**네트워크 프로그래밍  
Term Project Progress Report**

2020182042 최준하  
2020182004 고태경  
2020180026 이상유

**목차**

1. 애플리케이션 기획

2. High-Level 디자인

3. Low-Level 디자인

4. 팀원 별 역할 분담

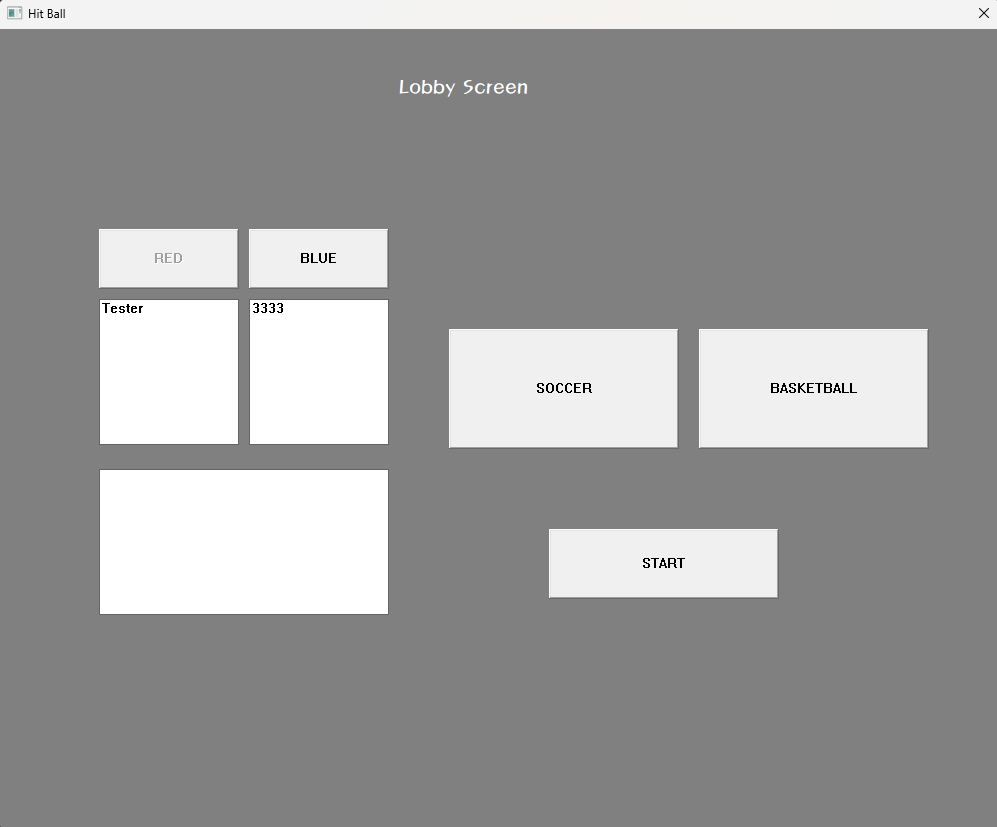
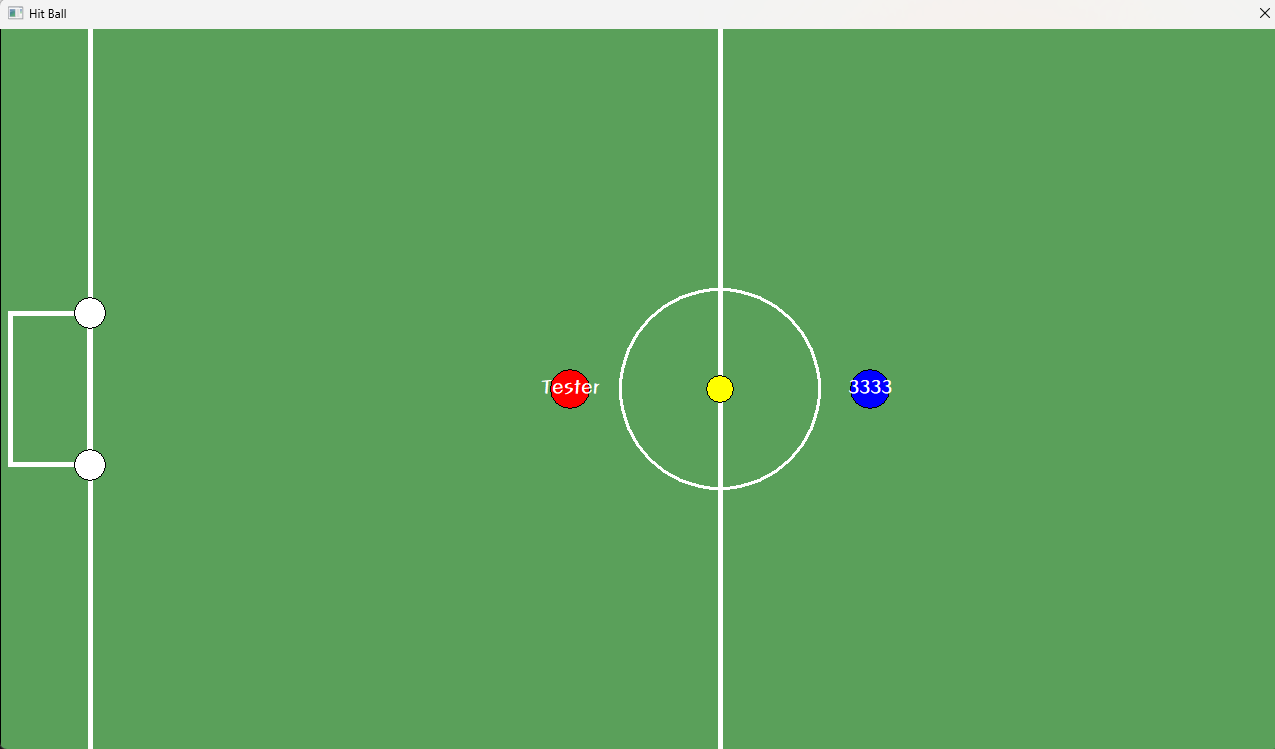
5. 개발 환경

6. 개인 별 개발 일정

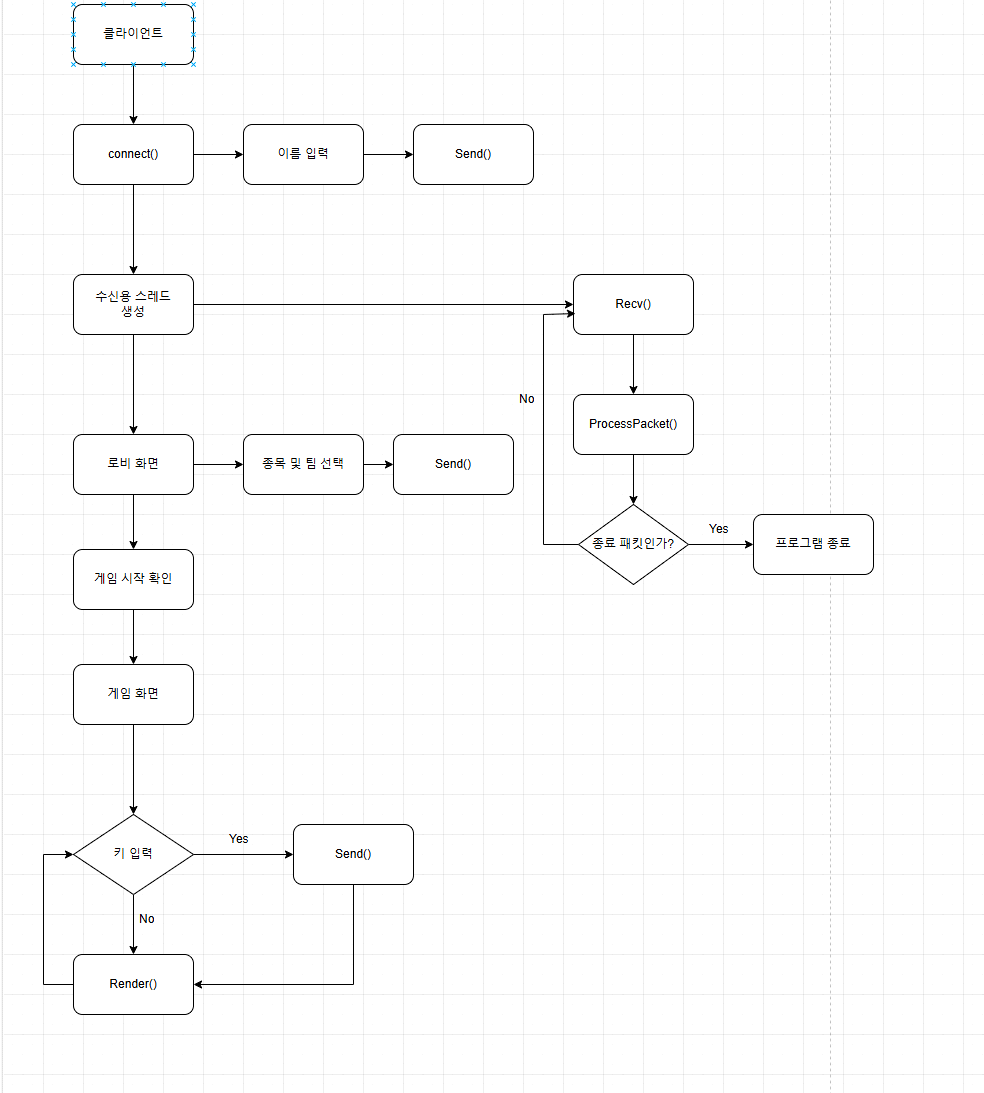
**애플리케이션 기획**  
고태경 학우가 2021-1 윈도우 프로그래밍에서 C++로 제작한 ‘Hit Ball’ 게임을 이용하여 프로젝트를 제작할 예정입니다.  
  
**게임소개**  
해당 게임은 ‘HaxBall’ 이라는 게임을 바탕으로 제작한 게임입니다.  
n대m 게임이며 방향키로 조작하여 상대방 골대에 골을 넣으면 이기는 게임입니다.  
탑뷰 시점으로 축구와 농구 2가지 종류를 플레이 할 수 있습니다.

**조작 방법**  
방향키로 이동하며, 스페이스바를 누르면 슈팅을 할 수 있습니다.

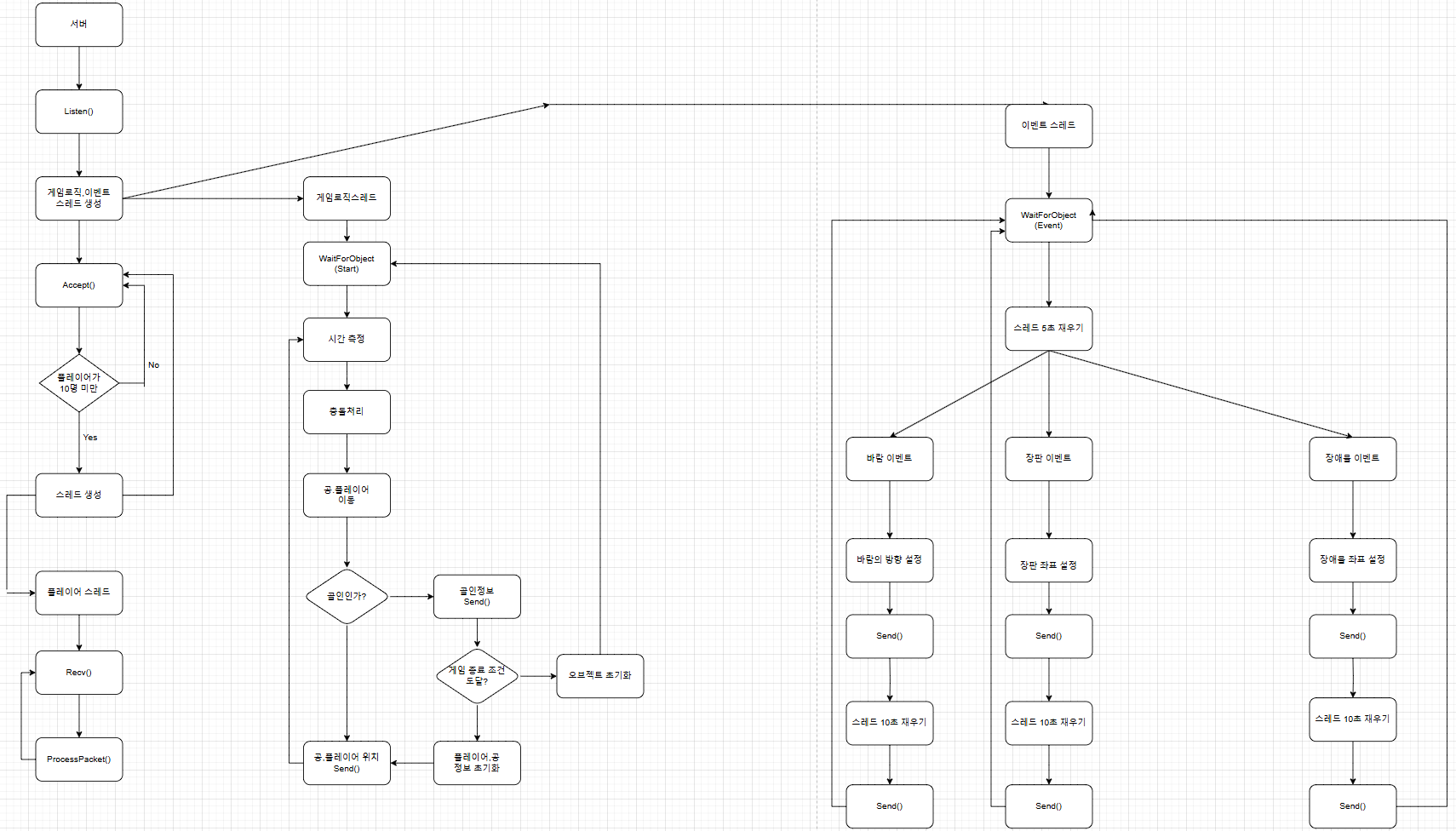
**인게임 스크린샷**

로비  
인게임  
****한 프로그램에서 2 ~ 10명의 플레이어가 동작하는 것을 네트워크를 활용하여 제작할 예정입니다.

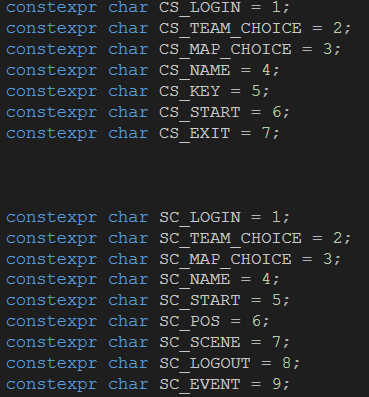
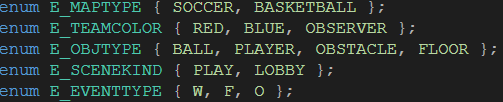
**High-Level 디자인 (클라이언트)**

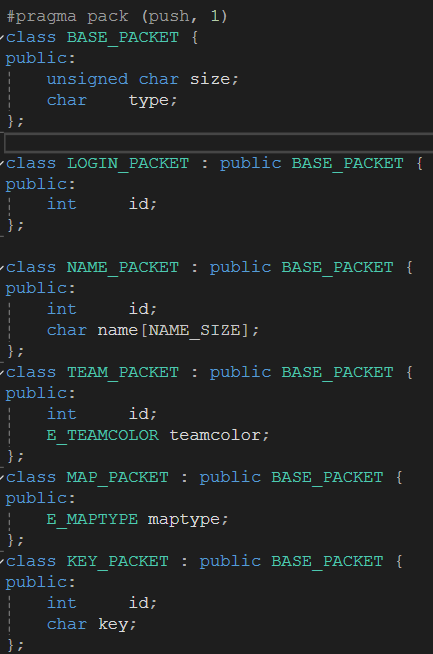
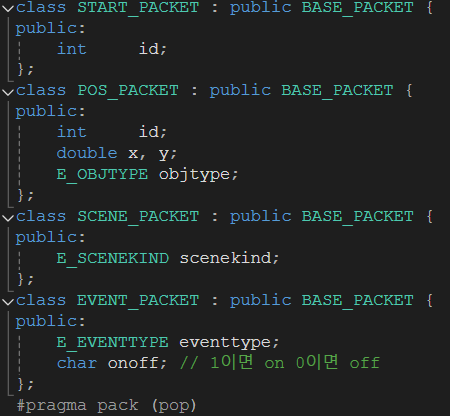


**High-Level 디자인 (서버)**

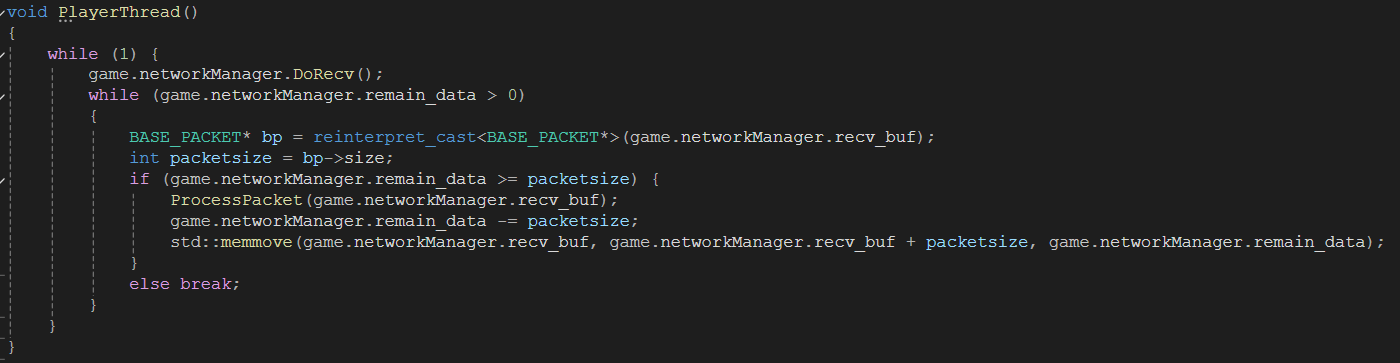
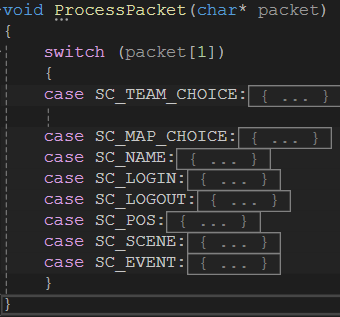
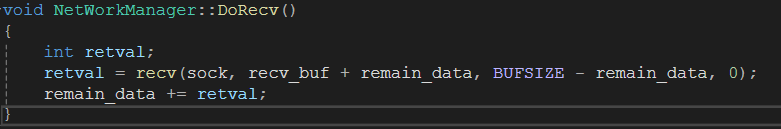


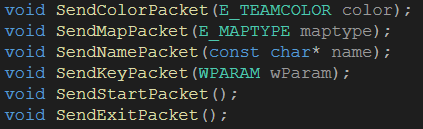
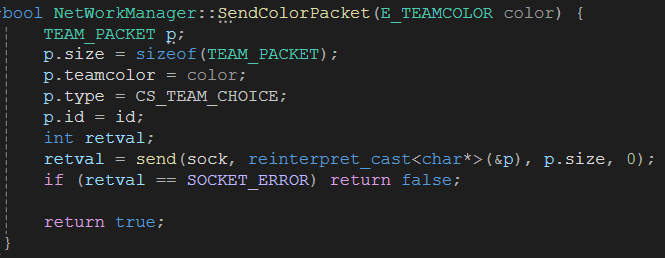
**Low-Level 디자인 (프로토콜)**

(패킷 타입)  
enum 타입

   
(패킷 구조체)

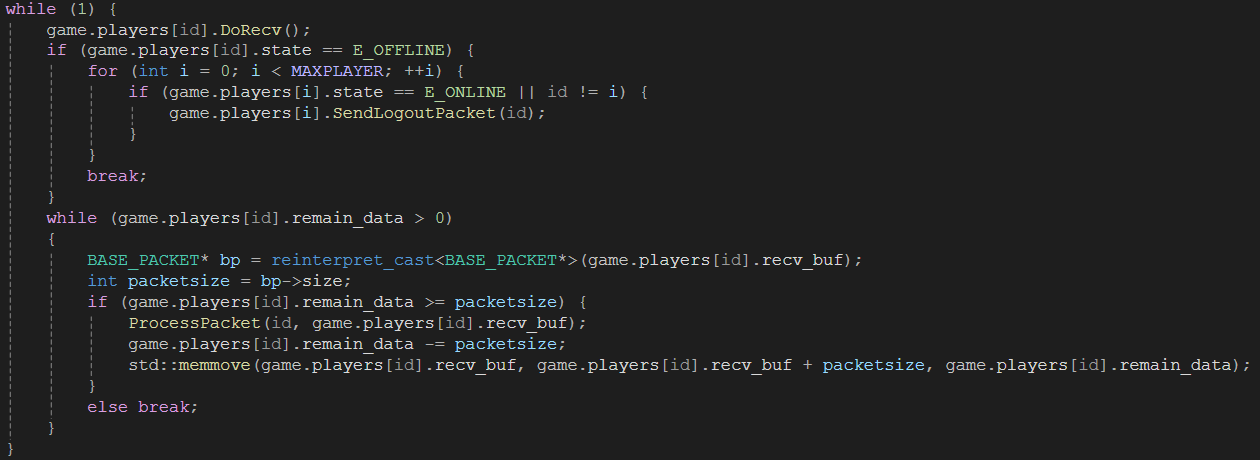
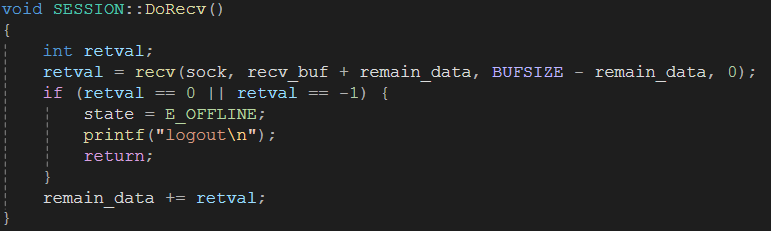
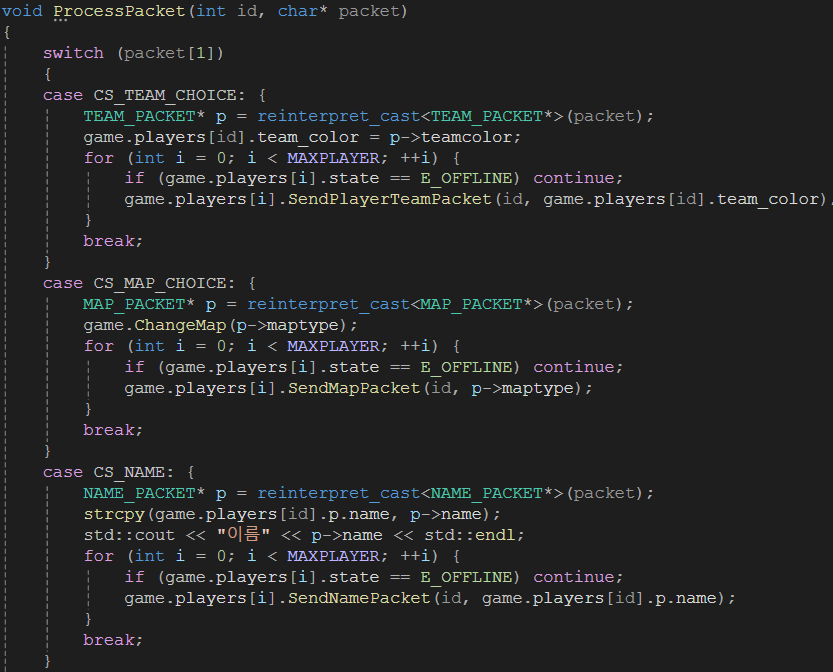
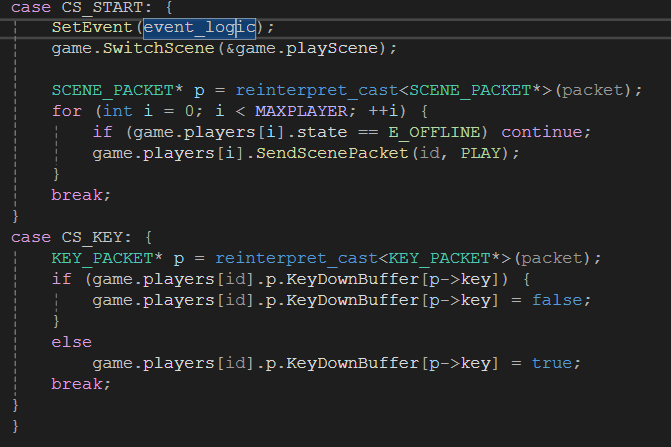
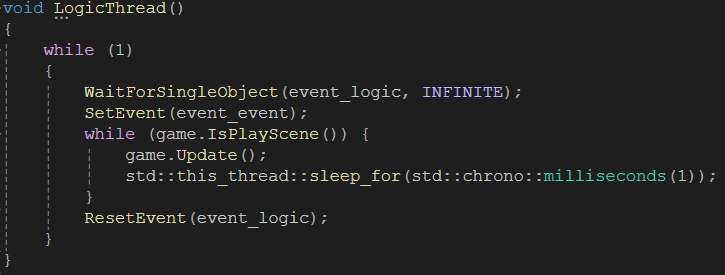
**Low-Level 디자인 (클라이언트)**

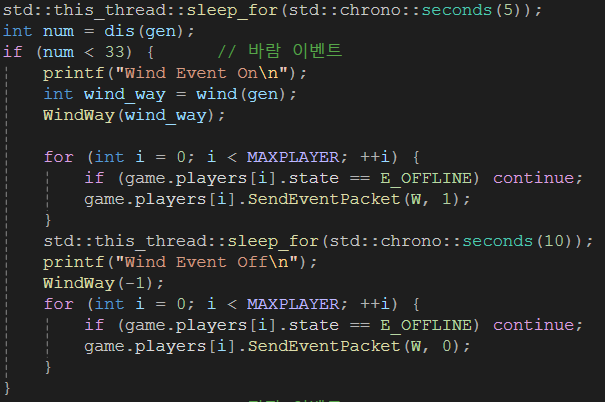
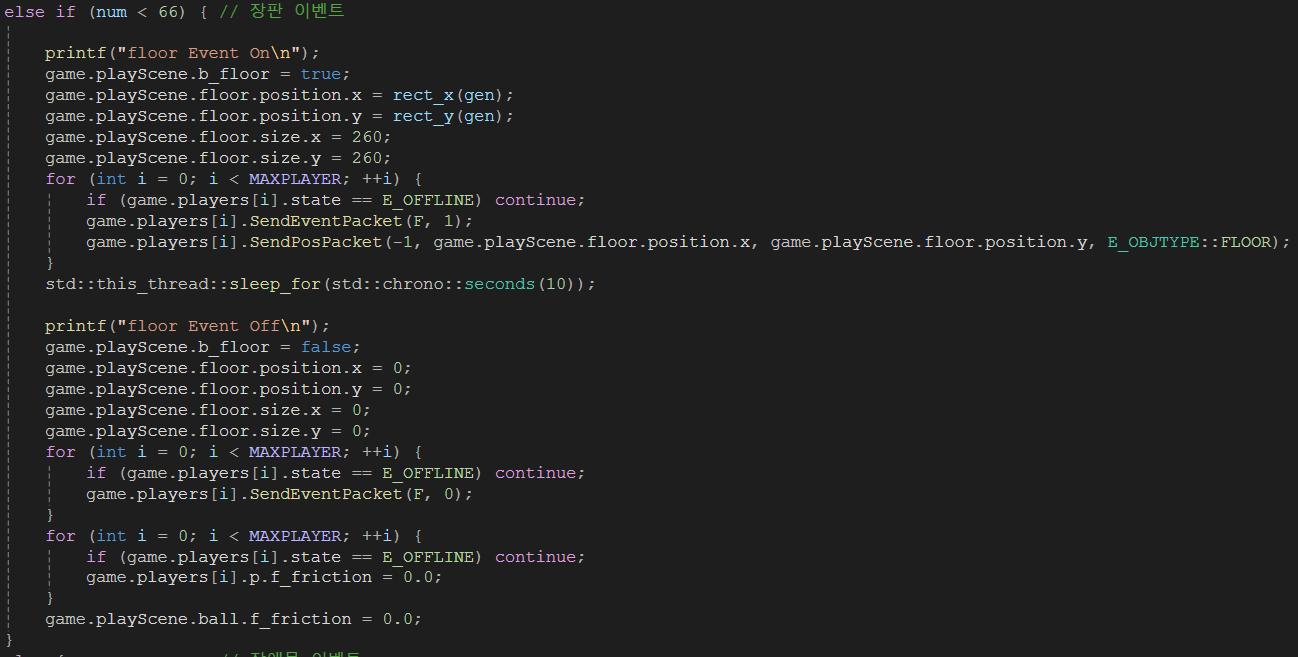
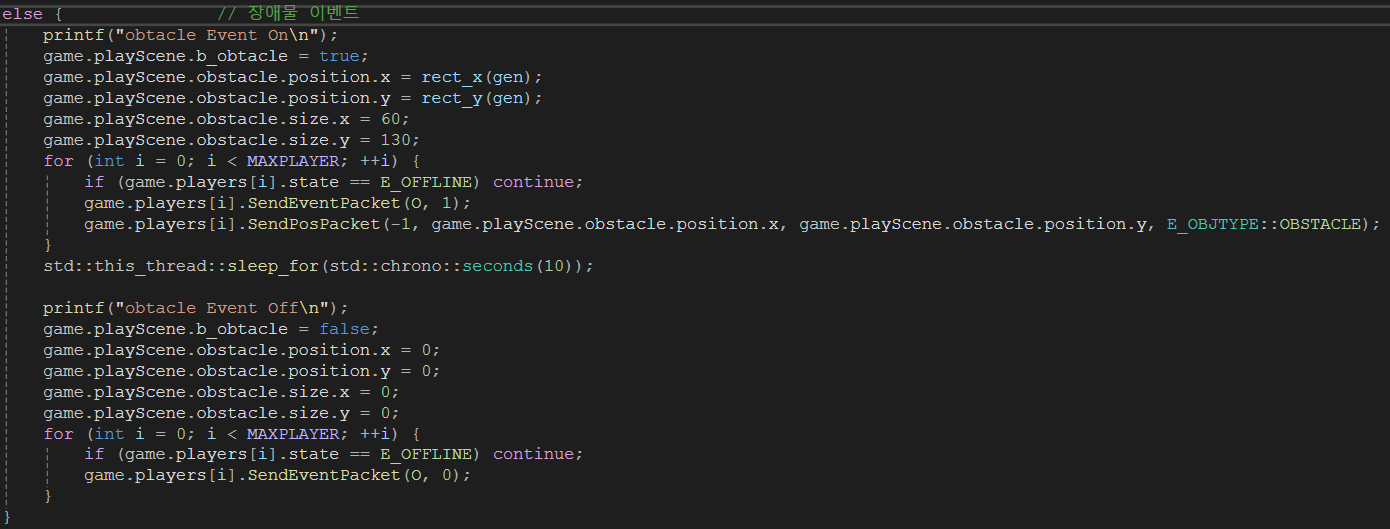
  
(Player스레드)  
  
(ProcessPacket 함수) 해당 패킷 타입에 맞는 패킷 처리를 한다.  
  
(DoRecv 함수)

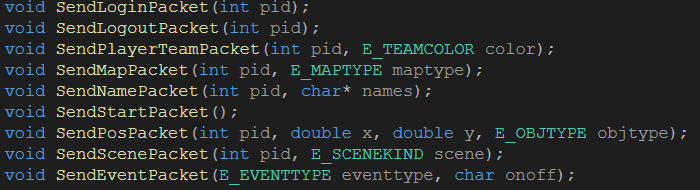
  
(클라이언트 Send 패킷 함수)  
  
(send 패킷 예시)

**Low-Level 디자인 (서버)**

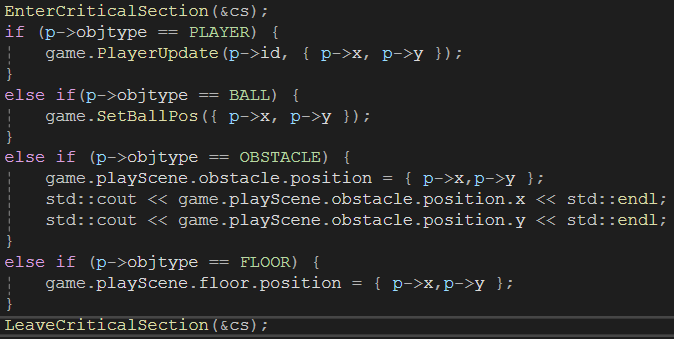
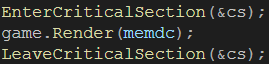
텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

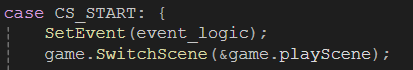
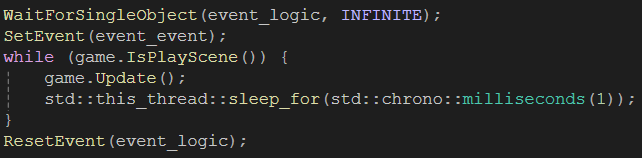
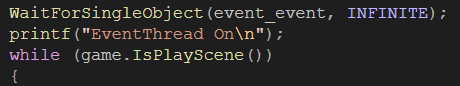
자동 생성된 설명  
(플레이어 배열에 빈자리가 있다면 스레드 할당)  
  
(PlayerThread)  
 (DoRecv)  
  
(processpacket)  
(로직 스레드)

 (바람 이벤트)(장판 이벤트)  
(장애물 이벤트)

  
(Send패킷 함수)

스레드 동기화

클라이언트  
메인스레드의 Render와 수신 스레드의 ProcessPacket이 같은 좌표라는 자원을 사용하고 있으므로 임계영역을 사용하여 동기화 할 예정입니다.  
  
(프로세스패킷에서 좌표를 업데이트하는 부분)  
 (메인스레드 렌더부분)

서버  
이벤트 스레드와 게임로직 스레드는 로비화면일때는 WaitForSingleObject로 대기 시켰다가 게임화면으로 넘어가면 SetEvent로 게임로직스레드를 깨운후 게임로직스레드에서 이벤트스레드를 깨우도록 설계하였습니다.  
(프로세스 패킷)  
(로직스레드)  
(이벤트스레드)

팀원 별 역할 분담  
최준하: 서버 프레임워크 구현, 플레이어 메인 스레드, 서버 이벤트 스레드 구현  
고태경: 클라이언트 프레임워크 구현, 서버 로직 스레드 구현  
이상유: 클라이언트 추가 기능 구현, 클라이언트 -> 서버 네트워크 기능 구현

개발 환경

언어: C++  
개발도구: Visual Studio 2022  
버전관리 툴: GitHub  
운영체제: Windows

**개발 일정**

최준하

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 11/1 | 2 |
| 3  부분 통과로 인한 추진계획서 수정 | 4  부분 통과로 인한 추진계획서 수정 | 5 | 6  Listen() accept() 구현 | 7 | 8  클라이언트 처리 담당 스레드 생성 DoRecv() 구현 | 9  Class Packet  (Base,Name, Team, Map) 제작 |
| 10  ~~1주차~~  ~~피드백~~ ProcessPacket() (Team)  구현 | 11 | 12 | 13  SendPlayer TeamPacket()  구현 | 14 | 15  ProcessPacket() (Map) 구현  SendPlayer TeamPacket()구현 (연기) | 16 SendMapPacket() 구현 |
| 17  2주차  피드백 | 18  ProcessPacket() (name) 구현 | 19 | 20  SendNamePacket() 구현 | 21 | 22  Class Packet (Key, Start) 제작 | 23  ProcessPacket() (Start) 구현 |
| 24  3주차  피드백 | 25  Class Packet (Pos,) 제작  ProcessPacket() (Key) 구현  SendStart GamePacket()  구현 | 26 | 27  클라이언트 스레드 동기화 ProcessPacket() (Start) 구현 | 28 | 29  이벤트스레드 생성 sendevent packet | 30 |
| 12/1  4주차  피드백 서버 스레드 동기화 | 2 | 3 | 4 | 5 | 6 | 7  클라이언트 뭉친 패킷처리 구조 설계 |
| 8  최종  피드백  바람,장판,장애물 이벤트 서버 뭉친 패킷처리 구조 설계 | 9 | 10 | 11  최종 점검 | 12 | 13 |  |

고태경

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 11/1  Player Class 설계 | 2  Object(Ball, Goalpost) class 설계 |
| 3  Map, Scene  Class 설계 | 4  Player의  이동, kick  입력 처리  (Play  Scene) | 5 | 6 | 7 | 8  Timer  추가 | 9 |
| 10  ~~1주차~~  ~~점검 및~~  부족한  진도 보강 | 11 | 12 | 13  Event class 설계  -> 10/14 | 14  Input Manager  (Start  Scene) | 15  Input Manager  (Lobby  Scene) | 16  Input Manager  (Play  Scene) |
| Event class  설계 (연기) |
| 17  2주차  점검 및  부족한  진도 보강 | 18  Client Manager  (Start  Scene) | 19  Client Manager (Lobby  Scene) | 20  Input Manager  (Lobby  Scene)(연기) | 21  기존프로젝트 코드 클라이언트로 옮기기 | 22 | 23 |
| 24  3주차  점검 및  부족한  진도 보강 | 25  Client Framework  (PlayScene  입력 처리) | 26  서버에서 Scene관리하도록 수정 | 27  멀티 플레이어  구현 | 28 | 29  팀 선택 입력 처리 및 recv 처리 | 30  Observer  추가 |
| 기존프로젝트 코드 서버로 옮기기 |
| 12/1  4주차  점검 및  부족한  진도 보강 | 2  Server에서  Goal Check  구현 | 3  카메라 구현 | 4  Server에서  kickOff 기능 추가 | 5 | 6  농구모드 추가 | 7 |
| 8  5주차  점검 및  부족한  진도 보강 | 9  StartScene 추가 | 10  버튼 동기화 추가 | 11  최종 점검 | 12 | 13 |  |

이상유

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 11/1 | 2 |
| 3 | 4 | 5 | 6 | 7  로비화면  추가 | 8 | 9  팀 선택,  종목 선택  추가 |
| 10  ~~1차~~  ~~피드백~~ | 11 | 12 | 13 | 14  바람 이벤트 구현  (12/5) | 15  Connect  함수 구현 | 16 |
| 17  2차  피드백 | 18 | 19  종목 선택 Send()  구현 | 20 | 21  수신용 스레드 생성  및 Recv()  구현 | 22 | 23  팀 선택 Send()  구현 |
| 24  3차  피드백 | 25 플레이어  이름  Send함수 구현 |  | 27 | 28  플레이어  목록 띄우기 | 29 | 30 |
| 12/1  4차  피드백  플레이어  레드 블루 리스트 추가 | 2  팀 선택 중복처리 | 3 | 4 | 5 | 6 | 7  플레이어  리스트 전체 오류 수정 |
| 8  이벤트최종 마무리 | 9 | 10 | 11  최종 점검 | 12 | 13 |  |