

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 76강 - 기존 오목 네트워크 게임 구성 분석하기

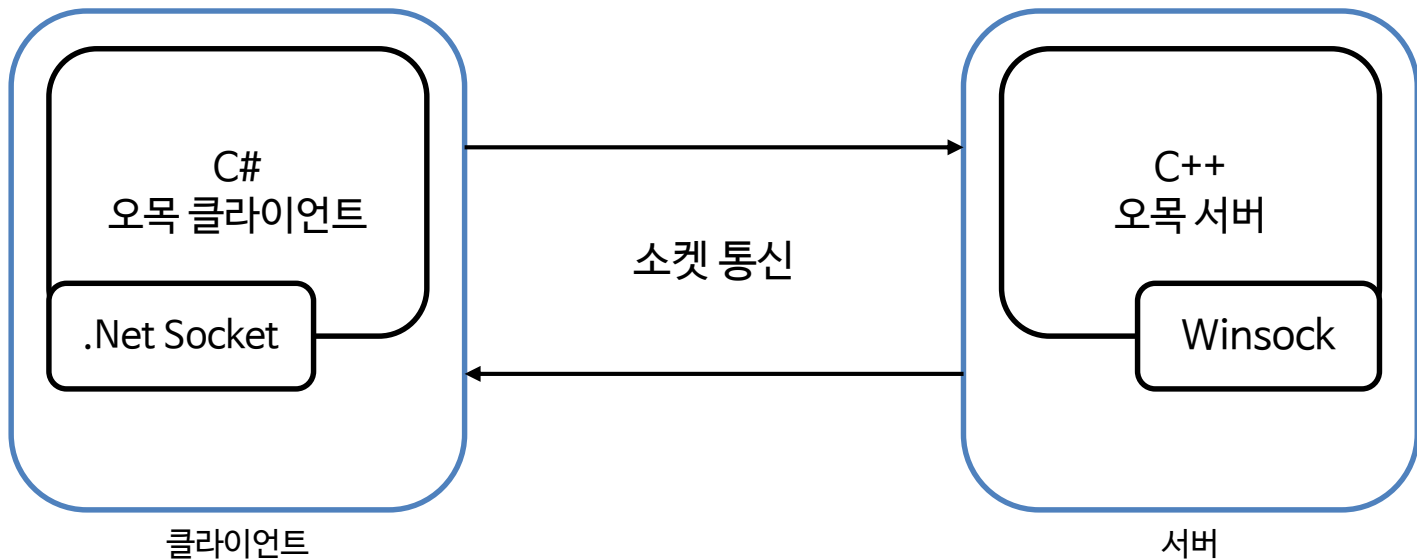
기존 오목 네트워크 게임 구성 분석하기

기존 오목 네트워크 게임 구성 분석하기

- 이전 시간에 만들었던 오목 네트워크 게임의 구성을 깊게 분석하여 게임 서버/클라이언트의 구조를 복습합니다.
- 각 기능에 따른 명령어 양식의 구성을 확인할 수 있습니다.
- 단순한 형태의 서버 구성을 자신이 원하는 형태로 재설계할 수 있습니다.

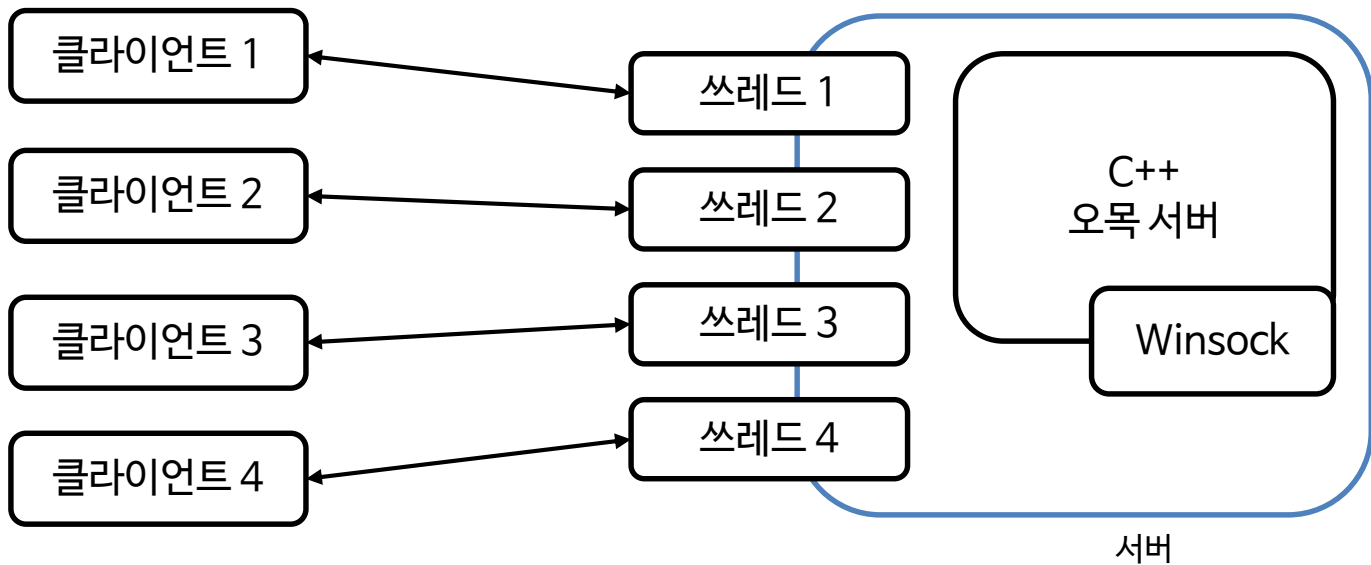
기존 오목 네트워크 게임 구성 분석하기

서버/클라이언트 구성도



기존 오목 네트워크 게임 구성 분석하기

서버/클라이언트 구성도



기존 오목 네트워크 게임 구성 분석하기

서버 프로그램 기능 분석 - 클라이언트 객체 정의

다음과 같은 클라이언트 객체를 정의한 이후에 접속한 모든 클라이언트를 Vector STL에 담아 관리합니다.

- 클라이언트 고유 번호: 클라이언트가 접속할 때마다 자동으로 1씩 증가
- 접속한 방 번호: 클라이언트가 접속 요청한 방 번호
- 클라이언트 소켓 객체: 연결된 소켓 객체

기존 오목 네트워크 게임 구성 분석하기

서버 프로그램 기능 분석 - 접속 요청

- 요청 형태: [Enter] {방 번호}
- 예시: [Enter] 123
- 해당 방 인원이 1명 이하인 경우 클라이언트에게 **[Enter]** 명령어를 전송합니다. (접속 성공)
- 해당 방 인원이 이미 2명인 경우 클라이언트에게 **[Full]** 명령어를 전송합니다. (가득 찬 방)
- 접속 이후에 방 인원이 2명이 되는 경우 **[Play]** 명령어를 전송합니다. 이 때 검은 돌 사용자에게는 **[Play] Black**을, 흰 돌 사용자에게는 **[Play] White**를 전송합니다. (게임 시작)

기존 오목 네트워크 게임 구성 분석하기

서버 프로그램 기능 분석 - 나가기 요청

- 요청 형태: 소켓 연결 해제
- 특정 클라이언트 소켓 연결이 해제되는 경우 방에 남아있는 사람에게 **[Exit]** 명령어를 전송합니다. (상대방 나감)

기존 오목 네트워크 게임 구성 분석하기

서버 프로그램 기능 분석 - 돌 놓기 요청

- 요청 형태: [Put] {방 번호}, {X 좌표}, {Y 좌표}
- 예시: [Put] 123,7,8
- 오목판의 특정한 위치에 돌을 놓았다는 정보를 방에 있는 사용자들에게 브로드캐스트(Broadcast)합니다. 사용자에게 보내는 명령어의 형태는 요청과 동일합니다.

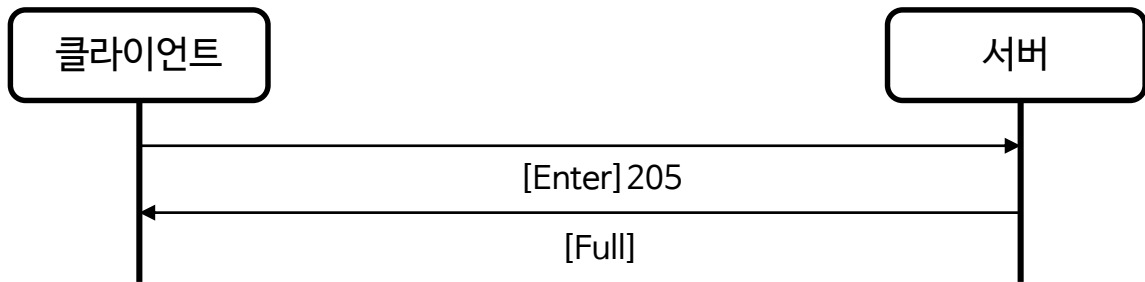
기존 오목 네트워크 게임 구성 분석하기

서버 프로그램 기능 분석 - 시작 요청

- 요청 형태: [Play] {방 번호}
- 예시: [Play] 123
- 특정한 방에 위치한 사용자의 시작 요청을 받아 **[Play]** 명령어를 전송합니다. 이 때 검은 돌 사용자에게는 **[Play] Black**을, 흰 돌 사용자에게는 **[Play] White**를 전송합니다. (게임 시작)

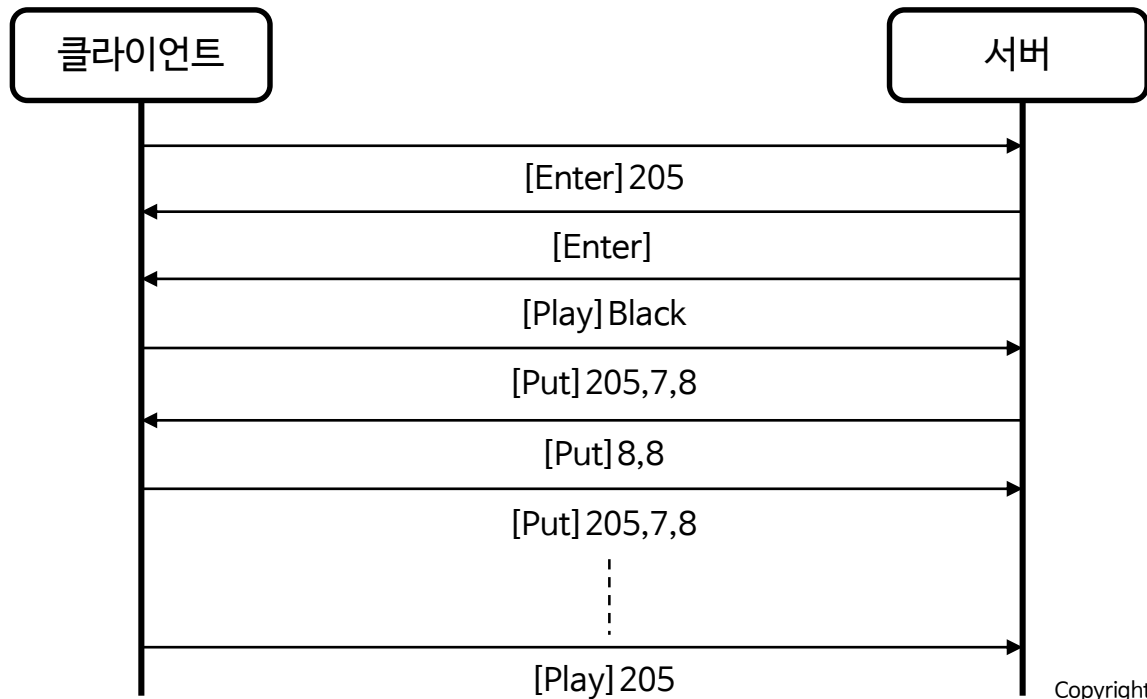
기존 오목 네트워크 게임 구성 분석하기

사용자 게임 진행 예시: 방이 가득 차서 접속 실패



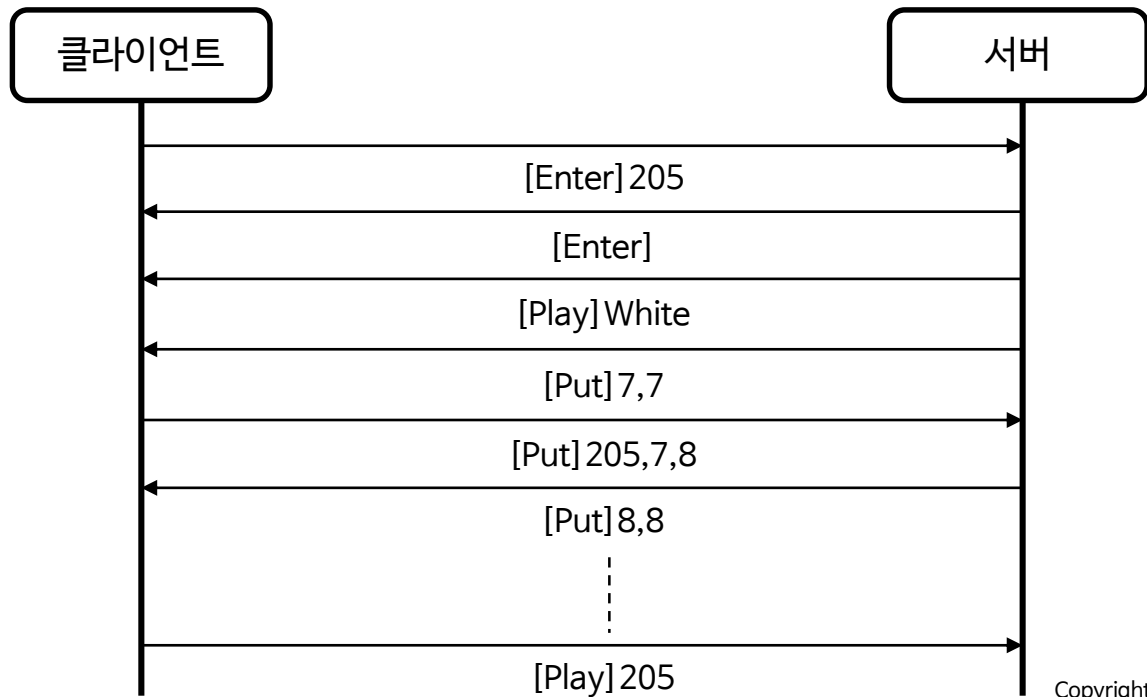
기존 오목 네트워크 게임 구성 분석하기

사용자 게임 진행 예시: 접속 및 게임 진행 (흑)



기존 오목 네트워크 게임 구성 분석하기

사용자 게임 진행 예시: 접속 및 게임 진행 (백)



기존 오목 네트워크 게임 구성 분석하기

개선점 고려하기

- 소스코드의 요청 및 응답의 형태가 너무 직관적이므로 분석하기 쉽습니다.
- 각 게임에 대한 승리 판정을 클라이언트가 대신하고 있습니다.
- 사용자의 수가 많아지는 경우 Vector STL이 비효율적으로 동작할 수 있습니다.

기존 오목 네트워크 게임 구성 분석하기

게임 서버 운영상의 특징

- 보안성을 향상시키기 위해서는 서버의 부담이 가중됩니다.
- 반대로 서버의 부담을 줄이면 클라이언트의 부담이 커지며, 보안성이 악화될 수 있습니다.
- 일반적으로 패키지 기반의 게임에서 게임 핵이 자주 등장하는 이유에 해당합니다.