플레이어 접속

**PrepareSocket**  
윈속 버전 : 2.2  
사용 프로토콜 : IPv4, TCP

**int Connecter(char\*, int)**  
**Return value** : player\_index  
**Parameter** : server\_ip, server\_port  
**Explain** :  
connect()와 RecvPlayerNum()을 통해 서버와 통신하고 CreateThread()를 사용하여 게임 플레이를 위한 쓰레드를 생성한다.

**int CheckGame()  
Return value** : game\_index  
**Parameter** : -  
**Explain** :  
PlayerManger의 게임들이 플레이어를 추가할 수 있는지 확인하고 추가한다. 불가능할 경우 MakeGame()을 부른다.

**int MakeGame()  
Return value** : game\_index  
**Parameter** : -  
**Explain** :  
PlayerManger의 게임배열에 새로운 게임을 추가한다.

**int MakePlayer(int)  
Return value** : player\_index  
**Parameter** : game\_index  
**Explain** :  
game\_index번째 게임에 통신자의 플레이어 오브젝트를 추가하고 해당 오브젝트의 인덱스 번호를 반환한다.

**void SendPlayerNum(int, SOCKET)  
Return value** : -  
**Parameter** : player \_index, client\_sock  
**Explain** :  
통신자에게 플레이어 오브젝트의 인덱스 번호를 넘겨준다.

Client Loop

**void DataReceiver()   
Return value** : -  
**Parameter** : -  
**Explain** :  
RecvThread()에서 불리는 함수.  
recv()를 통해 서버로부터 ObjectData의 배열을 받는다.

**void SetObjectData(ObjectData\*)  
Return value** : -  
**Parameter** : input\_array  
**Explain** :  
GameObjectManager. p\_Recv에 input\_array를 덮어씌운다.

**void GameObjectManager.SwapDataBuffer(DBufPointer)  
Return value** : -  
**Parameter** : location  
**Explain** :  
인자에 따라서 p\_Recv혹은 p\_Render를 p\_other와 교환한다.  
이 함수는 서로 다른 쓰레드에서 불릴 수 있기 때문에 임계영역으로 동기화한다.

**void Decoder()  
Return value** : -  
**Parameter** : -  
**Explain** :  
GameObjectManager.Decode()를 부른다.

**void GameObjectManager.Decode()  
Return value** : -  
**Parameter** : -  
**Explain** :  
GameObjectManager.p\_Render배열의 값들을 GameObjectManager. m\_GameObjects.Decode()의 원소로 넘겨준다.

**void GameObject.Decode(ObjectData)  
Return value** : -  
**Parameter** : object\_data  
**Explain** :  
object\_data의 값으로 자신의 멤버변수들을 Update 한다.

**void Render()**   
**Return value** : -  
**Parameter** : -  
**Explain** :  
RederThread()에서 불리는 함수.  
GameObjectManager.Render()를 부른다.

**void GameObjectManager.Render()**  
**Return value** : -  
**Parameter** : -  
**Explain** :  
GameObjectManager. m\_GameObjects들의 Render()를 부른다.

**void GameObject.Render()**  
**Return value** : -  
**Parameter** : -  
**Explain** :  
자신을 화면에 출력한다.

Server Loop

**void Update(float)**   
**Return value** : -  
**Parameter** : eTime  
**Explain** :  
GameThread()에서 불리는 함수.  
GameObjectManager.Updata()를 부른다.

**void GameObjectManager.Update(float)**  
**Return value** : -  
**Parameter** : eTime  
**Explain** :  
GameObjectManager. m\_GameObjects들의 Update()를 부른다.

**void GameObject.Updata(float)**  
**Return value** : -  
**Parameter** : eTime  
**Explain** :  
eTime에 따라 자신의 정보를 갱신한다.

**void Game.DataSender()**   
**Return value** : -  
**Parameter** : -  
**Explain** :  
GameThread()에서 불리는 함수.  
전역함수 Encoder()를 불러 얻은 배열을 game.PD[0], game.PD[1]에게 보낸다.

**ObjectData\* Encoder(GameObjectManager\*)**   
**Return value** : object\_data  
**Parameter** : GOMgr  
**Explain** :  
GOMgr.Encode를 불러 얻은 배열을 반환한다.

**ObjectData\* GameObjectManager.Encode(int)**  
**Return value** : object\_data  
**Parameter** : object\_index  
**Explain** :  
GameObjectManager.m\_GameObjects.Encode들의 Encode()를 불러 반환값들로 배열을 만들어 반환한다.

**ObjectData GameObject.Encode()**  
**Return value** : object\_data  
**Parameter** : -  
**Explain** :  
자신의 정보를 ObjectData 형식에 맞게 바꾸어 반환한다.

Key Event

**void KeySender(KeyData)**   
**Return value** : -  
**Parameter** : key\_data  
**Explain** :  
콜백함수 안에서 사용될 함수로 입력한 키와 눌림 여부를 서버에 송신한다.

**void KeyRecevier(KeyData)**   
**Return value** : -  
**Parameter** : key\_data  
**Explain** :  
각 플레이어들어게 배정된 PlayerThread()에서 사용될 함수로 Client가 입력한 KeyData를 받아 GameObjectManager.SetKeyBoardData()를 호출한다.

**void GameObjectManager.SetKeyBoardData(int, KeyData)**  
**Return value** : -  
**Parameter** : player\_num, key\_data  
**Explain** :  
입력받은 플레이어의 키 입력상태를 갱신한다.