1. 클라이언트 여러 씬 관리 - (2) 오류 해결



텍스트이(가) 표시된 사진

자동 생성된 설명

이전 작업에서 Scene마다 MagePlayer를 생성하면, CreateShaderResourceViews()에서 오류가 나는 일이 발생했다.

이전 작업에서 Scene마다 MagePlayer를 생성하면, CreateShaderResourceViews()에서 오류가 나는 일이 발생했다.

Player를 어떻게 Scene마다 생성해 둘 수 있을까 하며 CBV, SRV 크기를 늘려보려 했지만,

곰곰이 생각해보니 Scene마다 Player가 생성되면 Scene을 변경하는 중에 Player값이 바뀔 위험이 있어 게임 진행에 영향을 미치지 않을까 라는 의문이 들어서 최종적으로 게임 프레임 워크에서 하나의 m\_pPlayer만 가지고 게임을 진행을 하는 것이 더 타당한 방법이라 생각했다.

따라서 기존의 GameScene::m\_pPlayer를 제거하고 GameFramework::m\_pPlayer를 가지고 관리하게 구현했다.



GameScene의 update()에서 GameFramework::FrameAdvance()로 가져온 m\_pPlayer의 Update() 함수

1. 클라이언트의 Login (패킷 정리)

텍스트, 화이트보드, 문서이(가) 표시된 사진

자동 생성된 설명

클라이언트 – 서버 로그인 관련 패킷 연구

클라이언트에서 Login을 시도해서 Server로 로그인 관련 정보를 보낸다.

서버에서 받은 로그인 관련 정보로 로그인을 확인하고 성공/실패의 결과를 클라이언트에게 보내준다.

만약 성공이라면 서버에 접속이 성공했다는 의미이니, 서버에 플레이어를 생성하고, 클라이언트로 플레이어를 추가하라는 정보를 보낸다.

결과가 실패라면 로그인 시도를 다시 해달라는 화면 등을 알린다.

클라이언트는 플레이어 추가하라는 패킷을 받고 다른 클라이언트 플레이어를 추가해 준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트가 서버로 로그인 시도하는 패킷을 전송하는 모습이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

로그인의 성공 여부는 테스트를 위해 잠시 생략해두고 클라이언트가 접속하면 서버에선 플레이어를 추가하는 패킷을 보내게 했다.

첫 for문은 접속하는 클라이언트는 서버에 있는 모든 플레이어 정보를 받는 모습이고,

두번째 for문은 다른 클라이언트들에게 방금 접속한 클라이언트가 있다는 정보를 알려주는 모습이다.

당연하게 접속하는 클라이언트 본인의 정보는 필요 없으니 보내지 않도록 한다.

1. 서버 – 클라이언트 연결 (2) 개인 프로젝트에서 메인 프로젝트로 코드 병합

이전에 만든 개인 프로젝트 서버를 병합하고, 멀티 스레드로 만들어 둔 클라이언트를 싱글 스레드로 만든 Overlapped I/O CallBack 방식의 클라이언트 NetworkMGR를 병합하려 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이전에는 ClientRecvThread()를 만들어 블로킹 소켓과 멀티 스레드로 서버와 통신했지만, 멀티 스레드의 비용이 높다는 것을 이제 알기 때문에 네트워크 통신에서는 Overlapped CallBack I/O를 이용하여 쓸데없는 스레드 생성을 줄였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Start()가 불리면 클라이언트 본인의 닉네임을 입력하도록 했고(지금은 편의를 위해 name만 입력), 서버로 로그인을 시도하는 패킷을 보낸다.

그 후 do\_recv()를 호출하여 Overlapped I/O CallBack 방식의 Recv를 건다.



텍스트이(가) 표시된 사진

자동 생성된 설명

Recv가 성공하여 CallBack 함수가 불리면 이전에 구현해 둔 패킷 처리와 패킷 재조립을 하면서 받은 데이터를 처리한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

패킷 처리는 버퍼의 두번째 바이트를 확인하여 패킷 타입을 확인하고 처리할 패킷 종류로 버퍼를 타입 캐스팅하여 Scene등에 반영한다.

패킷 타입이 SC\_ADD\_PLAYER일 땐 게임 프레임워크의 m\_OtherPlayers에 플레이어 오브젝트를 생성한다.

현재는 테스트를 위해 다른 클라이언트의 로그인이 확인되면 연결된 클라이언트의 id를 x좌표로 하여 RedMage 모델의 오브젝트를 생성하도록 했다.

1. 서버 – 클라이언트 연결 (3) 클라이언트 Recv 오류

이렇게 서버와 클라이언트의 NetworkMGR을 구현하고 실행하면, 서버에 접속한 클라이언트 수만큼 접속한 클라이언트의 id를 좌표로 여러 RedMage 캐릭터가 생성돼야 할 것이다.

하지만 두 클라이언트를 접속시켜도 SC\_ADD\_PLAYER 패킷을 처리하지 않는다.

하늘, 야외, 산이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트에서 받아온 데이터의 1,2번째 바이트를 출력해본 결과 두 개 모두 0이 담아져 있었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

서버의 Send를 보면 size도 맞고 패킷 타입도 맞는 것으로 봐서 서버에서는 데이터를 올바르게 보낸다고 생각했다.

오류를 보면서 이것저것 시도해봤지만 원인을 찾기 어려웠고 다음 주까지 이어지게 됐다.