|  |
| --- |
| *School of Computer Science & Engineering* ***Network Programming*** *June 09 2019*  **Number Quiz Game(Up & Down)**  **Class day**(수) Student No. (2013013017) Name: (윤신웅(Yoon Shin Woong) |

1. **Motivation & Objectives**

* Motivation: Describe the problem you are going to solve herekl

Server/Client를 활용한 프로젝트를 기획하는 데 있어, 가장 흔하게 접할 수 있고 흔하게 떠올릴 수 있는 주제로 게임을 선정하였다. 그런데 주제의 후보로 선정했던 많은 게임들은 curses와 같은 라이브러리를 통해 스크린을 조작할 때 높은 완성도를 가져올 수 있다. 그렇지만 curses 라이브러리를 활용할 경우 Network Programming이 핵심인 프로젝트에서 거리가 멀어지게 될 것을 고려하여 Command Line Terminal 만을 통해 게임을 구현하고자 하였다. 그렇게 기획하게 된 게임이 Up & Down이었다. Command Line Terminal을 통해 Server와 Client 간의 숫자를 주고받고 하면서 Server가 출제한 문제를 맞히는 형태의 게임을 만들어보고자 하였다.

* Objectives: Describe the objective/purpose of this project

이러한 Number Quiz Game을 통해, 게임의 형태 자체는 가벼운 게임이더라도 이 게임이 동작되는 Server/Client 구조 자체는 절대 가볍지 않게 구현하고자 하였다. 탄탄하게 Server/Client 구조를 만들어 실제 게임이 구현되는 환경을 최대한 유사하게 구현하고자 하였다. 그래서 구체적인 목표와 목적성은 다음과 같이 두 가지로 정리된다.

1. Number Quiz Game을 Server/Client 구조를 통하여 실제 게임이 동작되는 모드와 최대한 유사한 형태로 구현하고자 한다(Server에서 제출한 문제를 Client에서 푸는 방식으로 구현한다).
2. 단순히 Server/Client 구조를 넘어, Client(User) 간의 경쟁 모드를 추가하여 게임 결과를 순위로 매길 수 있도록 한다.

* Expected results: Describe the results that are expected to be obtained from this project

이러한 Number Quiz Game을 통해 실제 터미널에서 잘 동작되는 게임의 형태를 완성해볼 것이다. 실제 패키지 게임이 동작되는 방식으로 구현하여 Server/Client 구조를 보다 깊게 이해하고 게임에서도 적용될 수 있는 형태를 이해하고자 한다. 나아가 이 Server/Client 구조에서 실제 문제를 내고 맞히는 과정을 조정하면 얼마든지 다른 게임의 형태로도 변형할 수 있도록 최대한 탄탄히 Server/Client 구조를 구현할 것이다.

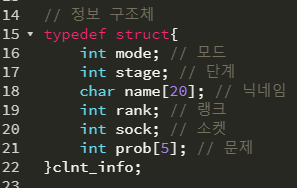
1. **Introduction of the Project**

* **Solutions of the problem**

이와 같은 게임의 형태를 구현하기 위해 해결해야 될 문제들이 많이 있었다. 많은 문제들 중 큰 문제들을 정리하면 다음과 같다.

1. Server에서 제출한 문제, Client에서 접속 시 희망한 User Name, Game Mode 등을 어떻게 공유할 것인가?
2. Client에서 제출한 Game Mode에 따라 Server/Client에서 어떻게 동작할 것인가?
3. 경쟁 모드로 동작 시 Client 간의 순위 공유는 어떻게 이루어 질 것인가?
4. 동작되는 동안 고려해야할 많은 예외들은 어떻게 처리할 것인가?

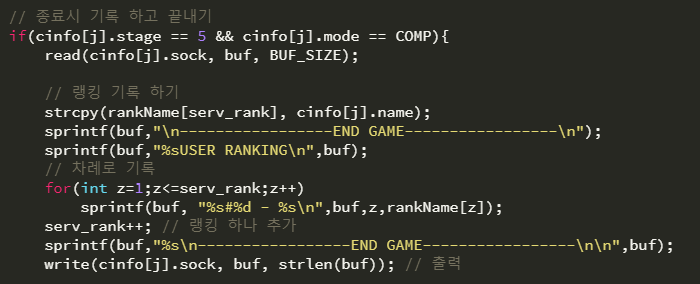
이 외에도 많은 문제들이 있었지만 다음과 같은 핵심적인 문제들에 대해서만 먼저 이야기하면, 먼저 첫 번째 ‘Server에서 제출한 문제, Client에서 접속 시 희망한 User Name, Game Mode 등을 어떻게 공유할 것인가?’ 문제에 대한 해결책은 Server/Client 간에 Struct 자료형으로 Data를 주고받는 방식을 사용하는 것이다.



다음과 같은 clnt\_info 라는 구조체 자료형을 정의하여 해당 자료형에 정보를 기록하여 Server/Client 간에 정보를 주고받고자 하였다. 이 때, Client는 자신 하나만의 정보를 기록하고 있으면 되고 Server는 해당 자료형을 Array로 기록하여 모든 Client(User) 정보를 기록하고 있어야 한다.

그리고 ‘Client에서 제출한 Game Mode에 따라 Server/Client에서 어떻게 동작할 것인가?’ 에 대한 해결책은 다음과 같다. Client에서는 일단 Game Mode에 따라 다른 점은 SOLO 모드 일 때는 바로 시작하고 COMP(경쟁) 모드일 경우에는 대기자가 전부 모였을 때 시작한다는 점과 게임이 종료된 후에 순위 기록을 받을 것인가에 대한 차이가 있다. 그래서 큰 틀에서 게임이 동작되는 방식은 통일시킨 후에 게임 시작 전과 후에 대해서 if 문을 통해 모드에 따라 다른 방식으로 처리되도록 한다. 그리고 Server에서가 많이 다소 복잡한 형태를 고려해야 하는데, 일단 기본적으로 Client와 소통하는 방식은 Select를 통한 Multiplexing 형태를 고려하였다. 여기서 Client에서 정보를 보냈을 때 Server 에서는 해당 Client에서 처음 보내는 데이터라고 한다면 해당 Client(User)의 정보를 read하여 관리하도록 한다. 이 때 맨 처음 접속한 Client라고 한다면 이를 고려하여 서버의 게임 동작모드를 선정한다. 이 때 SOLO 모드로 선정된다고 한다면 후에 들어온 다른 Client들도 SOLO 모드로만 동작될 수 있고 COMP 모드를 선정하면 종료되도록 처리한다. 반대로 COMP 모드로 서버가 동작 중인데 SOLO 모드로 들어온다면 종료되도록 처리한다. 그리고 마찬가지로 게임이 동작되는 형태는 통일시킨 후에 COMP 모드로 동작 중에는 게임이 종료되면 순위를 기록하여 Client에게 전송한다. (보다 자세한 사항은 다음 Overview of your solution을 참고해주세요)

그리고 다음으로 ‘경쟁 모드로 동작 시 Client 간의 순위 공유는 어떻게 이루어 질 것인가?’에 대한 해결책은 Server에서 Client가 게임을 클리어할 경우 순위를 차례로 기록하여 배열로 기록한다. 그리고 기록한 값을 바로 Client에게 바로 전송한다.

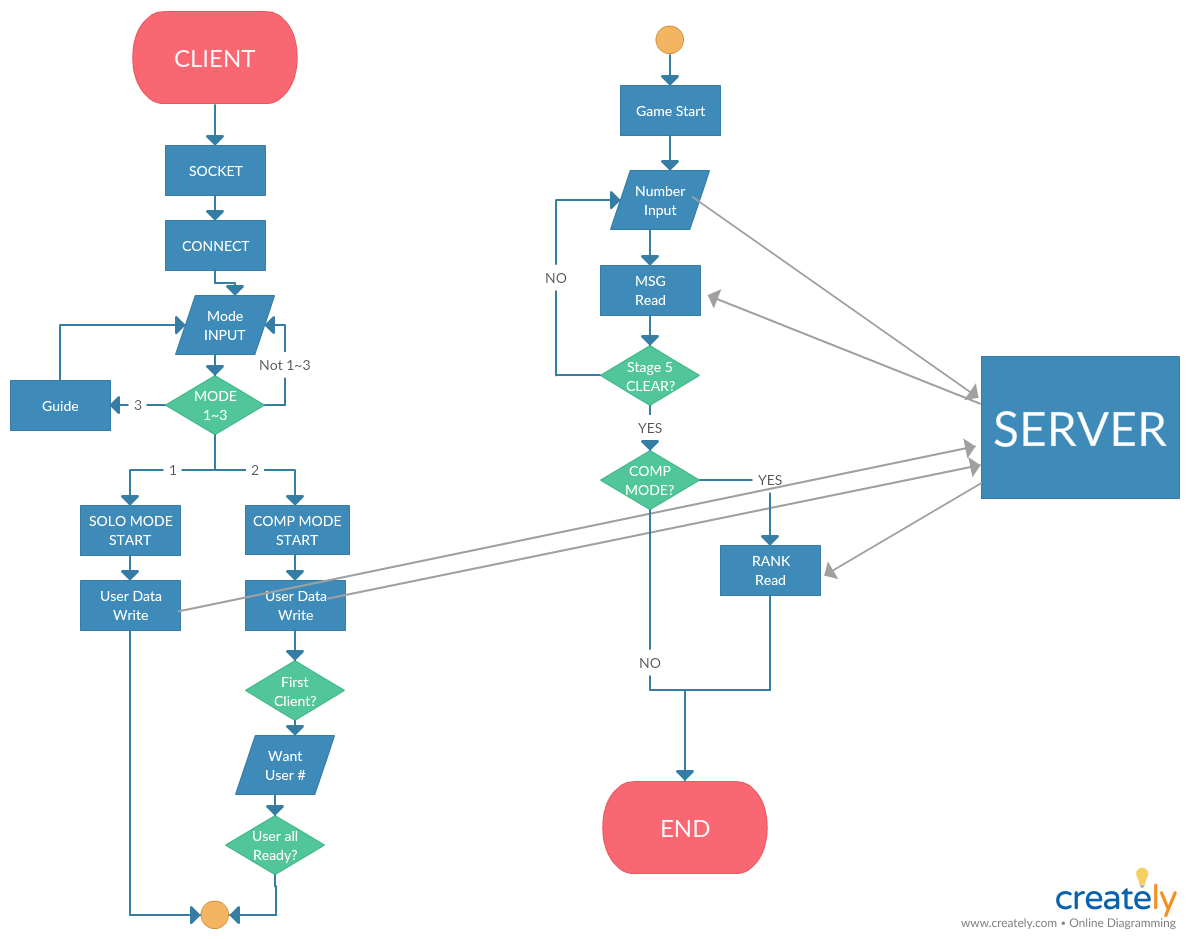


마지막으로, ‘동작되는 동안 고려해야할 많은 예외들은 어떻게 처리할 것인가?’에 대한 해결책은 고려할 수 있는 모든 사항들을 정리하였다. 입력에 대한 예외 처리(모드 1~3 외에 다른 숫자 입력 시, 숫자 범위인 1~100 외에 다른 숫자를 입력할 시 등)부터 Server/Client 구조에서의 동작 모드에 대한 예외 등을 모두 정리하여 구현할 수 있도록 하였다.

* **Overview of your solution: via flowchart, FSM, timeline tables, etc. (anything that can clearly describe your project)**

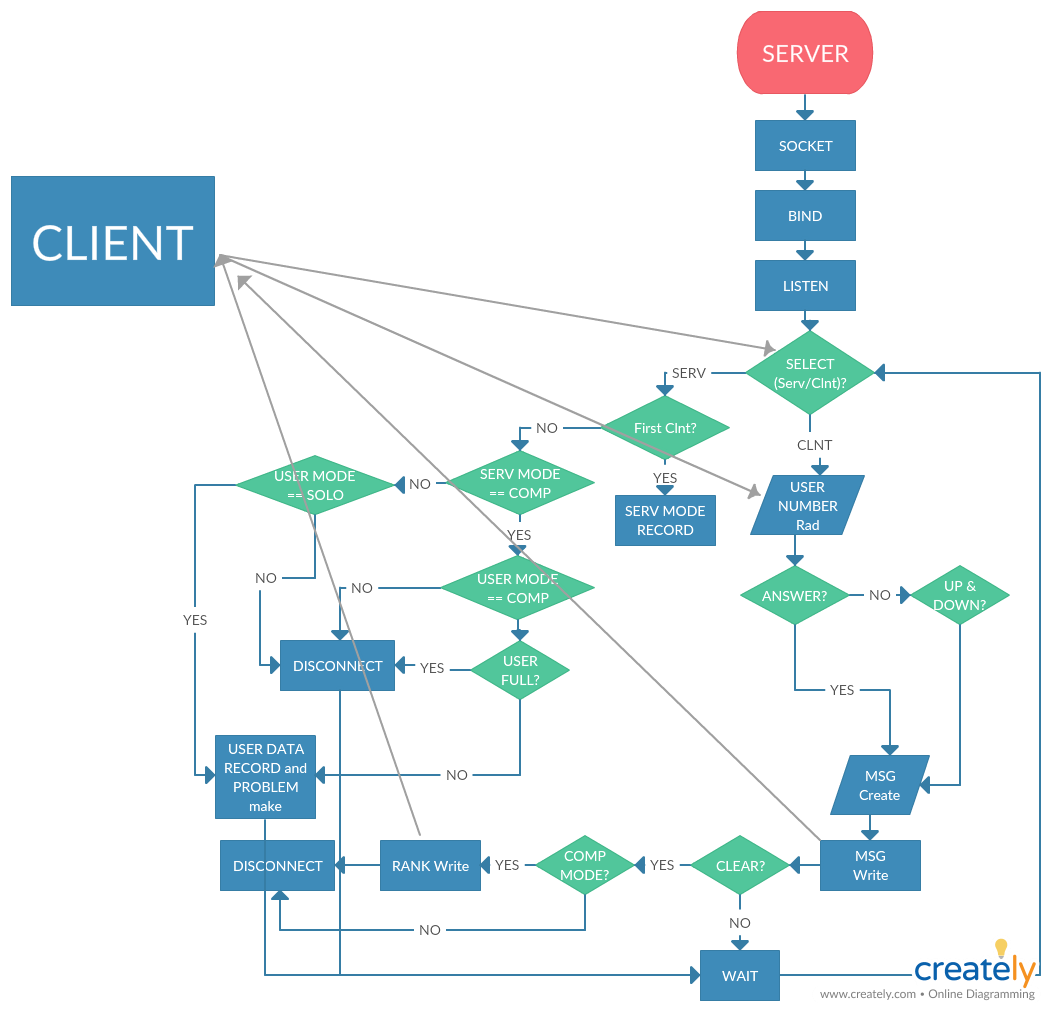
기본적으로 Server/Client가 동작되는 방식을 flowchart를 통해 표현하였다. 구체적으로는 보다 많은 상황이 들어있겠지만 동작되는 방식의 큰 흐름은 다음과 같이 표현할 수 있다..

CLIENT



먼저, Client에서는 시작하면 Socket과 Connect 처리 후 모드를 입력 받는다. 모드가 3일 경우 게임에 대한 소개를 후 다시 모드를 입력 받도록 한다. 만약 모드가 1이 들어오면 SOLO 모드로 바로 유저 정보를 서버에 보내고 게임을 시작처리를 한다(단, 여기서 Server에서 COMP 모드로 동작 중이면 예외처리를 통해 게임을 시작할 수 없고 연결이 끊긴다). 그리고 COMP 모드를 입력하면 유저 정보를 서버에 보낸다(이 때도 마찬가지로 Server에서 예외처리를 통해 서버에서 SOLO 모드로 동작 중이면 연결이 끊긴다). 그리고 만약 Server에 연결된 첫 유저라면 Room Manager가 되어 게임의 총 유저 수를 입력한다. 그리고 모든 유저가 게임을 시작할 수 있는 상태가 되면 게임이 시작된다. 숫자를 입력하면 해당 숫자를 Server에 출력한다. 그리고 Server에서 이를 보고 판단해주면 다시 Client에서 읽은 다음, 모든 Stage를 Clear할 때까지 반복한다. 그리고 모두 끝난다면 COMP 모드였을 경우에는 순위 또한 Server로부터 읽은 다음 출력하고 종료한다.

SERVER



Server에서는 다음과 같은 Flowchart로 동작한다. Socket, bind, listen 후에 Select를 통해 동작하는데 Server socket에 데이터가 들어오면 연결처리를 한다. 이 때, 여러 가지 예외상황을 고려하여 예외처리를 해주도록 하고, 연결이 허용된 경우에는 유저 정보를 읽어서 기록하고 연결한다. 예외상황들은 Server 모드와 User 모드가 다를 경우와 같더라도 Server 모드가 COMP 일 경우에 User가 포화상태일 경우에는 연결을 할 수 없도록 처리한다. 그리고 Select에서 Client socket으로 데이터가 들어오면 값을 읽어 들인 다음 판단하여 message를 작성해서 Client에 출력한다. 그리고 Client(User)가 모든 Stage를 Clear하면 종료를 하는데, 이 때 COMP 모드일 경우 순위를 기록하여 출력해준다.

* **Functionality of the project: what functions are realized in your project**

해당 프로젝트에서 구현된 기능은 크게 다음과 같다.

1. 게임의 형태로 동작하는 Multiplexing Server/Client 구조
2. Client 간의 Communication 구조
3. 2가지 다른 형태로 동작할 수 하나의 Server 구조

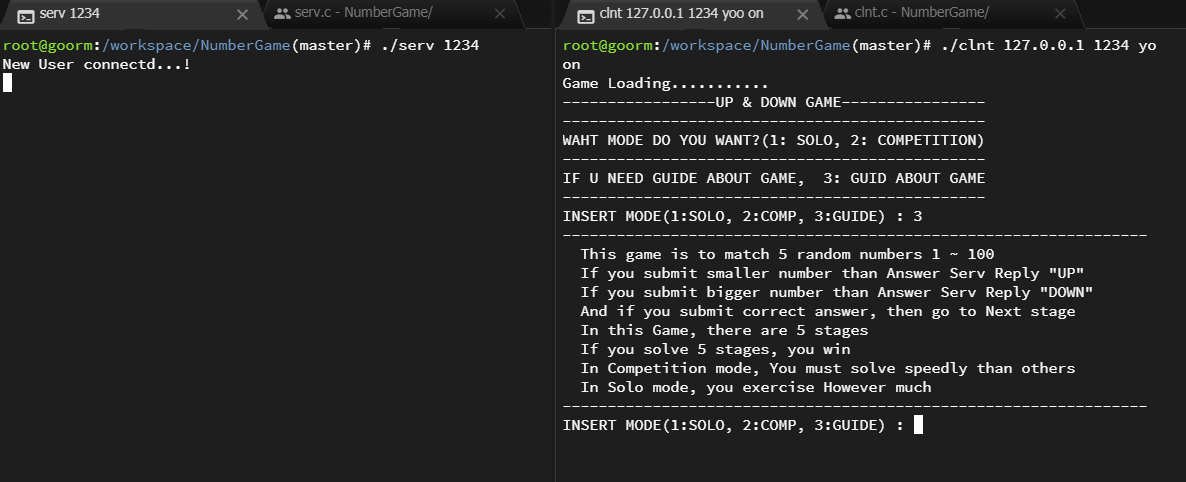
위와 같은 기능으로 구현된 프로젝트이기에 후에 다양한 형태의 게임 및 Server/Client 구조로 확장 및 변형이 용이할 것으로 판단된다. 추가적으로 Server/Client에서 단순히 char\* message 외에도 Struct 구조체를 주고 받는 것, 다양한 예외처리를 고려했다는 것 등 여러 기능적인 면 또한 고려된 프로젝트이다.

1. **Results**

* Please paste the screenshots of your execution results here and explain them

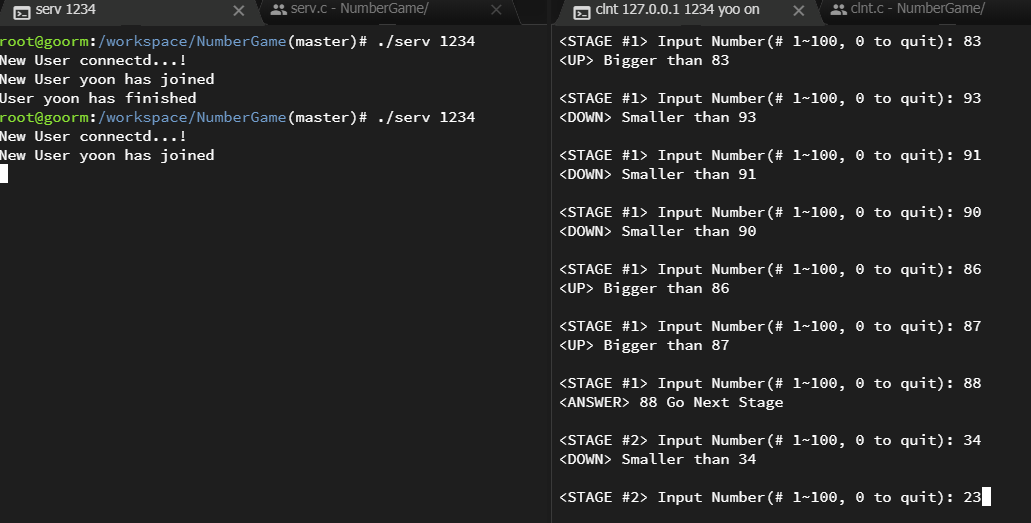
실행의 결과는 다음과 같다.

1. GUIDE MODE



기본적인 GUIDE MODE는 입력을 하면 다음과 같이 게임에 대한 Rule과 게임이 동작되는 방식에 대해서 간략히 안내를 받고 다시 모드를 입력할 수 있다.

1. SOLO MODE



SOLO MODE에서는 혼자서 게임을 시작하여 진행하는데 숫자를 총 5 Stage로 구성되어 있으며, 숫자를 입력하면 해당 숫자가 정답의 숫자보다 크다면 DOWN을 통해 낮추라고 말해주고, 정답보다 작다면 UP을 통해 높이라고 말해준다. 이러한 과정을 통해 5 Stage 모두 클리어하게 되면 게임은 종료되고 Client 프로그램도 닫히게 된다. Server에서도 더 이상 진행중인 게임이 없으면 자동으로 종료하도록 한다.

1. COMP MODE



COMP MODE 에서는 처음 클라이언트는 위와 같이 원하는 User 수를 입력하여 방을 꾸릴 수 있다. 위의 실행 결과에서는 User 수를 2로 입력하여 2명에서 경쟁하는 게임을 생성하였고 대기 중이다. 그리고 두 번째 클라이언트도 접속하게 되면 게임은 시작되고 2 Client 모두 Stage 1부터 시작한다. 둘 중 먼저 5 Stage를 클리어한 유저는 Rank 1로 기록되고 결과를 출력받고 종료한다. 2 번째 클라이언트도 5 Stage를 모두 클리어하면 Rank 2로 기록되어 1등, 2등의 순위를 Server로부터 받을 수 있다.