P o r t f o l i o

김기윤

서 버 프 로 그 래 머

Phone. 010 - 5100 - 2974

E-mail. kimkiyoun33@gmail.com

Education. 한국산업기술대학교 게임공학과 졸업예정

Birth. 1997. 09.10

Address. 경기도 의정부시

Skills

언어

- C
- C++
- python
- Lua Script
- java script
- react

클라이언트

- DirectX 12
- OpenGL
- Unity
- Unreal

서버

- IOCP
- 멀티쓰레딩
- Ms SQL
- PostgreSQL
- Overlapped I/O
- Select Thread

Etc.

- github
- Google cloud platform

CONTENTS

01 SSU(졸업작품) - 팀 프로젝트

개발 환경

- 사용 언어 : C, C++, Lua
- 클라이언트 : Direct X 12
- 서버 : IOCP
- MS SQL
- Unity(Height Map추출 및 오브젝트 위치 잡기)

02 게임 서버 프로그래밍 - 개인 프로젝트

개발 환경

- 사용 언어 : C, C++, Lua
- 클라이언트 : SFML
- 서버 : IOCP
- MS SQL

03 네트워크 게임 프로그래밍 - 팀 프로젝트

개발 환경

- 사용 언어 : C, C++
- 클라이언트 : 윈도우 프로그래밍
- 서버 : Socket Thread

00 Test 프로그램 하드웨어 성능

- CPU : AMD RYZEN 7 5800H
- RAM : 16GB

SSU(졸업작품)

01

게임 장르	MMORPG
작업 기간	2022.01~2022.07
작업 인원	3명 - 클라이언트 1명 - 서버 2명
구현내용 (역할)	서버 프로그래머 - 길찾기(A* 알고리즘) - 파티 시스템 - Raid 보스 - 섹터 분할(최적화 및 동접 향상)
깃허브	https://github.com/hjs0913/SSU

Npc 길찾기(A* 알고리즘)


```
#define REAL_DISTANCE 10
```

한칸은 실제 World에서 10만큼의 거리

```
// 쫓아가는 범위는 한 방향으로 120까지이다
pos now(12, 12);

int scoreG[25][25] = { 0 };
int scoreH[25][25] = { 0 };
int scoreF[25][25] = { 0 };
```

초기화 시 Now를 기준으로 좌우상하 12개씩으로 잡음

(최대 거리는 Now에서 총 12*REAL_DISTANCE = 120)

(REAL_DISTANCE를 늘리면 좌표가 정확하지 않고 줄이면 계산량이 늘어난다)

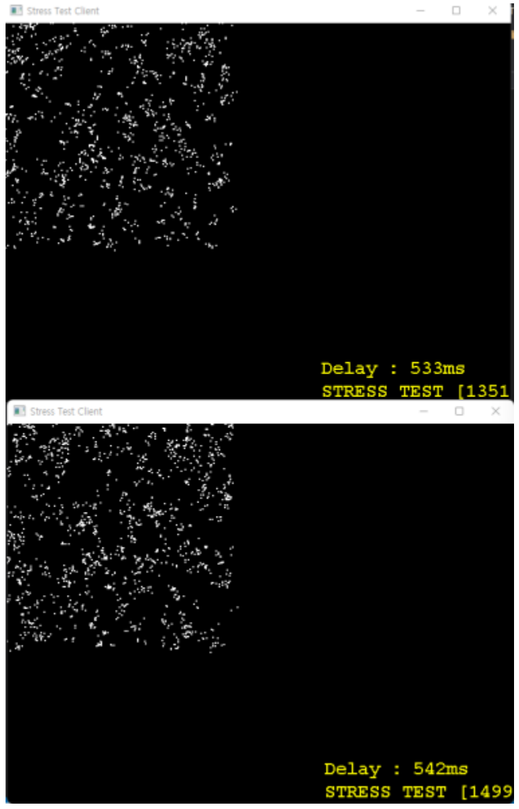
```
priority_queue < weight, vector<weight>, greater<weight>> open_q;
// 장애물이랑 부딪히는지 확인
if (false == check_move_alright(x + (p.first - 12) * REAL_DISTANCE, z + (p.second - 12) * REAL_DISTANCE, true, obstacles)) continue;

scoreG[now.first + dirX[i]][now.second + dirZ[i]] = scoreG[now.first][now.second] + cost[i];
scoreH[now.first + dirX[i]][now.second + dirZ[i]] = huristic(t_x, t_z, x + (p.first - 12) * REAL_DISTANCE, z + (p.second - 12) * REAL_DISTANCE);
scoreF[now.first + dirX[i]][now.second + dirZ[i]] = scoreG[now.first + dirX[i]][now.second + dirZ[i]] +
    scoreH[now.first + dirX[i]][now.second + dirZ[i]];

prior_point[now.first + dirX[i]][now.second + dirZ[i]] = pos(now.first, now.second);

weight w(scoreF[now.first + dirX[i]][now.second + dirZ[i]], pos(now.first + dirX[i], now.second + dirZ[i]));
open_q.push(w);
```

check_move_alright로 해당 부분에 장애물이 있다면 탐색을 하지 않는다.

Open_q에서 가장 ScoreF가 작은 값을 꺼내오며 prior_point로 길의 경로를 확인할 수 있도록 만들었다.

섹터분할(최적화 및 동접향상)

●섹터 분할 전(동접 2500)

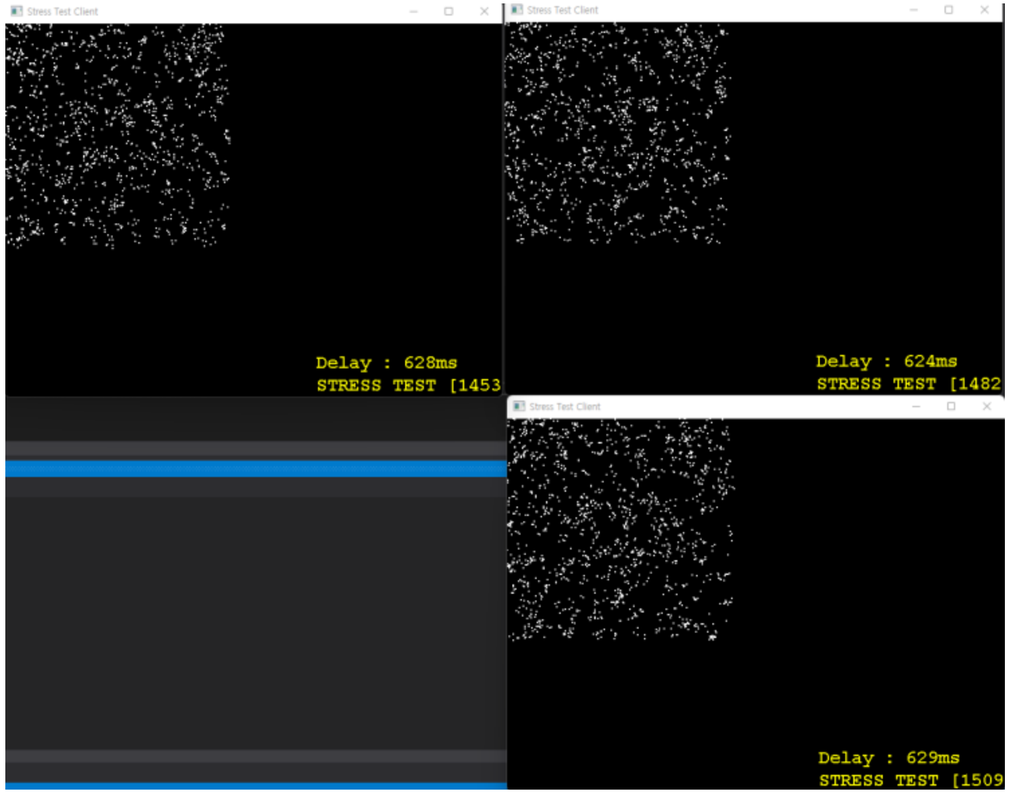


Put(Login)/Move시 시야처리

-> 모든 플레이어를 상대로 시야처리

((접속해 있는 플레이어 + Npc의 갯수) 번 탐색)

●섹터 분할 후(동접 4200)



Put(Login)/Move시 시야처리

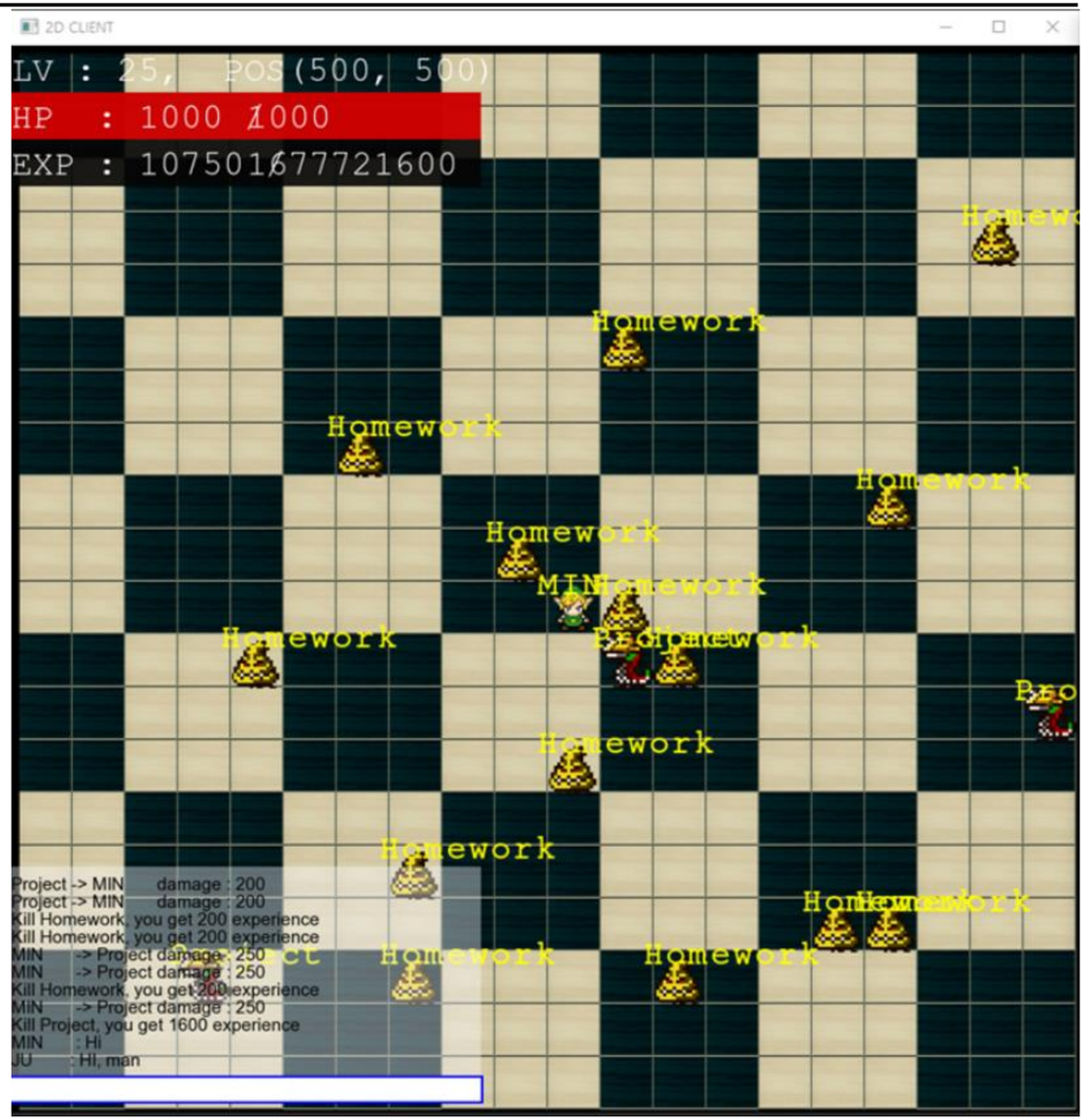
-> 전체 지역을 64등분 하여 **자신이 해당하는 섹터와**

그 주위의 8개의 섹터에 있는 플레이어에 한해 시야처리

(플레이어가 골고루 퍼져 있다는 가정하에 ((9/64)*섹터분할 전 탐색 수) 만큼 탐색)

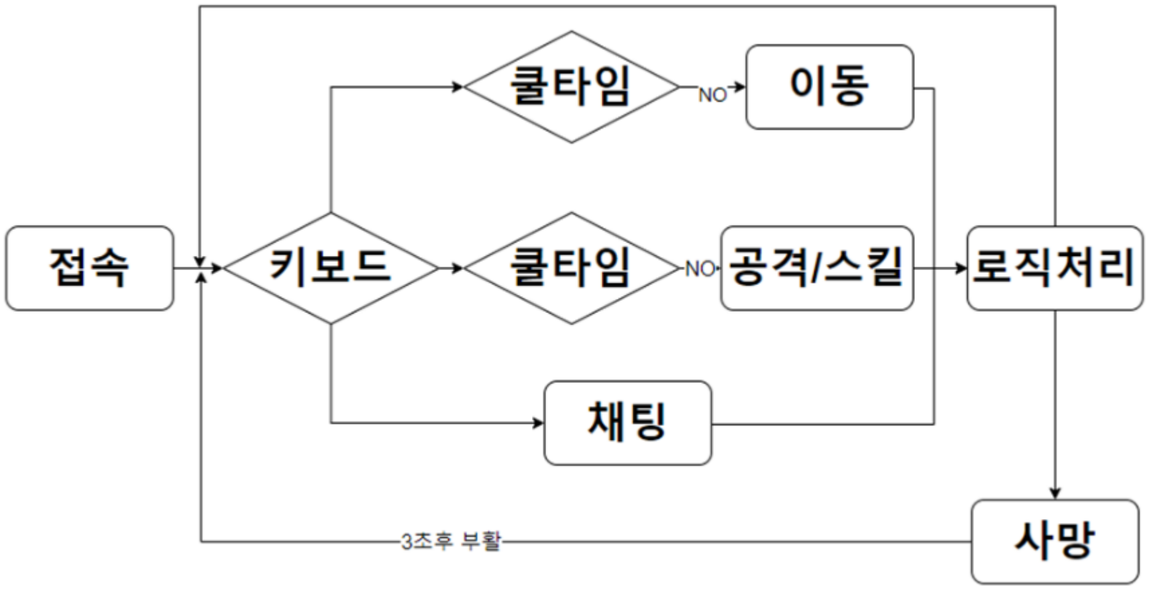
게임 서버 프로그래밍

게임 장르	2D MMORPG
작업 기간	2021.11~2021.12
작업 인원	1명(개인 프로젝트)
구현내용	기술 - 멀티쓰레드 IOCP이용 - DB(My SQL, SSMS 사용) - Lua Script 이용 컨텐츠 - 고정형 몬스터(때려야지 따라오는 몬스터) - Agro 몬스터(범위에 들어오면 따라오는 몬스터) - 기본 공격과 스킬 2개 - 채팅 기능
깃허브	https://github.com/Asias-ara/2021-2GameServer-TermProject

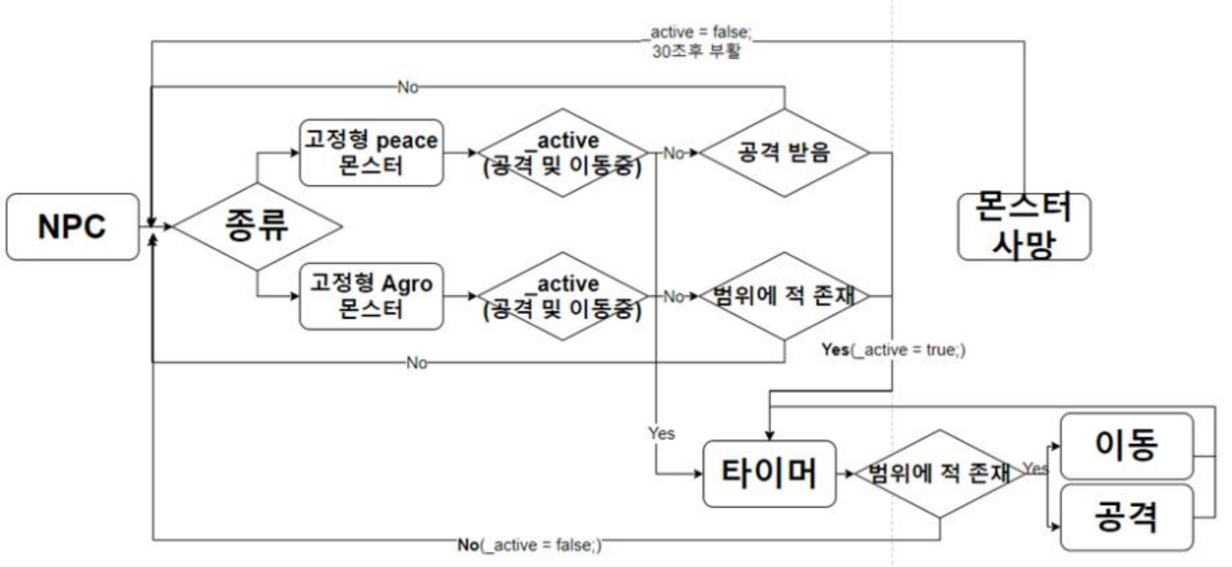


Game Flow Chart

Game Flow Chart (플레이어)



Game Flow Chart (Npc)

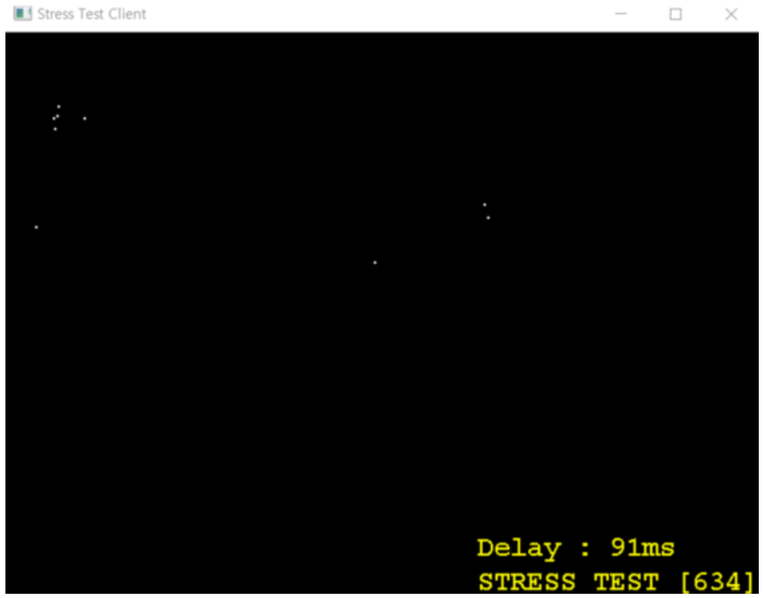


동접테스트

02

루프백 상황 - 평균 700명

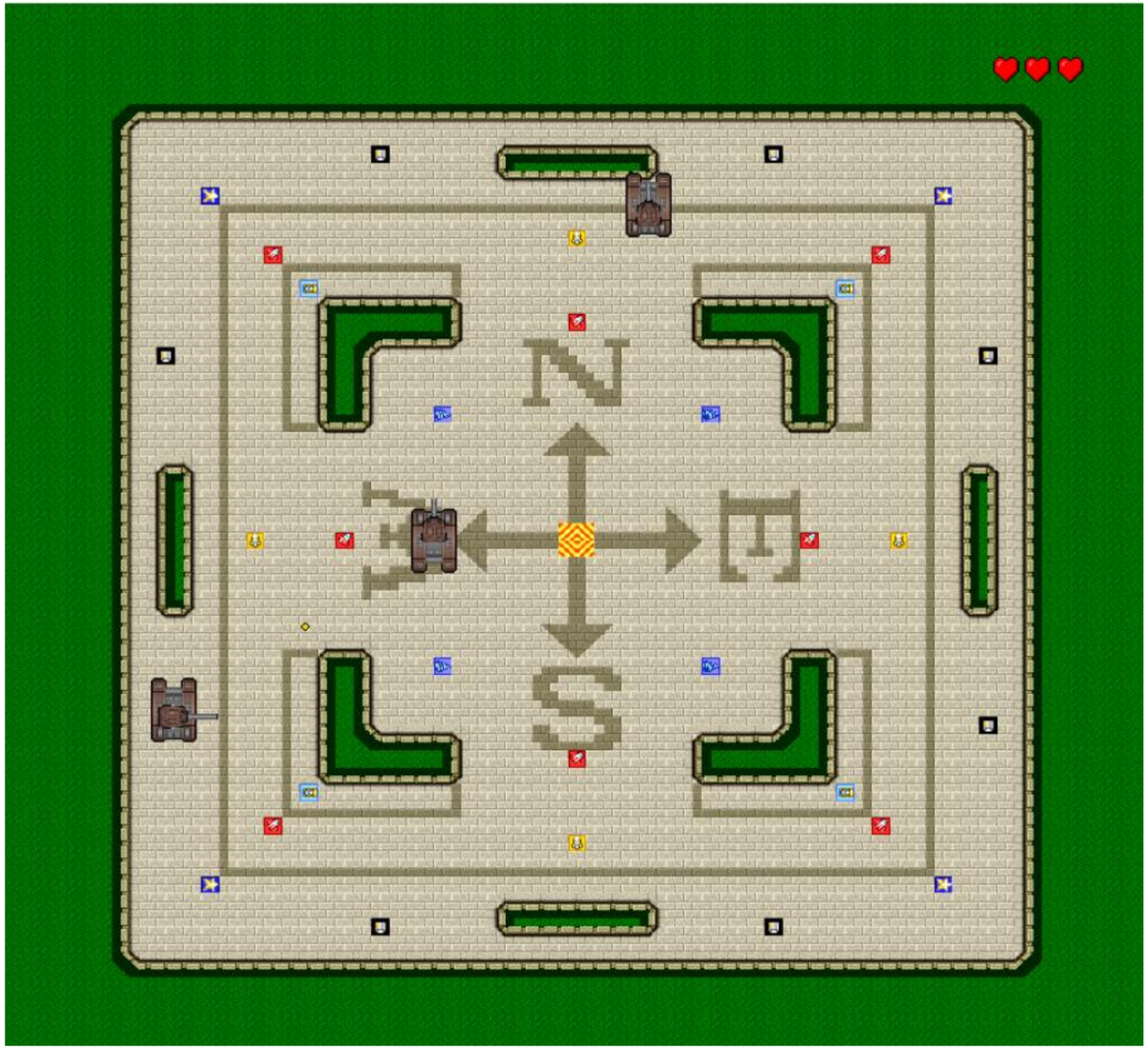
리모트 상황 - 평균 500명



- 루프백과 리모트간의 성능 차이가 심하다는 것을 알게 되었다
- Npc 20만 마리(그 중 Agro몬스터 1만마리)이므로 루프백일 경우에도 플레이어가 어디 위치에 있는가에 따라 최소 400 최대 1400의 성능 차이가 나는 것을 볼 수 있었다

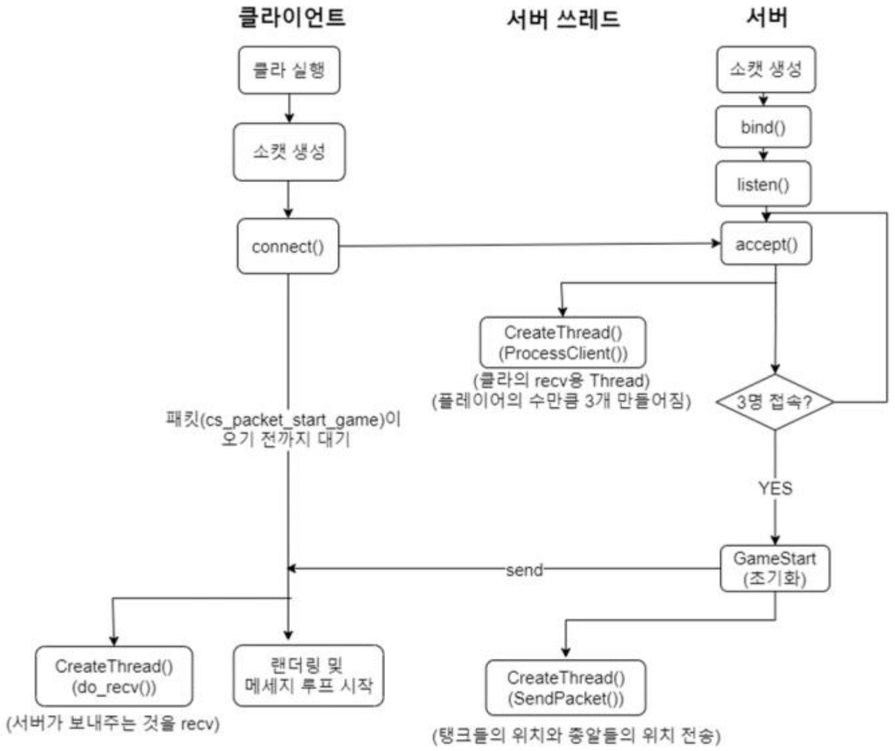
네트워크 게임 프로그래밍

게임 장르	3인 2D 어드벤처 게임
작업 기간	2021.11~2021.12
작업 인원	3명 - 클라이언트 1명 - 서버 2명
구현내용 (역할)	서버 프로그래머 - 게임 로그인 처리 - 서버 스레드간 동기화 - 총알 충돌처리
깃허브	https://github.com/Asias-ara/NGP_TermProject



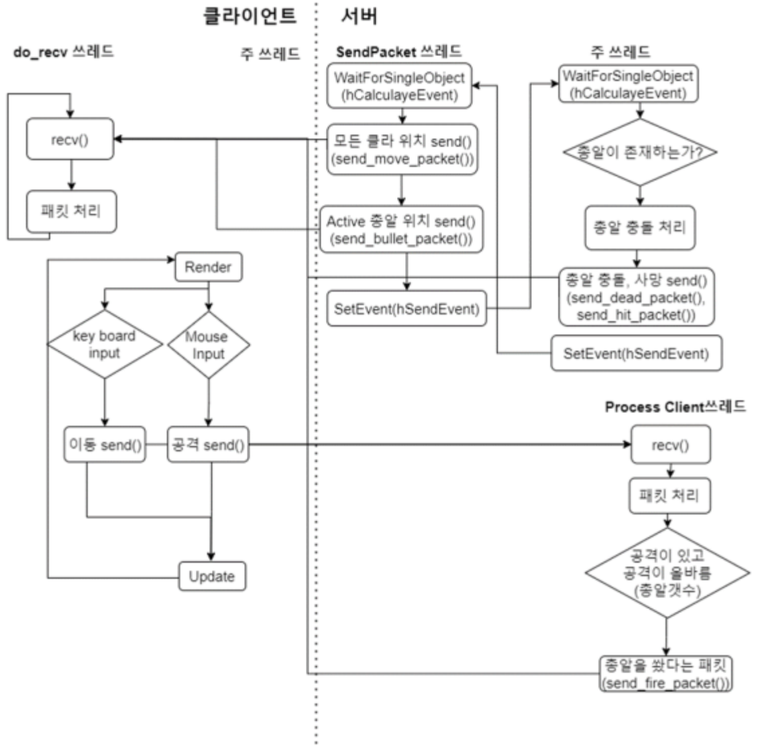
Game Flow Chart

Login Flow Chart



- 클라이언트가 접속시 각각에 대응하는 Thread(ProcessClient())를 만들며 이 쓰레드는 클라이언트가 보내는 패킷을 받는 역할을 한다
- 세명이 접속시 패킷을 보내는 Thread(SendPacket())를 하나 더 만들어 준다
- 마지막에는 총 5개의 쓰레드(주 쓰레드 1개, ProcessClient 쓰레드 3개, SendPacket 쓰레드 1개)가 생긴다

Game Flow Chart



- 팀원과 논의를 통해 가장 중요하게 생각한 것은 **충돌처리를 하는 주 쓰레드**이며 send는 주 쓰레드 한 후 처리하는게 옳다고 생각하여 SendPacket쓰레드와 주쓰레드를 **Event방식으로 공유자원에 대한 동기화**를 하였다.