**네트워크 게임 프로그래밍**

**Term-Project 기획서**

팀원

**1. 어플리케이션 기획**

- 게임 이름

Tank Fighters

- 게임 장르

멀티 탱크 대전게임

-게임 특징

최대 4인 참여가 가능한 탱크 대전 게임

DirectX 9을 사용

조작법 : 방향키, 스페이스 바

- 게임 방법

최대 4명의 플레이어의 탱크가 전투를 하고 최후에 살아남은 플레이어가 승리하게 된다.

- 게임 방식, 구조

Y

N

Y

Y

N

N

방향키를 눌럿는가

이동

포탄 발사

다른 플레이어가 모두 죽었는가

게임 종료

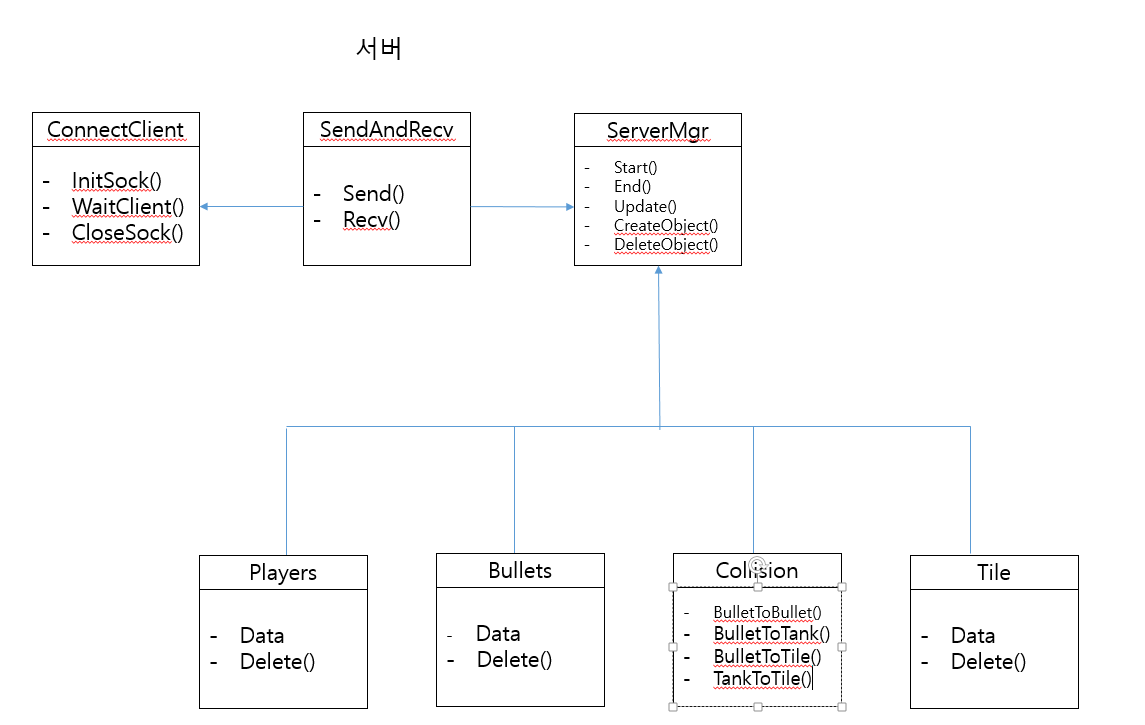
공격키를 눌럿는가

업데이트

게임 시작

**2. HIGH LEVEL DESIGN**

- 서버



* ConnectClient

클라이언트와의 접속을 위해서 InitSock()에서 소켓을 초기화,생성 해주고 WaitClient()에서 서버의 상태를 listen상태로 만든다.

* SendAndRecv

클라이언트와의 다양한 정보(체력, 좌표, 키값)를 송수신한다.

* ServerMgr

서버매니져로 게임의 시작과 종료, 객체의 생성과 파괴, 게임에서 발생하는 여러 상태들을 Update한다.

- Players

플레이어의 위치, 체력 정보

플레이어의 체력이 0이됬을 때 삭제한다.

- Bullets

포탄의 위치, 방향 정보

포탄이 화면을 벗어났을 경우, 탱크에 충돌했을 경우,타일에 맞았을 경우에 삭제한다.

- Collision

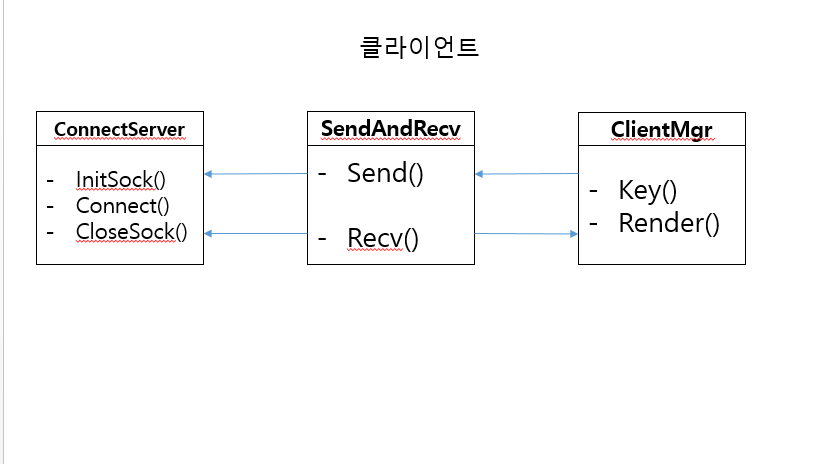
탱크와 포탄과 타일간의 충돌 판단

- Tile

타일 상태의 정보

포탄이 타일에 맞았을 경우에 삭제한다.

- 클라이언트



* ConnectServer

InitSock()으로 소켓을 초기화하고, Connect()로 서버와 접속을 시도한다.

* SendAndRecv

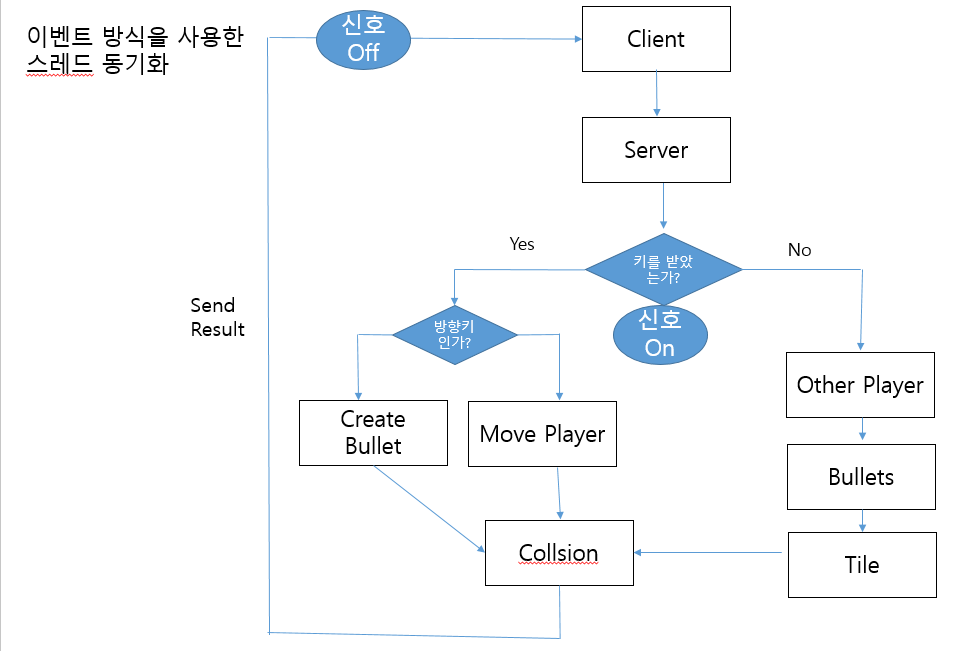
서버와 정보(체력, 좌표, 키값)를 송수신한다.

* ClientMgr

플레이어가 입력한 키정보를 전송하는 한편, 서버로부터 전송받은 정보를 Render해준다.

- 스레드 구조

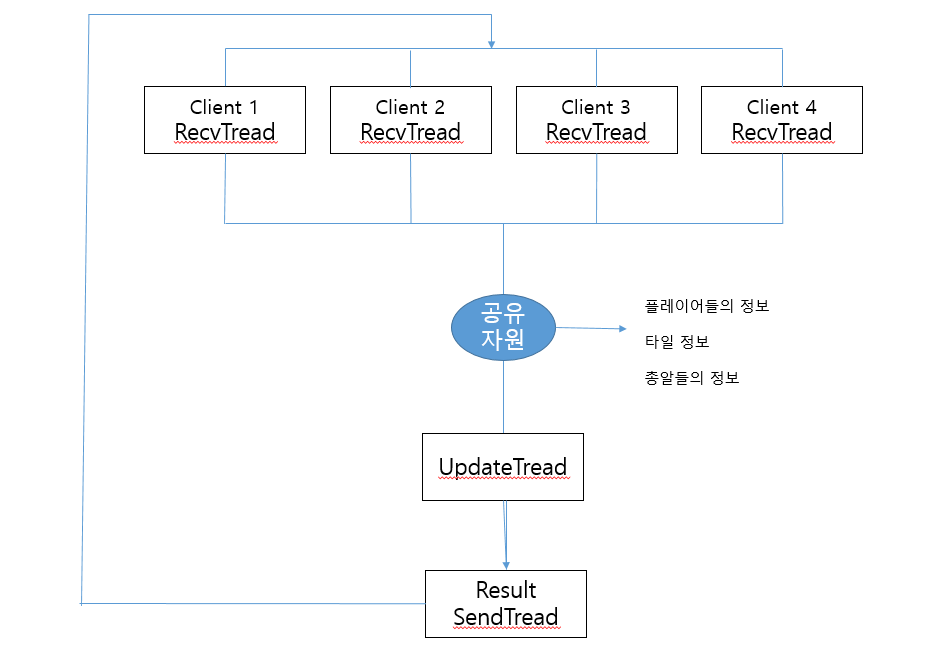
전체 동기화 구조



이벤트방식을 이용하여 스레드를 동기화

Update도중에 클라이언트에서 키를 입력 받으면 이벤트를 신호 상태로 바꾼다. 그후 서버에서 키 값 데이터를 공유자원에 저장을 한다. 이후에 그 키 값을 판단하여 총알, 플레이어, 타일들의 상태를 변환 , 충돌체크 단계를 거친 후에 이벤트를 비 신호 상태로 변환한다.

세부구조



클라이언트에서 RecvTread를 통해 정보를 받는다.

공유자원은 UpdateTread를 통해 Update 후에 최종 결과물을 SendTread로 클라이언트들에게 다시 전달(브로드캐스팅)한다.

**3. LOW LEVEL DESIGN**

서버 개요 :

TCP 서버 사용

스레드 동기화 : 이벤트 방식

바이트 정렬 방식 : 리틀 엔디안

전송방식 : 고정길이 + 가변길이

**4. 개발환경**

Windows 10 64 bit

Intel CPU

Visual studio 2017

DirectX 9

**5. 팀원 별 역할 분담**

|  |  |  |  |
| --- | --- | --- | --- |
| 역할  날짜 | 홍승필 | 황서윤 | 유창민 |
| 11/9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |
| 19 |  |  |  |
| 20 |  |  |  |
| 21 |  |  |  |
| 22 |  |  |  |
| 23 |  |  |  |
| 24 |  |  |  |
| 25 |  |  |  |
| 26 |  |  |  |
| 27 |  |  |  |
| 28 |  |  |  |
| 29 |  |  |  |
| 30 |  |  |  |
| 12 /1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 |  |  |  |
| 16 |  |  |  |
| 17 |  |  |  |

일정 확인표

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 |  | 10 |  | 11 |  | 12 |  | 13 |  | 14 |  | 15 |  |
| 16 |  | 17 |  | 18 |  | 19 |  | 20 |  | 21 |  | 22 |  |
| 23 |  | 24 |  | 25 |  | 26 |  | 27 |  | 28 |  | 29 |  |
| 30 |  | 1 |  | 2 |  | 3 |  | 4 |  | 5 |  | 6 |  |
| 7 |  | 8 |  | 9 |  | 10 |  | 11 |  | 12 |  | 13 |  |
| 14 |  | 15 |  | 16 |  | 17 |  |  |  |  |  |  |  |