**네트워크 게임 프로그래밍 텀 프로젝트 추진 계획서**

10팀

2018180022 방영규

2017180011 박정환

2019184102 박병준

목차

1.게임 소개

- 게임 개요

- 조작 방법

2.개발 환경

3.High Level

-client

-server

4.Low Level

-client

-server

5.개발 일정

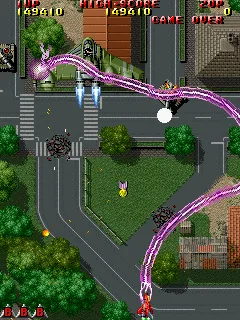
-역할 분담

-캘린더

1.1 게임 개요

클라이언트 프로그램 출처: 박정환 학생의 윈도우 프로그래밍 프로젝트

라이덴을 모티브로 한 게임이다



네트워크 게임 방향: 2인 멀티플레이 게임

게임을 실행하면 다른 플레이어와 매칭이 될 때 까지 대기한다

매칭이 성공하고 게임이 시작되면 각자 본인의 비행기를 조작하여 게임을 플레이한다

캐릭터는 스페이스바 키를 입력하여 총알을 발사한다

무작위 위치에 출현하는 적을 총알을 발사하여 제거한다

일정 점수에 도달할 때마다 환경이 변화한다(캐릭터 공격 강화, 다른 적 출현, 보스 출현)

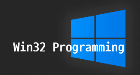
캐릭터가 적의 공격을 맞고 사망하거나 보스를 처치하면 게임이 종료된다

1.2 조작 방법

상하좌우 방향키 - 이동

스페이스바 - 공격

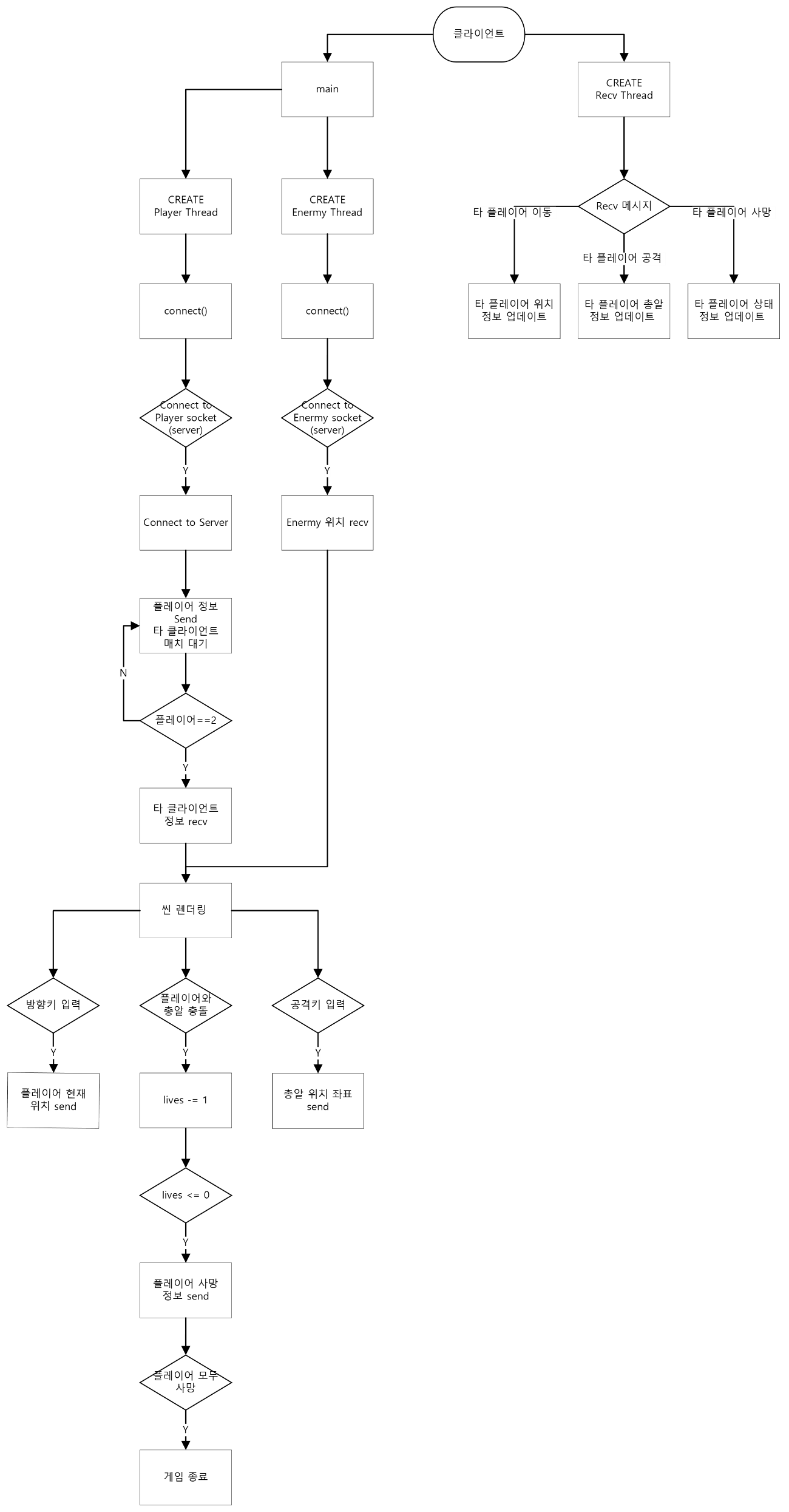
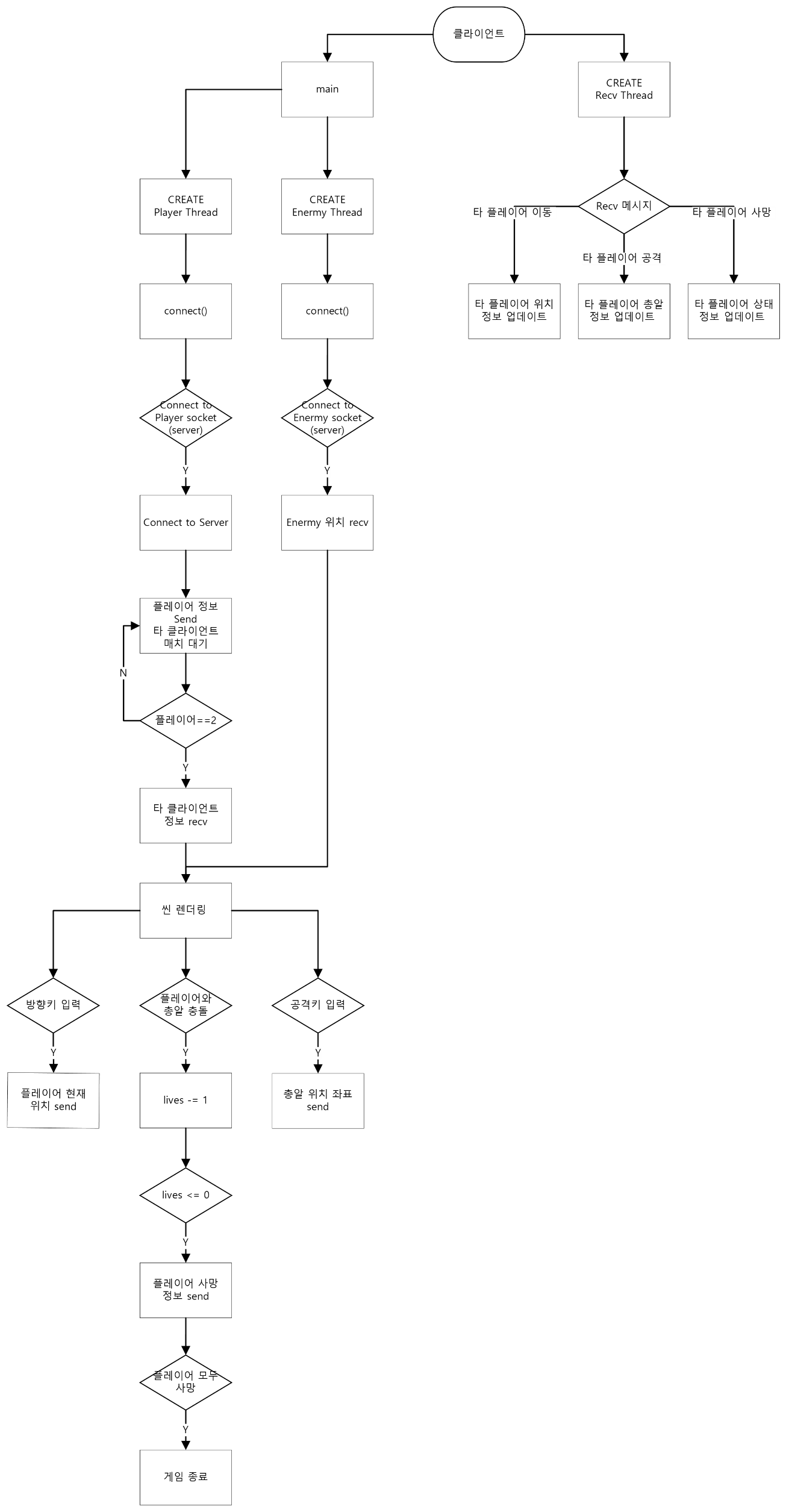
2 개발환경

Visual studio C++ TCP / IP Windows API Github Desktop

3. High Level

**Client High Level**



클라이언트는 main(), Player Thread(), Enemy Thread(), Recv Thread()로

총 4개의 Thread로 구성

main()

객체 렌더링과 서버와의 연결을 담당한다

Player Thread()

플레이어의 이동, 공격, 사망 정보를 담당한다

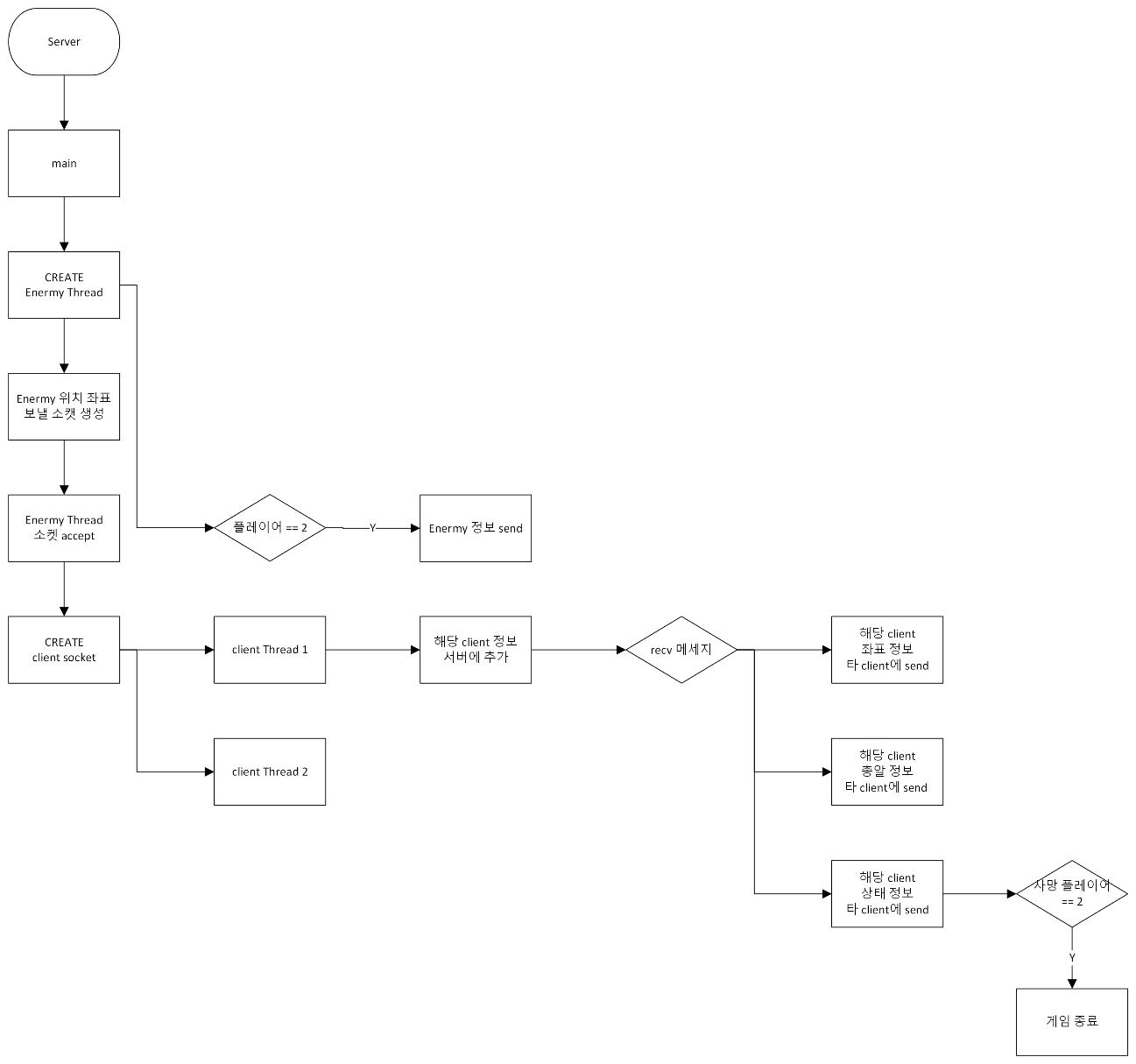
Enemy Thread()

몬스터의 생성정보를 담당한다

Recv Thread()

서버에 메시지가 왔는지 확인한다. 클라이언트 업데이트

**Sever High Level**



서버는 main(), Enemy Thread(), Client Thread() 총3개의 Thread로 구성

main()

클라이언트와 accpet하여 서버와 클라이언트를 연결한다

2명의 플레이어가 연결 될 때까지 대기한다

Enemy Thread()

무작위로 생성되는 몬스터의 위치를 생성하여 클라이언트에게 전송해준다

Client Thread()

클라이언트와 통신할 소켓을 받아 정보를 전송해준다

클라이언트에 대한 정보를 관리하고 데이터의 갱신이 필요하면 클라이언트에게 데이터를 전송해준다

4. Low Level

**4 – 1 Client Low Level**

-전역 변수

unordered\_map<USHORT, CEnemy> g\_enemy

서버로부터 받은 적의 정보를 관리하는데 사용되는 전역 변수.

CRITICAL\_SECTION g\_enemy\_cs

적의 정보를 동기화 해주기 위한 임계 영역

-구조체

struct Vec2 {float x;float y;bool dir; };

2D 공간의 벡터 또는 좌표를 표현하기 위한 구조체. 위치와 방향을 나타내는 데 사용.

struct CS\_MOVE\_PLAYER {BYTE playerID;Vec2 playerPos; };

플레이어의 이동 정보를 서버와 클라이언트 간에 전달하기 위한 구조체. 플레이어 ID, 현재 위치, 이동 방향을 포함.

struct CS\_PLAYER\_ATTACK {BYTE playerID;Vec2 bulletPos; };

플레이어의 공격 정보를 서버와 클라이언트 간에 전달하기 위한 구조체. 플레이어 ID, 총알의 초기 위치와 방향을 포함.

struct CS\_BULLET\_HIT\_ENEMY {BYTE playerID;USHORT enemyID; };

플레이어의 총알과 적의 충돌 정보를 저장하는 구조체. 충돌한 총알을 발사한 플레이어의 ID와 적의 ID를 포함.

struct CS\_PLAYER\_DEAD {BYTE playerID; };

플레이어가 사망했을 때 정보를 전달하는 구조체. 사망한 플레이어의 ID를 포함.

-함수

DWORD WINAPI SendPlayerInfo(LPVOID lpParam)

클라이언트가 서버로 플레이어의 정보를 전송하는 함수. 플레이어의 이동과 공격, 상태 정보를 전송하는 역할.

DWORD WINAPI RecvEnemyInfo(LPVOID lpParam)

서버에서 적의 정보를 수신하여 업데이트하는 함수. 적의 위치나 상태 변화를 수신하여 실시간으로 반영하는 역할.

DWORD WINAPI RecvThread(LPVOID lpParam)

서버로부터 다른 클라이언트의 데이터를 수신하는 함수. 다른 클라이언트의 위치정보, 총알정보, 상태정보를 수신하고 갱신하는 역할.

void PlayerMove(CS\_MOVE\_PLAYER& player)

플레이어의 이동을 처리하는 함수. 플레이어의 새로운 위치와 방향을 설정하고, 이를 서버에 전송하여 이동 상태를 업데이트.

void PlayerAttack()

플레이어의 공격을 처리하는 함수. 새로운 총알을 생성하고 서버에 해당 정보를 전송.

bool IsPlayerDead(const int lives)

플레이어의 남은 생명 수를 확인하는 함수. 클라이언트 내에서 생명 수를 관리하며, 생명 수가 0 이하일 경우 true를 반환하고, 플레이어가 사망했음을 서버에 전송.

**4 – 2 Sever Low Level**

-전역 변수

unordered\_map<USHORT, SESSION> g\_clients

클라이언트를 아이디에 따른 소켓으로 통신하기 위해 unordered\_map으로 클라이언트 소켓을 관리해줄 전역변수.

unorderd\_map<CEnermy> g\_enermy

SendEnermyInfo에서 생성한 정보를 아이디에 대해서 관리해줄 적 전용 전역변수.

CRITICAL\_SECTION cs

서버 동기화를 위한 임계영역

-구조체

struct SEND\_PLAYER\_MOVE {BYTE playerID;Vec2 playerPos; };

플레이어 이동 정보를 보내기 위한 패킷

플레이어 ID, 현재 위치, 이동 방향을 포함.

struct SEND\_PLAYER\_DEAD {BYTE playerID;};

플레이어 사망 시 그 플레이어를 삭제 요청하기 위한 패킷

사망한 플레이어의 ID를 포함

struct SEND\_PLAYERPOS {BYTE playerID;vec2 playerpos}

플레이어 시작 위치를 보내기 위한 패킷

플레이어 ID, 현재 위치를 포함.

struct SEND\_ENERMYPOS {BYTE playerID;USHORT enermyID;}

클라이언트에게 적 스폰 위치를 보내기 위한 패킷

적 ID와 보내줄 플레이 ID를 포함.

-함수

DWORD WINAPI SendEnermyPos (LPVOID lpParam)

서버에서 생성된 적의 위치를 연결된 클라이언트에게 전송하여

적의 위치가 똑같이 움직이도록 한다

DWORD WINAPI ClientThread (LPVOID lpParam)

클라이언트에게 데이터를 전송받고 다른 클라이언트에게 넘겨준다.

플레이어 이동, 사망, 총알, 시작위치 정보를 주고받는다.

bool SendMovePlayer, RecvMovePlayer(const CS\_MOVE\_PLAYER& packet)

플레이어 이동 정보를 클라이언트한테 받고 다시 보내줌으로서 모든 유저가 다른 유저의 움직임을 볼 수 있도록 한다.

bool SendRemovePlayer, RecvRemovePlayer(USHORT cid)

플레이어 사망 정보를 주고받아서 캐릭터의 사망을 실시간으로 동기화한다.

void SendPlayerBullet, RecvPlayerBullet(BYTE playerID;Vec2 bulletPos;)

플레이어 총알 정보를 클라이언트한테 받고 다시 보내줌으로서 모든 유저가 다른 유저의 총알을 볼 수 있도록 한다.

Void SendInitPos(BYTE playerID;Vec2 playerPos;)

플레이어 시작 위치 전송.

플레이어 ID와 플레이어 위치 포함.

5. 개발 일정

-역할 분담

|  |  |  |
| --- | --- | --- |
| 이름 | 클라이언트 | 서버 |
| 방영규 | PlayerMove()  IsPlayerDead()  Send\_PlayerInfo() | SendPlayerBullet()  recvPlayerBullet() |
| 박정환 | Recv\_EnermyInfo() | Send\_MonsterPos()  recvMovePlayer()  Client |
| 박병준 | RecvThread()  PlayerAttack() | SendRemovePlayer()  recvRemovePlayer()  SendInitPos()  SendMovePlayer() |

-캘린더

방영규

박정환

박병준

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 월 | 화 | 수 | 목 | 금 | 토 | 일 |
|  |  | 10/30  문서 제출 | 31 | 11/1 | 2  PlayerMove | 3  PlayerAttack |
| 4  Send\_MonsterPos  PlayerMove | 5 | 6 | 7  PlayerAttack | 8 | 9  Send\_PlayerInfo  Pos | 10  SendMovePlayer  Recv\_EnermyInfo |
| 11 | 12  SendMovePlayer | 13 | 14  Send\_PlayerInfo  Pos  Recv\_EnermyInfo | 15  Client | 16  SendRemovePlayer | 17  SendPlayerBullet |
| 18 | 19 | 20  Client | 21  SendPlayerBullet  SendRemovePlayer | 22  RecvThread  recvPlayerBullet | 23  recvMovePlayer | 24 |
| 25  recvPlayerBullet | 26 | 27 | 28  RecvThread  recvMovePlayer | 29  recvRemovePlayer | 30 | 12/1  IsPlayerDead  SendInitPos |
| 2 | 3 | 4 | 5 | 6 | 7  코드 리팩토링 | 8 |
| 9  코드 리팩토링 | 11 | 12 |  |  |  |  |