1. PQCS를 이용해 Timer Thread의 bottleneck 완화 시도

서버의 main thread는 worker\_thread를 생성하고 나서 Scene를 Update하는 thread로 구동하는데, 기존에는 이 과정에서 Player와 NPC의 변화에 따라 클라이언트를 loop하며 패킷을 전송하는 작업을 했다.

하지만 이 과정에서 플레이어와 몬스터의 움직임, 회전, 애니메이션 상태 등의 패킷 전송 작업을 많은 클라이언트에서 진행하기 때문에 하나의 thread에 많은 작업이 몰리는 bottleneck이 일어났다.

따라서 하나의 thread(main thread)에서 작업하지 않고 여러 thread에서 이를 처리하기 위해 PQCS를 이용하여 Event로 넘겨 처리하려고 했다.



Event를 받아 해당 작업을 처리하는 함수이다.

사용 예시(main thread(timer thread))



PQCS를 이용하여 Event를 보내는 모습으로, 예시로 Animation type Event를 처리하라는 메시지를 m\_ioType에 담아 보내는 모습이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Worker\_thread에서 PQCS로 인한 Event가 들어오고, 그 Event의 m\_ioType를 확인해 IO\_RECV가 아니면(PQCS로 온 Event) 작업을 처리한다.

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

IO\_TYPE과 어떤 클라이언트인지 확인하고 Event를 처리한다. 예시로 Animation type Event를 받아 패킷을 보내는 모습이다.

이 작업을 통해 한 스레드에서 처리하는 많은 작업을 여러 스레드에 나눠 처리할 수 있게 되어 bottleneck 현상이 완화될 것이다.

오류

플레이어의 이동, 회전, 애니메이션 작업은 잘 작동하지만, 몬스터의 작업 때문에 프로그램이 작동하지 않는다.

처음에는 PQCS를 이용한 방식 자체가 잘못되었다 생각했지만 몬스터 작업을 주석하고 테스트해보니 플레이어는 잘 작동하지만, 몬스터 작업을 진행할 때 문제가 생긴다는 것을 알아냈다.



PQCS의 3번째 인자인 key값을 넣을 때, 플레이어 작업일 경우 클라이언트의 주소를 넣어 worker\_thread에서 map으로 클라이언트를 꺼내 작업한다.





몬스터 또한 비슷한 방식으로 몬스터의 주소를 key로 넣고 Event를 처리할 때 p\_Client자리에 key 값을 넣어 어떤 몬스터의 작업인지 표시하려 했다.

하지만 몬스터의 Event를 PQCS로 전송하고 나서 처리하는 과정에서 오류가 발생한다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Key에 들어가는 주소 값이 쓰거나 읽을 때 잘못되었나 하고 몬스터의 id를 key로 넣어 리스트에서 찾아 Process\_Timer\_Event()의 인자로 넣어 작동해봤지만, 여전히 같은 오류가 난다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

멀티 스레드라 크게 의미는 없겠지만 디버그로 돌려 나온 오류부분이다. remoteClients의 b\_Enable을 읽을 때 오류가 난다.

몬스터의 관련하여 문제가 생겨야 하는 것이 당연하다고 생각하는데 뜬금없이 remoteClients에서 클라이언트를 읽을 때 에러가 발생하는 것으로 보인다.

이러한 에러는 멀티 스레드 특성상 한 스레드가 작업할 때 문제가 생겼는데 그 와중에 컨텍스트 스위치가 일어났기 때문으로 보인다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Process\_Timer\_Event()를 주석하여 작업을 진행하지 않아도 오류가 나는 것을 알아냈다.

몬스터 Event의 PQCS만으로 에러가 검출된다는 것이다.

‘