

Missile War Project



Overview

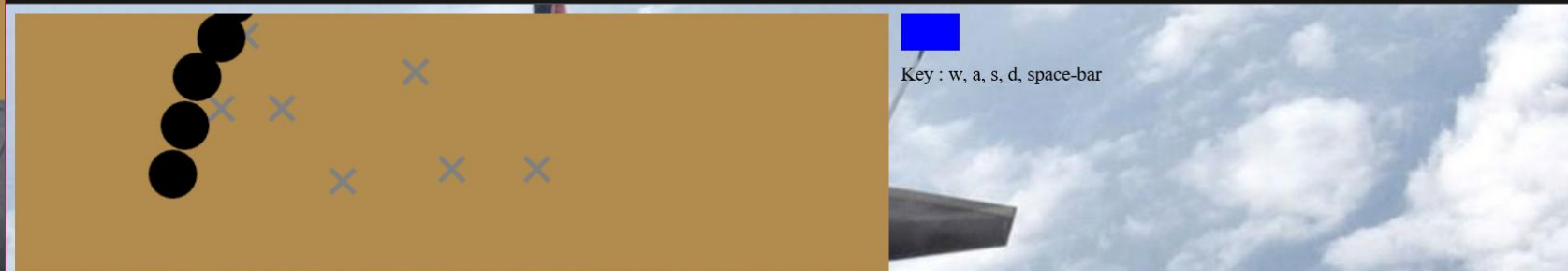
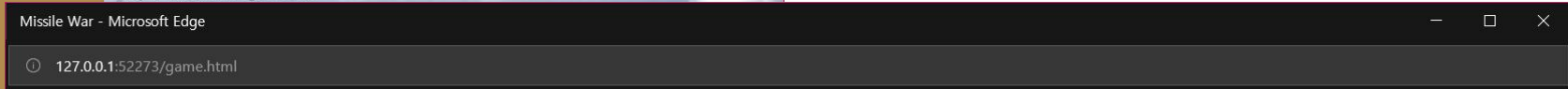
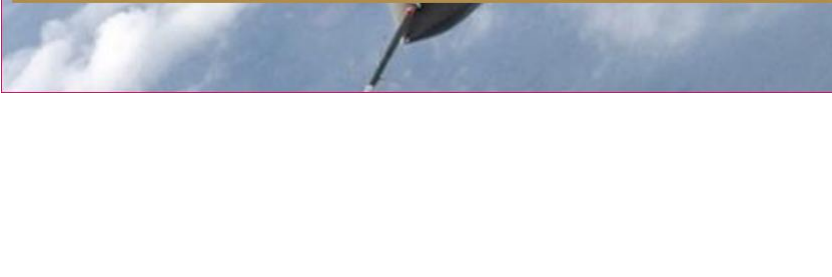
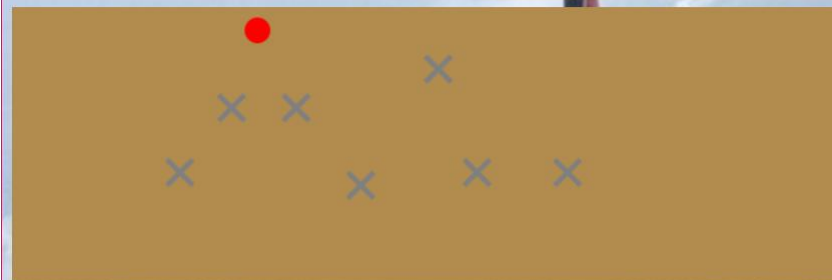
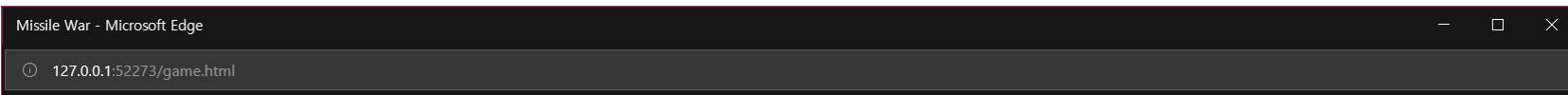
미사일 대전 프로젝트란?

- 미사일 대전은 웹 브라우저에서 서비스하는 실시간 다대다 슈팅 액션 게임이다.
- 게임 뿐만 아니라 채팅과 게시판 등 커뮤니티 기능과 로그인, 회원가입 등 회원관리 시스템을 총 통틀어서 “미사일 대전 프로젝트”라고 칭한다.

게임에 대해서 자세히

- Red팀과 Blue팀 양쪽으로 나누어져 서로 미사일을 발사하고 피하며 싸우는 방식
- 플레이어의 수에는 제한이 없다.
- 마나(Mana) 시스템이 적용되어 시간당 발사할 수 있는 미사일의 수가 제한

게임에 대해서 자세히



회원관리 시스템

- 로그인 - 회원이라면 아이디와 비밀번호를 기입하여 접속 가능
- 회원가입 - 회원이 아니라면 회원 등록 가능
- 카카오 로그인 - 회원이 아니어도 카카오 로그인을 통하여 접속 가능
- 프로필 - 각 회원은 자신의 프로필이 있고 조회할 수 있음

회원관리 시스템

로그인 - Microsoft Edge

127.0.0.1:52273/login.html

아이디(ID)를 입력해주세요

비밀번호(Password)를 입력해주세요

로그인 회원가입

카카오계정으로 로그인

회원가입 - Microsoft Edge

127.0.0.1:52273/sign_up.html

아이디(ID)를 입력해주세요

비밀번호(Password)를 입력해주세요

비밀번호(Password)를 다시 입력해주세요

회원가입

accounts.kakao.com - Microsoft Edge

https://accounts.kakao.com/login?continue=https://kauth.kakao.com/oauth/at

kakao 로그인 도움말 문의하기

chlcgy@naver.com

●●●●●●●●

☐ 로그인 상태 유지

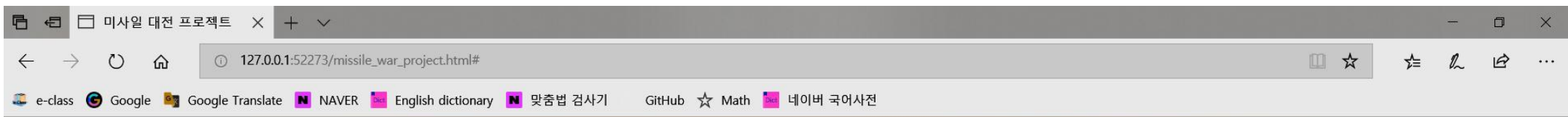
로그인

회원가입 카카오계정 찾기 | 비밀번호 재설정

공지사항 | 이용약관 | 개인정보 처리방침 | 운영정책 | 한국어 ▾

Copyright © Kakao Corp. All rights reserved.

회원관리 시스템



Kuenhwi | 로그아웃



Rank:
SUPER
Score: 47

홈

게임

커뮤니티

채팅방

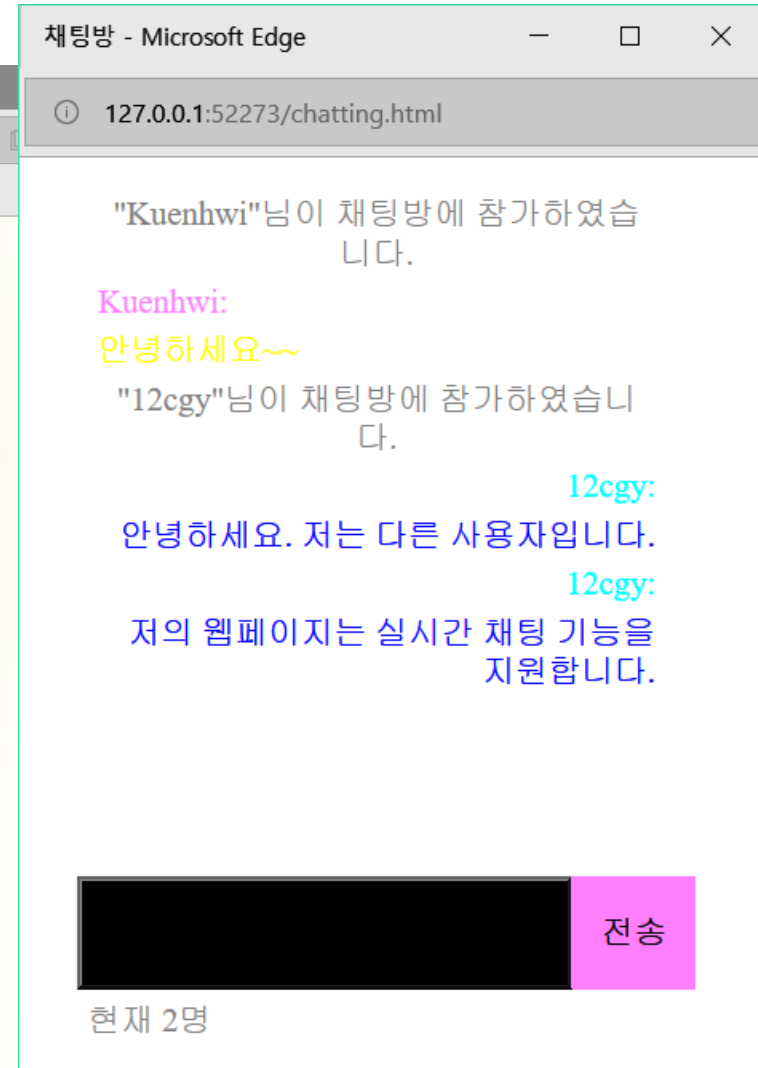
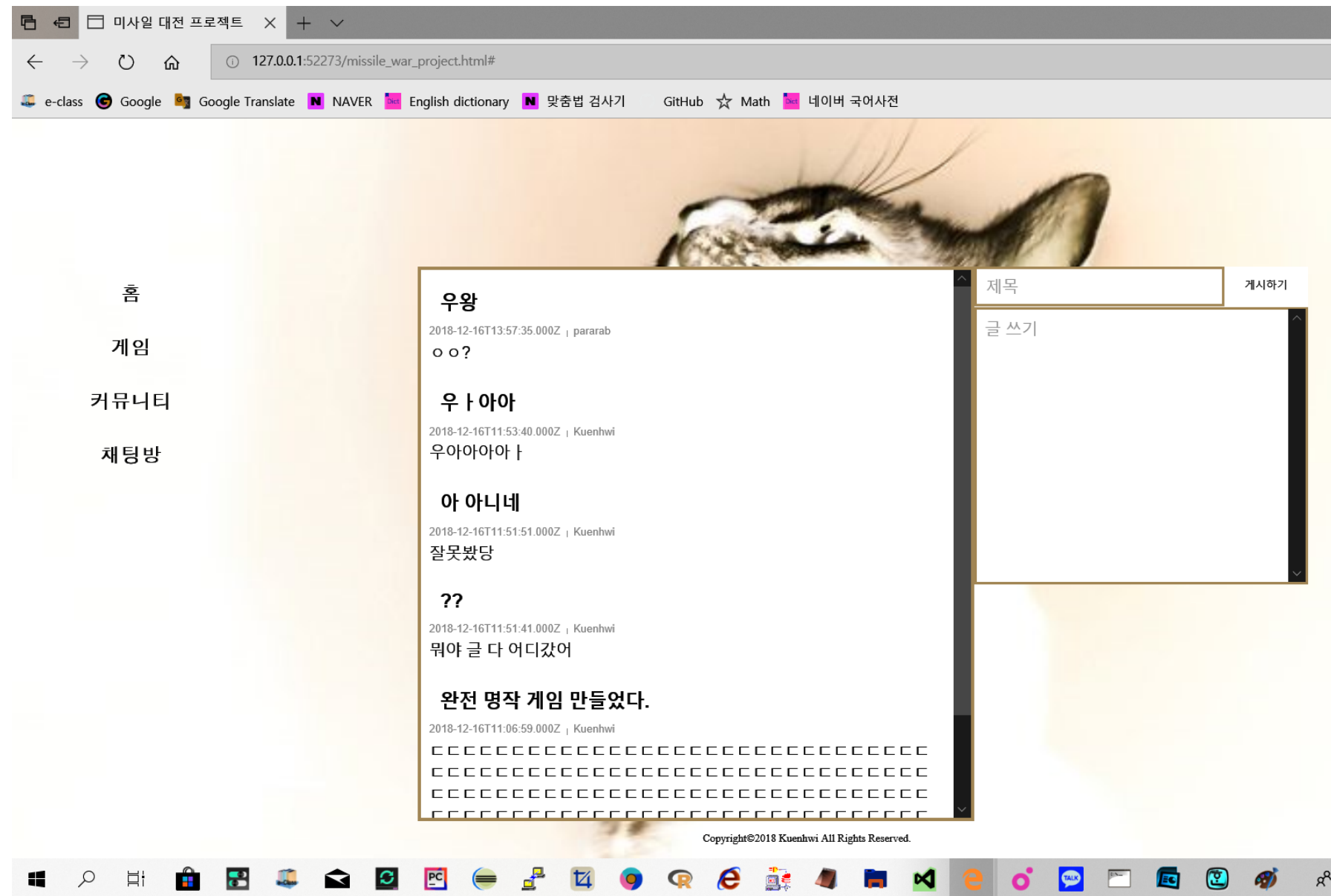
Copyright©2018 Kuenhwi All Rights Reserved.



커뮤니티 기능

- **게시판** - **회원**은 **글**을 써서 **게시**할 수도 있고 다른 회원들이 게시한 글들을 **읽을** 수도 있다.
- **채팅** - **실시간**으로 접속한 다른 회원과 채팅을 할 수 있다.
게시판과 다르게 **기록이 남지 않는다**.

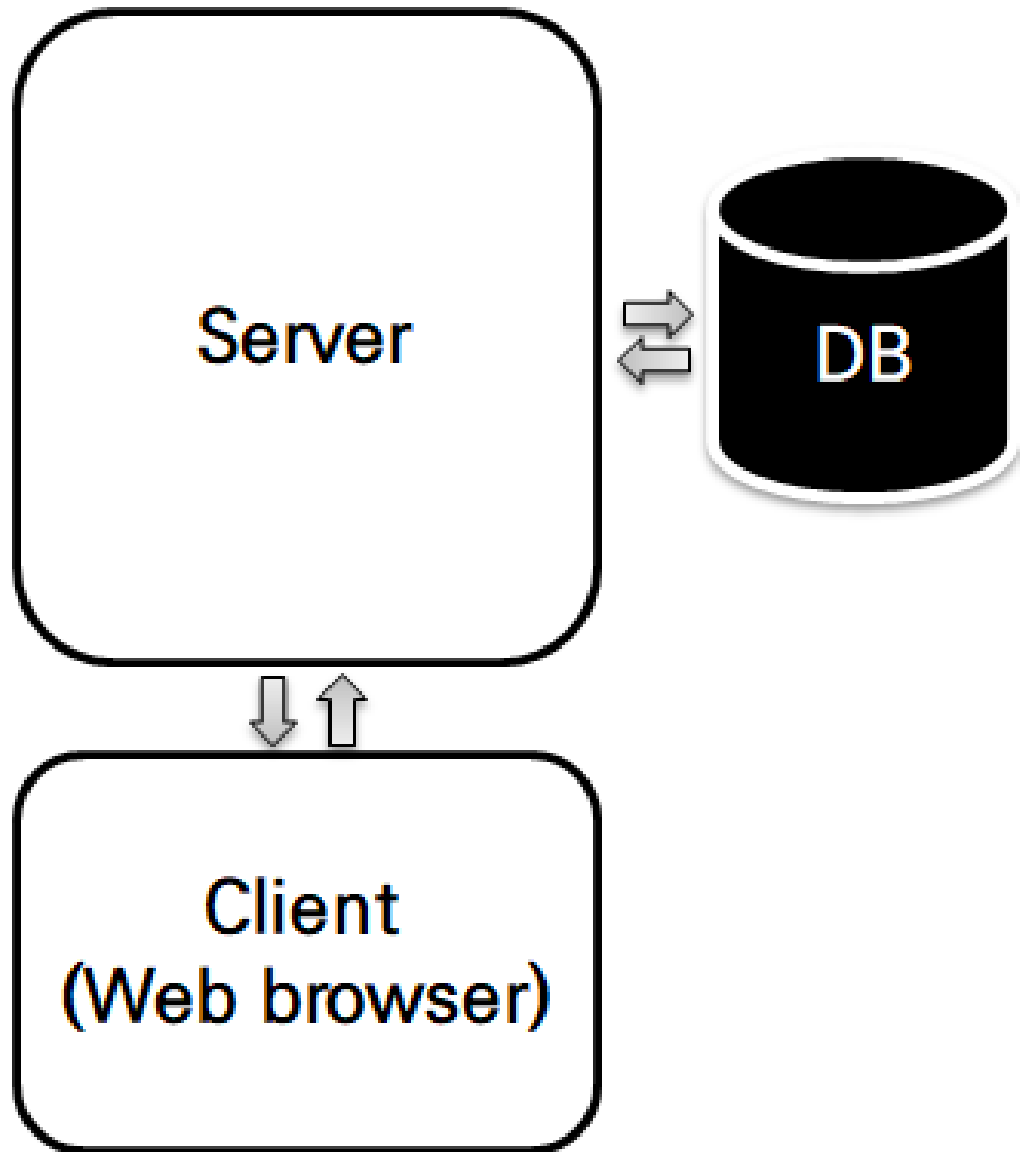
커뮤니티 기능 - 좌)게시판, 우)채팅



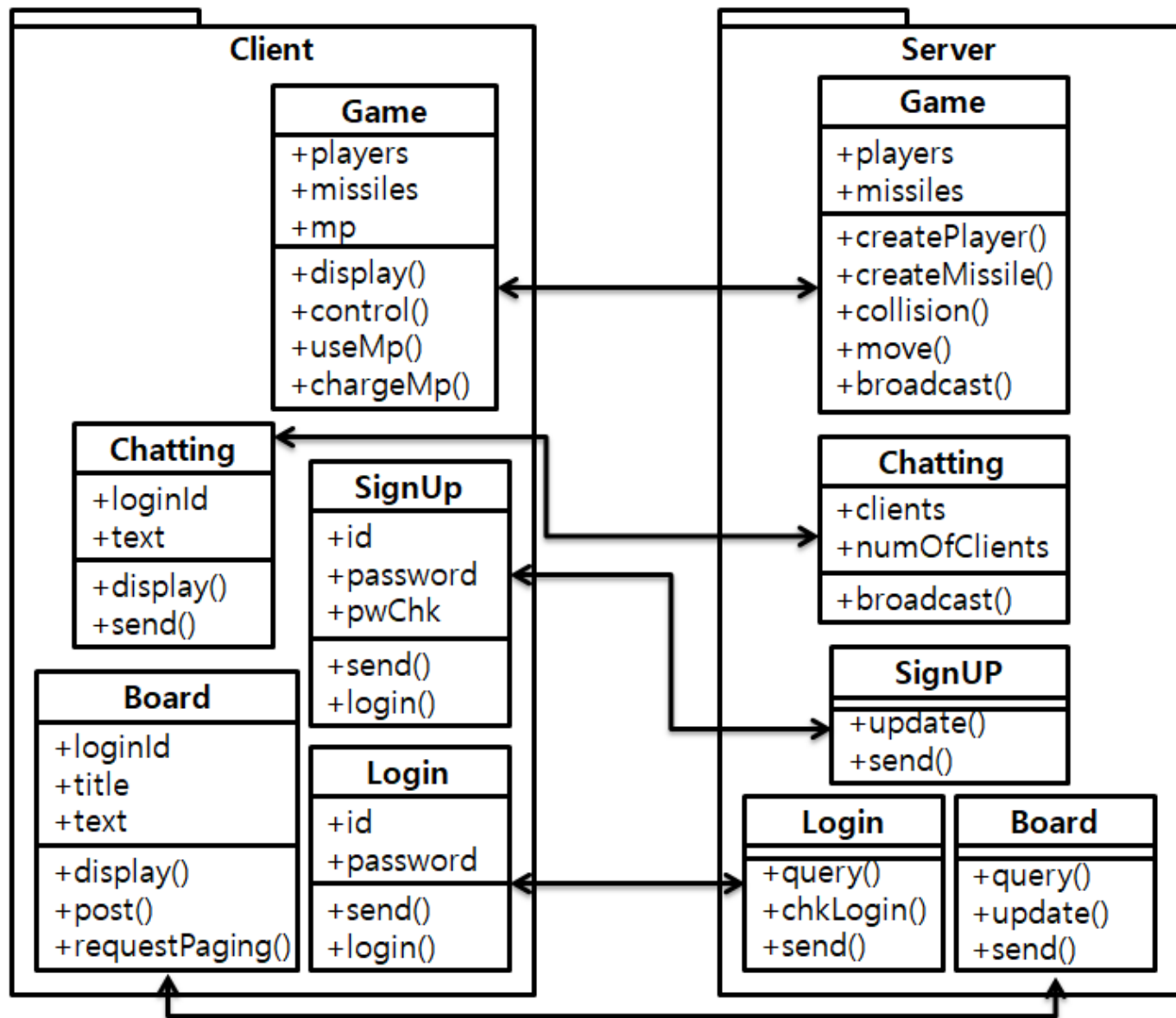


Project Design

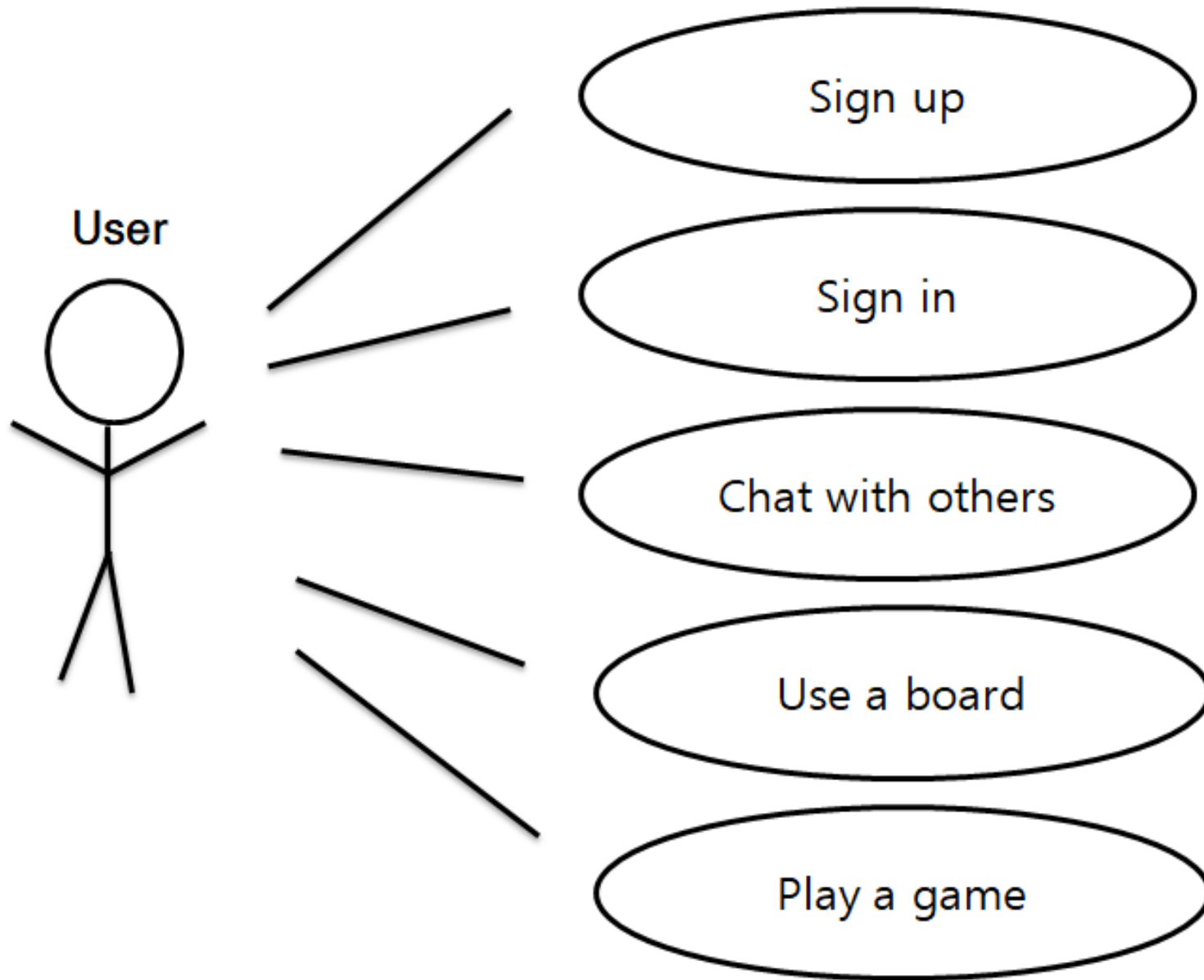
시스템 아키텍처



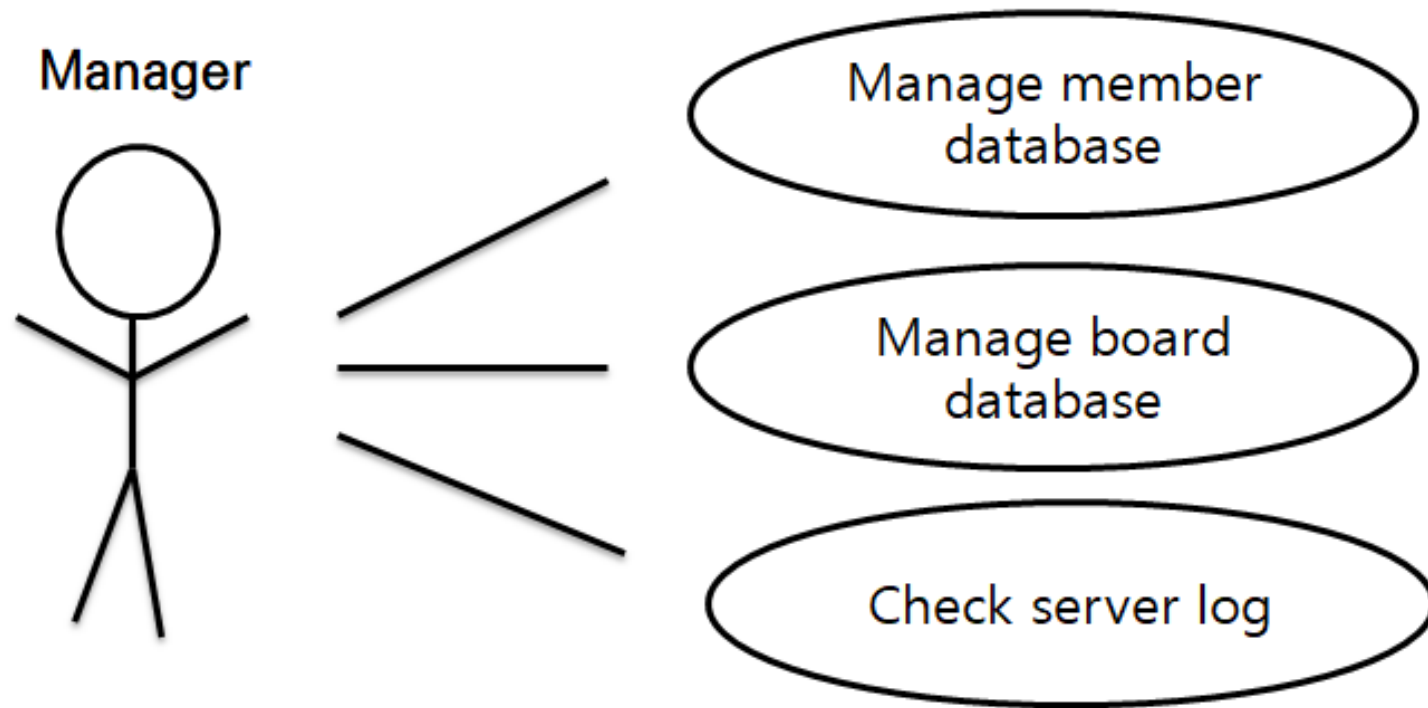
클래스 뷰



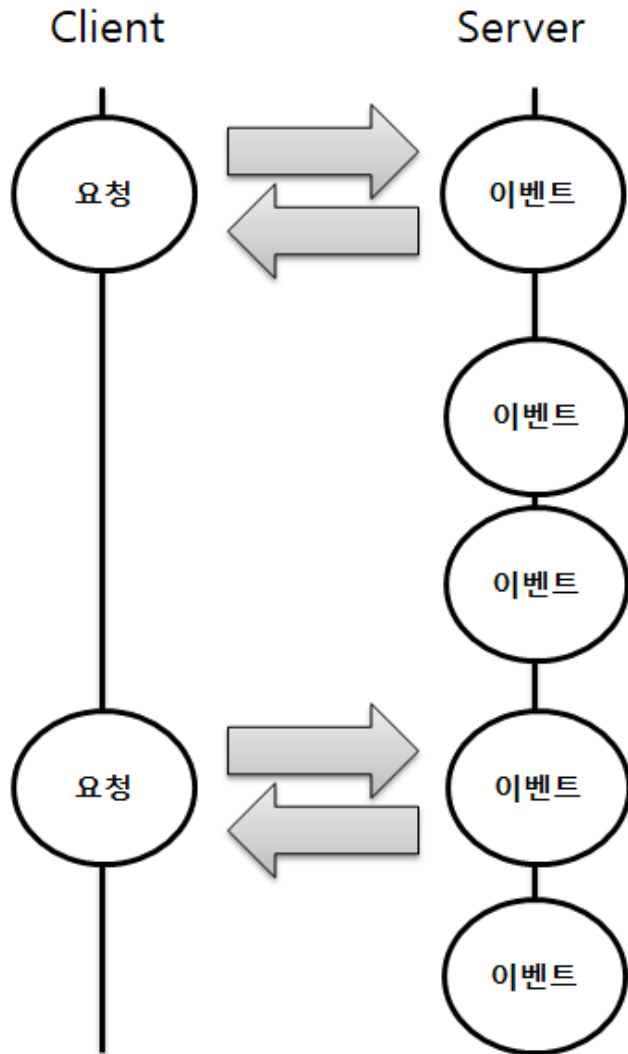
사용자 요구 사항



서버 관리자 요구사항



로그인, 회원가입, 게시판, 프로필 조회에서 클라이언트-서버간 상호작용 다이어그램

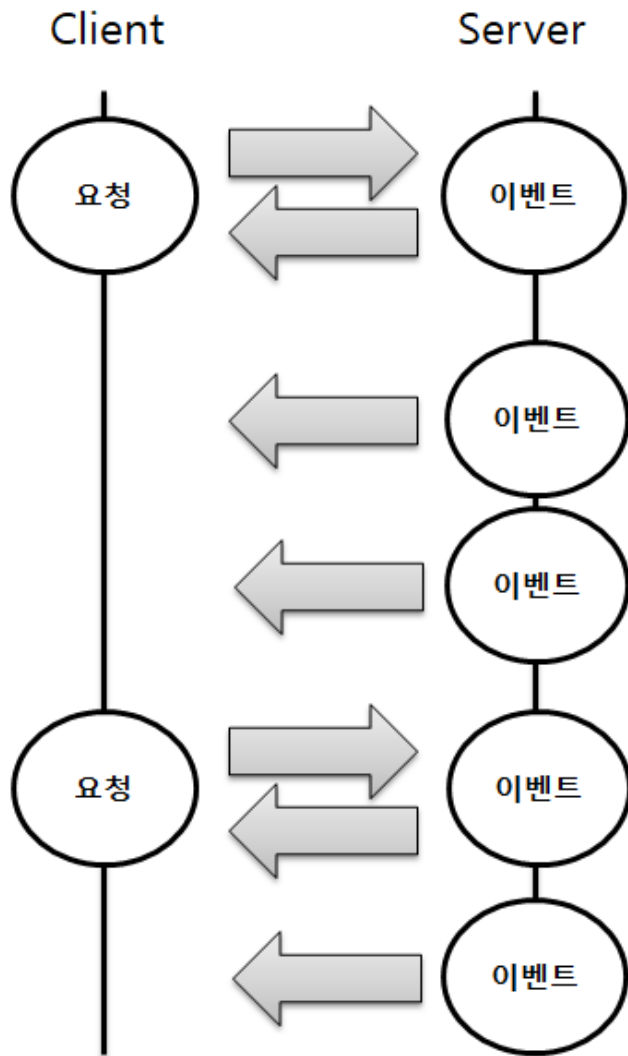


-> 클라이언트-서버간 **비동기적** 통신

-> **Polling** 기법

-> **Ajax**

게임, 채팅에서 클라이언트-서버간 상호작용 다이어그램



-> 클라이언트-서버간 **동기적** 통신

-> **Streaming** 기법

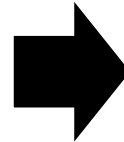
-> **Web-socket**



Details

클라이언트 요청 -> 서버 응답 (Ajax)

```
$.ajax({
  url: './profile/' + decodeURIComponent($.cookie('login_info')),
  type: 'GET',
  success: function (data) {
    my_rank = data['rank'];
    my_score = data['score'];
    var text = "";
    text += '<p>Rank: ' + my_rank.toUpperCase() + '</p>';
    text += '<p>Score: ' + my_score + '</p>';
    $('#profile_rank').html(text);
    $('#profile_card').css({ 'visibility': 'visible' });
  }
});
```



```
125 app.get('/profile/:id', function (request, response) {
126     var id = request.params.id;
127     db.query('select authority, score from member '
128         + " where id = binary('" + id + "')"
129     , function(error, rows, fields){
130         if (error) {
131             console.log(error);
132             response.send({
133                 result: 'failed',
134                 failType: error
135             });
136         }
137         else {
138             if (rows.length == 0) {
139                 response.send({
```

게시판 페이징

```
156 // 게시판 응답
157 app.get('/board/:offset', function (request, response) {
158     var offset = Number(request.params.offset);
159
160     db.query('select * from board\n'+ 'order by post_date desc\n' + 'limit ?, 5'
161         , [offset], function (error, rows, fields) {
162             if (error) {
163                 console.log(error);
164                 response.send({
165                     result: 'failed',
166                     failType: error
167                 });
168             }
169             else {
170                 var items = rows.length;
171                 var titles = [];
172                 var post_dates = [];
173                 var writers = [];
174                 var contents = [];
175                 for (var i = 0; i < items; i++) {
176                     titles.push(rows[i].title);
177                     post_dates.push(rows[i].post_date);
178                     writers.push(rows[i].writer);
179                     contents.push(rows[i].content);
180                 }
181             }
182         });
183     }
184     console.log('board data sent ');
185     response.send({
186         'items': items,
187         'titles': titles,
188         'post_dates': post_dates,
189         'writers': writers,
190         'contents': contents
191     });
192 });
```

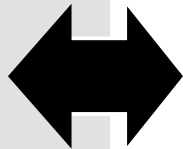
```
181 console.log('board data sent ');
182 response.send({
183     'items': items,
184     'titles': titles,
185     'post_dates': post_dates,
186     'writers': writers,
187     'contents': contents
188 });
189 }
190 });
191 });
```

:offset

-> 동적 라우팅 요청 매개변수

클라이언트 - 서버 웹 소켓 요약

```
18 // 웹소켓 생성
19 var ws = new WebSocket("ws://127.0.0.1:52274");
20 // 웹소켓이 생성되면
21 ws.onopen = function (event) {...};
38 // 응답
39 ws.onmessage = function (raw) {...};
70 // 연결상태 체크
71 function heartbeat() {...}
74 ws.on('open', heartbeat);
75 ws.on('ping', heartbeat);
76 ws.on('close', function() {...});
79 // 예외 처리
80 ws.onerror = function (error) {...};
```



```
234 // 채팅방 연결
235 chatSocket.on("connection", function (ws) {
236     ws.on('pong', heartbeat);
237     console.log("chatting socket conneted:" + Date());
238     //ws.send(chatDataForm('join', ws));
239     ws.on("message", function (raw) {
240         var data = decodeChattingData(raw);
241         var type = data['type'];
242         var who = data['who'];
243         var message = data['message'];
244
245         chatSocket.clients.forEach(function each(client) {
246             try {
247                 encodeChattingData('peoples', 'system', peoples); // < 인원 수 정보
248                 client.send(encodeChattingData(type, who, message));
249             } catch (e) {
250             }
251         });
252     });
253 });
```

게임 관련 처리 함수들(서버)

```
307  /*-----*/
308  ⊕var createRedPlayer = function ()...
317  ⊕var createBluePlayer = function ()...
326  ⊕var moveRedPlayer = function (redPlayer)...
351  ⊕var moveBluePlayer = function (bluePlayer)...
376  ⊕var createRedMissile = function (redPlayer)...
384  ⊕var createBlueMissile = function (bluePlayer)...
392  ⊕var moveRedMissile = function (redMissile)...
400  ⊕var moveBlueMissile = function (blueMissile)...
408  ⊕var collision = function (player, missile)...
415  ⊕var redPlayersCollision = function (redPlayers, blueMissiles)...
428  ⊕var bluePlayersCollision = function (bluePlayers, redMissiles)...
441  ⊖/*-----
442  | /* Game Thread -----
```

게임 렌더링(클라이언트) - 1

```
40 var display = function () {
41     if (!context)
42         context = canvas.getContext("2d");
43     context.clearRect(0, 0, canvas.width, canvas.height);
44     context.fillStyle = "#b28c4e";
45     context.fillRect(0, 0, width, redLine);
46     context.fillRect(0, blueLine, width, height);
47
48     for (var player in players) {
49         if (players[player]['status'] == die) {
50             // 회색 X자
51             context.beginPath();
52             context.moveTo(players[player]['x'] - radiusOfPlayer
53                 , players[player]['y'] - radiusOfPlayer);
54             context.lineTo(players[player]['x'] + radiusOfPlayer
55                 , players[player]['y'] + radiusOfPlayer);
56             context.moveTo(players[player]['x'] + radiusOfPlayer
57                 , players[player]['y'] - radiusOfPlayer);
58             context.lineTo(players[player]['x'] - radiusOfPlayer
59                 , players[player]['y'] + radiusOfPlayer);
60             context.lineWidth = 4;
61             context.strokeStyle = "grey";
62             context.stroke();
63         }
64         else {
65             context.beginPath();
66             context.arc(players[player]['x'], players[player]['y'], radiusOfPlayer,
```

```
66             context.arc(players[player]['x'], players[player]['y'], radiusOfPlayer,
67                 0, Math.PI * 2, false);
68             if (player == seed)
69                 context.fillStyle = "#00cf37";
70             else if (players[player]['team'] == red)
71                 context.fillStyle = "red";
72             else if (players[player]['team'] == blue)
73                 context.fillStyle = "blue";
74             context.fill();
75             context.closePath();
76         }
77     }
78     for (var missile in missiles) {
79         if (missiles[missile]['status'] == die) {
80             /* 의도적으로 비움 */
81         }
82         else {
83             context.beginPath();
84             context.arc(missiles[missile]['x'], missiles[missile]['y'], radiusOfMissile,
85                 0, Math.PI * 2, false);
86             context.fillStyle = "black";
87             context.fill();
88             context.closePath();
89         }
90     }
91 }
```

게임 렌더링(클라이언트) - 2

```
101 ws.onmessage = function (raw) {  
102     var data = JSON.parse(raw.data);  
103     var type = data['type'];  
104  
105     switch (type) {  
106         case 'seed':  
107             seed = data['value'];  
108             console.log(seed);  
109             break;  
110         case 'players':  
111             players = JSON.parse(data['value']);  
112             break;  
113         case 'missiles':  
114             missiles = JSON.parse(data['value']);  
115             break;  
116     }  
117     display();  
118 };
```




Future Improvements

개선 사항 - 1

게임 플레이 개선

- 새로운 공격, 회피기술 추가
- 그래픽 개선
- 배경음악과 효과음 추가
- 게임에서 참여하거나 승리할 시 회원에게 점수를 주도록 회원관리 시스템과 연동

개선 사항 - 2

보안

- 로그인, 회원가입 과정과 로그인 유지를 위한 쿠키를 암호화
- 회원가입 시 본인 인증 시스템 도입

수익 모델

- 광고 시스템 도입

Q & A