



쉽고 간단한

오델로 프로토콜

보안제품개발트랙

한종호

INDEX

1. 특징

2. 프로토콜

2.1 접속

2.2 대국

2.3 종료

3. 정리

1. 특징

O	T	H
E	L	L
O	●	○

포맷

메시지

accept

전송코드

color:b

Key:Value

token:4K2odNg834j2pd0GbWm

Client (20자리)

board:00 ... 0bw0 ... 00

보드정보 0,b,w (64개)

2. 프로토콜

2.1 접속

2.2 대국

2.3 종료

O T H

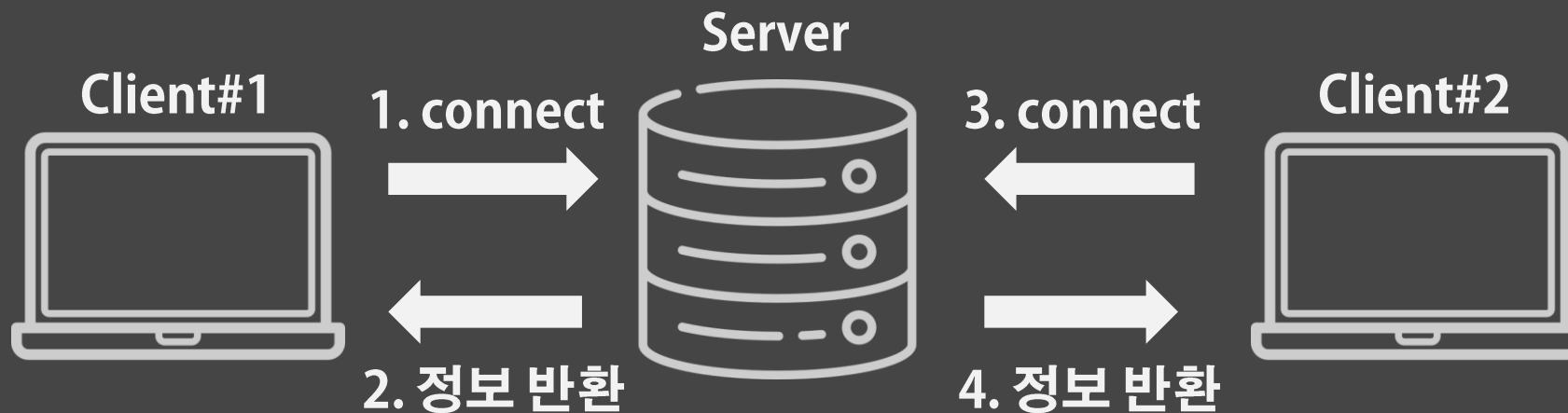
E L L

O ● ○

접속

프로토콜

1. 클라이언트가 서버로 접속
2. 먼저 접속한 순서대로 **흑, 백**
3. 2명이 접속하면 더 이상 접속불가



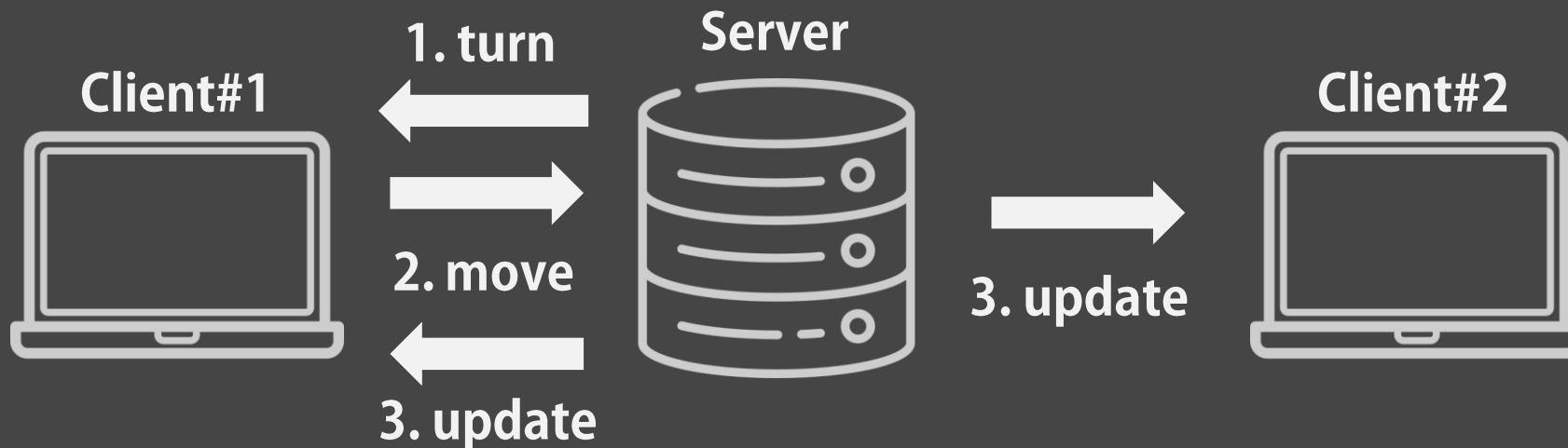
접속 프로토콜



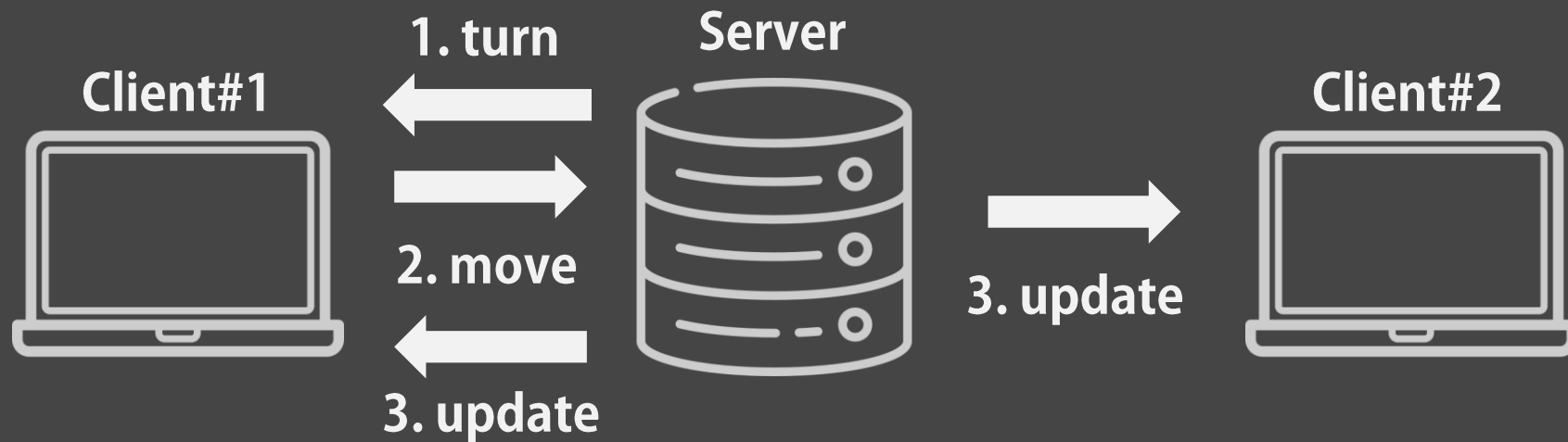
2. 정보 반환

```
accept  
color:b  
token:4k2odng834j2pd0gbwm  
board:0000 ... 0bw0 ... 0wb0 ... 0000
```

1. 서버가 턴과 놓을 수 있는 위치를 알려줌
2. 20초 안에 클라이언트가 돌을 놓을 위치를 Token과 함께 전송
3. 양쪽 클라이언트에 보드 업데이트



대국 프로토콜

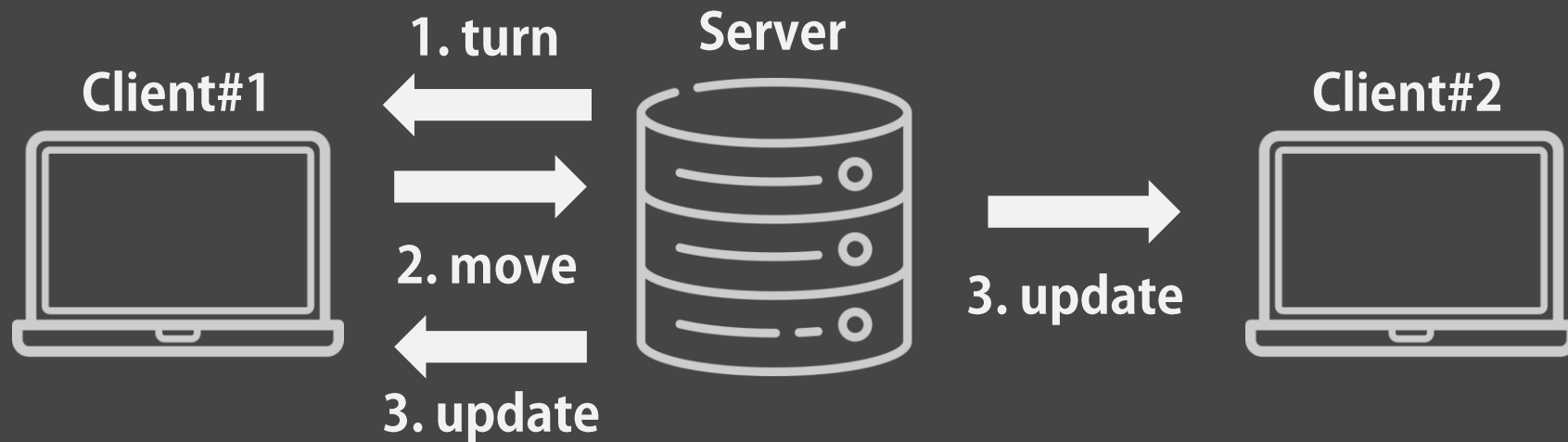


1. turn

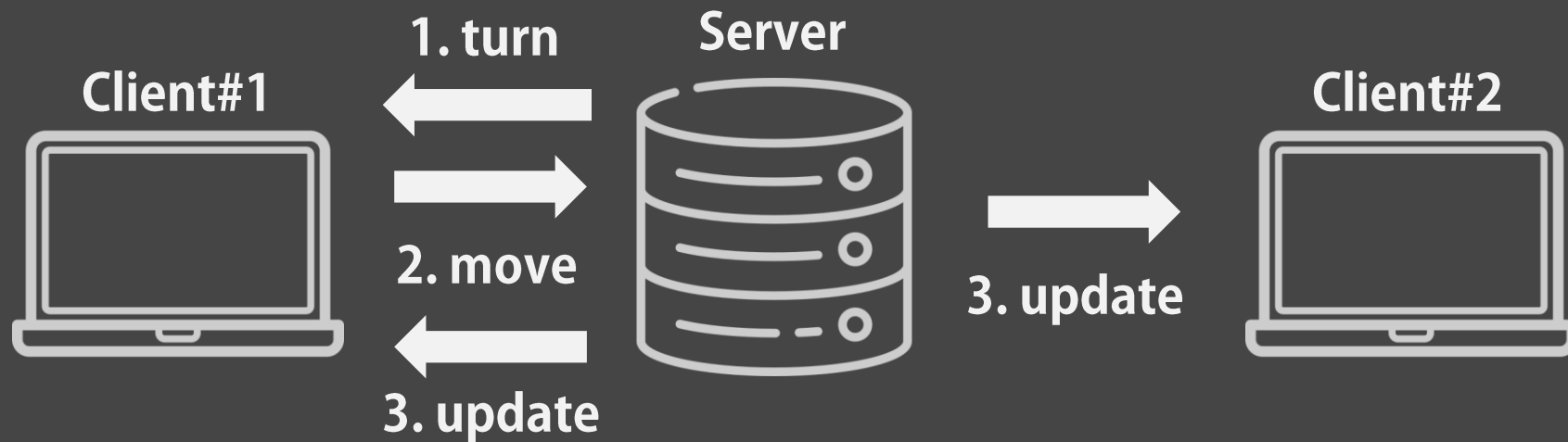
turn
available:1A 2B 3C 4D

※ 데이터를 보낸 후 20초 카운트 시작

대국 프로토콜



대국 프로토콜

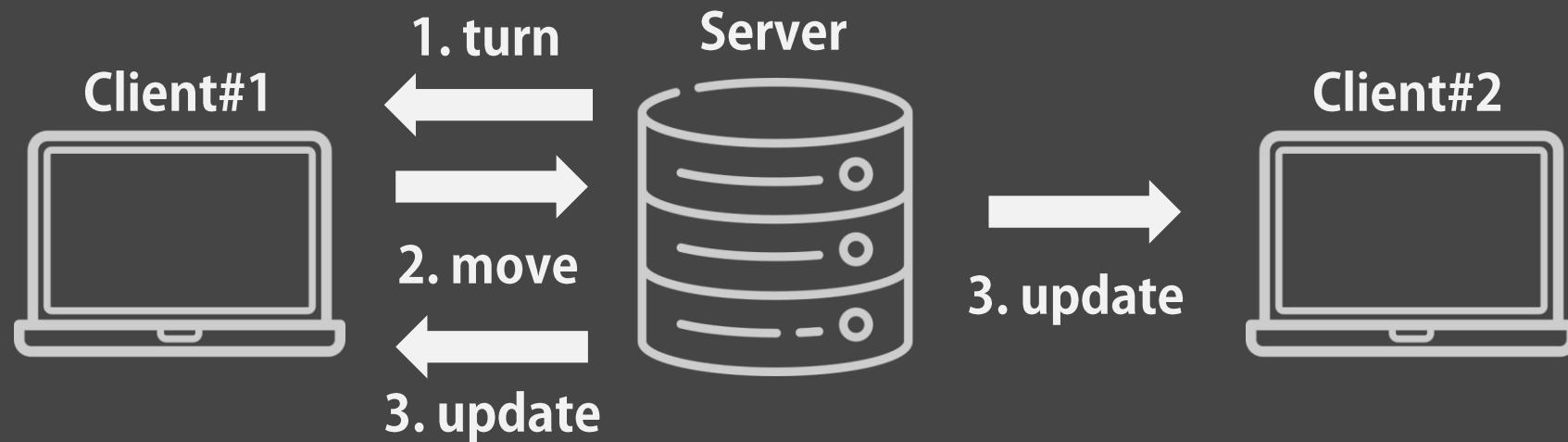


3. update

update

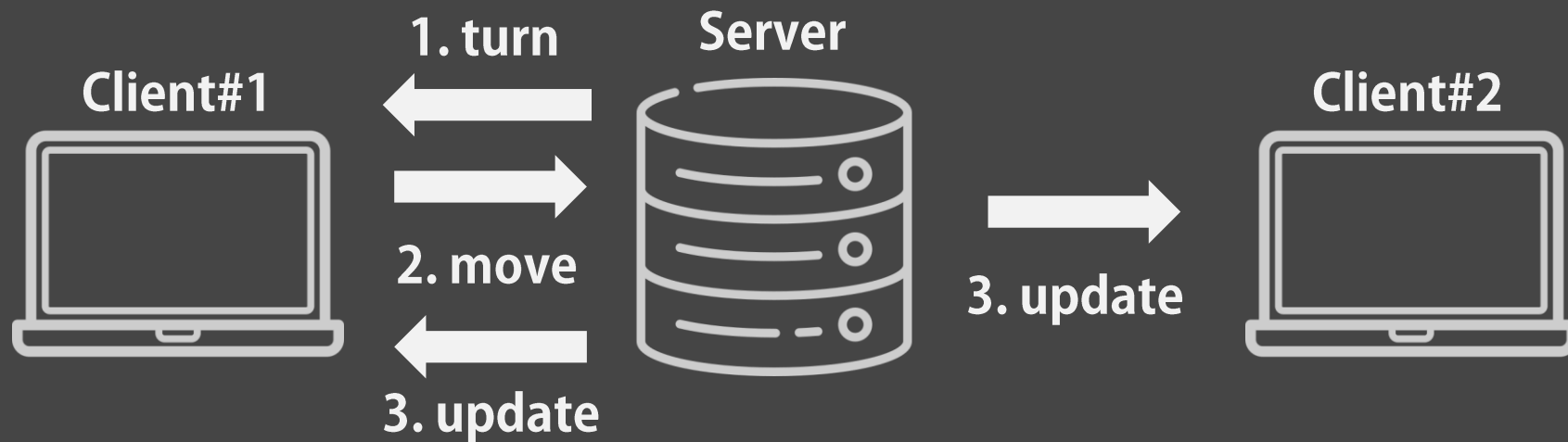
board:b000...bwwbbw...000

패스 프로토콜



패스가 일어날 시,
다시한번 같은 클라이언트에 turn을 보낸다.

물수패

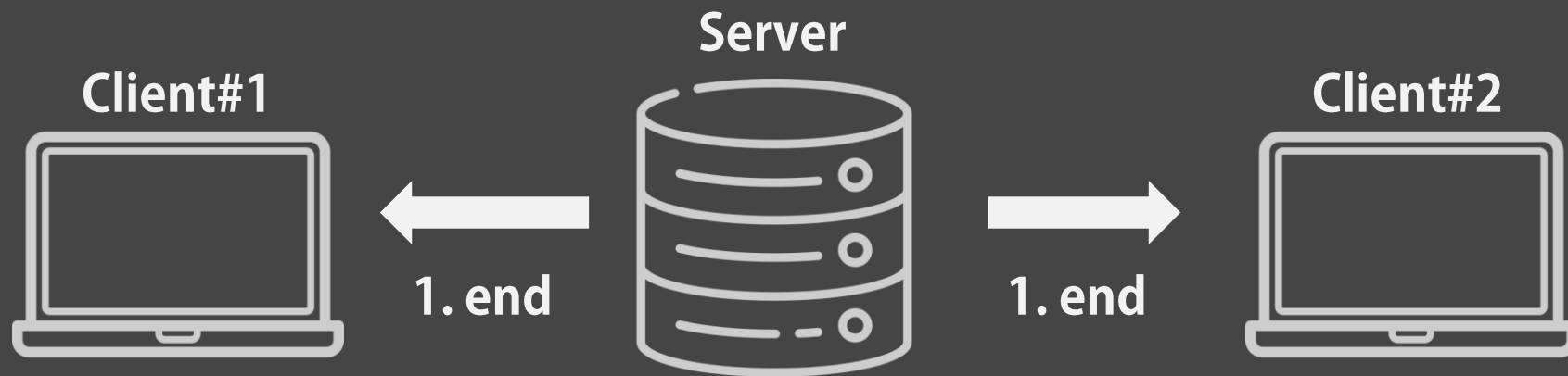


1. 잘못된 곳을 놓으려 하거나

2. 20초의 시간이 모두 지나면

패배

종료 프로토콜



1. end

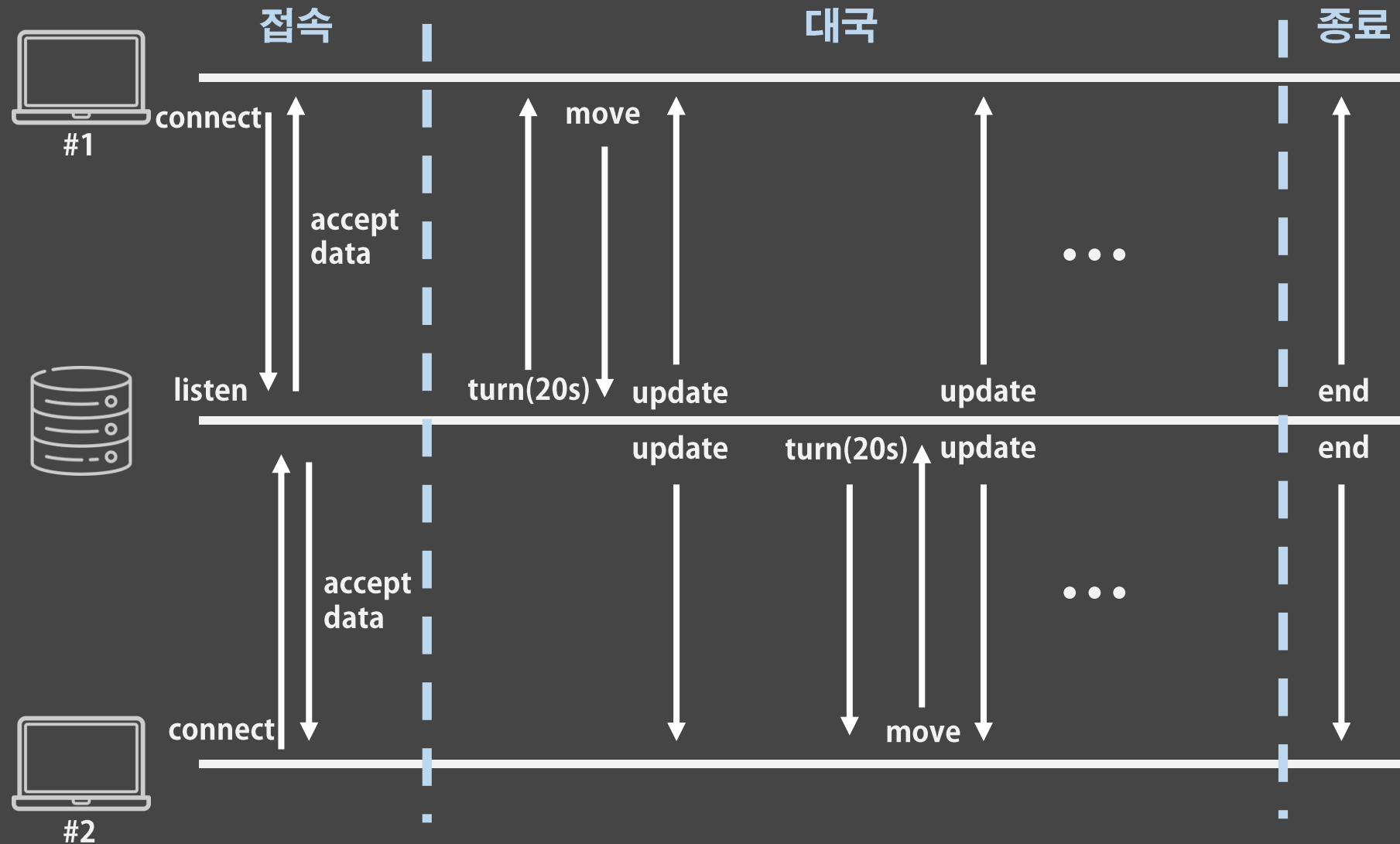
```
end
status:win
score:20b 44w
board:bbwwb...bwbwbbwbb
```

3. 정리

O	T	H
E	L	L
O	●	○

정리

플로우 차트



정리

특이사항

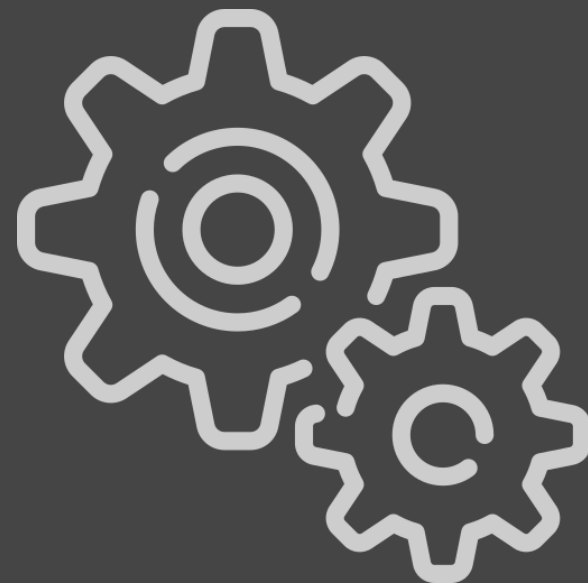
1. 돌을 놓을 때 클라이언트 키를 함께 전송
2. 20초 안에 놓을 곳을 전송하지 못하거나 잘못된 곳에 돌을 놓으려고 하면 몰수패
3. 5종류의 간단한 메시지를 통한 쉬운 프로토콜

정리

제공항목



프로토콜 명세 문서
(PDF)



라이브러리 코드
(C++, Python)

Q & A

O T H
E L L
O ● ○

부록

OTH
ELL
O ● ○

부록

메시지 구성

accept

첫 줄에는 code가 온다.

color:b

다음 줄부터는 Key:Value가 온다.

부록

메시지 구분

accept

color:b

각 줄은 \n으로 구분된다.

Key와 Value는
:으로 구분된다.

부록

좌표

	A	B	C	D	E	F	G	H
1								
2								
3								
4				●	○			
5				○	●			
6								
7								
8								

→ 1H

{행}{열} 순서대로
좌표를 표시한다.

전체 보드를 표현할 때는
비어 있는 곳을 0, 흑 돌을 b,
백 돌을 w로 표현하며,
1A부터 8H까지 가로로 표현한다.

예시: 00000000000000000000000000000000wb000000bw000000000000000000000000000000

부록

메시지

사용되는 코드(5종류)

accept, turn, move, update, end

사용되는 Key(7종류)

color, token, board, available,
move, status, score

부록

각 code별
사용하는
Key

accept
color
token
board

turn
available

move
move
token

update
board

end
status
score
board

부록

accept

```
accept  
color:{color}  
token:{token}  
board:{board}
```

{color}에는 'b' 또는 'w' 가 입력된다.
{token}에는 20자리 랜덤문자가 입력된다.
{board}에는 0,b,w로 구성된 64자리 문자열이
입력된다.

부록

turn

turn
available:{available}

{available} 에는 놓을 수 있는 좌표들이
하나의 space로 분리되어 입력된다.
ex) 1A 3B 4C 2E

부록

move

```
move  
move:{move}  
token:{token}
```

{move} 에는 클라이언트가 돌을 놓기로
선택한 좌표가 입력된다.
{token} 에는 서버가 클라이언트에 전달한 token값이
입력된다.

부록

update

```
update  
board:{board}
```

{board}에는 0,b,w로 구성된 64자리 문자열이
입력된다.

부록

update

```
end  
status:{status}  
score:{score}  
board:{board}
```

{status}에는 win, lose 또는 tie가 입력된다.
{score}에는 두 색깔 각각의 돌 개수가 다음의 형식으로 입력된다. “{black}b {white}w”

ex)20b 44w

{board}에는 0,b,w로 구성된 64자리 문자열이 입력된다.

부록

API

othello.py(cpp)에 있는 othello class를 사용하여 코드를 작성하면 됩니다.

생성자: othello(IP, Port) - IP(문자열) Port(숫자)

wait_for_turn() 함수와
move(string)함수를 사용.

부록

API

wait_for_turn() 함수

update 메시지를 자동으로 처리하여 board를 갱신한다.
turn, end 메시지에 return 한다.

- return 값과 타입

Python: code(str), data(dict) 튜플.

C++: code(string), data(map)의 pair.

Type: `std::pair<std::string, std::map<std::string, std::string>>`

wait_for_turn() 함수

code가 **turn** 이면, move 함수를 이용하여 돌을 놓고,
end 이면 경기를 끝낸다.

turn일 때, data의 key는 **available**만 있고,
end일 때, data의 key는 **status, score, board**이다.

move(str) 함수

돌을 놓을 좌표를 문자열로 전달한다.

매개변수 타입

Python: str

C++: std::string