

# 게임서버와 클라이언트 연동

편집: 김혜영

## 5.7 | 게임 플레이 이외의 네트워킹

- 로그인 과정에서 클라이언트와 서버가 대화하는 주요 절차
  - 로그온 요청 메시지를 서버로 전송한다.
  - 서버는 파일이나 데이터베이스에서 해당 유저의 ID와 비밀번호를 받아서 식별한다.
  - 식별 결과, 즉 로그인 처리 결과를 클라이언트에 통보한다.
  - 클라이언트가 이 통보를 받으면(예를 들어 로그인 성공이라는 통보를 받으면) 플레이어 정보를 데이터베이스에서 로딩해서 게임 서버 메모리에 보관한다. 이 과정은 있을 수도 있고 없을 수도 있다.

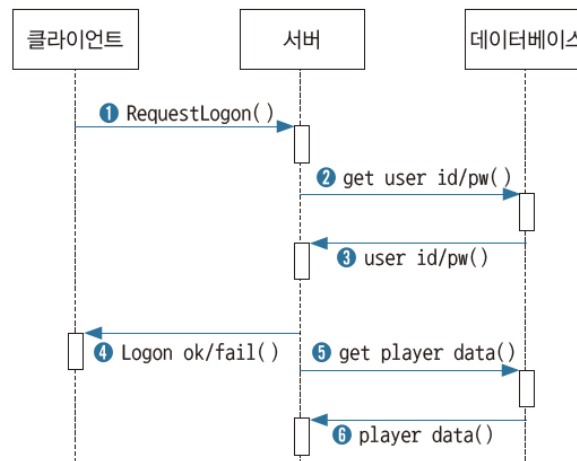
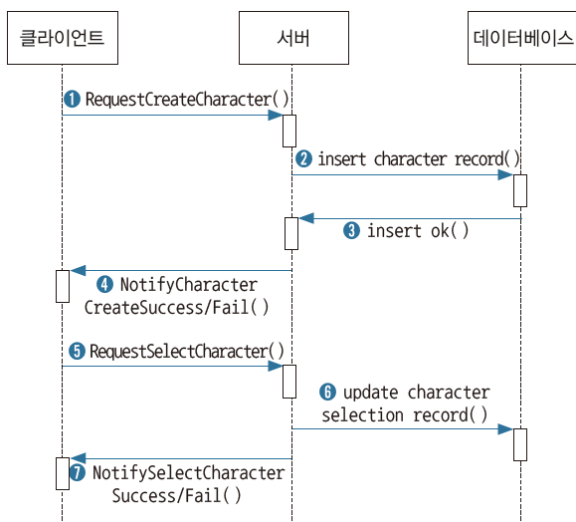


그림 5-19 로그인 과정을 시퀀스 다이어그램으로 표현

## 5.7 | 게임 플레이 이외의 네트워킹

### • 캐릭터를 만드는 과정



- 1 클라이언트가 서버에 "내 캐릭터를 만들어라"라고 요청한다.
- 2 서버는 데이터베이스에 "새 캐릭터에 대한 데이터 개체, 즉 레코드를 만들어라"라고 요청한다.
- 3 데이터베이스는 이에 대한 응답을 한다. (여기서는 "성공했다"라는 응답을 받았다고 함)
- 4 서버는 클라이언트에 "캐릭터를 만드는 것이 성공했다"라고 알려 준다.
- 5 클라이언트는 서버에 "내 캐릭터 중 000를 선택하겠습니다"라고 요청합니다.
- 6 서버는 데이터베이스에 "캐릭터 000를 선택했다고 기록하자"라고 요청한다.
- 7 서버는 클라이언트에 "캐릭터 000를 성공적으로 선택했다"라고 알려 준다.

그림 5-20 캐릭터를 만드는 과정을 시퀀스 다이어그램으로 표현

데이터베이스에 기록하되 그 결과를 기다릴 필요가 없을 때는  
다음 코드와 같이 데이터베이스에 기록하는 함수를 별도의 스레드나 비동기 함수 호출로 처리해도 된다.

```
func()
{
    data.change(xxx);
    thread.doAsync() = >
    {
        db.write(xxx);
    });
}
```

## 5.7 | 게임 플레이 이외의 네트워킹

- 매치메이킹\_플레이어가 수동으로 방을 만들고, 다른 플레이어가 수동으로 방에 들어가는 방식 설계

- 해킹을 방지하고자 방 만들기 혹은 들어가기 정보는 클라이언트에서 판단하지 말고 서버에서 모두 판단할 것.
- 클라이언트에서는 일방적으로 판단하지 말고 서버에 요청하여 그 결과에 따라서 행동할 것.
- 방 만들기 혹은 방 들어가기로 서버 내부의 방 목록이나 방 안의 플레이어 목록이 변할 때 클라이언트는 그 변화를 통보 받을 것.

서버에서는 방 목록과 각 방에 들어가 있는 플레이어, 즉 방 안의 플레이어 목록을 갖고 있어야 하며, 게임 플레이 중인 방의 상태 데이터도 갖고 있어야 한다.

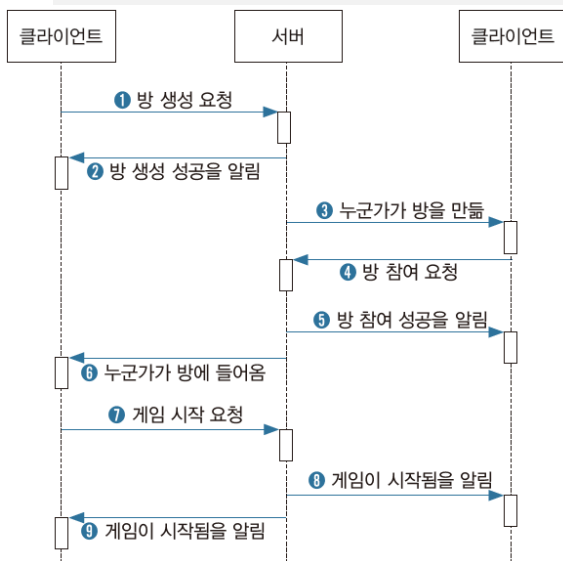


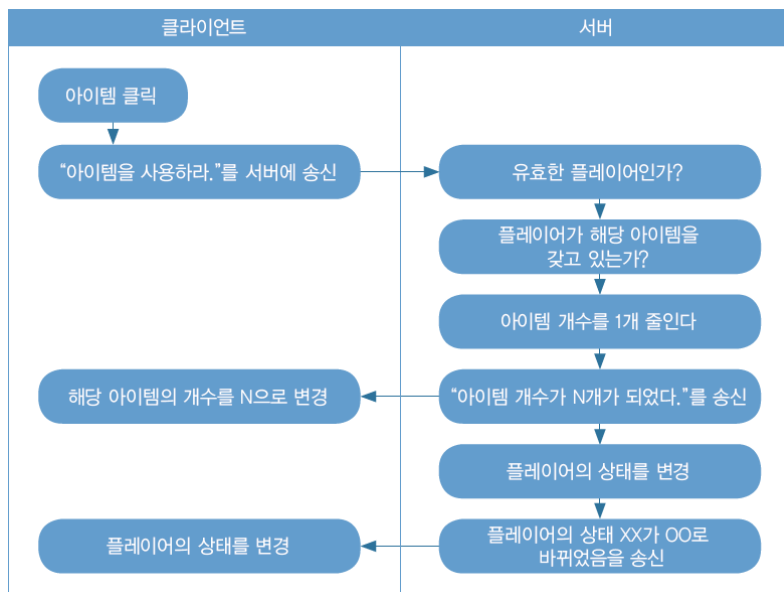
그림 5-21

두 플레이어가 방을 만들고 들어가는 과정의 시퀀스도

- 왼쪽 클라이언트는 서버에 “방 하나를 만들어라.”라고 요청.
- 서버는 방을 만들고 만든 방을 자신의 메모리에 보관하고 클라이언트에 “방 하나를 만들었다.”라고 응답.
- 이미 서버 접속 상태를 유지하고 있던 오른쪽 클라이언트에 “방 하나가 새로 만들어졌으니 너도 알고 있어야.”라고 통보.
- 오른쪽 클라이언트의 플레이어는 이 방에 들어가고 싶다. 클라이언트는 서버에 “나는 이방에 들어가겠다.”라고 요청.
- 서버는 오른쪽 클라이언트에 “방에 성공적으로 들어왔다.”라고 응답한다. 동시에 방 정보와 방에 들어간 왼쪽 클라이언트의 존재도 알려 줌.
- 오른쪽 클라이언트가 방에 들어갔음을 왼쪽 클라이언트도 이제 알아야 한다. 따라서 서버는 자체 클라이언트에 “오른쪽 클라이언트가 네 방에 들어왔다.”라고 통보. 이제 양쪽 클라이언트는 서로의 존재를 알고 있으며, 둘 다 방에 들어간 상태.
- 왼쪽 클라이언트는 서버에 “게임을 시작하자.”라고 요청.
- 서버는 오른쪽 클라이언트에 “게임을 시작하라.”라고 통보한다.
- 서버는 왼쪽 클라이언트에 “게임을 시작하라.”라고 통보한다.

## 5.7 | 게임 플레이 이외의 네트워킹

- 게임 로직을 개발하는 과정에서 서버와 클라이언트의 대화 규칙
  1. 클라이언트에서는 요청을 보낸다.
  2. 서버에서는 그 요청을 받아 결과를 판단한다.
  3. 요청 결과에 영향을 받을 다른 클라이언트가 서버에 있으면 그 클라이언트에 통보한다.
- 아이템 사용하기 과정 설계하기
  1. 클라이언트에서는 사용하는 아이템의 식별자(ID)를 인자로 담은 메시지를 서버에 전송한다.
  2. 서버에서는 해당 아이템을 플레이어가 사용할 수 있는 상태인지 판단해서 아이템 사용 결과를 판정한다.
  3. 그 결과를 클라이언트에 알려 주어 아이템이 사용됨을 고지한다.



거시적인 흐름을 표현해야 할 때는 시퀀스 다이어그램을 사용하고, 세부적인 로직을 표현해야 할 때는 액티비티 다이어그램을 사용한다.

그림 5-22  
아이템 사용 처리를  
액티비티 다이어그램으로 표현