네트워크 게임 프로그래밍

텀 프로젝트 추진계획서

윤승규

신동원

김석현

목차

[1. 게임 기획 3](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084126)

[1.1 게임 개요 3](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084127)

[1.2 조작 방법](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084128) 3

[1.3 구성 요소](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084129) 4

[2. High Level Design 5](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084131)

[2.1 클라이언트 5](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084132)

[2.2 서버 6](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084133)

[3. Low Level Design 7](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084136)

[3.1 프로세스 7](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084137)

[3.2 함수 8](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084139)

[4. 팀원 역할 9](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084143)

[5. 개발 일정 10](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084145)

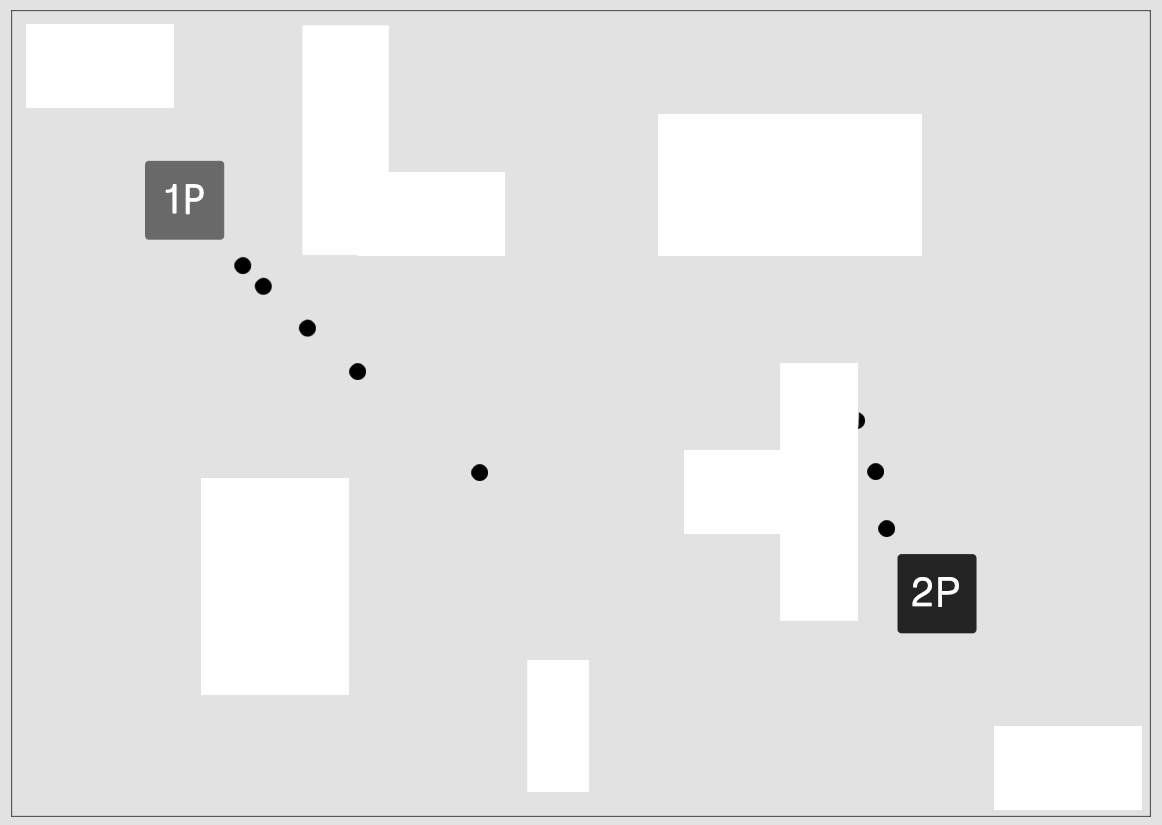
[5.1 윤승규 일정 10](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084146)

[5.2 신동원 일정 10](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084147)

[5.3 김석현 일정 10](file:///C:\Users\tjrgu\Desktop\학교\넷겜플\새%20폴더\네트워크%20게임%20프로그래밍%20팀프.docx#_Toc55084148)

1. **게임 기획**

**1.1 게임 개요**



2명의 플레이어가 총알을 발사해 상대방을 공격하며, 제한시간 동안 가장 적게 사망한 플레이어가 승리하는 멀티 플레이 게임이다. 하얀 벽에서는 총알이 막히며, 플레이어가 벽 뒤에 숨는 등의 전략적인 행동이 가능하다.

* 1. **조작 방법**

|  |  |
| --- | --- |
| 조작키 | 설명 |
| W, A, S, D | 플레이어의 이동 방향키 W-위/A-왼/S-뒤/D-오 |
| ↑,←,↓,→ | 플레이어의 탄환이 해당 방향으로 발사되는 공격 키 |

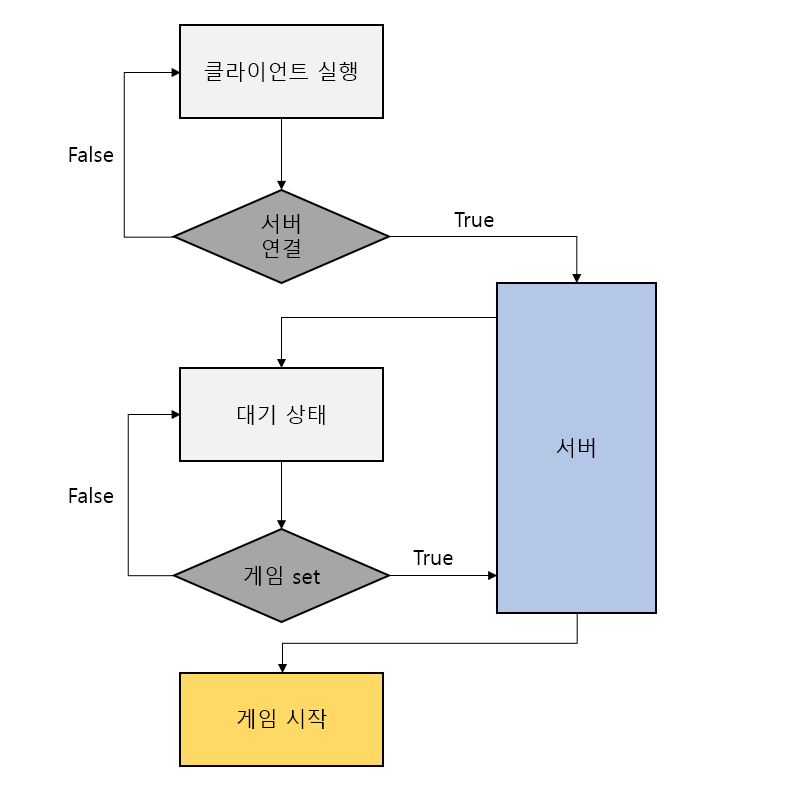
* 1. **구성 요소**

|  |  |  |
| --- | --- | --- |
|  | 이름 | 설명 |
|  | 1 Player | 플레이어 1 이 조종하는 캐릭터이며 탄환에 닿거나, 다른 플레이어에게 닿으면 피해를 입거나 사망한다. |
|  | 2 Player | 플레이어 2 이 조종하는 캐릭터이며 탄환에 닿거나, 다른 플레이어에게 닿으면 피해를 입거나 사망한다. |
|  | 벽 | 플레이어가 이동할 수 없게 만든 벽  탄환은 벽을 통과할 수 없어, 플레이어 간의 전투시 전략적으로 이용이 가능하다. |
|  | 탄환 | 플레이어의 조작키로 발사되는 탄환이다.  탄환에 맞은 플레이어는 피해를 입고, 일정 피해를 입으면 플레이어는 사망한다. |

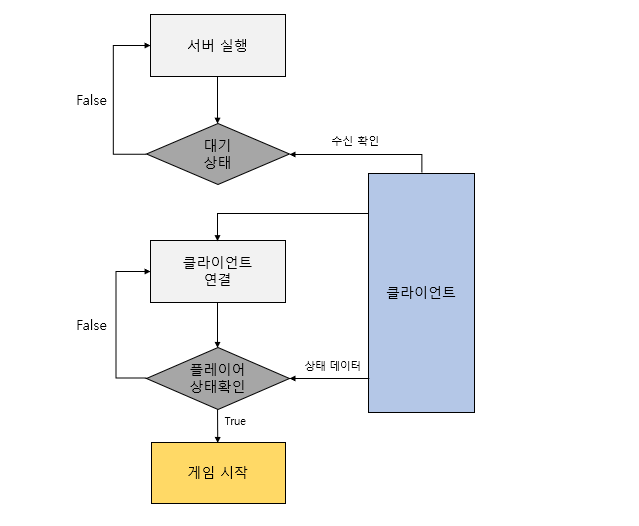
1. **High level design**

**2.1 클라이언트**

UDP 프로토콜 방식

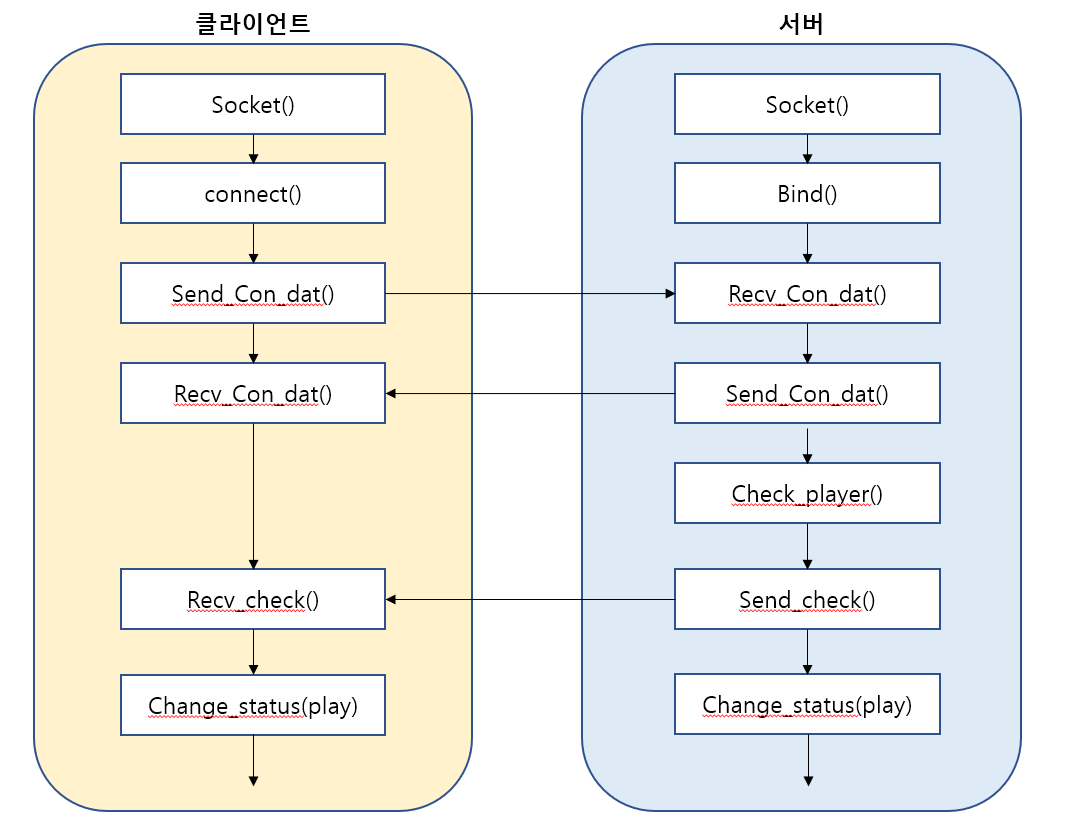


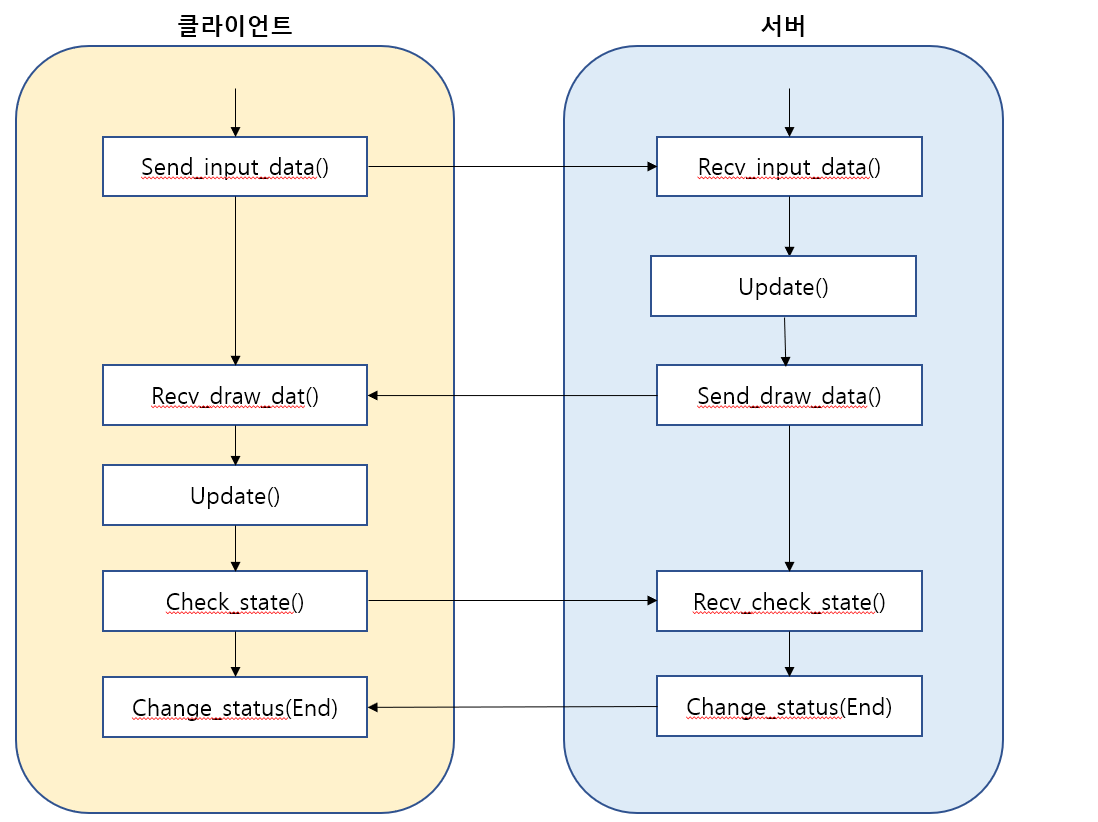
* 1. **서버**



1. **Low level design**

**3.1 프로세스**





* 1. **함수**

**클라이언트**

|  |  |  |
| --- | --- | --- |
| 구분 | 함수 | 설명 |
| 송신 | Recv\_Con\_dat() | 서버 로부터 데이터 소켓을 전달받는 함수 |
| Recv\_draw\_data() | 서버에서 받은 Update 데이터를 그리는 함수 |
| Recv\_check() | Player가 모두 준비되었는지 데이터를 받는 함수 |
| 수신 | Send\_Con\_dat() | 서버에 데이터 소켓을 전달하는 함수 |
| Send\_input\_data() | 입력 받은 데이터를 서버에 전달하는 함수 |
| 처리 | Change\_status() | 게임의 상태를 play, wait, end 로 바꿔주는 함수 |
| Update() | 입력 받은 데이터를 확인해 최신화 하는 함수 |
| Check\_state() | 게임의 진행 상황을 서버가 확인하도록 요청하는 함수 |
| Timeout() | 일정시간 안에 입력 데이터의 전송이 되지 않고 지연 시 False로 처리하는 함수 |

서버

|  |  |  |
| --- | --- | --- |
| 구분 | 함수 | 설명 |
| 송신 | Recv\_Con\_dat() | 클라이언트 로부터 데이터 소켓을 전달받는 함수 |
| Recv\_input\_data() | 클라이언트 로부터 입력 데이터를 받는 함수 |
| Recv\_check\_state() | 클라이언트로부터 실시간으로 게임 상태를 확인하도록 요청 받아 확인하는 함수 |
| 수신 | Send\_Con\_dat() | 클라이언트 에 데이터 소켓을 전달하는 함수 |
| Send\_draw\_data() | 서버에서 Update 처리한 데이터를 클라이언트에 보내는 함수 |
| send\_check() | 2명의 player의 상태를 보내는 함수 |
| 처리 | Change\_status() | 게임의 상태를 play, wait, end 로 바꿔주는 함수 |
| Check\_player() | 2명의 player가 서버와 연결되어 준비상태인지 확인하는 함수 |
| Update() | 입력 받은 데이터를 확인해 최신화 하는 함수 |
| Check\_state() | 게임의 진행 상황을 서버가 확인하도록 요청하는 함수 |
| Timeout() | 일정시간 안에 입력 데이터의 전송이 되지 않고 지연 시 False로 처리하는 함수 |

1. **팀원 역할**

|  |  |
| --- | --- |
| 이름 | 역할 |
| 윤승규 | 서버 송/수신 함수 및 클라이언트 송신 함수 구현 |
| 신동원 | 서버 프레임 워크 및 송신 함수 구현 |
| 김석현 | 클라이언트 프레임워크 및 수신함수 구현, 문서 작성 |