로우 레벨 디자인

<클라이언트>

테이블이(가) 표시된 사진

자동 생성된 설명

* SetPlayerInfo(): 서버로 보낼 플레이어의 정보

테이블이(가) 표시된 사진

자동 생성된 설명

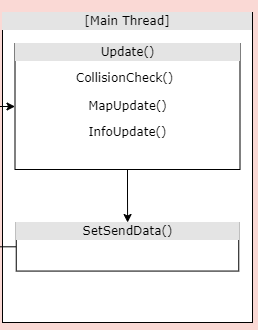
* UpdatePlayerInfo(): 서버에서 받아온 정보로 플레이어 정보 갱신. 객체 위치 조정, 어떤 임무를 수행했는지를 받아오면 해당 임무를 수행완료로 바꾸기.
* UpdateMapInfo(): 서버에서 받아온 정보로 맵 갱신. 임무가 완료되었으면 완료된 표시. 임무 진척도 증가 등 서버에서 변한 정보 확인하여 바꾸기.

<서버>

테이블이(가) 표시된 사진

자동 생성된 설명

* CheckPlayerInfo(): 클라이언트에서 받아온 정보로 플레이어의 타입을 파악하여 어떤 함수를 호출할 지 정하기. Ex> 플레이어1이면 무슨 임무를 수행해야하는지 등등



* CollisionCheck(): 충돌체크하는 함수. 객체 타입에 따라 충돌체크 다르게 함. 플레이어가 범인일 경우 모든 플레이어와 범인, 플레이어가 일반인일 경우 임무 오브젝트와 충돌체크한다.
* MissionCheck(): 충돌체크 이후 호출하는 함수로, 일반인일 경우 어떤 임무를 수행했는지 확인한다.
* SetDead(): 충돌체크 이후 호출하는 함수로, 범인일 경우 어떤 플레이어와 충돌했는지 확인하여 충돌한 플레이어 사망상태로 바꾼다.
* MapUpdate(): 충돌처리된 데이터로 맵을 갱신한다. 임무, 플레이어 사망, 플레이어 이동 등
* InfoUpdate(): 임무진척도, 각 플레이어의 남은 임무 등을 갱신한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* SetSendData(): 갱신된 정보를 보낼 데이터의 구조체에 맞게 채워 넣는다. 이 데이터는 클라이언트가 받아서 클라이언트에서 데이터를 갱신하는데 사용한다.