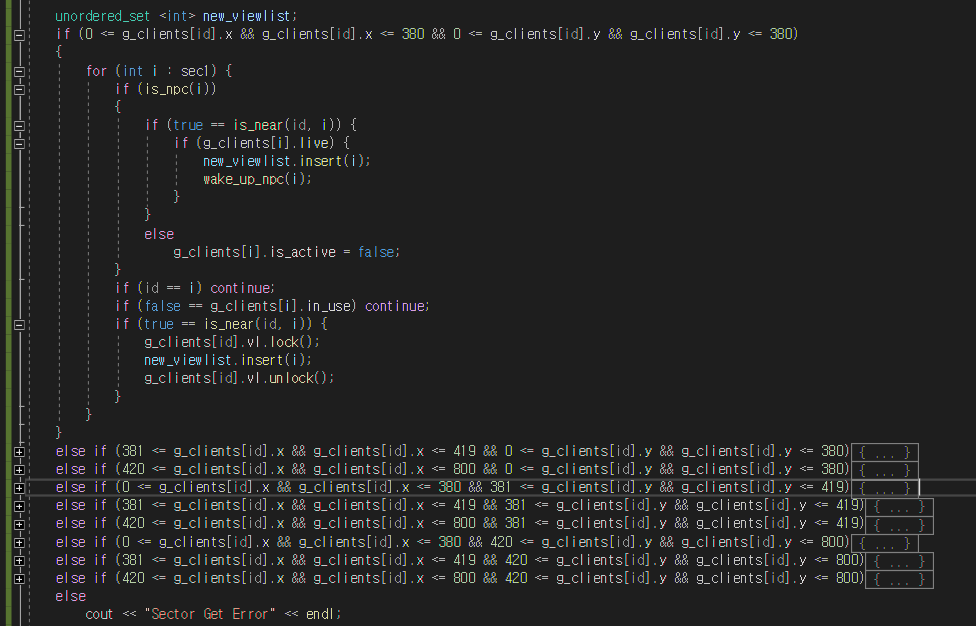


process\_move 함수에서

플레이어는 입력받은 방향 처리를 하며 이동하면서 장애물 및 아이템과 충돌처리를 하면서 이동합니다.



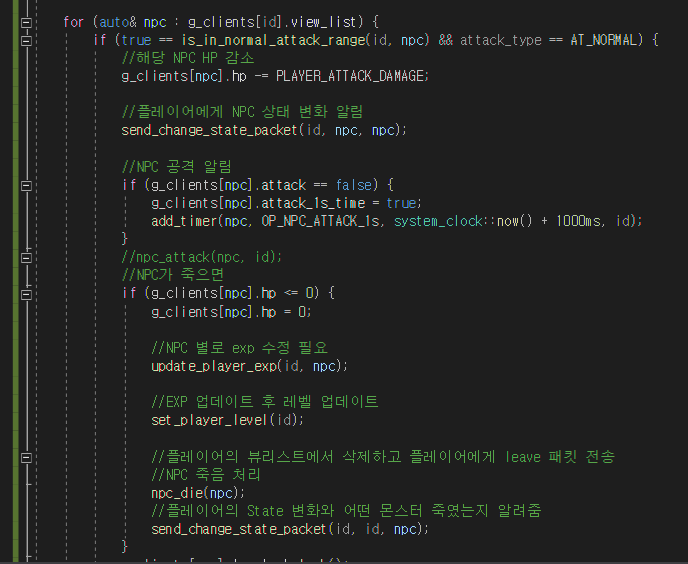
이동후에는 변경된 좌표를 섹터에 맞게 저장하고 다른 섹터에서 삭제합니다.



해당 클라이언트 주변에 있는 NPC와 다른 플레이어를 view\_list에 저장합니다. 섹터별로 나누어 해당 섹터에 있는 NPC와 플레이어들만 검색하여 최적화 시켰습니다.



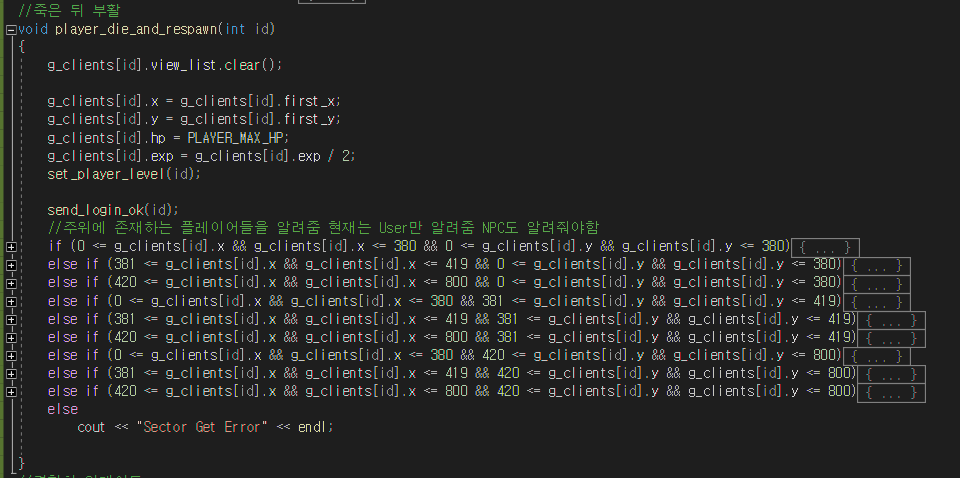
함수 마지막에 다시 타이머를 설정하고 이벤트를 할당하여 타이머에 추가하여 1초 뒤 다시 움직임을 받아 처리하도록 하였습니다.



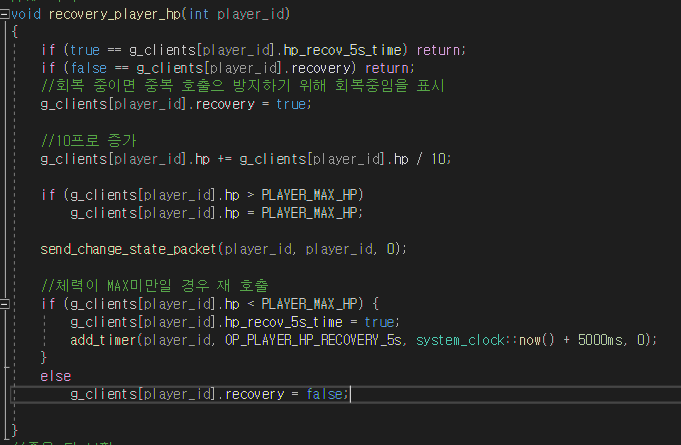
플레이어가 공격한 경우 공격 타입에 따라 범위에 있는 몬스터의 HP를 감소시키고 몬스터의 HP변화를 플레이어에게 알린 뒤 공격받은 몬스터에게 플레이를 공격하라는 이벤트이 타이머를 추가합니다. 몬스터의 HP가 0이되면 플레이어의 경험치를 변화시키고 몬스터의 처리를 한 뒤 변화된 데이터를 플레이어에게 전송합니다.



move함수와 마찬가지로 1초 후에 공격 입력을 받을 수 있도록 타이머를 등록합니다.

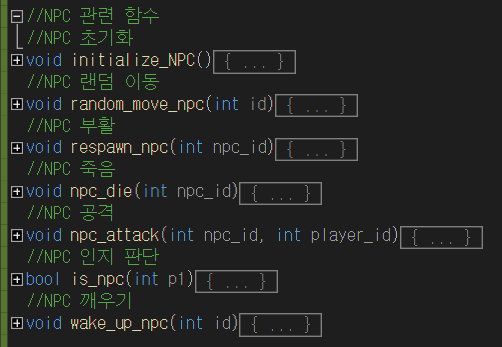
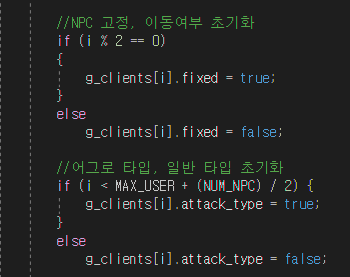


플레이어가 몬스터의 공격으로 HP가 0이 된 경우 해당 플레이어의 view\_list를 초기화 하고 초기 위치 설정 및 HP를 복구, 경험치를 반으로 줄입니다. 이후 login\_ok 패킷을 보내면 플레이어의 전체 데이터를 보내줄 수 있습니다. 이후 변화된 위치에 따라 시야처리를 합니다.

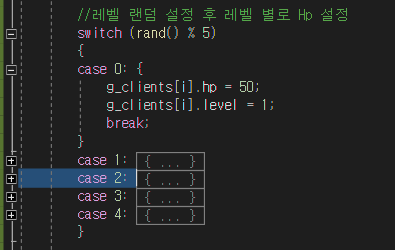


체력회복 함수가 호출된 경우 현재 HP의 10%를 회복시키고 변화한 것을 플레이어에게 알려줍니다. HP가 MAX로 찬 경우 더 이상 회복을 하지 않습니다.

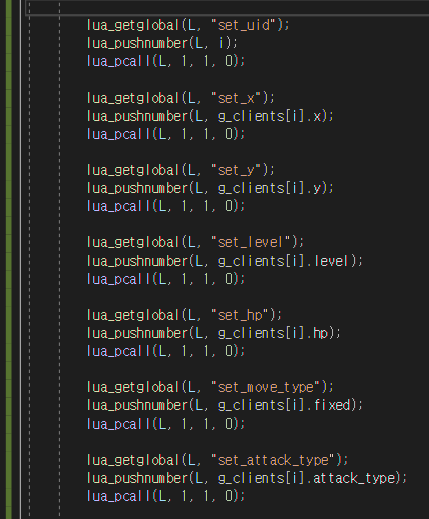
1. NPC(몬스터) 관련 함수



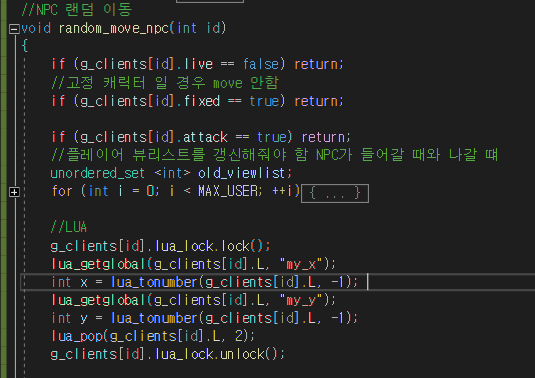
몬스터를 초기화 하면서 몬스터의 이동 타입과 공격 타입을 ID에 따라 골고루 분산시켜 초기화 합니다.



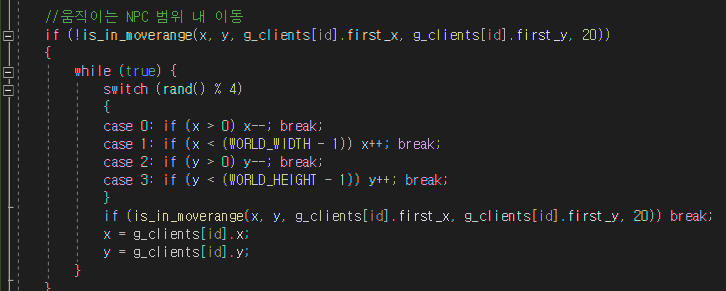
레벨 별로 HP를 다르게 초기화 합니다.



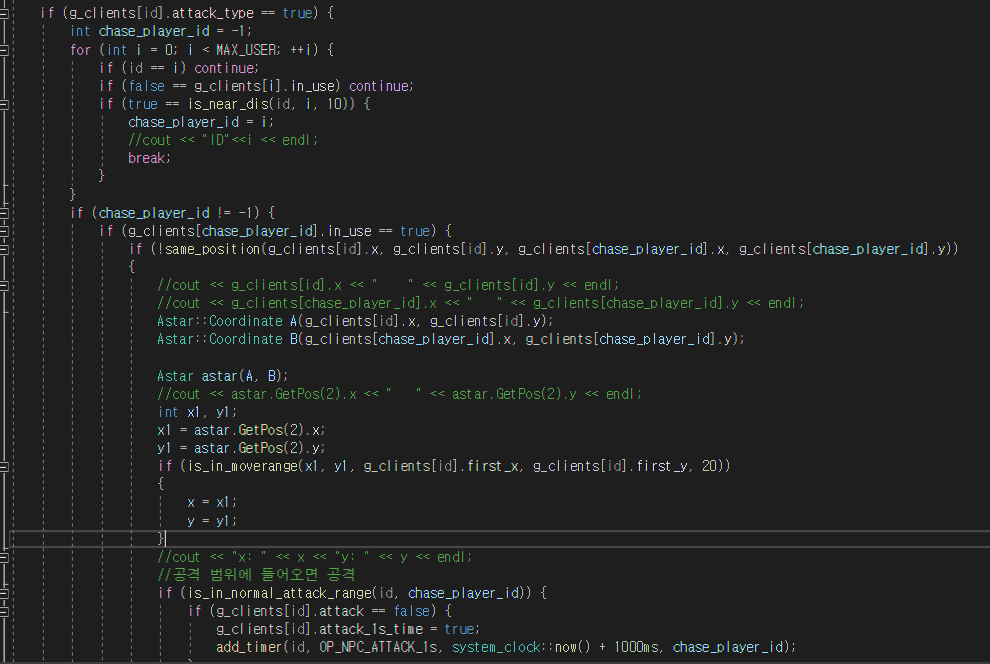
초기화된 몬스터의 정보를 LUA 스크립트로 저장하여 관리하도록 합니다.



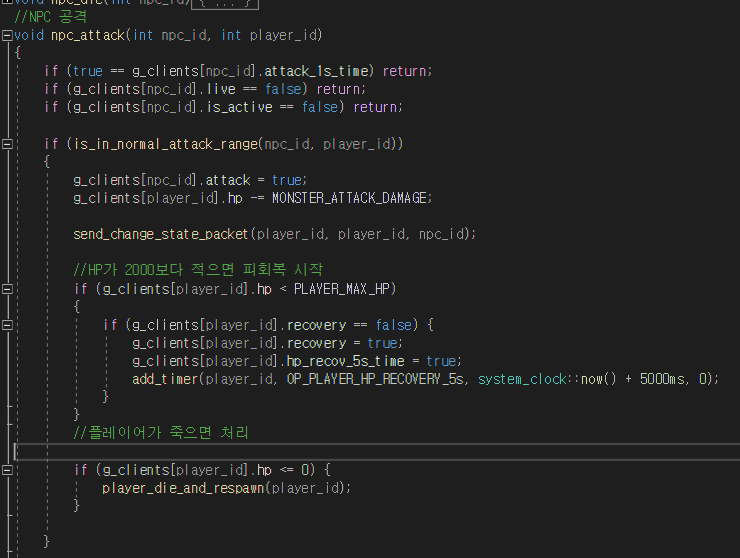
이동 요청을 받은 몬스터가 고정이거나 공격 중인 경우 이동을 거부하고 이동할 때 좌표는 스크립트에서 받아옵니다.



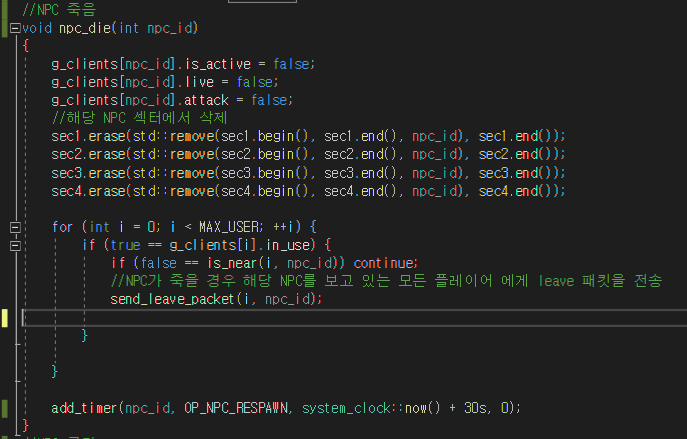
몬스터는 20\*20범위 안에서 이동을 하도록 합니다.



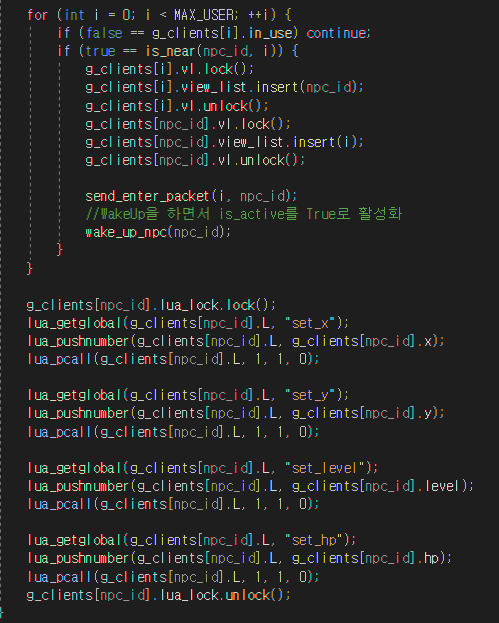
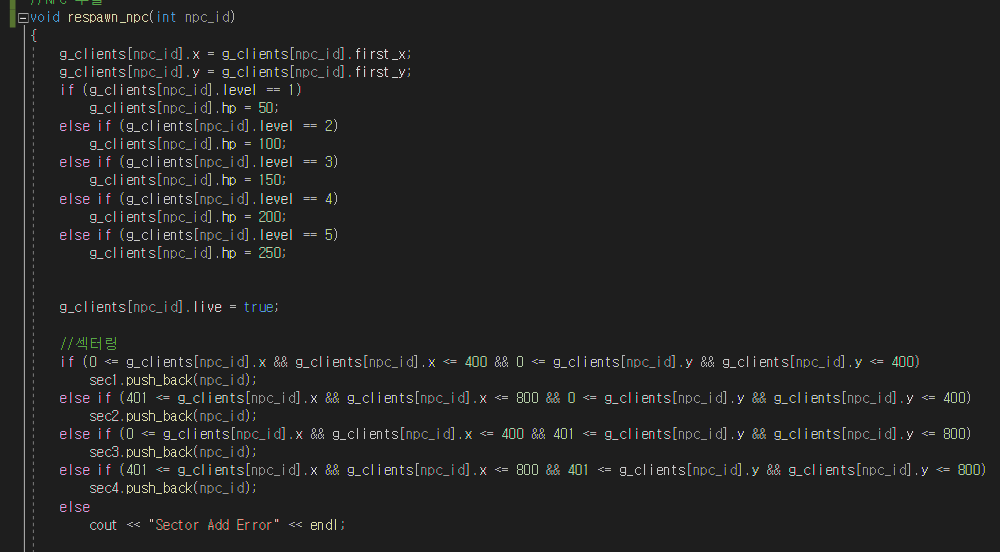
공격 타입이 Agro 인 몬스터는 플레이어가 10\*10 시야에 들어오면 A\* 알고리즘을 사용하여 플레이어를 쫓아간 뒤 공격범위 안에 들어오면 플레이어를 공격합니다. 이후 이동처리는 플레이어 함수와 비슷하게 섹터링을 활용하여 동작합니다.



플레이어와 비슷하게 공격범위 안에 들어올 경우 플레이어의 HP를 감소시키고 플레이어의 HP가 줄어들면 HP 회복 이벤트를 실행합니다. 또한 플레이어의 HP가 0이 된 경우 부활처리를 합니다.

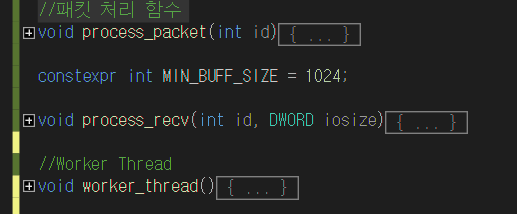


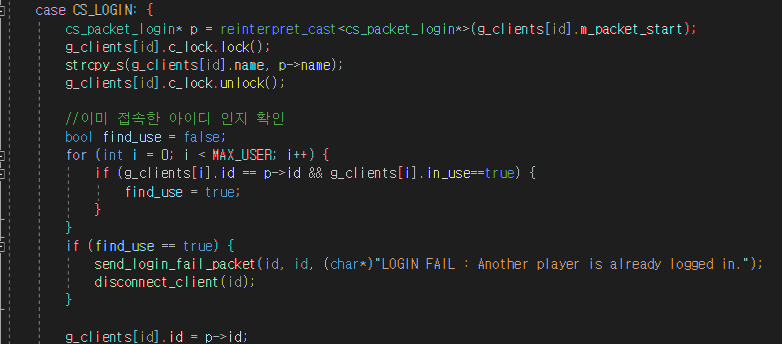
몬스터가 죽으면 몬스터의 is\_active, live, attack을 false로 바꾸고 모든 섹터에서 몬스터를 삭제합니다. 또한 주변 플레이어에게 몬스터가 사라짐을 알립니다.



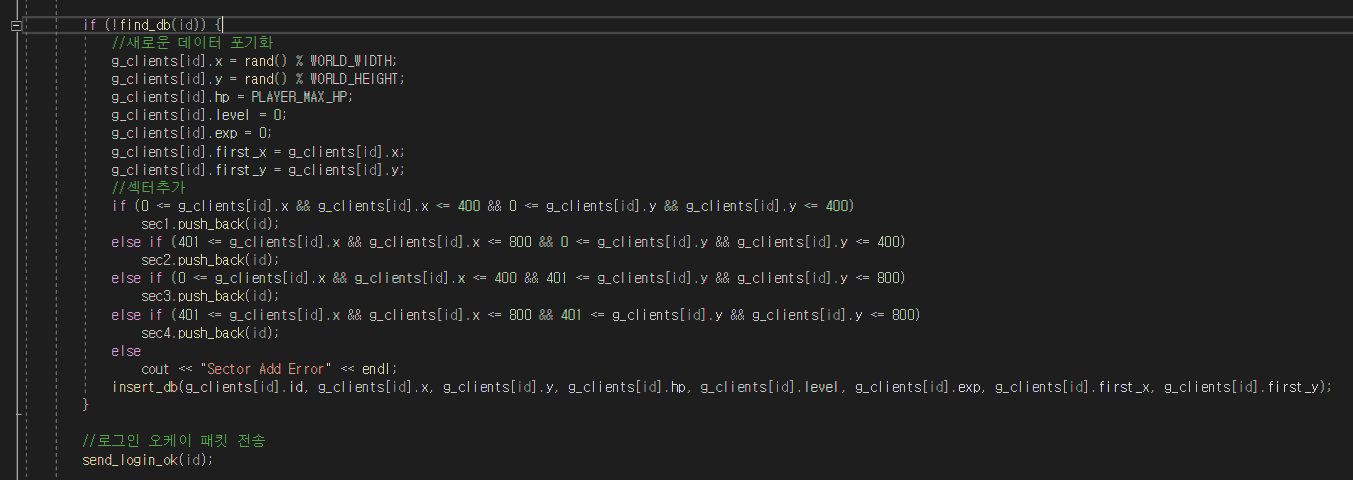
몬스터가 부활하면 초기 위치로 설정한 뒤 레벨에 맞게 HP를 초기화 합니다. 이후 좌표에 따라 섹터를 나누어 저장한 뒤 근처 플레이어에게 enter\_packet을 보냅니다. 이후 몬스터를 깨워 이동을 주위에 플레이어가 있으면 이동시킵니다. 마지막으로 변경된 좌표를 스크립트에 저장합니다.

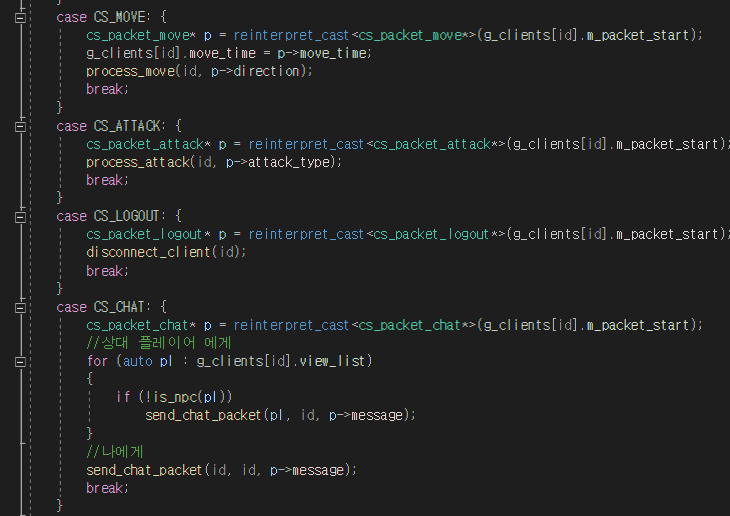
1. 패킷 처리 함수 및 Worker\_thread 함수



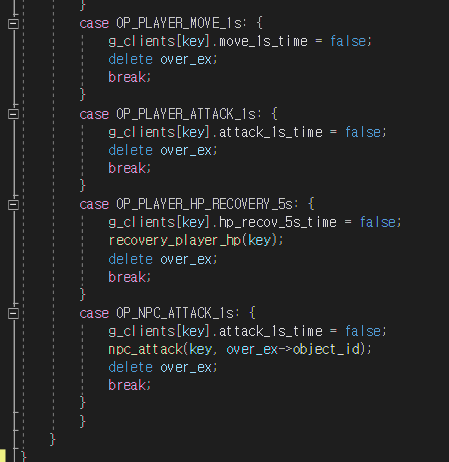
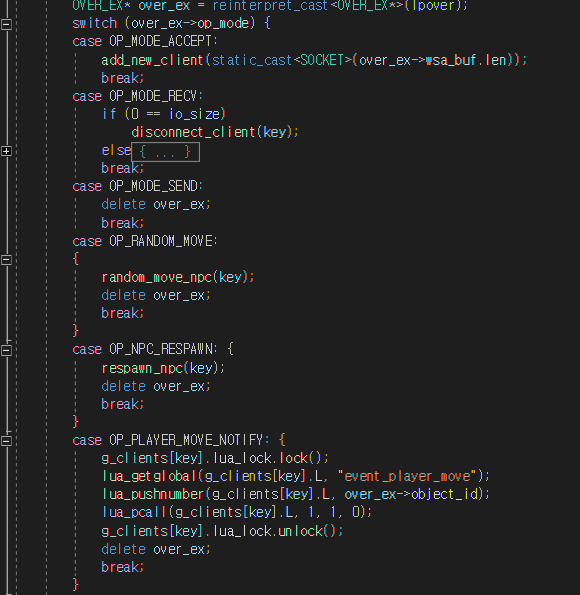


클라이언트가 로그인 패킷을 보내면 아이디를 확인하여 사용중인 플레이어 인지 확인하고 사용중인 경우 로그인 실패를 알리고 소켓 연결을 종료합니다.



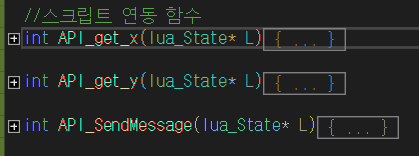
데이터베이스에 해당 ID를 검색하여 데이터가 있는 경우 그 데이터를 그대로 받아 사용하고 없을 경우 새로 초기화 한 뒤 섹터에 저장하고 login\_ok를 알립니다. 

클라이언트의 이동, 공격, 로그아웃은 해당 함수를 호출하여 처리하고 채팅의 경우 주변에 있는 플레이어들에게 채팅을 전송해줍니다.

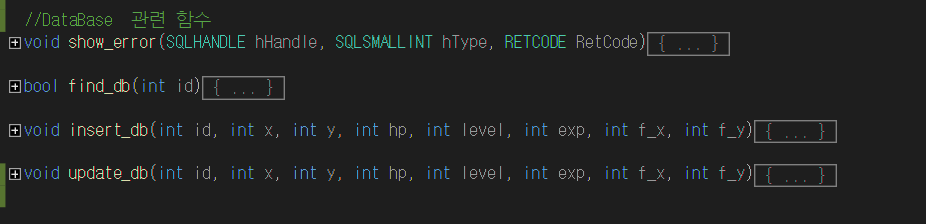


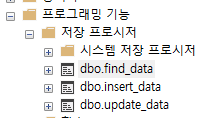
worker\_thread에서 PostQueuedCompletionStatus로 들어온 이벤트를 해당 함수를 호출하거나 플레이어의 이동 또는 공격의 경우 bool 자료형를 활용해 1초 이후에 처리되도록 합니다. NPC의 공격과 플레이어의 회복 또한 중복호출을 막기위해 bool 자료형을 통해 중복호출을 방지하였습니다.

1. 스크립트 연동 함수

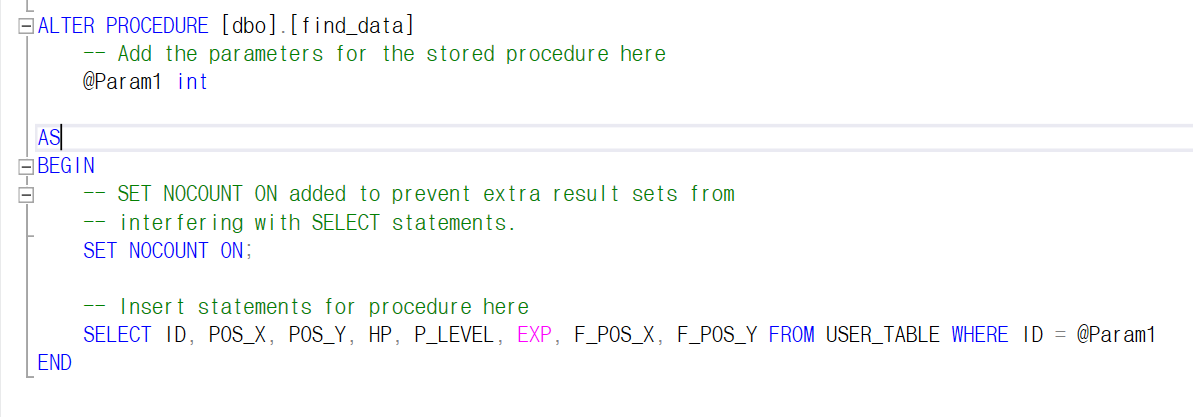


1. 데이터베이스 처리 함수



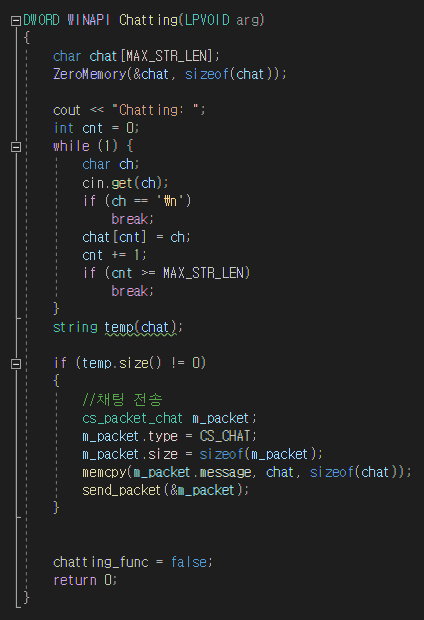
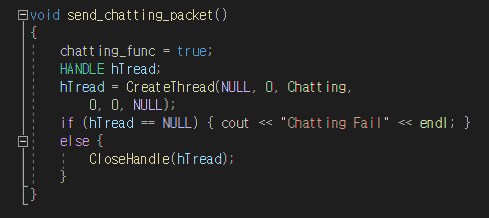






쿼리는 Stored Procedure을 사용하여 실행속도를 높였습니다.

* 클라이언트 채팅 기능



클라이언트에서 채팅을 입력할 때는 새로운 스레드를 생성하여 채팅을 입력 받도록 합니다. 이는 채팅 때문에 게임이 중단되는 것을 막기 위함 입니다. 채팅을 보낸 이후에는 해당 스레드를 반환합니다.