

# 네트워크 게임 프로그래밍 팀프로젝트 추진 계획서

4팀

# 목차

## 1. 애플리케이션 기획

- 게임 이름
- 게임 흐름
- 조작법

## 2. High-level 디자인

- 캐릭터 선택 화면
- 게임 화면

## 3. Low-level 디자인

- 클라이언트
- 서버

## 4. 팀원별 역할 분담/주차별 할 일

## 5. 개발 환경

## 6. 개발 일정

# 애플리케이션 기획

게임 이름: Head Soccer - 손성훈(윈도우 프로그래밍)

## 게임 흐름: 캐릭터 선택 > 게임 진행

캐릭터 선택 창에서 마우스 좌클릭으로 캐릭터를 설정할 수 있다. 가운데 버튼을 클릭해 게임을 준비 상태로 만들고 두 플레이어가 모두 준비 상태에 들어갔으면 게임을 시작한다. 게임은 1분간 진행된다. 게임이 끝나면 결과 화면이 뜨고 재시작과 게임 종료를 선택할 수 있다.

## 조작법

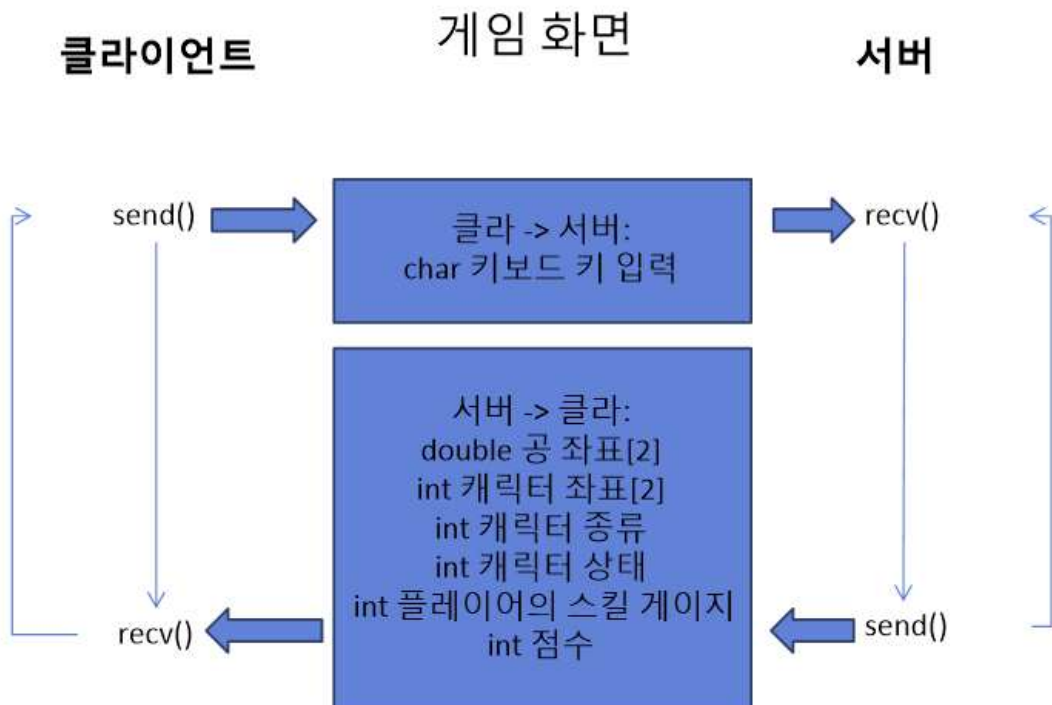
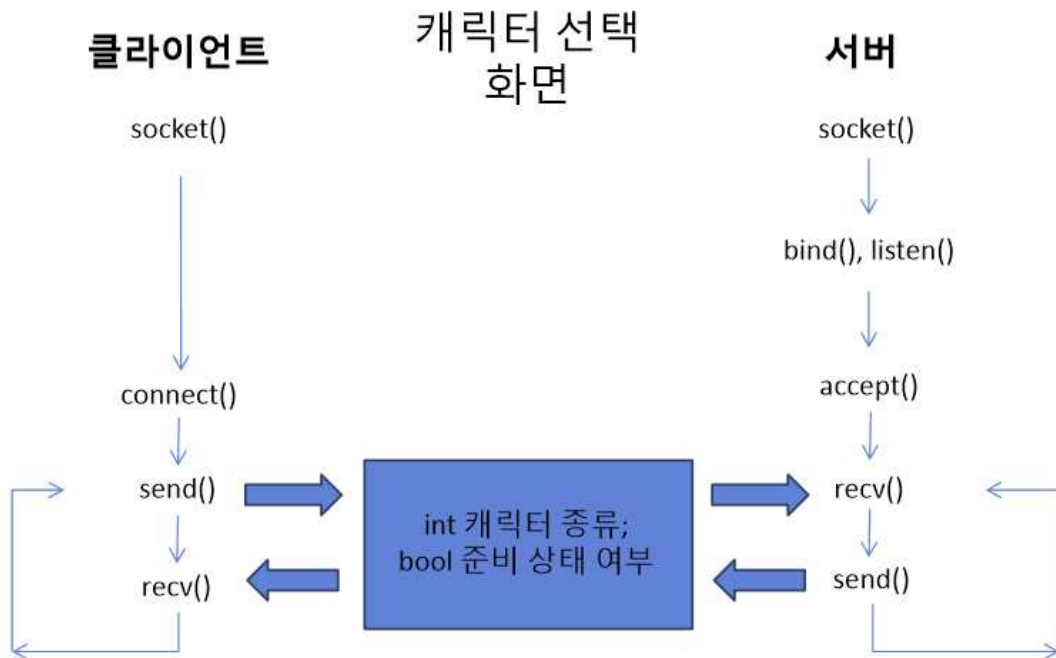
A, D - 좌우 이동

W - 점프

S - 킥

F - 파워 슈트

## High-Level 디자인



## Low-Level 디자인

클라이언트:

//윈속 초기화

//소켓 생성

//connect()

//keydown일 때 키보드 입력 값을 서버에게 send()

//WM\_Paint 앞부분에서 캐릭터, 공 좌표 등을 서버로부터 recv() 후 그리기

서버:

//윈속 초기화

//소켓 생성

//bind(), listen()

//CreateThread() : 스레드 2개 생성

//키 입력을 클라이언트로부터 recv(), 계산 이후 공 좌표, 캐릭터 좌표 등을 클라이언트로 send()

DWORD WINAPI ClientThread(LPVOID arg){

recv()

공 좌표, 캐릭터 좌표 등 계산()

스레드 p1과 스레드 p2 동기화

send()

}

// 공의 움직임을 제어하는 클래스

class Ball{

//좌표

//속도

//반지름

}

Ball::Ball() // 좌표, 속도, 크기

```
// 좌표 리턴
```

```
Ball::xPos()
```

```
Ball::yPos()
```

```
// 공을 초기 위치로 되돌림
```

```
Ball::Reset()
```

```
// 충돌처리와 물리 엔진
```

```
Ball::Physics(Character* p1, Character* p2)
```

```
//캐릭터 움직임을 제어하는 클래스
```

```
class Character{
```

```
//좌표
```

```
//파워 게이지
```

```
//능력치(점프, 속도, 파워)
```

```
//점수
```

```
//캐릭터 상태
```

```
}
```

```
Character::CharPos() //캐릭터 위치 리턴
```

```
Character::CharScore() //캐릭터 점수 리턴
```

```
Character::PwGaugeFull() // 파워게이지 증가
```

```
Character::Kick(int playerNum) // 캐릭터 상태 변경
```

```
Character::Goal() // 득점
```

```
Character::Goaled() // 실점 시 파워게이지 증가
```

```
Character::ResetPos(int playerNum) // 플레이어 위치 초기화
```

```
Character::PowerGauge() // 현재 게이지 값 리턴
```

```
Character::PowerShoot() // 스킬 사용 후 게이지 0으로
```

```
CharacterName::CharacterName(int playerNum) {  
int jump, speed, power; //10개 캐릭터 능력치 설정  
}
```

## 4. 팀원 별 역할 분담

이준석:

기존 게임 코드에서 클라이언트 코드 분리 및 수정

- 게임 화면
  - 키 입력 값 send(), 오브젝트 값들 recv()
  - draw()

기존 게임 코드에서 서버 코드 분리 및 수정

- 게임 화면
  - 계산된 오브젝트 값(캐릭터 좌표, 공 좌표)을 send()

이진수:

서버가 TCP 통신 가능하게 기본 코드 생성(send, recv, listen, accept)

서버에서 캐릭터 클래스 생성

CreateThread() : Player(Client) 스레드 1, 2 생성

기존 게임 코드에서 클라이언트 코드 분리 및 수정

- 캐릭터 선택 화면
  - 캐릭터 종류, 준비 상태 send(), recv()

기존 게임 코드에서 서버 코드 분리 및 수정

- 게임 화면
  - 계산된 오브젝트 값(캐릭터 좌표, 공 좌표)을 send()

손성훈:

클라이언트가 TCP 통신 가능하게 기본 코드 생성(send, recv, connect)

클라이언트에서 캐릭터 클래스 생성

기존 게임 코드에서 서버 코드 분리 및 수정

- 게임 화면
  - recv() 후 그 값에 의한 오브젝트 이동 및 충돌/점수 계산



## 주차 별 할 일

1주차: TCP 서버, 클라이언트 코딩 (게임 기능 x), 기존 게임 코드 분석

2주차: 캐릭터 리소스, draw를 위한 클래스, 능력치 변수 추가

3주차: 캐릭터 선택 화면에서의 통신 기능 추가, 캐릭터 선택 화면 렌더링, 공 클래스 추가

4주차: 게임 화면에서의 통신할 데이터 추가

5주차: 클라이언트에서 게임 화면 렌더링

6주차: 문제점 개선

## 5. 개발 환경

Visual Studio 2022, Github

## 6. 개발 일정

1주차	10월 27일	10월 28일	10월 29일	10월 30일	10월 31일	11월 01일	11월 02일
손성훈			명령행 인자로 서버의 IP를 받아 통신을 시도하는 클라이언트 프로그램 제작				

2주차	11월 03일	11월 04일	11월 05일	11월 06일	11월 07일	11월 08일	11월 09일
손성훈		클라이언트용 캐릭터 클래스 생성	좌표 관련 클래스 함수 상속	리소스 관련 클래스 함수 상속			

3주차	11월 10일	11월 11일	11월 12일	11월 13일	11월 14일	11월 15일	11월 16일
손성훈		서버 공 클래스 생성	공과 캐릭터의 충돌처리	공과 골대의 충돌처리			

4주차	11월 17일	11월 18일	11월 19일	11월 20일	11월 21일	11월 22일	11월 23일
손성훈			클라이언트로 부터 키 입력 recv() -> 좌표 변경	recv()와 충돌처리의 순서 확인 및 개선			

5주차	11월 24일	11월 25일	11월 26일	11월 27일	11월 28일	11월 29일	11월 30일
손성훈		클라이언트 공 클래스 생성	클라이언트 기타 오브젝트 draw()				

6주차	12월 01일	12월 02일	12월 03일	12월 04일	12월 05일	12월 06일	12월 07일
손성훈			문제점 개선	문제점 개선			

1주차	10월 27일	10월 28일	10월 29일	10월 30일	10월 31일	11월 01일	11월 02일
이준석			원본게임 구조 분석		원본게임 main() 분석	원본 게임 전체 코드 분석	

2주차	11월 03일	11월 04일	11월 05일	11월 06일	11월 07일	11월 08일	11월 09일
이준석			캐릭터 1개 로직(서버)/랜 더(클라) 나누기		게임 클라이언트 부분 분리		

3주차	11월 10일	11월 11일	11월 12일	11월 13일	11월 14일	11월 15일	11월 16일
이준석			게임 서버 부분 분리			클라이언트- 캐릭터 종류 준비 상태 send(), recv()	

4주차	11월 17일	11월 18일	11월 19일	11월 20일	11월 21일	11월 22일	11월 23일
이준석			클라이언트에 서 키 입력 send()		클라이언트에 서 오브젝트 값들 recv()	클라이언트에 서 캐릭터 2명 draw()	

5주차	11월 24일	11월 25일	11월 26일	11월 27일	11월 28일	11월 29일	11월 30일
이준석			클라이언트- 게임 화면 draw()				

6주차	12월 01일	12월 02일	12월 03일	12월 04일	12월 05일	12월 06일	12월 07일
이준석			문제점 개선		문제점 개선		

1주차	10월 27일	10월 28일	10월 29일	10월 30일	10월 31일	11월 01일	11월 02일
이진수						서버- 기본 코드 send(), recv(), listen(), accept()	서버- 멀티쓰레드 구현 CreateThread( )

2주차	11월 03일	11월 04일	11월 05일	11월 06일	11월 07일	11월 08일	11월 09일
이진수						서버- 캐릭터 클래스 1~5작성	서버- 캐릭터 클래스 6~10작성

3주차	11월 10일	11월 11일	11월 12일	11월 13일	11월 14일	11월 15일	11월 16일
이진수						클라이언트- 캐릭터 종류 준비 상태 send(), recv()	

4주차	11월 17일	11월 18일	11월 19일	11월 20일	11월 21일	11월 22일	11월 23일
이진수						서버- 계산된 오브젝트 값(캐릭터 좌표, 공격 좌표)을 클라이언트로 send()	

5주차	11월 24일	11월 25일	11월 26일	11월 27일	11월 28일	11월 29일	11월 30일
이진수			클라이언트- 게임 화면 draw()			클라이언트- 게임 화면 draw()	

6주차	12월 01일	12월 02일	12월 03일	12월 04일	12월 05일	12월 06일	12월 07일
이진수						문제점 개선	문제점 개선