|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 8주차 | **기간** | 2024.10.24~  2024.10.30 | **지도교수** | (서명) |
| 이번주 한일 요약 | - 아이템 상자 삭제 동기화  - Send를 lock-free로 구현 | | | | |

<상세 수행내용>

- 아이템 상자 제거되는 부분 동기화

- Send, Recv를 lock-free로 구현

이번주에는 저번주에 아이템 생성 동기화 작업을 구현하는 과정에서 아이템 상자를 제거하는 부분의 동기화 작업을 해주지 않은 것이 생각나, 이 부분을 우선 구현하였다. 캐릭터가 아이템 픽업함수를 호출할 때 서버로 어떤 상자를 열었는지 보내서, 모든 클라이언트에서 상자를 제거하도록 작업 완료하였다.

다음으로는 기존에 Send, Recv 큐를 만들면서 스레드간의 Data Race가 발생하지 않도록 lock을 걸어 구현하였던 것을 수정하였다. 기존의 lock을 걸어서 만든 코드는 아무래도 블록킹이 발생을 하여 성능에 영향을 줄 수밖에 없다고 판단하였다. 그래서 lock을 사용하지 않고, Concurrent\_queue와 CAS연산을 사용하여 구현해 보았다. Concurrent\_queue와 CAS연산 모두 원자적 연산을 통해 다중 스레드 환경에서 Data Race없이 데이터 추가/삭제 또는 수정이 가능하기 때문에 이를 이용해서 lock을 삭제하고 lock-free로 구현할 수 있었다.

유튜브: <https://youtu.be/6rm55IGNDxA>

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 9주차 | **다음기간** | 2024.10.31~2024.11.06 |
| **다음주 할일** | - 매칭 시스템 만들기 | | |
| **지도 교수**  **Comment** |  | | |