섹터매니저

플레이어의 시야에 들어오는 데이터만 서버가 전송

시야를 기준으로 잡는다.

이 시야의 등분을 나눈다 표준 9개

플레이어가 작으면 더 나눠야 한다.

섹터당

영역좌표를 가지고,

안에 있는 오브젝트들을 관리하는 list (monsterlist,playerlist,itemlist 등)

그리고 시야에 들어오는 섹터 리스트도 갖는다.

(View sector list)

섹터는 맵에 모양에 따라서 재량껏 나누면 된다.

섹터 여러 개를 묶어서 1영역 2영역 등으로 해도 된다.

섹터매니저에서 이 섹터들을 관리한다

Sector list

영역을 나눈다면 sector group list가 있을거임.

Init 될때 전체 섹터 정보들이 로딩되어야한다. 전체 초기화 작업.

플레이어를 특정 섹터에 스폰한다면 해당 섹터의 정보를 playerlist에 넣는다.

플레이어도 현재 위치한 섹터 정보를 가지고 있어야 함.

Session에 sector\* 를 멤버로 가지고 있기.

Sectormanager에서

어떤 플레이어가 행동을 했을 때 시야 범위 내에 모든 오브젝트에 정보를 뿌릴 때 필요한

Sendpacket함수가 있어야 함다.

Sendpacket(session\*,protocol,moveflag)

{

Sector의 viewlist를 순회하면서 그 섹터들의 playerlist를 전부 순회 하면서 정보를 전송한다.

}

서버가 주는 정보로 렌더링한다.

레이턴시 발생은 보간법을 통해 쪼개서 하는데 일단

원칙은 서버의 정보로 렌더링 하는것임.

섹터 이동시 기존의 섹터 리스트 말고도 이동해서 보이게 된 섹터까지 정보를 업데이트를 해줘야 한다. 이런 검사를 sendpacket에서 하는데 moveflag가 true인 경우에

섹터이동이 있는지 체크하는 함수를 호출해서 검사하고 이동이 되었다면 이동전

Viewlist 의 정보를 변경해 주고 이동 후의 섹터 list에 넣고 하는 작업을 한다.

현재 session의 sector\*도 바뀐다.

섹터가 바뀌면 이전 섹터리스트에도 전부 보내고, 이동 후 추가된 섹터 리스트에도 보내야한다.

이 모든 작업은 sendpacket 내에서 수행.

몬스터 매니저에서도 섹터매니저를 호출 할것이다.

몬스터도 동일하게 돌아간다.

Ai 심플하게 할거면 서버에서 해도 되는데

복잡하면 클라에서 구현한다

하나의 클라이언트를 호스트 삼아서 그 호스트에서만 몬스터 ai를 돌리고

그 값을 서버에 보내고 그 정보를 서버가 다른 클라들에 뿌려서 렌더링함.

전체적으로 연산이 많은 작업은 호스트 한명에 시켜서 서버가 받아서 전체에 뿌리는 식으로 가야한다.

섹터 구조는 생각하기. 2차원일때는 vector가 편하다.

db에서 euckr 로 했는데 이렇게하면 한글은 2byte 영문은 1byte로 들어오는데

우리는 문자열 유니코드로 하기 때문에 디비에서 가져온 데이터를

유니코드로 변경해주는 함수를 사용해서 유니코드로 처리할수 있도록 할 수 있다.

아스키까지 통틀어서 영 1 한 2 =멀티바이트

2바이트 문자 셋 = 유니코드 와이드캐릭터타입.

Multibytetowidthchar => 멀티 > 와이드

Widthchartomultibye => 와이드 > 멀티

로비지나서 게임을 접속해서 맵에 들어갈 때 스폰되어서 해당 섹터로 들어갈 때

그 섹터에 포함시키고 시야내의 섹터들에게 시야내에 있어야 할 데이터들을

전부 전송해줘야 한다.

패킷의 크기가 4096을 넘어가지 않도록 패킹한다.

클라는 시야에 대한 렌더링만 하기 때문에 만약 시야밖의 오브젝트의 상태 변화는 처리 안하고 있는다. 하지만 서버에서는 그 데이터를 가지고 있음.

그래서 만약 다시 시야내로 들어오게 되면 그 정보를 전송.

만약 플레이어가 이동했을 때 변화한 시야에 대한 섹터에 대한 정보를 이동한 오브젝트에 정보를 보내줘야 한다.

그니까 오브젝트가 섹터를 이동했을 때 시야에서 제외되는 섹터들의 정보 함수, 시야에 추가되는 섹터 정보 함수를 만든다.

안보이는 오브젝트들에 대해서는 상태를 off 시켜야한다. 보이게 되면 on 시킨다.

On 시키면서 서버가 준 정보로 이동 시킴.

각 오브젝트들을 아이디가 있어야 한다 상호작용 오브젝트들에는 전부 id 부여.

이 게임에만 쓰이고 저장할 필요 없는 경우 캐릭터의 인벤에 계속 저장할 필요는 없다.

그냥 메모리에서만 저장했다 쓰면 됨.

이동에 대한 연산은 클라에서 하고 서버에서는 못가는 block 섹터를 만들어 두고

서버에서 승인을 해주면 연산에 대한 이동을 수행하고 승인이 안나면 연산에 대한 이동을 수행하지 않는다.