1. 서버 – 클라이언트 통신 – 플레이어 기본 움직임, 카메라 방향

현재 클라이언트의 키 입력 상태를 서버에서 알 수 있다.

클라이언트가 플레이어를 움직이게 할 때 패킷 통신을 어떻게 해야 할까?

아래 처음 나오는 방법은 이번 주 차 초반에 생각하고 구현했던 방식이었지만, 문제점을 발견하여 다른 방식으로 바꿨다.

클라이언트에서 움직임을 정하지 말자

클라이언트가 매 프레임마다 플레이어를 움직이려고 키를 입력할 때마다 패킷으로 저장해서 보낸다면, 클라이언트마다 프레임 속도가 다르기 때문에 같은 값의 속도여도 클라이언트마다 이동하는 양이 달라질 수 있다.

또한 서버로 연결하는 클라이언트가 매우 많은 것을 가정하기 때문에 서버에서 감당하지 못할 만큼 많은 패킷을 처리해야 할 수 있다. (네트워크 낭비로 예상했다.)

서버에서 클라이언트의 키 입력 상태로 움직임을 정하자.

위의 이유로 클라이언트는 KEYDOWN과 KEYUP으로 서버에 송신하고, 서버에선 클라이언트의 방향키가 눌렸을 때 해당하는 플레이어 움직임을 처리하며 한 프레임마다 모든 클라이언트에게 모든 Player의 위치를 보내주는 것이 합당하다 생각했다.

텍스트, 전자제품이(가) 표시된 사진

자동 생성된 설명

클라이언트의 플레이어의 정보를 저장하기 위해 클라이언트에서 사용하던 틀을 가져와 사용했다.

단, Render부분은 서버에서 필요 없으니 제거했다.

텍스트이(가) 표시된 사진

자동 생성된 설명 Player의 위치 상태와 방향 등의 정보가 있는 Component

텍스트이(가) 표시된 사진

자동 생성된 설명PlayerMovementComponent의 start()

텍스트이(가) 표시된 사진

자동 생성된 설명

PlayerMovementComponent의 update()

본인 클라이언트의 KeyInput을 추적하여 움직임을 처리하는 모습이다.

텍스트이(가) 표시된 사진

자동 생성된 설명 실행 화면

움직이는 값은 얼추 제대로 나온 것처럼 보이지만 아직 해결해야 할 점이 있다.

플레이어의 회전이 적용되지 않았다.

현재 w/a/s/d는 오브젝트의 x/z 값을 +/- 방향으로 단순히 바꿔주고 있다. 하지만 플레이어의 바라보는 방향에 따라 움직이는 방향도 달라져야 한다. 즉, 클라이언트가 카메라를 움직일 때 서버에서도 방향을 바꿔줘야 한다.

통신 방법 변경

문제가 생겼다.

서버에서 클라이언트로 플레이어의 위치 정보를 보내려면 한 프레임마다 Scene의 정보를 다 보내야 하는데, 한 프레임마다 모든 클라이언트를 루프해야 하는 점과 이벤트가 없는 플레이어의 정보도 보내야 한다는 점에서 비용이 매우 크게 든다는 것을 간과했다.

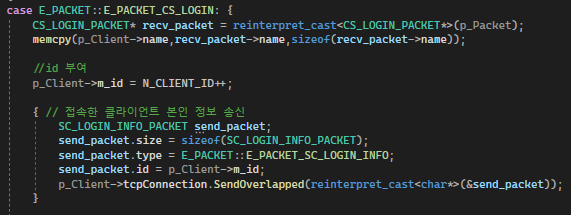
따라서 매 프레임마다 서버에서 클라이언트로 패킷을 전송하는 것이 아니라, 클라이언트가 서버로 패킷을 전송하면 바로 처리하여 다른 클라이언트들에게도 결과를 전송하기로 했다.

그 전에 플레이어의 ID를 부여하는 방식이 빠져 있어서 그 부분부터 해결했다.

텍스트, 장치, 게이지, 계량기이(가) 표시된 사진

자동 생성된 설명

서버가 클라이언트에게 딱 한번 보내주는 패킷으로, 서버 입장에서 번호표를 주는 것과 같다.



서버에서 클라이언트가 로그인 할 때 그 클라이언트에게 ID값을 전송하는 모습이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

서버가 부여한 ID를 본인 Player 객체의 멤버변수에 작성한다.

플레이어 움직임 변경

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트에서 서버로 움직임 정보를 보내기 위한 패킷을 정의했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트가 움직이려 할 때 패킷 전송하는 모습이다.

여기서 CS\_MOVE\_PACKET의 dirX, dirY, dirZ 값은 크기를 1로 정규화 하여 플레이어가 어느 방향벡터로 보낼 지 정해준다.

즉, 전과 다르게 서버에서 플레이어의 LookVector 정보를 따로 받지 않아도 어느 방향으로 갈 지 알 수 있다.

텍스트, 장치이(가) 표시된 사진

자동 생성된 설명

서버에서 클라이언트로 플레이어의 위치를 정해주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

서버에서 CS\_MOVE\_PACKET 처리 방법

클라이언트로부터 온 방향 벡터 정보를 이용하여 플레이어를 움직이고, 다른 클라이언트들에게 움직인 정보를 보내준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트에서 서버가 정해준 플레이어의 위치를 처리하는 방식

받아온 패킷의 id를 확인하고 본인의 id일 때, GameFramework::m\_pPlayer(본인의 플레이어 객체)의 위치를 변경하고, 그렇지 않다면 OtherPlayer 리스트를 탐색하여 존재하는 id를 갖는 Player가 있다면 그 객체의 위치를 바꾼다.

플레이어 방향 변경

텍스트, 장치, 계량기이(가) 표시된 사진

자동 생성된 설명

텍스트, 장치이(가) 표시된 사진

자동 생성된 설명

클라이언트가 카메라 방향이 변경될 때(LookVector) 그 방향의 정보를 담는 패킷



텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트의 Player::Rotate에서 결과로 나온 Player의 LookVector를 패킷에 담아 서버로 보내준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

서버에서 CS\_LOOK\_PACKET을 처리하는 모습이다.

받아온 LookVector를 이용하여 RightVector와 UpVector를 다시 맞추고, 세 Vector를 플레이어의 방향 정보로 넣어 처리한다. 그리고 LookVector를 다른 클라이언트들에게도 전송한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

클라이언트에서 SC\_LOOK\_PLAYER\_PACKET을 처리하는 모습이다.

Move와 비슷하게 id로 처리할 플레이어를 찾고, 서버에서 처리한 것과 같이 LookVector로 RightVector, UpVector를 다시 만들고 플레이어의 방향으로 정해준다. 이 때 카메라에도 같은 값을 넣어서 플레이어가 바라보는 방향의 입장에서 카메라가 보도록 했다.

실행 화면

텍스트, 잔디, 운동 경기, 스포츠이(가) 표시된 사진

자동 생성된 설명

움직임은 플레이어의 방향대로 움직이긴 하나, 카메라가 매우 버벅이는 모습이 보인다.

이유가 무엇일까?

카메라를 고정시키고 Player 객체를 회전시켜보니 그 객체 자체가 버벅인다.

서버로부터 받은 LookVector의 값이 제각각인 것이다.

그래서 서버에서 LookVector값을 어떻게 주는지 관찰해봤다.

텍스트이(가) 표시된 사진

자동 생성된 설명 분명 한 방향으로 카메라를 돌렸지만 LookVector에 증감이 생긴다.

1. 서버 – 클라이언트 통신 – Player ID 부여

클라이언트가 닉네임을 입력하여 접속하면, 서버에서 Player에 받아온 닉네임을 저장하여 생성하고 서버는 클라이언트에게 고유의 ID(번호표 같은 개념)를 부여한다.