1. 서버 – 클라이언트 연결 (3) 클라이언트 Recv 오류

저번 주에 이어 서버 – 클라이언트 데이터 송수신의 오류를 처리하였다.

하늘, 야외, 산이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

저번 주의 작업에서 서버에서 Send할 때 보내는 데이터는 올바르게 보내는 것처럼 보였고, 클라이언트에서 Recv할 때 제대로 받아오지 못하는 것으로 유추됐다.

텍스트이(가) 표시된 사진

자동 생성된 설명

코드를 보니 Recv하기 전 Recv와 Overlapped 객체를 초기화 해주지 않고 있었다.

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

따라서 do\_recv()에 Recv에 관련된 객체들을 모두 초기화 해주고 WSARecv()을 호출하게 만들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 로그인을 하니 SC\_ADD\_PLAYER 패킷이 성공적으로 수신됐다.

텍스트이(가) 표시된 사진

자동 생성된 설명

하지만 SC\_ADD\_PLAYER 패킷을 받으면 임의로 ‘RedMage’ 모델 오브젝트가 생성되게 만들었는데 화면 어디에도 생성되지 않았다.

1. 서버 – 클라이언트 키 입력

클라이언트는 최소한의 데이터를 서버로 보내고, 서버는 그에 맞는 처리 결과를 클라이언트로 보내는 것이 합당하다 생각하여 클라이언트의 키보드 입력 상태를 가지고 서버가 데이터를 처리하려고 했다.

구조체

텍스트이(가) 표시된 사진

자동 생성된 설명

서버에서 클라이언트마다 키 상태를 저장하기 위한 Input 구조체로, 주로 keys의 값이 0인지 1인지로 키 상태를 구분할 예정이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트에서 서버로 KEYDOWN과 KEYUP 데이터를 보내기 위한 패킷 구조체이다.

클라이언트

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트가 키를 입력하면 입력한 키(WPARAM)를 가지고 패킷을 만든다.

해당 작업이 한 프레임마다 일어나면 안 되기 때문에 Input 구조체를 이용하여 입력 상태를 저장해두었다.

서버

텍스트이(가) 표시된 사진

자동 생성된 설명

이전 개인 프로젝트에서 사용하던 Input class를 이용하여 서버에 접속한 클라이언트의 키 정보를 RemoteClient의 멤버 변수로 만들어 저장한다.

텍스트, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

서버에서는 클라이언트로부터 KEYDOWN과 KEYUP 정보가 들어오면 해당하는 클라이언트의 key를 업데이트한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트로부터 패킷이 제대로 도착하는 모습

1. Player pool 추가

전 작업에선 NetworkMGR에서 Object를 동적 할당하여 생성하려 했을 때, 아무런 모델도 띄워지지 않았다.

그래서 다른 클라이언트분들과 선배 한 분에게 여쭤본 결과, 모델을 생성하는 LoadAnimationModel()이 Scene 작업중 동적으로 생성되는 장면을 고려하지 않고 구현된 것 일 수도 있다는 의견이 나와서 직접 함수에 수정이 필요해 보이는데, 현재 상황에서 다른 클라이언트분들은 더 중요한 작업이 남아있고 내가 해결할 수 있는 능력도 없기 때문에 이 문제를 어떻게 해결해야 하나 고민했다.

우선 게임을 진행하면서 복잡한 모델이면서, 동적으로 생성되어야 할 객체들이 어떤 것이 있고 한 번의 Frame에 최대로 동시에 얼마나 사용하는 지 찾아봤다.

게임 진행을 하는 Player (최대: 4)

Player나 Monster가 사용하는 투사체 공격(최대: 객체 하나 당 약 10~20)

이펙트(또는 파티클) 효과는 간단한 구조의 모델이나 Texture를 이용하기 때문에 고려하지 않았다.

Player는 접속/종료로 사용 여부가 정해져 있고 그때마다 new/delete를 사용하는 것은 모델의 비용이 크기 때문에 비효율적이라 판단했다.

따라서 동적 생성을 줄여 프로그램 속도를 향상시키고 객체를 재사용하기 위해 Pool을 이용했다.





Pool을 생성하기 전, Pool에있는 객체를 사용할 것인가 구분하기 위해 char 형의 변수를 생성했다.



PlayersPool과 Update와 Render등의 작업을 진행하는 Player 컨테이너이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

PlayersPool에 객체를 생성하는 모습이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

서버로부터 새로운 클라이언트가 접속하여 Player를 생성하라는 패킷이 오면 m\_OtherPlayersPool에 저장해둔 포인터 중 하나를 m\_OtherPlayers로 옮겨 Player 하나를 생성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

다른 클라이언트 Player 배정을 성공적으로 완료한 모습이다.

배정은 완료했으나 서버에서 패킷에 Player의 정보를 담아주지 않았고, 받은 정보가 없으니 배정하는 Player 객체는 기본 값으로 만들어진 모습이다.

따라서 다음 작업은 접속하는 플레이어의 정보를 담아야 하고 서버에서 클라이언트의 키 입력을 바탕으로 움직임 등이 가능하도록 만들어야 한다.