**게임 서버 프로그래밍 숙제 #7**

**DB 연동 보고서**

**게임공학부**

**엔터테인먼트 컴퓨팅과**

**2016184041 차지환.**

1. **구현 방식.**
2. **컴퓨터 사양 및 테스트 설정.**
3. **DB정보 및 결과**

**1 구현방식.**

DB쿼리 문은 stored procedure로 진행하였으며,

DB작업은 Blocking작업이기 때문에 DB작업을 위한 Thread Pool로 넘겨 db 작업용 thread에서 실행하도록 만들었습니다. 각 스레드는 db connection을 가지고 있습니다..

DB작업이 끝나면 DB쓰레드에서 PQCS를 이용해서 서버의 워커 스레드로 작업을 넘겨주어 게임에 반영하도록 하였습니다.

**DB Thread Pool 구성**

DB 조회( select ) Thread \* N개

DB 변경 (update, delete 등 ) 1개

DB 변경 쓰레드는 혹시라도 순서가 바뀔 것을 고려해서 한 개로 놔두고 천천히 처리하도록 Queue를 주었습니다.

**User 로그인시..**

DB에 아이디가 없다면 아이디를 생성.

후 다시 로그인 하여 접속하도록 하였습니다.

**User 좌표 업데이트 주기**

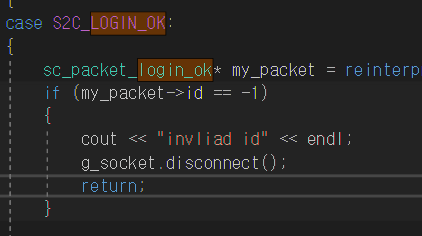
매우 자주하면 DB부하가 클 것 같아 서버에서 판단 시

첫번째 sector가 변경될 때와 접속 종료 시와

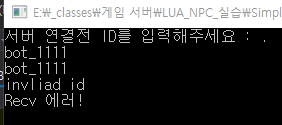
두번째 상시 이동시 업데이트

두가지 경우를 테스트 하기로 했고 했습니다.

**클라이언트 변경 사항.**

****

프로토콜을 추가 해도 되는지 여부가 확실치 않아 서버에서 -1을 보내면 접속이 실패한 것으로 간주하였습니다.



로그인 실패의 경우.

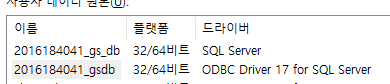
**2.컴퓨터 사양.**

Cpu : Ryzen 3700x ( 8 core 16thread )

Ram : 32기가

SSD : 500 + 250

DSN 이름 : 2016184041\_dbgs

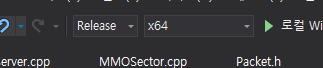


수업에서 사용한 ‘SQL Server’ 드라이버 사용하여 연결 시 테스트는 통과하는데 실제 서버에서 접속 시 time out이 발생하여 ODBC Driver 17 for SQL Server 을 사용하고 있습니다.

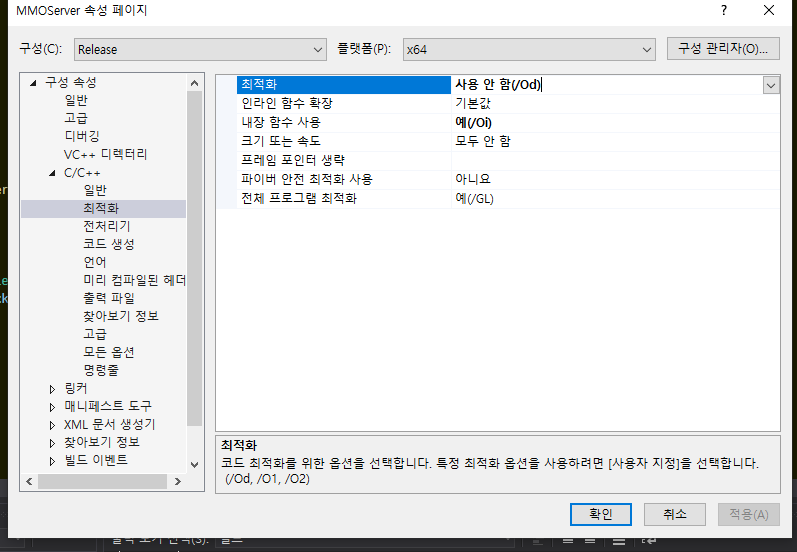
**2.1 필수 Visual Studio 2019 환경.**

**필수!!**

**Release Mode x64**

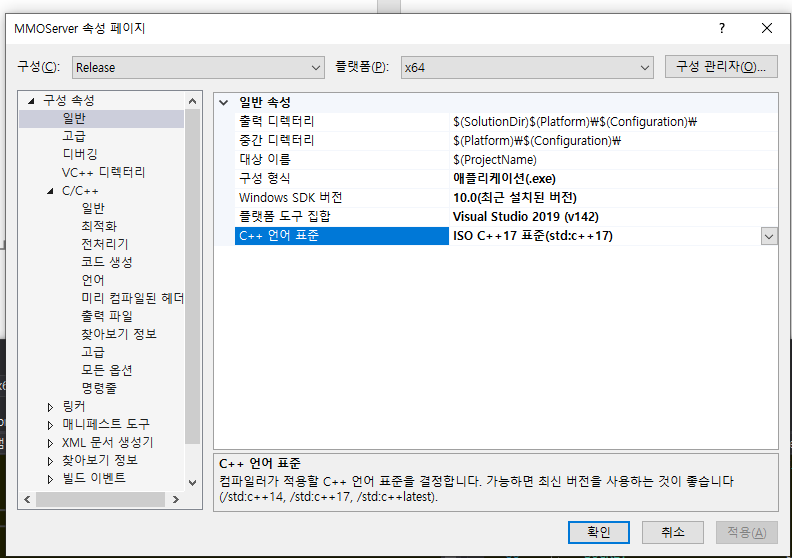
**.**

**구성 속성 -> C / C++ 최적화 -> 최적화 사용안함.**

****

**필수 : 컴파일러 최적화를 꺼주시기 바랍니다.**

**구성 속성 -> 일반 -> C++ 언어 표준 c++ 17 표준.**

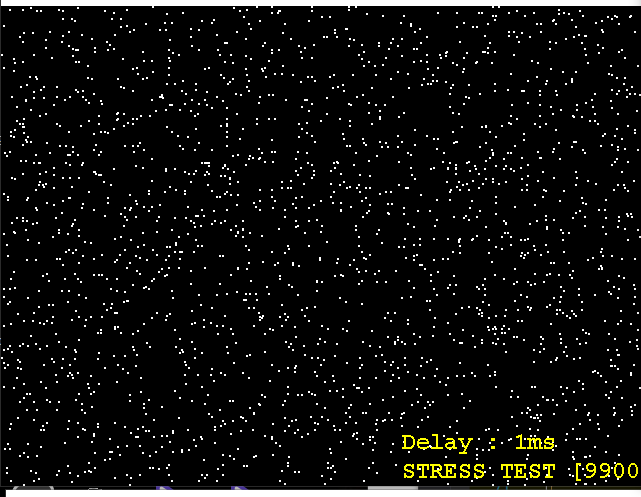
****

**3 .결과**

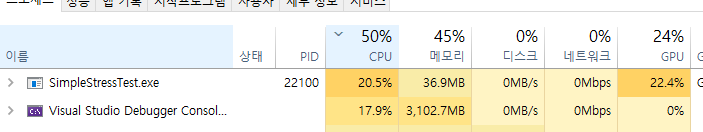
DSN 이름 : 2016184041\_dbgs

NPC 2만명.

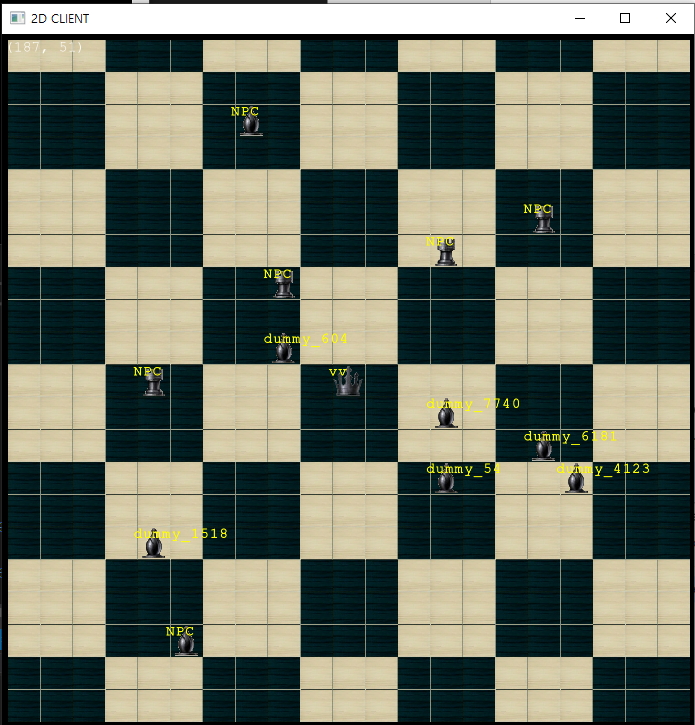
**1번 섹터 이동시와 종료시에만 포지션 업데이트.**

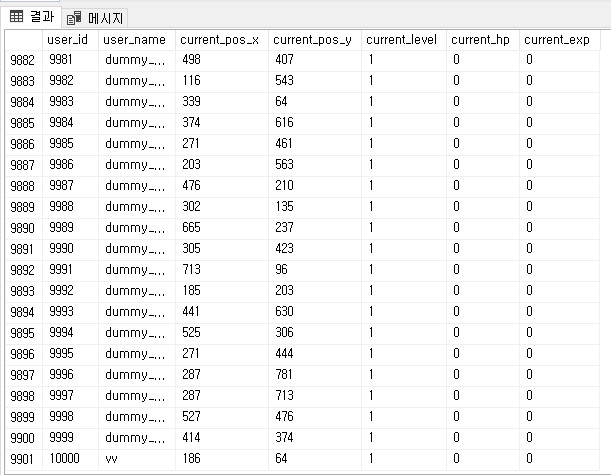
평소처럼 서버에 큰 부하 없이 돌아가고 있습니다.

서버나 클라이언트 DB 점유율

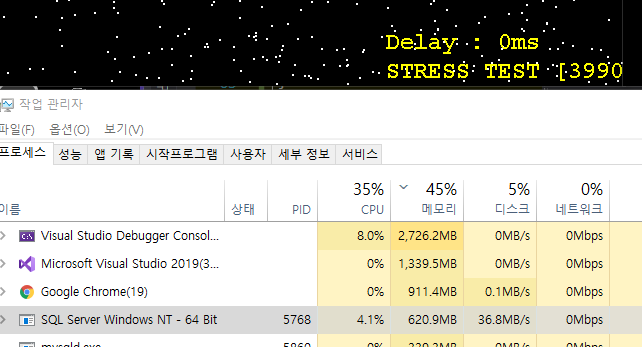


큰 이상 없이 돌아 가고 잇습니다.





2번 구현 이동시마다 좌표 업데이트.



( 2번구현)



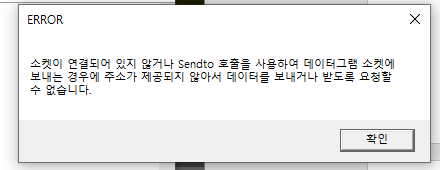


(1번 구현)

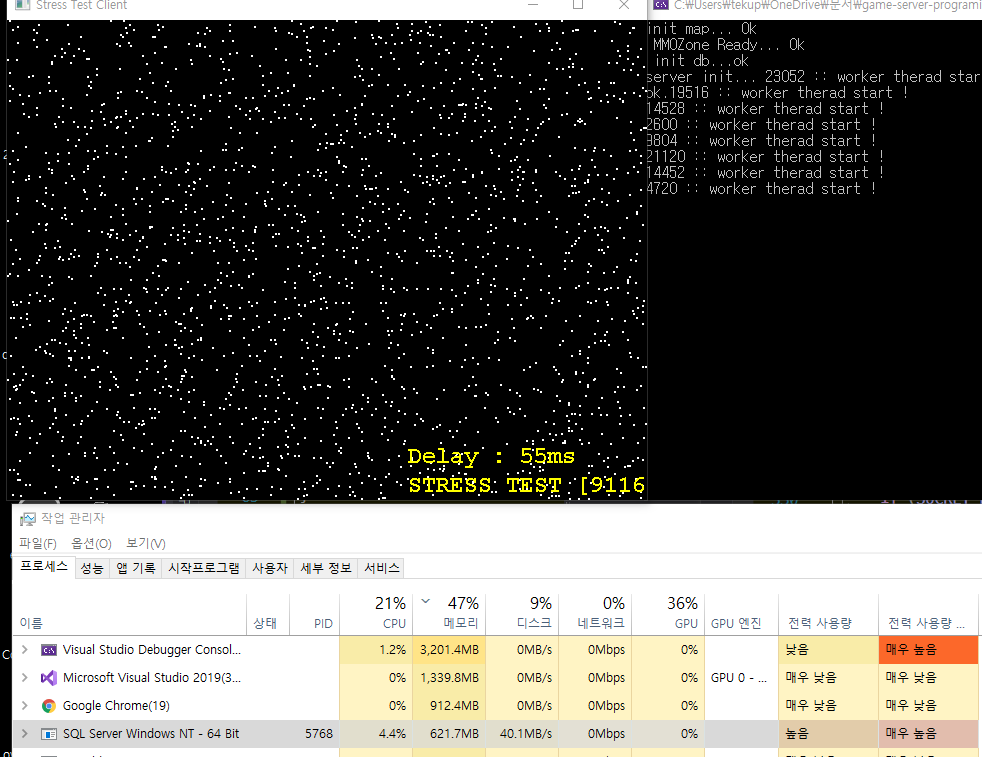
매 이동시마다 좌표 업데이트 시.

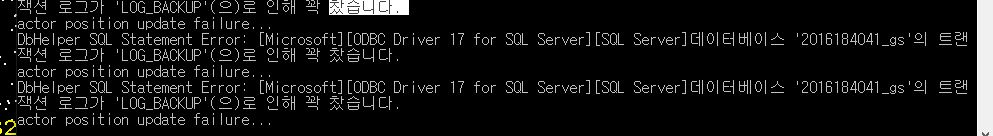
4천명 정도밖에 접속하지 않았지만 1번처럽 업데이트를 최소화 했을 때에 비해 디스크 IO가 3배 정도는 더 사용하고 있었다.

9천명 정도 접속하자



접속을 거부하는 경우가 발생했으며… 알고 보니 이는 DB에러로..

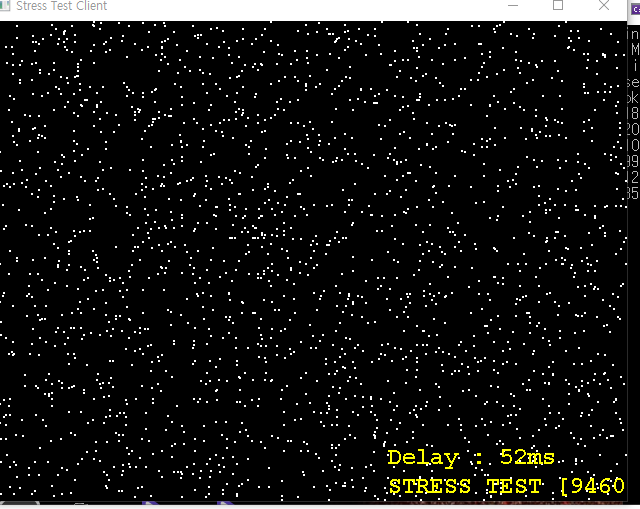




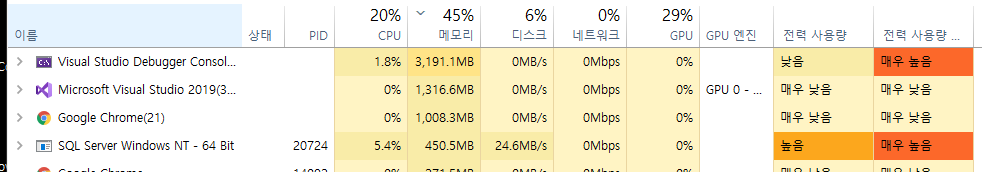
DB 로그가 꽉 찼다는 에러로

트랜잭션 로그를 주기적으로 비우지 않고 계속 저장만 해서 결국 디스크가 꽉차서 생긴 문제였다.

SQL server 설정을 변경해서 로그를 4기가로 제한하고 다시 실행하였다.



DB 로그 문제를 해결했지만 여전히 9천 중반대가 되면 DB쪽이 병목이 되어 갑자기 접속이 느려지고 딜레이가 심해지는 문제가 발생하였다.



현재는 1번 구현 방향으로 하기로 하였다.