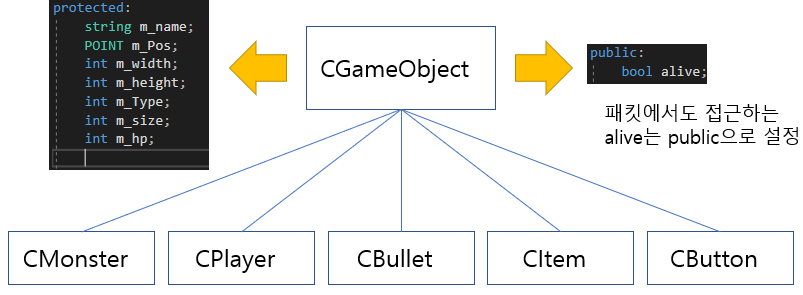
게임실행



**- 게임 실행**

<KeyDown 시>

- void SendData(PlayerInfo , EnemyInfo**) - 모든 OBJECT들의 정보 송신**

**- 게임 실행**

~~Int IsPlayerAlive()~~ **~~–플레이어의 목숨 개수를 반환하여 플레이어의 생존을 확인하고, 남은 목숨을~~****~~그린다.~~**

Player의 GetHp()함수를 사용하여 m\_Hp가 0보다 크면 InGame->Render()에서 렌더링한다

~~bool IsEnemyAlive()~~ **~~-살아있는 적을 판단하여 그린다.~~**

~~void IsDrawItem(bool)~~ **~~-활성화된 아이템을 판단하여 그린다.~~**

~~void IsDrawBullet(bool)~~ **~~– 화면 안에서 적 오브젝트와 충돌하지 않은 총알을 그린다.~~**

객체의 IsAlive()함수를 사용하여 alive가 true면 InGame->Render()에서 렌더링한다

void RecvInGameData(PlayerInfo, EnemyInfo) **–서버에서 받은 정보로 객체 정보를 업데이트 해준다**

**- 게임종료**

void IsAllPlayerDead() **–게임 종료를 위해 모든 플레이어가 죽었는지 확인 후 서버에서 받은 값을 현재 플레이어의 생존, score정보를 저장하고 Score 페이지로 씬을 넘긴다.**

~~void RecvRankData()~~ **~~-10위권 내의 랭킹 정보를 송신받는다.~~**

MyMenu의 생성자에서 정보를 송신받는다

~~void DrawScore()~~ **~~-획득한 점수를 그린다~~.**

~~void DrawRank()~~ **~~-서버에서 송신 받은 랭킹정보를 그린다~~.**

MyMenu->Render();에서 서버에서 송신받은 닉네임, 랭킹,점수 정보를 그립니다

|  |  |  |
| --- | --- | --- |
| 김천기 | 이소현 | 장은선 |
| 예상 역할분담 | | |
| 서버 프레임워크 제작 | 프레임워크 제작 | 클라이언트 보조 |
| 레디 송수신 구현 | 클라이언트 메인 | 동기화문제 해결 |
| 스레드 함수 설계 및 구현 | 클라이언트측 송수신 구현 | 스레드 함수 설계 및 구현 |
| 충돌 체크 함수 설계 및 구현 | 스코어 송수신 구현 | 충돌 체크 함수 설계 및 구현 |
| 실제 역할분담 | | |
| 서버 프레임워크 제작 | 클라이언트 프레임워크 제작 | 서버 프레임워크 제작 |
| 보조 클라이언트 | 메인 클라이언트 | 보조 클라이언트 |
| 레디 송수신 구현 | 레디 송수신 구현 | 레디 송수신 구현 |
| 스레드 함수 설계 및 구현 | 동기화 문제 해결 | 스레드 함수 설계 및 구현 |
| 충돌체크 함수 설계 및 구현 | 충돌체크 함수 및 구현 | 스코어 송수신 구현 |
| 공유자원 설계 |  | 공유자원 설계 |
|  |  |  |

