1. Room 준비

우리 게임은 하나의 Scene으로 접속하는 게임이 아니라 LobbyScene에서 방을 만들어 접속을 원하는 클라이언트끼리 상호 작용할 수 있도록 해야 한다.

그러기 위해 Room이라는 클래스를 만들어 클라이언트와 Scene을 관리하기로 했다.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* Room class 내부 모습

Start() – 방이 생성될 때 호출해야 하는 함수이다. 예로는 NPC, Monster 객체 생성이 있다.

Update() – 주기적으로 호출하는 Scene Timer 역할이다. Player나 Monster의 상태 등이 바뀔 때 데이터를 통신하는 역할 등을 한다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Room class를 만듦으로 Main Timer 코드는 매우 단축된 모습이다.



Room class 안에 Scene 객체가 각각 있어 다른 Room과의 게임 상태 영향이 미치지 않지만, 성능적으로 문제가 되는 부분을 찾았다.

모든 Room이 같은 처리를 해야 하는 부분에서 불필요한 메모리 사용이 문제가 된다.

예를 들어 Player가 움직이지 않는 바위, 나무 등과의 객체와의 충돌처리 등 모든 Room에서 사용하는 같은 데이터가 Room마다 갖고 있게 된다면 메모리 낭비가 된다.

그래서 Scene 클래스의 정적 객체를 남겨두고, 돌이나 나무, 맵 등 수정되지 않는 데이터를 저장한다음 모든 Room에서 사용할 수 있도록 했다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

그리고 Monster는 Room 마다 다르기 때문에 Monster를 생성할 때 어떤 Room에 생성할 지 정해주는 생성자를 만들었다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* 서버가 시작할 때 임시로 Room을 생성하는 모습

지금은 서버가 시작할 때 임시로 Room을 만들어 모든 클라이언트가 그 Room으로 접속하게 했다.

이 작업은 나중에 로비 화면이 만들어 지면 클라이언트가 방을 만들 때 생성하는 것으로 바꿔야 한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 클라이언트가 접속할 때 임시로 만든 Room(0번)에 추가하는 모습

마찬가지로 로비 화면이 만들어 지면 클라이언트가 방을 접속할 때 이 작업이 이루어져야 한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

그리고 Room마다 다른 객체라는 구분의 편의성을 위해 Player와 Monster에 m\_roomNum 변수를 둬서 어떤 방에 있는 객체인지 구분할 수 있다.

1. Far Monster, Rush Monster 서버에 정의

서버에서 정의된 Monster Type은 Close Type만 정의돼 있었다.

클라이언트는 Far Type, Rush Type까지 정의되어 있고 사용하고 있으니 서버에서도 제대로 작동하기 위해 정의해야 한다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명 텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

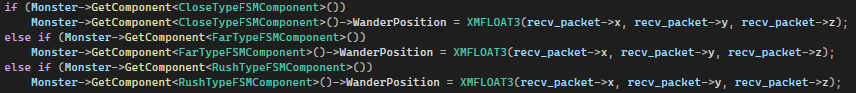
* 클라이언트에서 정의된 Component와 State를 불러온 모습

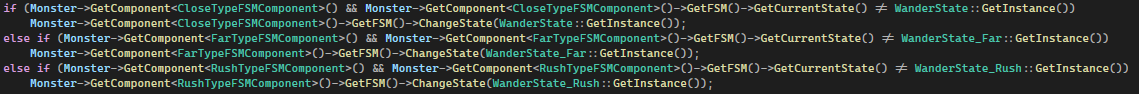
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명





그리고 Close Type과 마찬가지로 서버 통신을 받을 때 공격, Target, 배회 방향, 그 외 상태 변경 등 추가했다.

1. 해결해야 할 버그

다른 코드와 병합하고 나서 UILayer를 출력할 때 프로그램이 중단된다. 즉, 2DText 출력 과정에서 문제가 생긴다.

그리고 이전에 잘 작동하던 DB가 연결되지 않는다..

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

DB와 연결하는 부분에서 제대로 작업하지 않는다는 것을 알았다.