|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **작성자** | 2016182005 김경욱  2017184008 김덕현  2017184028 임해인 | **팀명** | WT |
| **주차** | **5주차** | **기간** | **2023.01.30~02.05** | **지도교수** | **정 내 훈**(서명) |
| **이번주 한일** | **김경욱**   * **플레이어 이동 관련 패킷 수정** * **스트레스 테스트 작업 시작**   **김덕현**   * **오브젝트 데이터 Exporter 공부 및 변경, Exporter를 통한 오브젝트 데이터 생성** * **Input Key 클라이언트에서 처리하는 방식으로 변경** * **여러 클라이언트 접속 상태에서 캐릭터들이 방향 동기화되던 버그 수정**   **임해인**   * **중점 연구과제 1번, 멀티쓰레드 클라이언트 구현 완료( 멀티 커멘드 리스트 구현 )** | | | | |

**<상세 수행내용>**

# 김경욱[서버]

* **플레이어 이동 관련 패킷 수정**

**기존의 60프레임 단위 업데이트에서 30프레임 단위로 변경 및 npc 이동 패킷 전송 또한 30프레임 단위로 변경**

**스트레스 테스트 기본 작업 수행 – 이때 최대 인원 약 700명 정도 접속 확인**

# 김덕현[클라이언트, 게임 렌더링 엔진]

* **오브젝트 데이터 Exporter 공부 및 변경, Exporter를 통한 오브젝트 데이터 생성**
* **Input Key 클라이언트에서 처리하는 방식으로 변경**
* **여러 클라이언트 접속 상태에서 캐릭터들이 방향 동기화되던 버그 수정**

# 임해인

* **멀티 커멘드 리스트를 이용한 클라이언트 구현 완료**

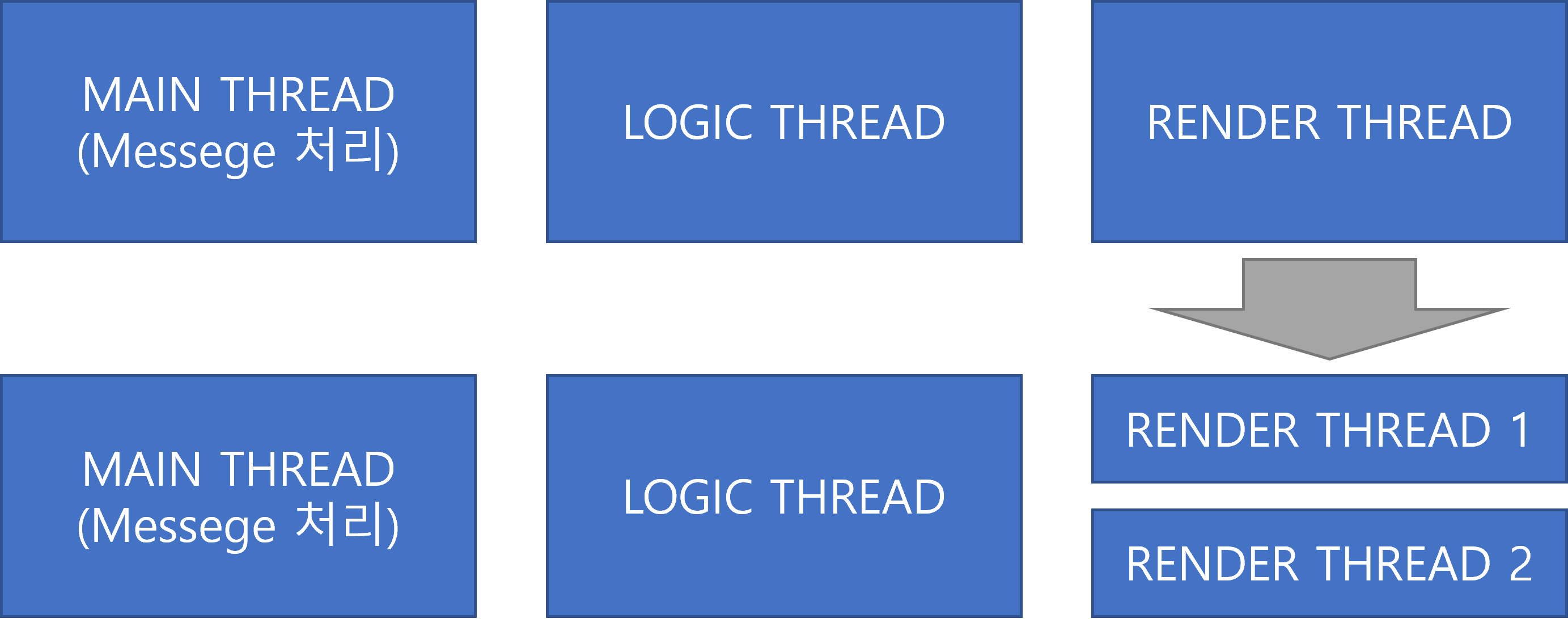


그림 쓰레드 구성 변경사항

기존의 Main/Logic/Render이렇게 3가지 구성으로 된 쓰레드를 Render 쓰레드를 2개로 나눠 분리함

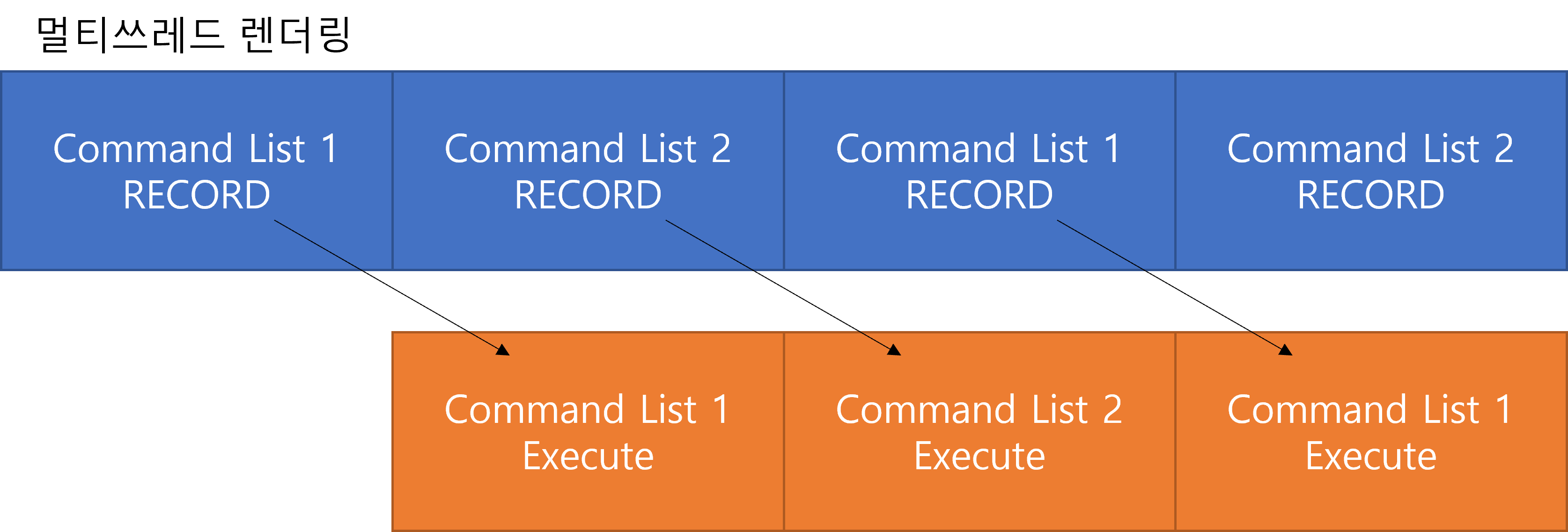


그림 RenderThread의 동작방식

* 렌더링 쓰레드는 호모 지니어스 방식으로 동작하되, 각각의 커맨드 리스트를 가지고 동작
* Execute와 Record의 끝은 이벤트를 통해서 상태를 동기화
* 결과 프레임 2배 상승

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | **해결 방안** |  |
| **다음 주차** | **6주차** | **다음 기간** | 0206~0212 |
| **다음주 할 일** | 1. 클라이언트 툰 쉐이딩 적용을 위한 코드 분석 2. 용->거미로 변경 | | |
| **지도교수**  **Comment** |  | | |