**3. 서버**

실질적인 게임을 실행하는 부분이다. 웹소켓을 이용해서 클라이언트에서 받은 정보로 대기실을 운영하거나 게임을 진행하고 처리한 결과를 다시 클라이언트로 보낸다.

**3-1. 소스파일**

main.cpp : main문이 있고 사용자가 웹페이지 접속 또는 나갔을 때와 클라이언트에서 메시지를 받는 부분

message.cpp : 웹소켓과 통신하기위해 쓰이는 메시지를 처리하는 부분. 받은 메시지를 분류하고 프로그램(서버)에서 처리한 결과를 메시지로 보낸다.

message.hpp : message.cpp의 헤더파일

room.cpp : 대기실과 관련된 작업을 처리하는 부분. 방 만들기, 방 입장, 방 나가기 등을 처리한다.

room.hpp : room.cpp의 헤더파일

map.cpp : 맵과 관련된 일을 하는 부분. 주어진 조건대로 Vertex들과 Edge들을 생성하여 맵을 구성한다.

map.hpp : map.cpp를 위한 헤더파일

game.cpp : 게임과 관련된 일을 하는 부분. 게임을 진행시키고 게임 정보를 저장한다.

game.hpp : game.cpp를 위한 헤더파일

**3-2. 메시지 리스트**

|  |  |  |  |
| --- | --- | --- | --- |
| 방향 | 이름 | 설명 | (서버에서 받았을 때 )부르는 함수 |
| 클->서 | Request\_username | 사용자가 정한 닉네임을 보낸다. | setName() |
| 서->클 | Player\_init | 정해진 사용자의 ID와 닉네임을 보낸다. |  |
| 클->서 | Request\_room\_list | 대기실들의 정보를 요청한다. | roomList() |
| 서->클 | Room\_list | 대기실들 정보를 보낸다. |  |
| 클->서 | Request\_room\_create | 방 생성을 요청한다. | createRoom(), joinRoom() |
| 서->클 | Room\_join\_accept | 방에 입장됨을 알리고 방 정보를 보낸다. |  |
| 서->클 | Room\_player\_join | 방에 있는 유저들에게 새로 들어온 유저 정보를 보낸다. |  |
| 클->서 | Request\_room\_join | 방 입장을 요청한다. | checkroom(), joinRoom() |
| 서->클 | Room\_join\_reject | 방 입장이 거절됨을 알리고 거절 사유 코드를 보낸다. |  |
| 클->서 | Request\_room\_leave | 방 나가기를 요청한다. | checkroom(), leaveRoom() |
| 서->클 | Room\_player\_leave | 방에 있는 유저들에게 나간 유저 정보를 보낸다. |  |
| 클->서 | Request\_game\_start | 게임 조건을 보내고 게임 시작을 요청한다. | startGame(), makeMap(), createVertex(), excludeVertex(), createVertex(), excludeVertex() |
| 서->클 | game\_start | 게임이 시작됨을 알리고 정해진 게임 정보를 보낸다. |  |
| 클->서 | Request\_role\_select | 선택한 역할을 요청한다. | selectRole(), allSelected() |
| 서->클 | Role\_select | 게임 유저들에게 유저가 선택한 역할을 보낸다. |  |
| 서->클 | Game\_role\_data | 게임 유저들에게 모든 유저의 역할 정보를 보낸다. |  |
| 클->서 | Request\_map\_data | 맵 정보를 요청한다. |  |
| 서->클 | Game\_map\_data\_start | 맵 정보 보내기가 시작됨을 알린다. |  |
| 서->클 | Vertex\_create | 맵의 Vertex 정보를 보낸다. |  |
| 서->클 | Edge\_create | 맵의 Edge 정보를 보낸다. |  |
| 서->클 | Game\_map\_data\_end | 맵 정보 보내기가 끝남을 알린다. |  |
| 클->서 | Request\_agent\_create | 말 생성을 요청한다. | createPlayer() |
| 서->클 | Agent\_create | 게임 유저들에게 생성된 말 정보를 보낸다. |  |
| 클->서 | Request\_turn\_start | 턴 시작을 요청한다. | findTurn() |
| 서->클 | Agent\_move\_turn | 게임 유저들에게 이동할 차례의 말 정보를 보낸다. |  |
| 클->서 | Request\_agent\_move | 말 이동을 요청한다. | movePlayer(), arrested(), gameWon(), findTurn() |
| 서->클 | Agent\_move | 게임 유저들에게 말 이동 정보를 보낸다. |  |
| 서->클 | Agent\_caught | 게임 유저들에게 말이 잡혔음을 알리고 잡힌 말 정보를 보낸다. |  |
| 서->클 | Game\_end | 게임 유저들에게 게임이 끝남을 알리고 승패 정보를 보낸다. |  |

# **서버 설계**

실질적인 게임을 실행하는 부분이다. WebSocket을 이용해서 클라이언트에서 받은 정보로 대기실을 운영하거나 게임을 진행하고 처리한 결과를 다시 클라이언트로 보낸다.

## **파일 구성**

main.cpp : main문이 있고 사용자가 웹페이지 접속 또는 나갔을 때와 클라이언트에서 메시지를 받는 부분

message.cpp : 웹소켓과 통신하기위해 쓰이는 메시지를 처리하는 부분. 받은 메시지를 분류하고 프로그램(서버)에서 처리한 결과를 메시지로 보낸다.

message.hpp : message.cpp의 헤더파일

room.cpp : 대기실과 관련된 작업을 처리하는 부분. 방 만들기, 방 입장, 방 나가기 등을 처리한다.

room.hpp : room.cpp의 헤더파일

map.cpp : 맵과 관련된 일을 하는 부분. 주어진 조건대로 Vertex들과 Edge들을 생성하여 맵을 구성한다.

map.hpp : map.cpp를 위한 헤더파일

game.cpp : 게임과 관련된 일을 하는 부분. 게임을 진행시키고 게임 정보를 저장한다.

game.hpp : game.cpp를 위한 헤더파일

## **통신 프로토콜**

서버와 클라이언트 간 통신을 위한 메시지 프로토콜은 다음과 같이 정의된다.

|  |  |  |  |
| --- | --- | --- | --- |
| 방향 | 이름 | 설명 | 대응 함수 |
| 클->서 | request\_username | 사용자가 정한 닉네임을 보낸다. | setName() |
| 서->클 | player\_init | 정해진 사용자의 ID와 닉네임을 보낸다. |  |
| 클->서 | request\_room\_list | 대기실들의 정보를 요청한다. | roomList() |
| 서->클 | room\_list | 대기실들 정보를 보낸다. |  |
| 클->서 | request\_room\_create | 방 생성을 요청한다. | createRoom(), joinRoom() |
| 서->클 | room\_join\_accept | 방에 입장됨을 알리고 방 정보를 보낸다. |  |
| 서->클 | room\_player\_join | 방에 있는 유저들에게 새로 들어온 유저 정보를 보낸다. |  |
| 클->서 | request\_room\_join | 방 입장을 요청한다. | checkroom(), joinRoom() |
| 서->클 | room\_join\_reject | 방 입장이 거절됨을 알리고 거절 사유 코드를 보낸다. |  |
| 클->서 | request\_room\_leave | 방 나가기를 요청한다. | checkroom(), leaveRoom() |
| 서->클 | room\_player\_leave | 방에 있는 유저들에게 나간 유저 정보를 보낸다. |  |
| 클->서 | request\_game\_start | 게임 조건을 보내고 게임 시작을 요청한다. | startGame(), makeMap(), createVertex(), excludeVertex(), createVertex(), excludeVertex() |
| 서->클 | game\_start | 게임이 시작됨을 알리고 정해진 게임 정보를 보낸다. |  |
| 클->서 | request\_role\_select | 선택한 역할을 요청한다. | selectRole(), allSelected() |
| 서->클 | role\_select | 게임 유저들에게 유저가 선택한 역할을 보낸다. |  |
| 서->클 | game\_role\_data | 게임 유저들에게 모든 유저의 역할 정보를 보낸다. |  |
| 클->서 | request\_map\_data | 맵 정보를 요청한다. |  |
| 서->클 | game\_map\_data\_start | 맵 정보 보내기가 시작됨을 알린다. |  |
| 서->클 | vertex\_create | 맵의 Vertex 정보를 보낸다. |  |
| 서->클 | edge\_create | 맵의 Edge 정보를 보낸다. |  |
| 서->클 | game\_map\_data\_end | 맵 정보 보내기가 끝남을 알린다. |  |
| 클->서 | request\_agent\_create | 말 생성을 요청한다. | createPlayer() |
| 서->클 | agent\_create | 게임 유저들에게 생성된 말 정보를 보낸다. |  |
| 클->서 | request\_turn\_start | 턴 시작을 요청한다. | findTurn() |
| 서->클 | agent\_move\_turn | 게임 유저들에게 이동할 차례의 말 정보를 보낸다. |  |
| 클->서 | request\_agent\_move | 말 이동을 요청한다. | movePlayer(), arrested(), gameWon(), findTurn() |
| 서->클 | agent\_move | 게임 유저들에게 말 이동 정보를 보낸다. |  |
| 서->클 | agent\_caught | 게임 유저들에게 말이 잡혔음을 알리고 잡힌 말 정보를 보낸다. |  |
| 서->클 | game\_end | 게임 유저들에게 게임이 끝남을 알리고 승패 정보를 보낸다. |  |

## **Class, 전역변수**

Class

|  |  |
| --- | --- |
| **Class 이름** | **Class 설명** |
| **Attribute 이름** | **Attribute 설명** |
| **User** | **게임을 이용하는 유저 클래스** |
| webscoketpp::connection\_hdl id;  std::string user\_name; | user id  user 이름 |
| **Room** | **함께 게임할 유저들이 모이는 방 클래스** |
| unsigned int room\_id;  std::string title;  int capacity;  std::map <int, User> users;  user owner; | Room id  Room 제목  Room 최대수용인원  방에 입장한 유저들  방장 User |
| **Edge** | **맵을 구성하는 edge 클래스** |
| vertex\_id str;  vertex\_id dst; | edge의 시작 Vertexd id  edge의 도착 Vertex id |
| **Vertex** | **맵을 구성하는 vertex 클래스** |
| vertex\_id id;  int x  int y  std::vector <Edge> edges; | vertex id  Vertex x좌표  Vertex y좌표  Vertex에 연결된 Edge들의 vector |
| **Game** | **게임 클래스** |
| unsigned int game\_id;  unsigned int room\_id  int cop\_num;  int rob\_num;  Vertex\* map;  int map\_size;  std::multimap <Role, User> users  std::list <Player> players | 게임 id  게임이 시작된 Room id  게임 Cop 수  게임 Rob 수  게임 맵  맵 Vertex 개수  게임 플레이 중인 유저들  유저들이 조정하는 Player들 |
| **Player** | **게임 말 클래스** |
| unsigned int player\_id;  user user;  Role role;  vertex\_id position;  int turn; | Player id  Player를 조종하는 user  Player의 Role  Player이 위치한 Vertex id  말 이동 순서 |

전역변수

|  |  |  |
| --- | --- | --- |
| **자료구조** | **변수명** | **설명** |
| Std::list <User> | **user\_list** | 현재 접속된 유저들을 저장해둔 리스트 |
| Std::list <Room> | **rooms** | 현재 존재하는 대기실들을 저장해둔 리스트 |
| Std::list <Game> | **games** | 현재 진행되고 있는 게임들을 저장해둔 리스트 |