struct fvec2{ //플레이어 좌표 구조체

float x;

float y;

}

enum Scene {

Lobby,

Char\_sel,

Main\_game,

End\_game

}

enum CharState {

Idle,

Walk,

Attack,

Attacked,

Dash,

Skill,

Win,

Lose,

}

enum ProtocalInfo{ //현재 전송하는 정보

Timer,

PlayerData,

Scene\_Data

//….

}

struct Gtimer{

float \_left\_time;

float \_skill\_cooltime1;

float \_skill\_cooltime2;

}

struct SC\_lobby\_send{ //새로운 acc 있을 때마다 보내줌

int \_acc\_count;

int \_my\_num;

}

struct player\_data{

int \_char\_type;

fvec2 \_location;

int \_look;

int \_state;

bool \_coin;

}

//공통

struct SC\_scene\_send{ // 씬데이터

int \_scene\_num;

}

struct CS\_ingame\_send{

int vkey;

// GetAsyncKeyState(vkey)로 동시키입력 여부를 확인하기 위한 변수

SHORT pressedVkey;

}

struct CS\_lobby\_send {

CS\_ingame\_send \_input;

bool \_ready;

}

1. **서버 네트워크**

int InitServer() : 소켓을 생성한다

void SendData(player\_data&) : 클라이언트에게 플레이어 데이터를 전송한다.

void WaitForClientConnect() : 접속한 순서대로 클라이언트에게 넘버를 부여하고 클라이언트가 3명이 모두 접속 했는지 확인한다.

static DWORD WINAPI RecvCLData (LPVOID) : 클라이언트로부터 데이터를 받고 업데이트한

정보를 클라이언트에 전송한다.

void SVSendTimer() : 타이머(탑 화면의 시간, 스킬 쿨타임)를 전송한다.

DWORD UpdateTimer() : 타이머 업데이트

void SVSendTotal() : 현재 전송하는 Protocal Info와 접속한 Player Data를 전송한다.

void SVSendScene() : 다음 넘길 Scene을 전송한다.

void SVSendCharSel() : 캐릭터 선택창에서의 플레이어 데이터(선택한 캐릭터, Ready 여부)를 전송한다.

void SVSendMainGm() : 메인 게임 내에서 플레이어 데이터(선택한 캐릭터, 위치, 방향, 상태, 코인 여부) 전송한다.

void UpdatePLInfo() : 플레이어 데이터(상태, 좌표값, 방향, 코인여부)를 업데이트한다.

void UpdateScene() : Scene 정보를 업데이트한다.

void PLCollision() : 플레이어와 플레이어, 플레이어와 코인, 플레이어와 장애물 사이의

충돌여부에 따라 플레이어 상태를 업데이트한다.

void PLDamaged() : 스킬 키를 입력이 들어왔을 때 플레이어가 스킬에 의한 데미지를 입었는지 검사하고 플레이어 상태를 업데이트한다.

1. **클라이언트 네트워크**

void CreateSocket() : 소켓을 생성한다

bool CLConnect() : 네트워크에 연결하고 연결 성공 유무에 따라 값을 반환한다.

void CLSendData (LPVOID) : 서버로 패킷을 전송한다.

void CLLobbySend(CS\_lobby\_send ) : 로비에서의 데이터(레디유무,선택캐릭터)를 서버에게

보낸다.

void SendKeyInfo(CS\_ingame\_send& ) : 입력한 키값을 서버에게 보낸다.

void RecvTimer() : 탑 화면 시간과 스킬 쿨타임 정보를 서버로부터 받는다.

void UpdatePL(player\_data ) : 서버로부터 넘겨받은 플레이어 정보로 해당 값을 업데이트한다.

void UpdateScene() : Scene을 업데이트한다.