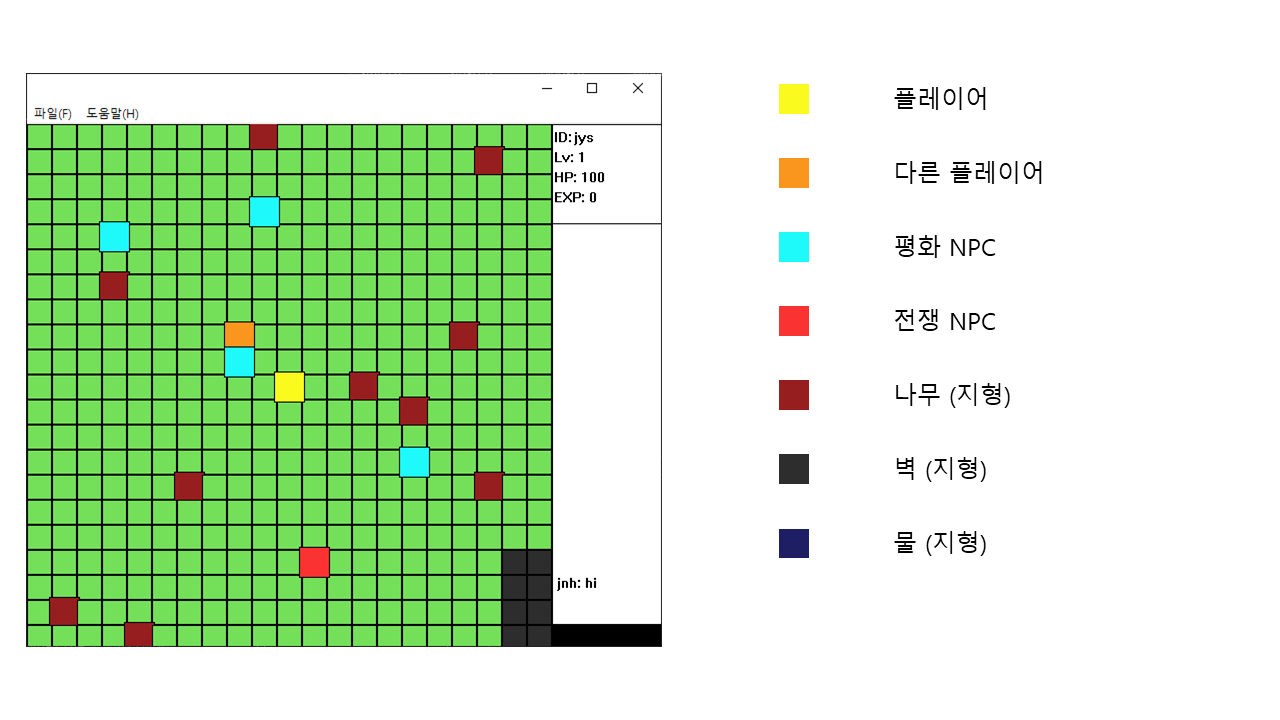
2019-2게임서버프로그래밍 텀프로젝트 설명서

**2015184024 장윤수**

DB사용할 때 use\_DB bool 값을 true로 바꾸고 실행

스트레스 테스트할 때 servertest bool 값을 true로 바꾸고 실행

**[게임소개]**

****

클라이언트 프로그램은 winapi를 이용하여 제작하였습니다.

화면 우측 상단에 플레이어의 스탯이 표시되고 그 아래에는 채팅이 출력된다.

화면 하단에 전투 메시지가 출력됨.

맵 크기는 800X800이고 맵 상에 지형이 존재한다. 나무(갈색), 벽(회색), 물(파랑)은 이동 불가능

플레이어 자신의 캐릭터 -> 노랑색

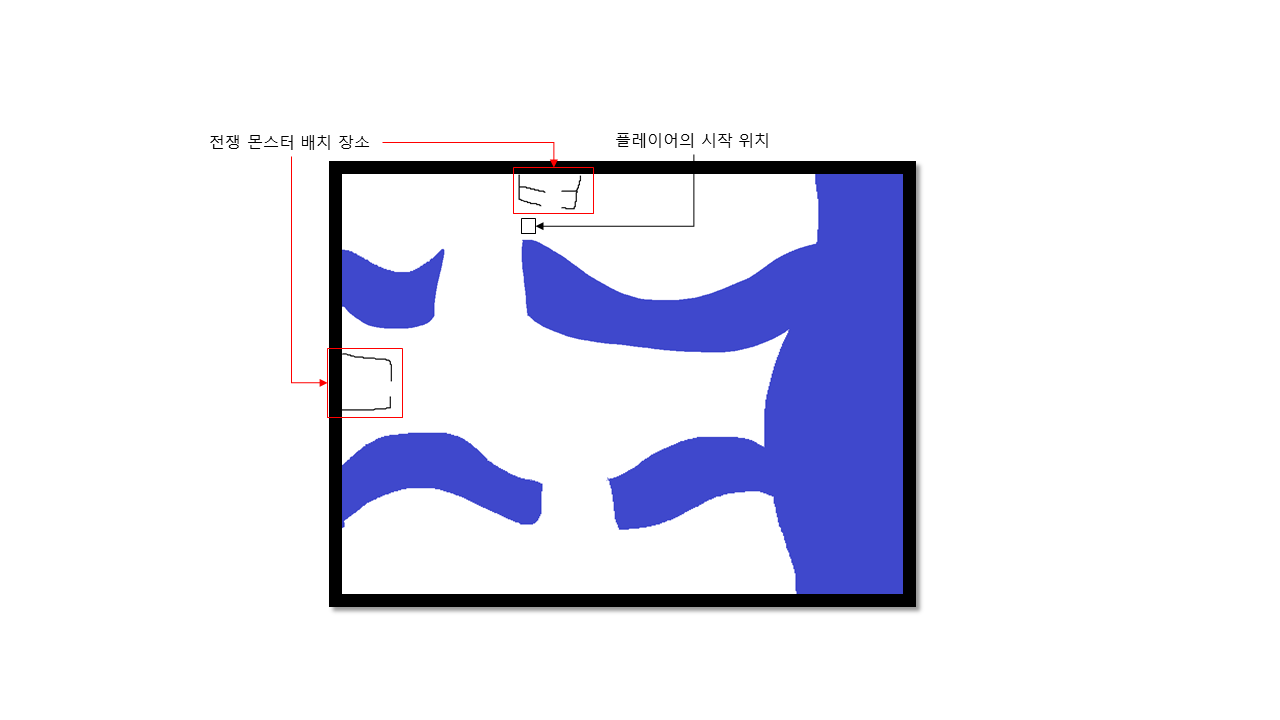
다른 플레이어의 캐릭터 -> 주황색

평화 NPC -> 청록색

전쟁 NPC -> 빨간색

NPC는 평화NPC와 전쟁NPC 두가지 존재 NPC들은 1초에 한 칸 랜덤으로 이동. 평화NPC는 플레이어를 공격하지 않고 플레이어는 모든 NPC를 공격가능. 전쟁NPC는 플레이어가 자신 주변 4X4 영역 안으로 들어오면 어그로가 활성화되어 플레이어를 따라가고 1초에 한번씩 주변 4방향에 공격을 한다.

평화NPC는 맵 전체에 랜덤으로 배치되고 전쟁NPC는 특정 지역에만 배치가 됩니다.

 간략하게 나타낸 월드맵

**[조작법]**

최초 접속시에 DB에 저장 되어있는 id를 입력하고 엔터키를 누르면 서버에 접속 요청

DB에 저장되어있는 접속가능한 id 목록 - [ jys ] , [ jnh ]

Id 가 DB에 존재하지 않을 경우 DB에 새 아이디를 만들어 접속

이미 접속해 있는 아이디 일 경우 접속 불가능

상하좌우 방향키로 플레이어 이동 이동은 0.5초에 한 칸씩 가능

‘a’키를 이용해 공격 가능

스페이스바를 누르면 화면 우측하단에 채팅 입력창이 생긴다. 채팅 입력 후 스페이스 바를 다시 누르면 채팅이 접속중인 모든 플레이어에게 전송됨.

**[프로토콜]**

서버 -> 클라이언트

**SC\_SEND\_ID**: 서버가 클라이언트를 accept했을 때 클라이언트에 id를 부여하기 위해 보냄

**SC\_LOGIN\_OK**: 로그인 시도한 아이디가 DB에 존재할 경우 로그인 성공의 의미로 보냄

**SC\_PUT\_PLAYER**: 모든 플레이어와 NPC의 정보를 보냄

**SC\_REMOVE\_PLAYER**: 플레이어와 NPC를 삭제 해야 할 때 보냄

**SC\_POS**: 플레이어와 NPC의 움직임을 알려주기 위해 보냄

**SC\_CHAT**: 채팅을 보내기 위함

**SC\_STAT\_CHANGE**: 플레이어의 스탯변화를 알려주기 위해 보냄

**SC\_PLAYER\_DEAD**: 플레이어가 죽었을 때 보냄

**SC\_LOGIN\_FAIL**: 클라이언트의 로그인 요청이 실패했을 때 보냄

클라이언트 -> 서버

**CS\_LEFT, CE\_RIGHT, CS\_UP, \_CS\_DOWN**: 플레이어의 이동을 위해 보냄

**CS\_REQUEST\_LOGIN**: 클라이언트가 ID로 접속을 요청하기 위해 보냄

**CS\_ATTACK**: 플레이어가 공격을 하기 위해 보냄

**CS\_CHAT**: 플레이어의 채팅을 보내기 위해 보냄

**[자료구조]**

SOCKETINFO clients[MAX\_OBJECT]: 플레이어와 NPC의 정보를 가지고 있는 배열

vector<DATABASE> vec\_database: DB에서 로드한 데이터들을 저장해놓는 벡터

priority\_queue<EVENT> timer\_queue: 모든 이벤트들을 가지고 있는 우선순위큐

**[게임흐름]**

1. 초기화 – 월드맵 로드, DB로드, NPC생성 후 클라이언트의 접속을 기다린다.
2. 로그인 – 클라이언트가 아이디를 입력하고 로그인 (DB에 있는 아이디면 로그인성공, DB에 없을 경우 새로운 아이디를 DB에 생성하고 로그인, 아이디가 이미 접속 중일 경우 접속 차단)
3. 게임 진행 – 플레이어는 이동하며 NPC몬스터를 사냥한다, 전쟁 NPC에게 죽을 경우 5초뒤에 경험치를 절반으로 깎고 부활한다.
4. 채팅 – 플레이어가 보내는 채팅은 접속 되어있는 모든 플레이어들에게 전송됨
5. 종료 – 접속을 종료하면 게임 내에서 플레이어가 사라지며 정보를 DB에 업데이트하고 소켓을 닫는다.

[알고리즘]

1. 시야처리: 처음 접속시에 플레이어의 위치 시야범위 내에 있는 플레이어들에게 put\_player패킷을 보내고 플레이어가 이동할 때 마다 이동 전 시야 리스트와 이동 후 시야리스트를 만들어 두개의 시야리스트를 비교하며 시야범위 안으로 들어온 플레이어게게 put\_player패킷을 보내고 시야에서 사라진 플레이어에게는 remove\_player패킷을 보내고 계속해서 시야안에 있는 플레이어들에게는 pos패킷을 보내 이동한 위치를 알린다.

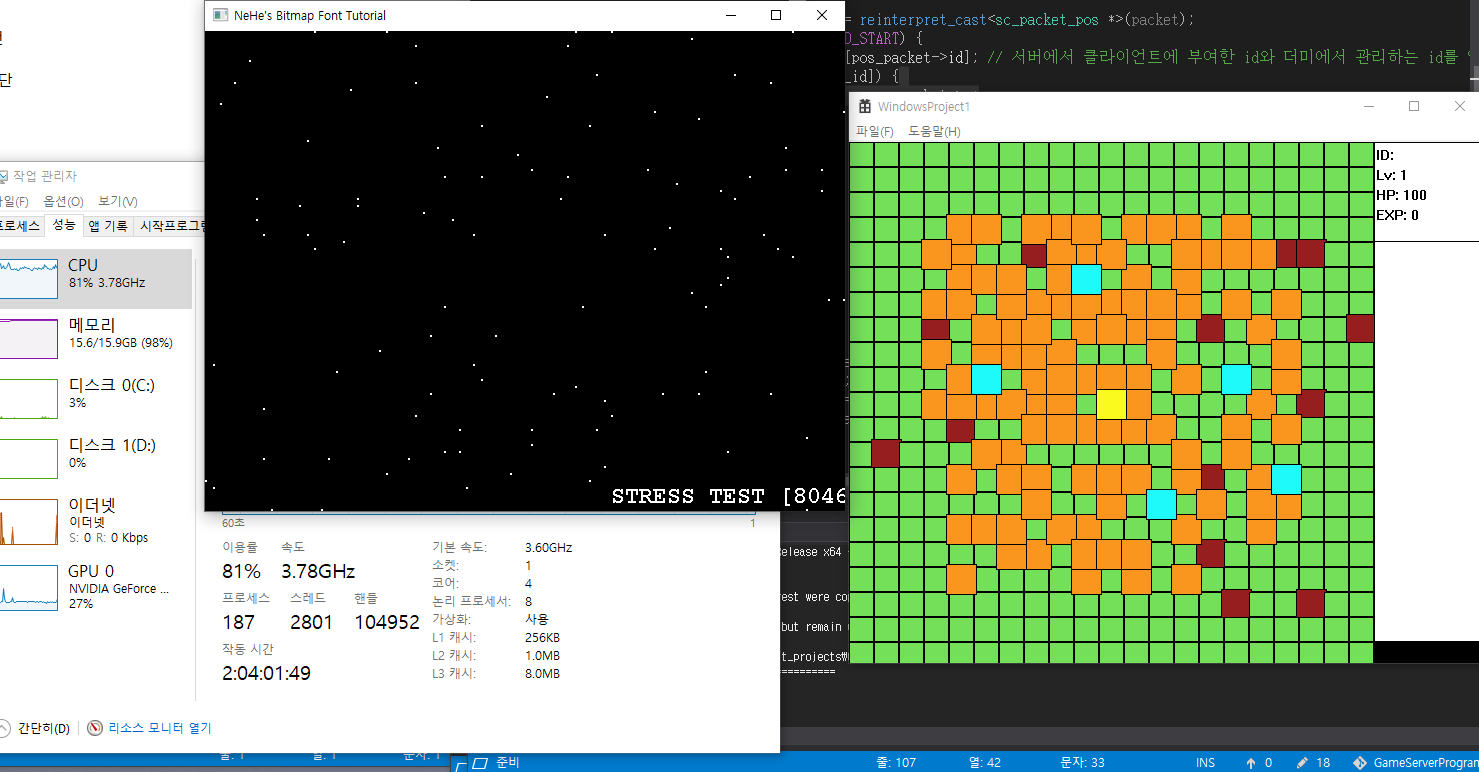
2. NPC 시야처리: NPC는 플레이어보다 더 넓은 시야를 가지고 NPC시야범위 안에 플레이어가 들어오면 active상태로 바뀌어서 랜덤이동을 시작한다. 시야안에 플레이어 한 명도 없다면 active상태를 false로 만들어 움직임을 멈춘다.

3. 전쟁NPC 추적: 전쟁 NPC의 어그로 범위안에 플레이어가 들어오면 플레이어의 위치와 자신의 위치를 X축 Y축으로 나누어서 비교하여 거리 차이가 더 큰 축을 먼저 이동하면서 플레이어에게 1초에 한 칸씩 다가간다.

**[구현목록]**

|  |  |  |
| --- | --- | --- |
| 기능 | 구현 | 미구현 |
| 이동 | 플레이어는 0.5초에 한칸씩 이동가능 |  |
| 지형 | 이동이 불가능한 나무, 벽, 물 지형 존재 |  |
| NPC | 평화NPC, 전쟁NPC 구현 |  |
| NPC이동 | 평화 NPC는 1초에 한칸씩 이동  전쟁 NPC는 1초에 한번씩 이동하다 플레이어가 4X4 영역안에 들어오면 어그로가 활성화되어 플레이어를 쫓아간다. (장애물은 피하지 못함) | 플레이어를 따라가는 A\* 길찾기 |
| 전투 | 플레이어는 한번에 4방향을 공격 가능  전투메세지 출력 |  |
| DB | DB에 플레이어의 모든정보 저장, 로그인시 ID로 DB에 존재하는 ID만 접속가능  DB에 없는 아이디 일 경우 DB에 아이디 새로 생성 후 로그인  클라이언트 접속 종료시 마지막 정보 DB저장  이미 로그인 되어있는 ID로 로그인이 시도될 경우 차단 |  |
| SCRIPT | Lua script로 NPC 채팅 전달 | Lua script로 npc위치 지정 |
| Chatting | 플레이어들의 채팅이 모든 클라이언트에 전송 |  |
| 기타 | 5초마다 체력 회복  플레이어가 죽으면 5초뒤 처음위치에서 부활  NPC가 죽으면 30초 뒤 부활 |  |
|  |  |  |

**[스트레스 테스트] – 모든 테스트는 릴리즈 모드로 수행 함.**

방법1. 전체 월드에 고르게 플레이어들의 시작 위치를 지정

* 2천명 정도부터 조금씩 딜레이가 생기는 것을 플레이어가 느낄 수 있다.
* 5천명 이후부터 0.5초 전후로 딜레이가 생김.
* 7천명부터 딜레이가 1초 이상, 딜레이 때문에 원활한 게임플레이가 불가능.
* 8천명에서 cpu점유율 90퍼센트 전후를 왔다 갔다 함 이미 7천명에서부터 1초 이상의 딜레이로 원활한 플레이가 불가능 하다고 판단하여 스트레스 테스트 종료 (서버가 먼저 다운되지는 않았습니다.)

방법2. 일정 지역에 밀집하여 플레이어들의 시작 위치를 지정



* 800명부터 렉이 느껴짐.
* 1000명부터 0.5초 이상의 딜레이.
* 1300명부터 1초 이상의 딜레이가 느껴짐, 딜레이가 길어서 원활한 플레이 불가능.
* 1600명에서 딜레이가 2초 이상.
* 1800명에서 cpu 점유율 100퍼센트 도달 딜레이가 2초가 넘게 되어서 정상적인 플레이 불가능하여 테스트 종료.