|  |  |  |  |
| --- | --- | --- | --- |
|  | 2025-06-17 |  | |
|  | |  | |
| 게임서버프로그래밍  텀프로젝트 게임 설명서 | | | |
|  |  | |  |
|  |  | | 2020180006 김성준 |

목차

[1. 게임 조작 2](#_Toc201020396)

[2. 프로토콜 2](#_Toc201020397)

[3. 게임 흐름 2](#_Toc201020398)

[4. 자료구조 및 알고리즘 3](#_Toc201020399)

[5. 기타 4](#_Toc201020400)

# 1. 게임 조작

이동: 방향키  
공격: A

# 2. 프로토콜

1. sc\_packet\_avatar\_info

2. sc\_packet\_move

3. sc\_packet\_enter

4. sc\_packet\_leave

5. sc\_packet\_chat

6. sc\_packet\_stat\_change

7. sc\_packet\_login\_fail;

8. cs\_packet\_login

9. cs\_packet\_move

10. cs\_packet\_attack;

11. cs\_packet\_chat

# 3. 게임 흐름

NPC: Peace. Agro 두 타입 존재

Peace NPC: 플레이어가 공격할 때 까지는 랜덤하게 이동, 플레이어가 공격하는 순간 마지막으로 공격한 플레이어를 끝까지 추격

Agro Npc: 10 \* 10 시야 범위에 플레이어가 있다면 자동으로 추격

플레이어, NPC는 모두 공격시 레벨 + (10 범위의 랜덤값) 의 데미지를 받고,  
 NPC 사망 시 플레이어는 레벨 \* 레벨 \* 2의 경험치 획득  
플레이어는 5초마다 레벨 \* 5의 체력을 회복

# 4. 자료구조 및 알고리즘

타이머 쓰레드:  
텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

std::multimap과 std::mutex 사용 insert와 erase는 std::mutex로 보호, 실행해야 할 작업들은 key값으로 판별하며 std::upper\_bound를 통해 실행해야 할 이벤트의 범위를 얻은 뒤 범위 내의 이벤트를 복사 하여 모두 한꺼번에 실행하도록 구현하였다.

데이터 베이스:

concurrency::concurrent\_queue와 atomic\_bool 변수 사용  
 DB 작업은 타이머 쓰레드에 비해 매우 적으므로 atomic\_bool의 wait 함수를 통해 실제 작업이 있거나 들어올 때만 실행되도록 구현하였다.

# 5. 기타

ODBC DSN 이름: 2020180006\_GameServer\_TermProj