

# **CSCI 399**

## **Assignment 2**

**Student name: LiZhao Xiaoxiong**

**Student number:4520841**

## Table of Contents

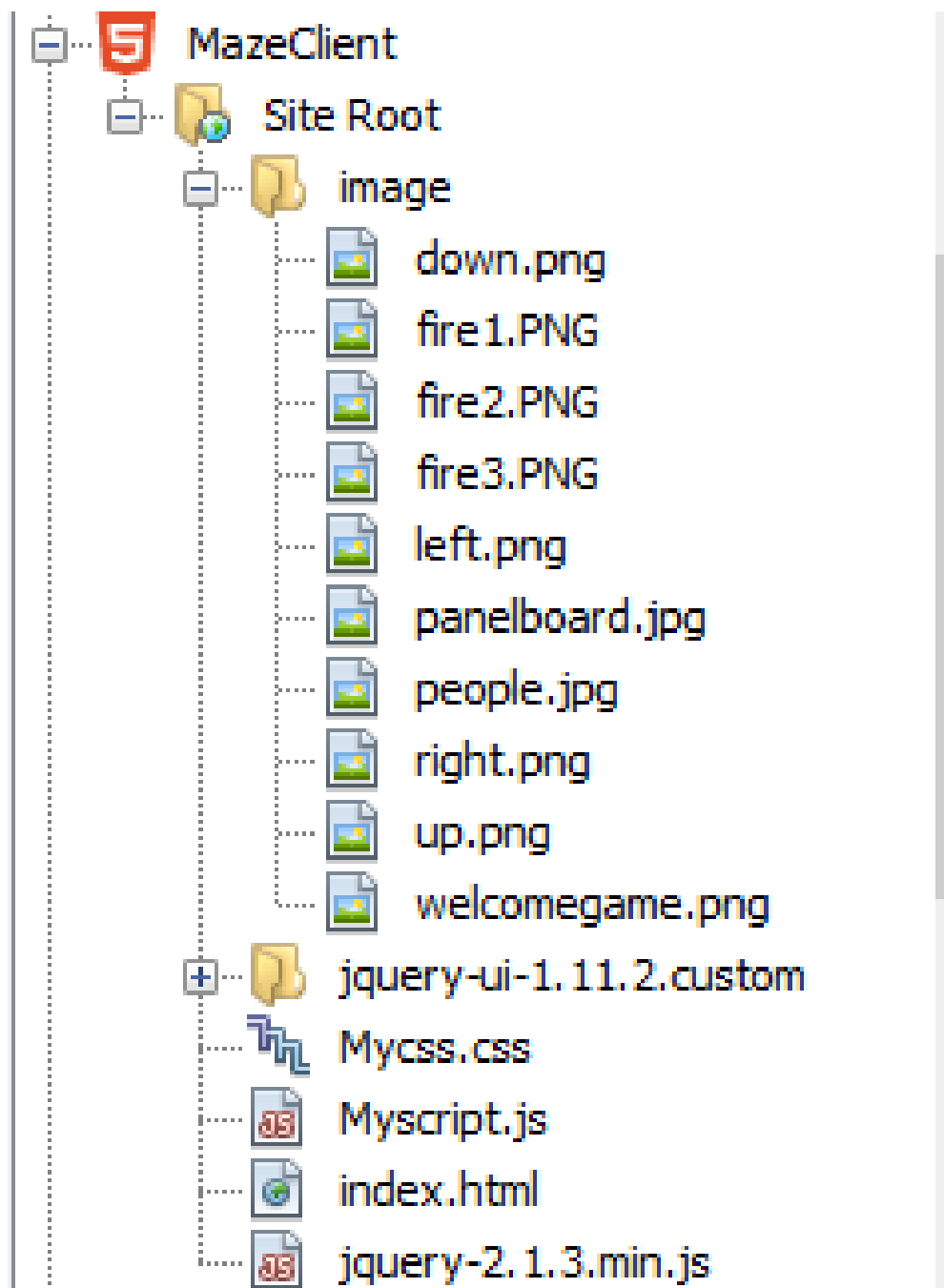
Part 1 Introduction.....	3
Part 2 Overview.....	4
Part 3 Demonstration of working site .....	6
Part 4 Display and explanation of code.....	37

## **Part 1 Introduction**

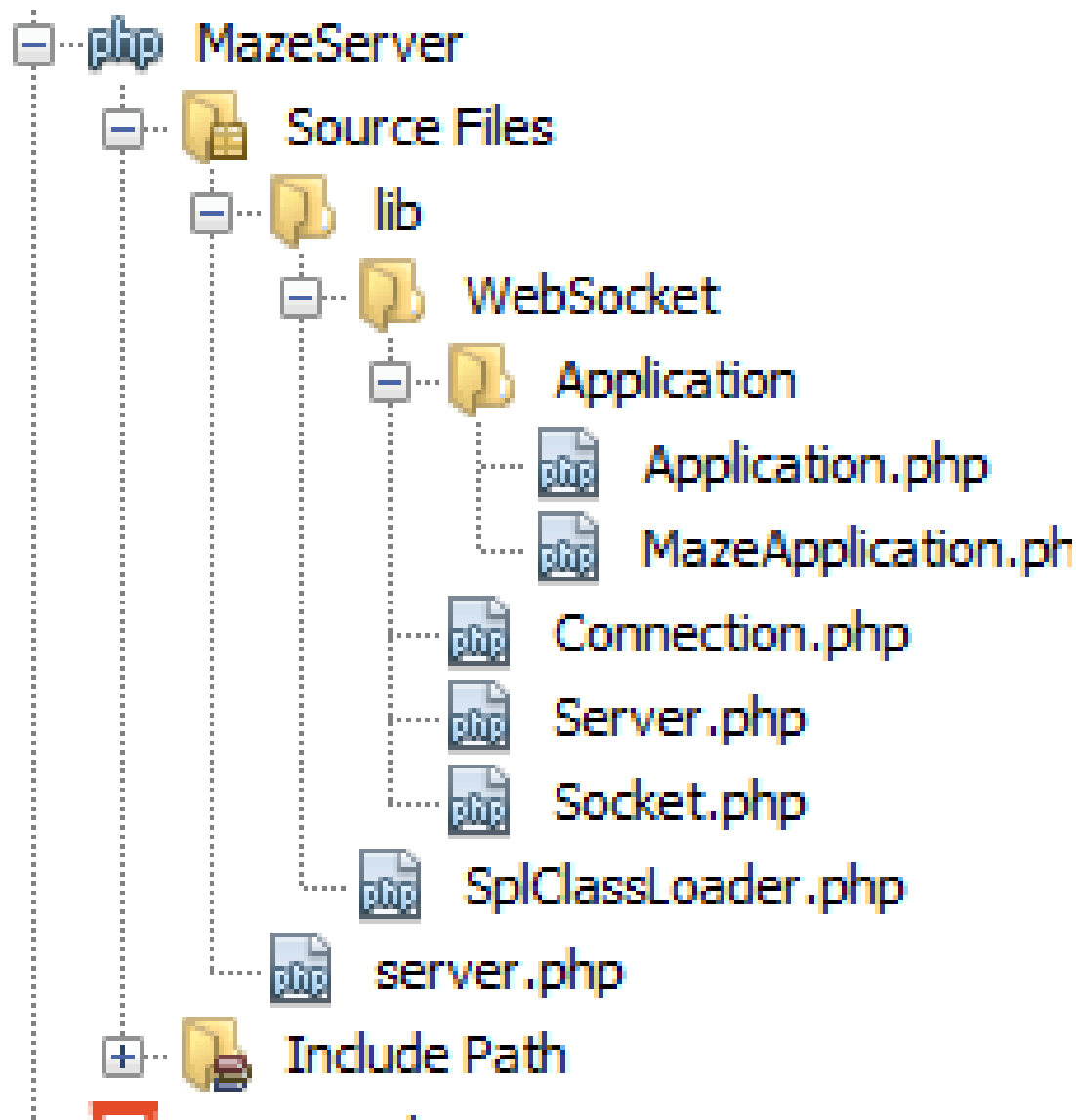
This is assignment give me some experience of using HTML5 new and magnificent features like canvas and web socket, and I'm using these new technology to develop a multi-player online game, it's just so amazing of this assignment, it afford me great interest in learning. I hope more, lol, really. I just can't help be the first one to finish this.

## **Part 2 Overview**

This is my netbeans project client side files screen shot



And this is server side netbeans project



### Part 3 Demonstration of working site

First go to htdoc directory in my apache folder use command line, and use “php -f” to run the server side php code to create the server.

```
Victor@VICTOR-PC c:\xampp\htdocs\MazeServer
# php -f server.php
2015-04-03 10:22:47 [info] Server created
```

And then run the client side in the web browser.

First client connected

```
Victor@VICTOR-PC c:\xampp\htdocs\MazeServer
# php -f server.php
2015-04-03 10:22:47 [info] Server created
2015-04-03 10:27:12 [info] [client 127.0.0.1:45031] Connected
2015-04-03 10:27:15 [info] [client 127.0.0.1:45031] Performing handshake
2015-04-03 10:27:15 [info] [client 127.0.0.1:45031] Handshake sent
```

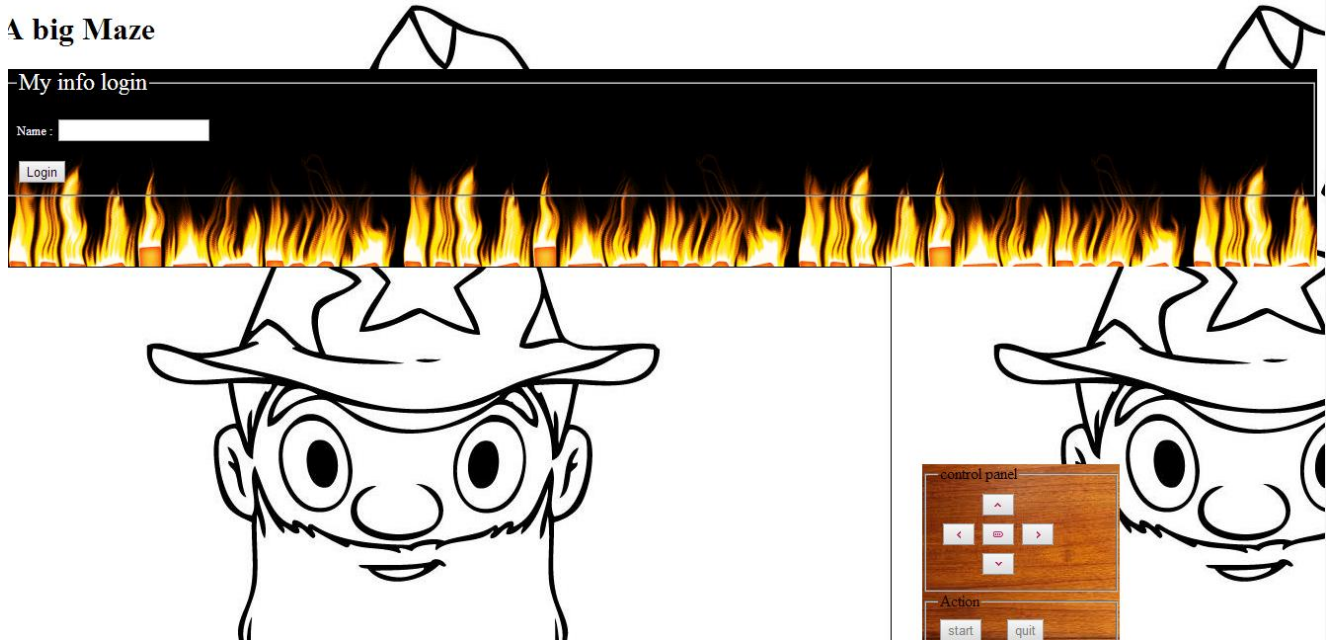
Second and third and fourth client connected, actually there could be 100 clients which set in “ \$server->setMaxClients(100);” php code, but I just use four as example, because I will give their name Alice, Bob , Carol , and Oscar

Now four of them connected to server

```
Victor@VICTOR-PC c:\xampp\htdocs\MazeServer
# php -f server.php
2015-04-03 10:22:47 [info] Server created
2015-04-03 10:27:12 [info] [client 127.0.0.1:45031] Connected
2015-04-03 10:27:15 [info] [client 127.0.0.1:45031] Performing handshake
2015-04-03 10:27:15 [info] [client 127.0.0.1:45031] Handshake sent
2015-04-03 10:31:18 [info] [client 127.0.0.1:45124] Connected
2015-04-03 10:31:18 [info] [client 127.0.0.1:45124] Performing handshake
2015-04-03 10:31:18 [info] [client 127.0.0.1:45124] Handshake sent
2015-04-03 10:31:21 [info] [client 127.0.0.1:45129] Connected
2015-04-03 10:31:21 [info] [client 127.0.0.1:45129] Performing handshake
2015-04-03 10:31:21 [info] [client 127.0.0.1:45129] Handshake sent
2015-04-03 10:31:23 [info] [client 127.0.0.1:45134] Connected
2015-04-03 10:31:23 [info] [client 127.0.0.1:45134] Performing handshake
2015-04-03 10:31:23 [info] [client 127.0.0.1:45134] Handshake sent
```

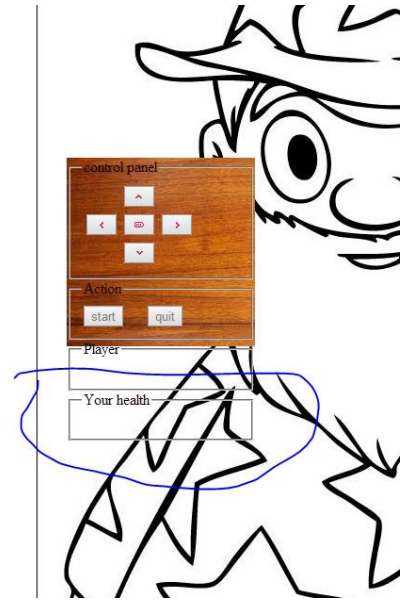
This is how client side pages look like

### A big Maze



What is difference from Mr Neil is I have a place which can show how many health remain, I have already circled in blue lines in screen shot, it initially 10, I will show u later



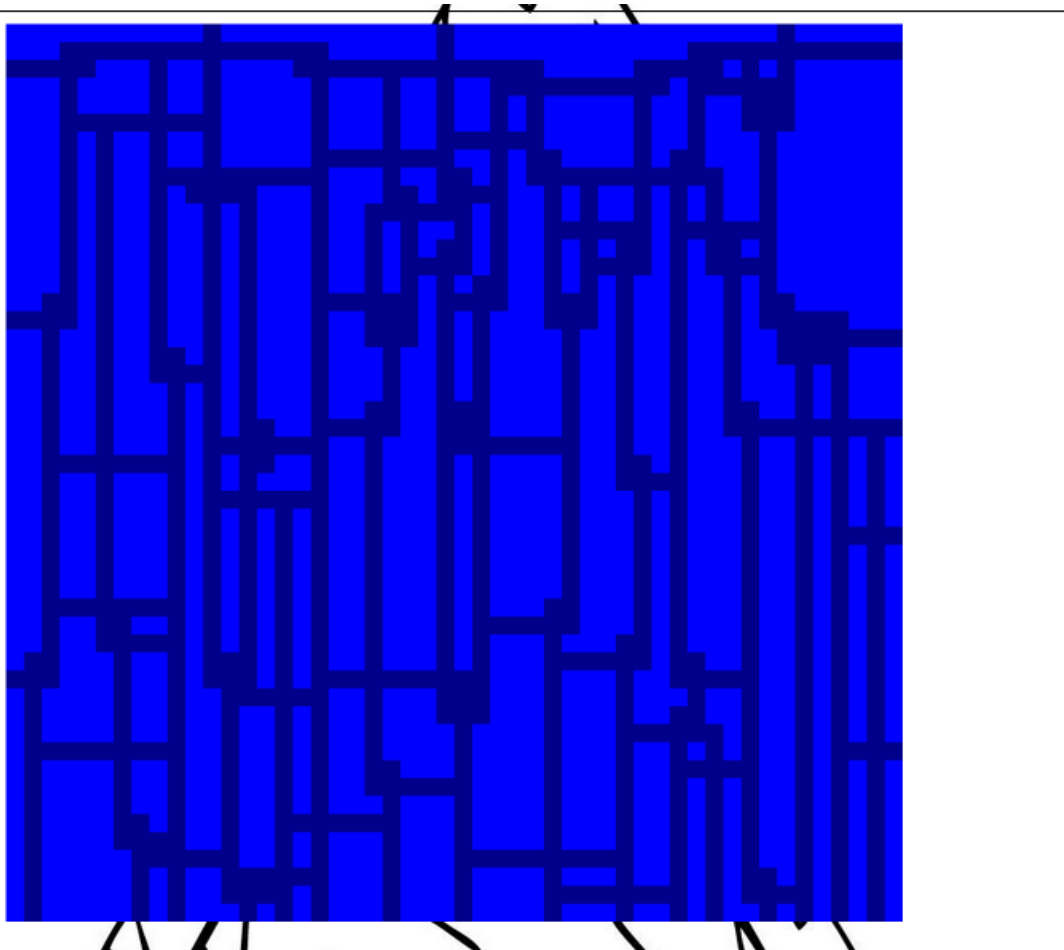


Ok let's login for each user

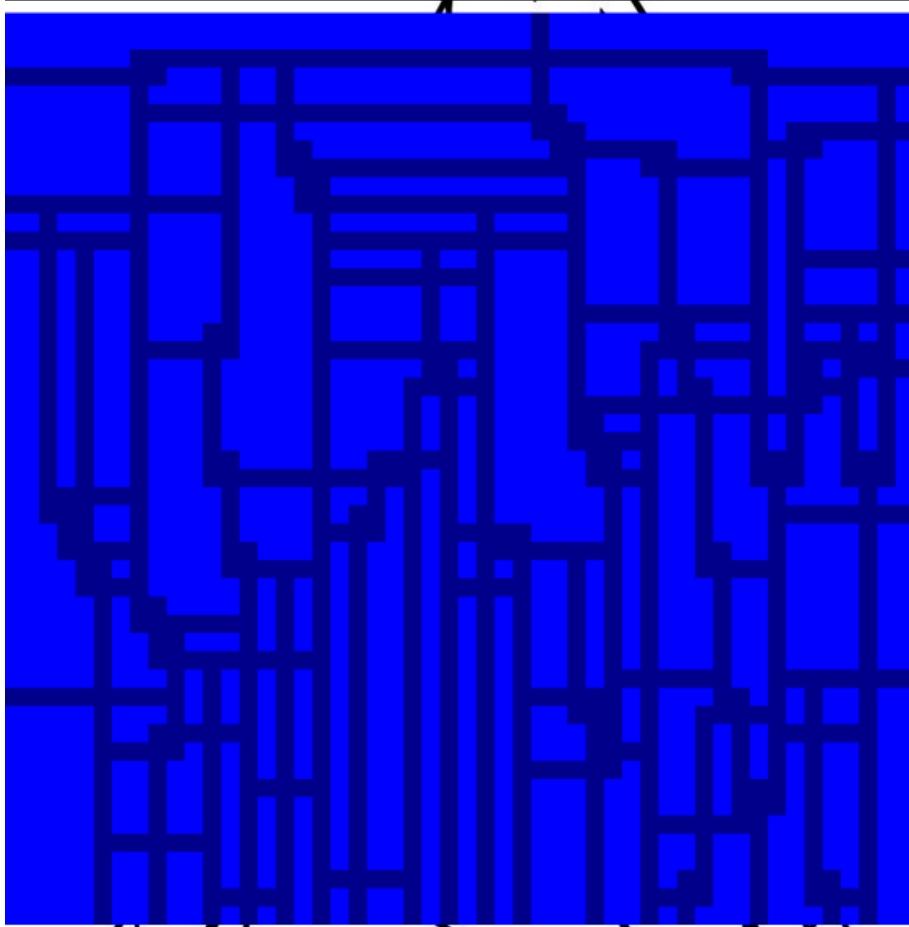
First Alice login, and maze generate, I specifically write an algorithm to generate the maze which random generator maze everytime, u will see in my code

This just to show my maze is random from everytime, still havn't start the game and each client maze should be same, and I will show u later on.

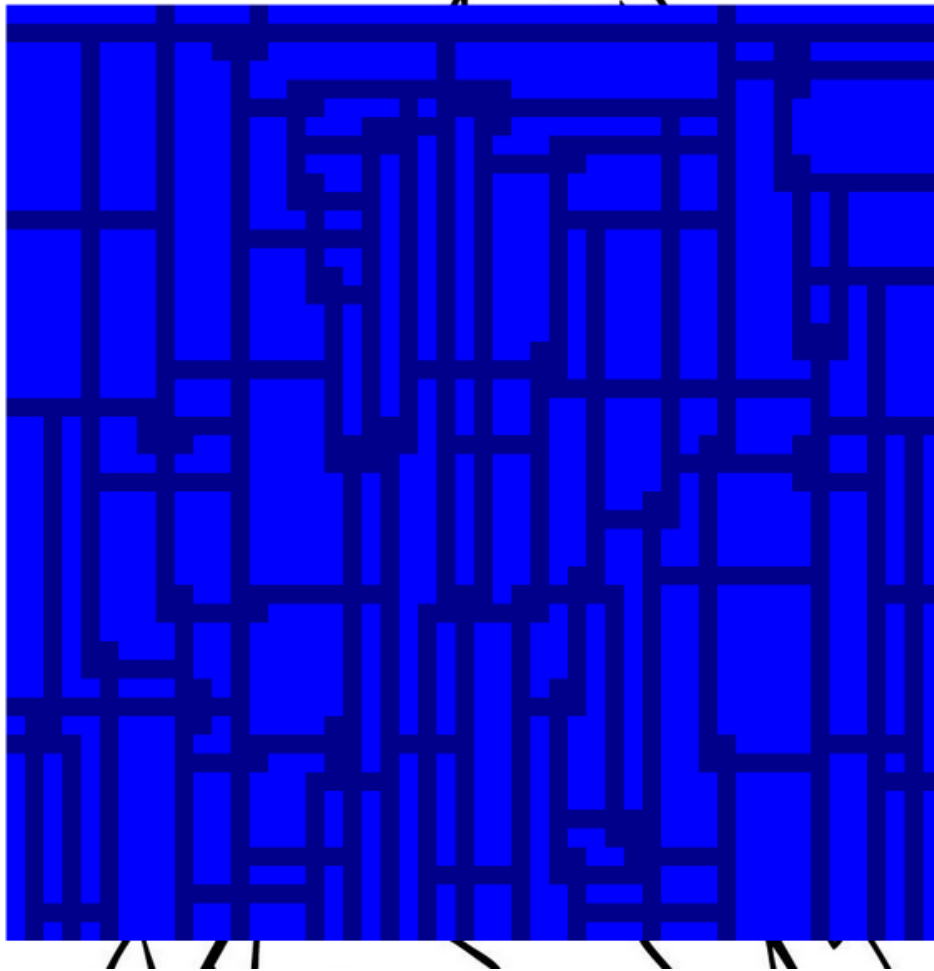
Random maze 1:



Random maze two



Random maze 3, alright I think that's enough good to show this example



Now let's login four of them Alice,Bob,Carol,Oscar

In my project I almost echo everything from server, u will see more later in my show

```

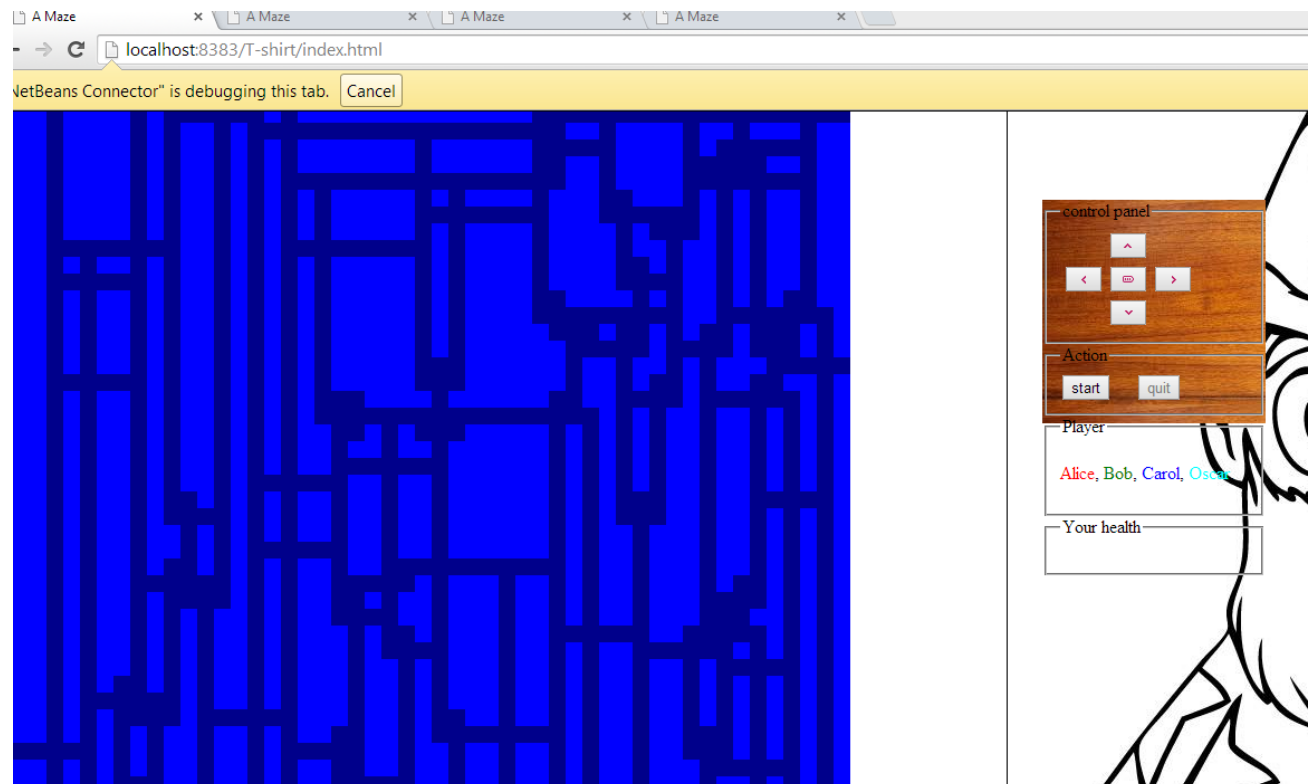
Victor@VICTOR-PC c:\xampp\htdocs\MazeServer
# php -f server.php
2015-04-03 10:53:33 [info] Server created
2015-04-03 10:54:00 [info] [client 127.0.0.1:45379] Connected
2015-04-03 10:54:00 [info] [client 127.0.0.1:45379] Performing handshake
2015-04-03 10:54:00 [info] [client 127.0.0.1:45379] Handshake sent
2015-04-03 10:54:03 [info] [client 127.0.0.1:45388] Connected
2015-04-03 10:54:03 [info] [client 127.0.0.1:45388] Performing handshake
2015-04-03 10:54:03 [info] [client 127.0.0.1:45388] Handshake sent
2015-04-03 10:54:05 [info] [client 127.0.0.1:45397] Connected
2015-04-03 10:54:05 [info] [client 127.0.0.1:45397] Performing handshake
2015-04-03 10:54:05 [info] [client 127.0.0.1:45397] Handshake sent
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406] Connected
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406] Performing handshake
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406] Handshake sent
Alice is login now
Bob is login now
Carol is login now
Oscar is login now

```

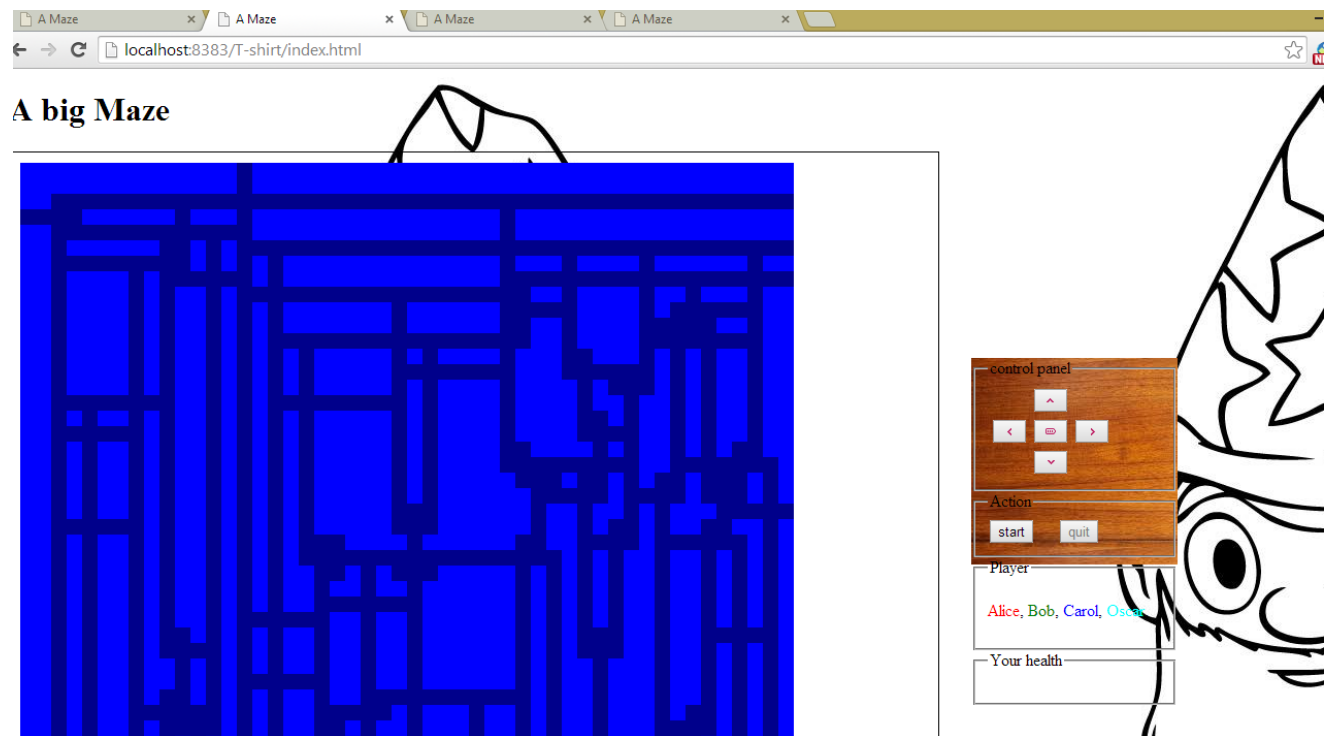
And how page look like for every of them

Notice the browser tab for each client and maze are totally same

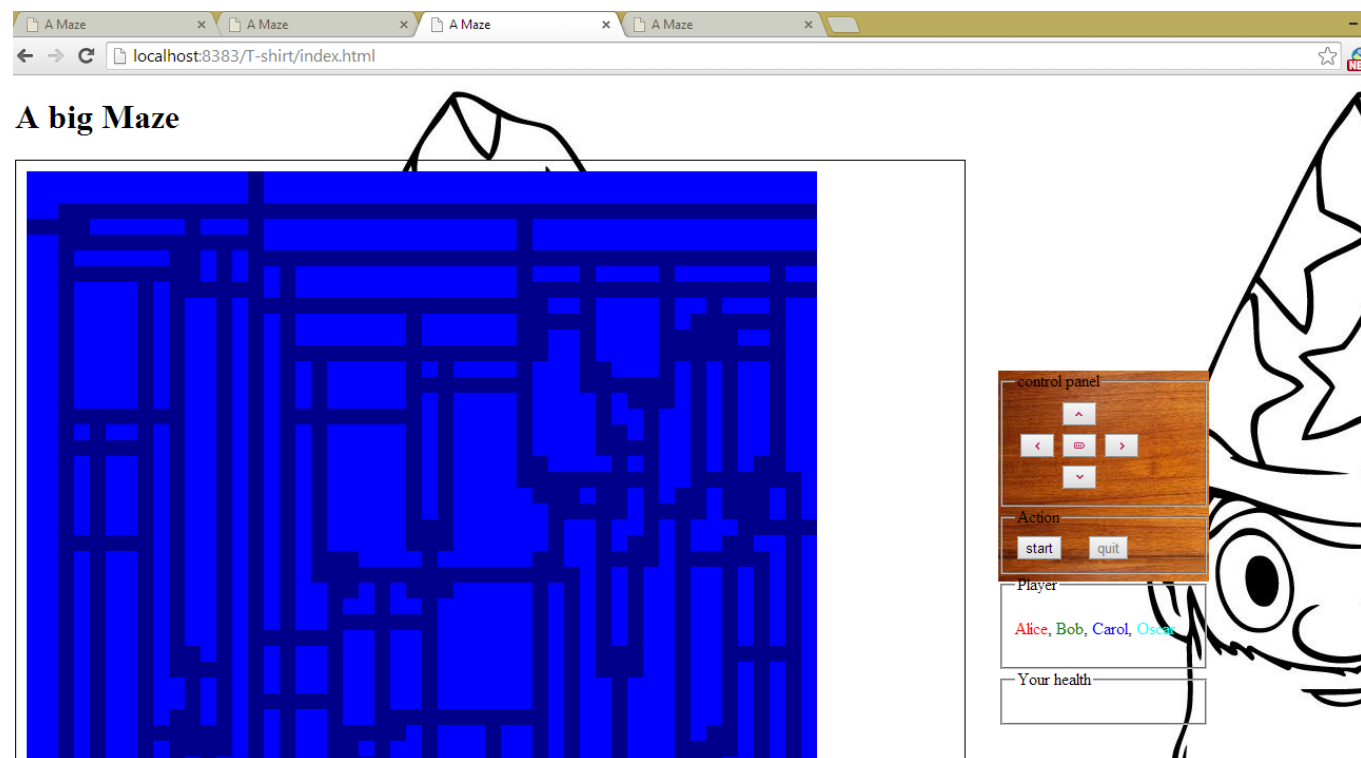
Alice client color red:



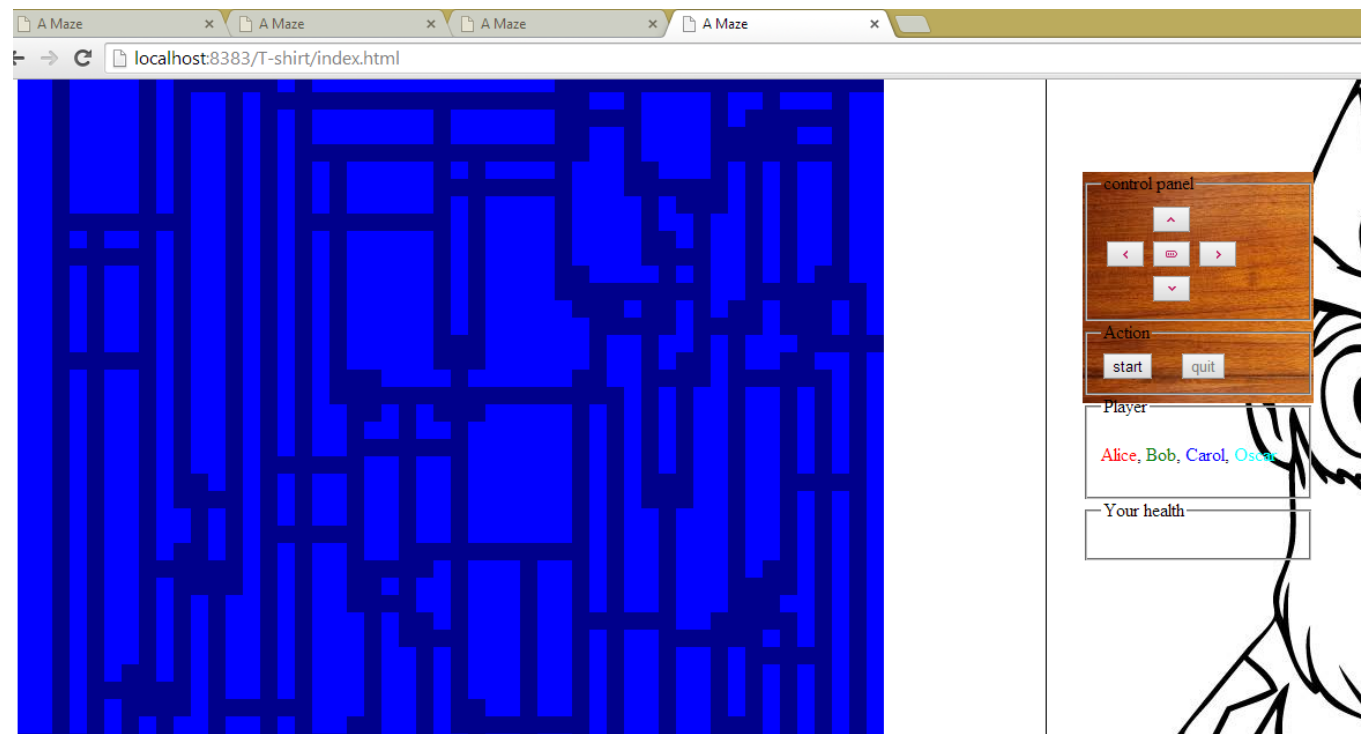
Bob client color green:



Carol client color deep blue:



Oscar client color light blue:

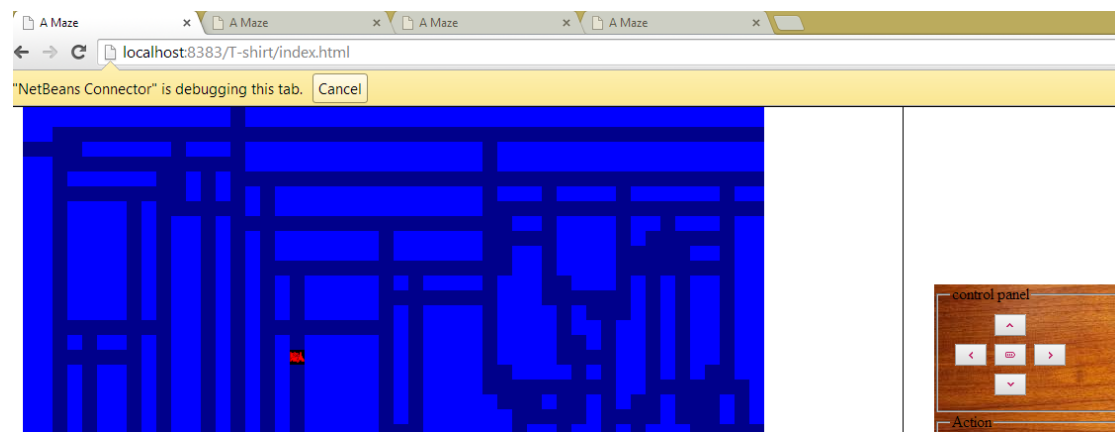


And not lets press start to start game:

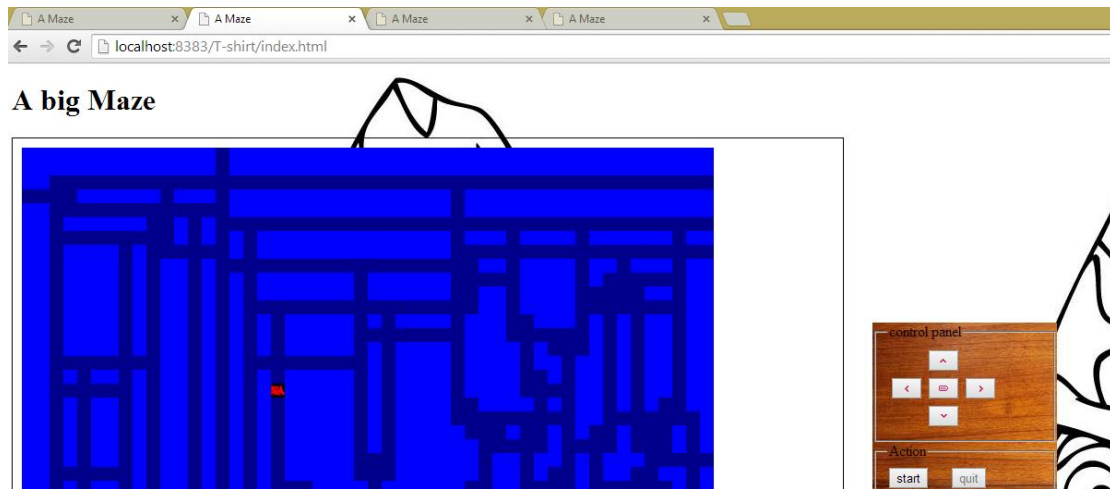
Let start Alice first, notice the browser tab ,each client can see alice

now

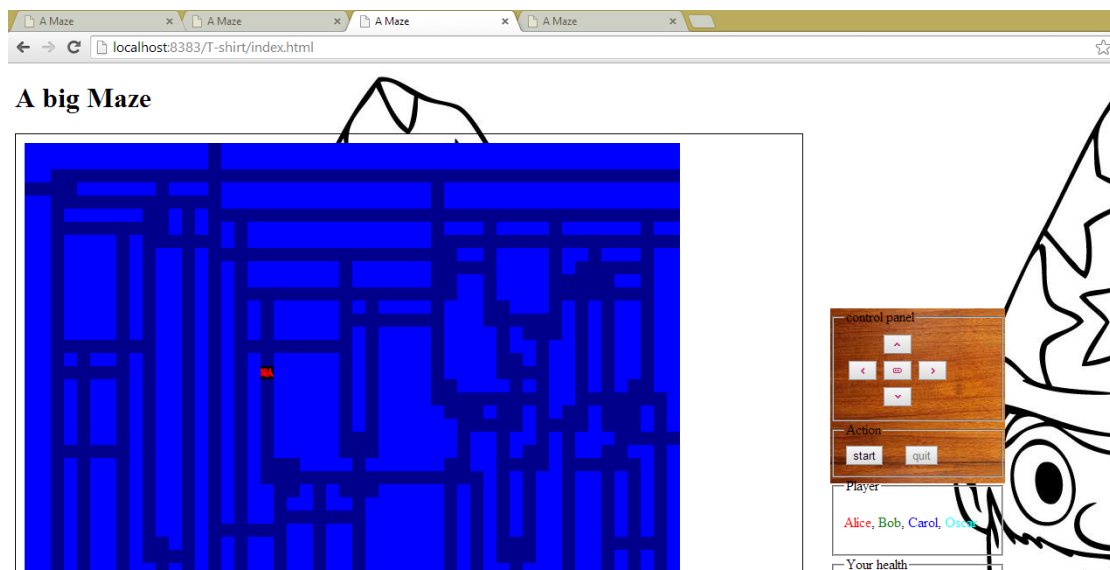
Client 1



## Client 2

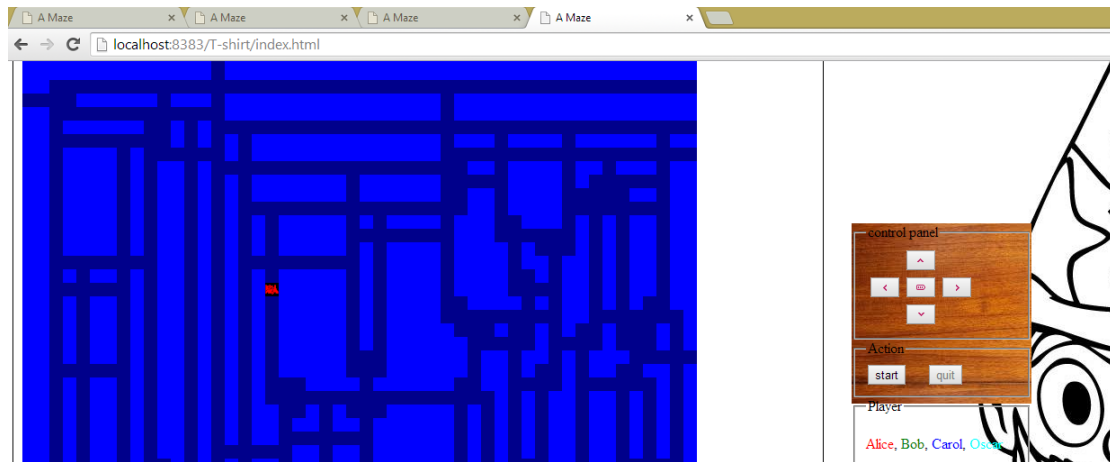


## Client 3

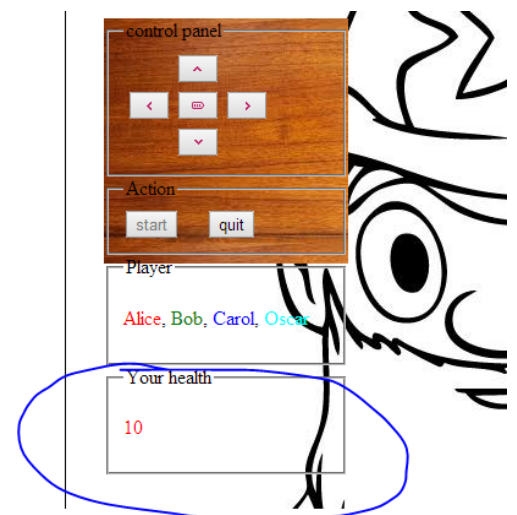
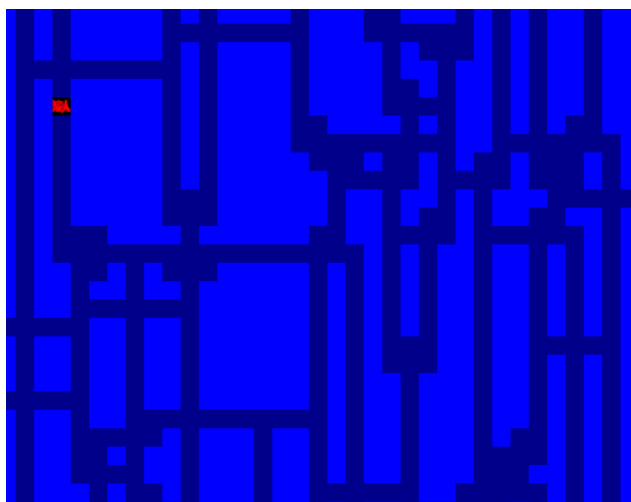


## Client 4





And in alice tab u can see the health shows ten now, I circle in blue line:



See the echo:

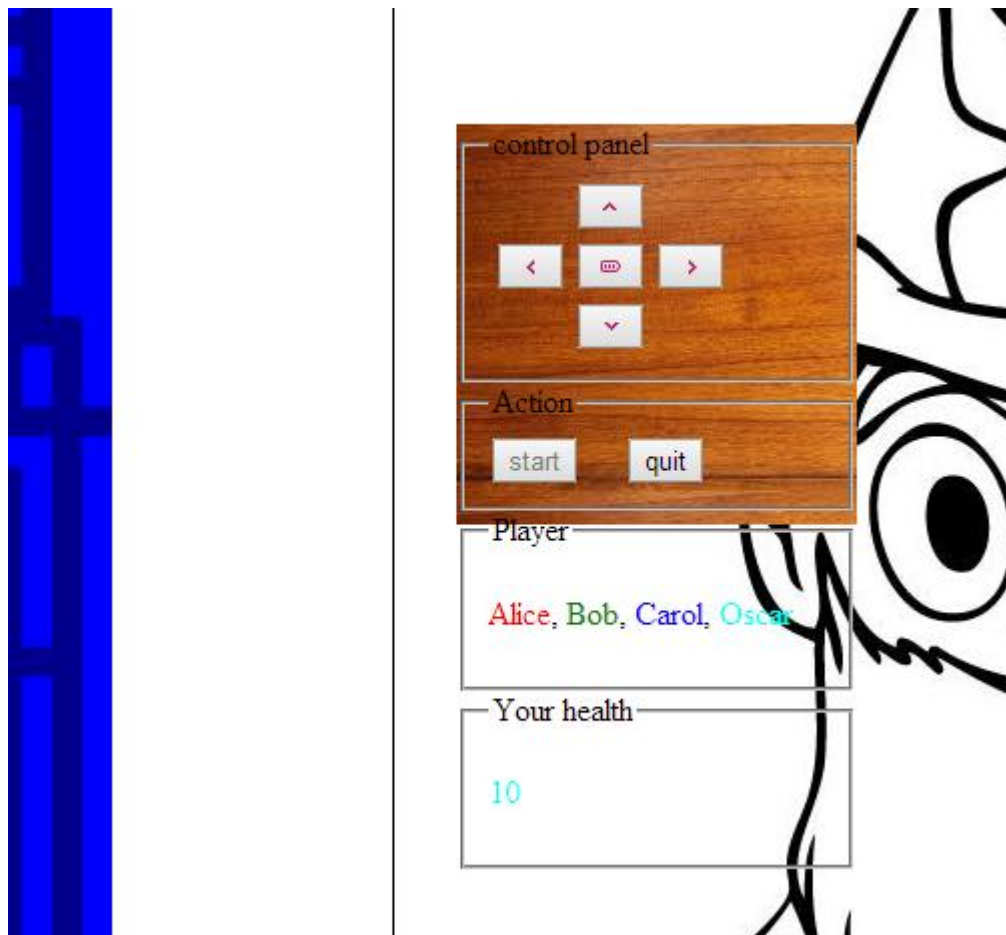
```

2015-04-03 10:54:00 [info] [client 127.0.0.1:45379] Ha
2015-04-03 10:54:03 [info] [client 127.0.0.1:45388] Co
2015-04-03 10:54:03 [info] [client 127.0.0.1:45388] Pe
2015-04-03 10:54:03 [info] [client 127.0.0.1:45388] Ha
2015-04-03 10:54:05 [info] [client 127.0.0.1:45397] Co
2015-04-03 10:54:05 [info] [client 127.0.0.1:45397] Pe
2015-04-03 10:54:05 [info] [client 127.0.0.1:45397] Ha
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406] Co
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406] Pe
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406] Ha
Alice is login now
Bob is login now
Carol is login now
Oscar is login now
Alice is now starting the game

```

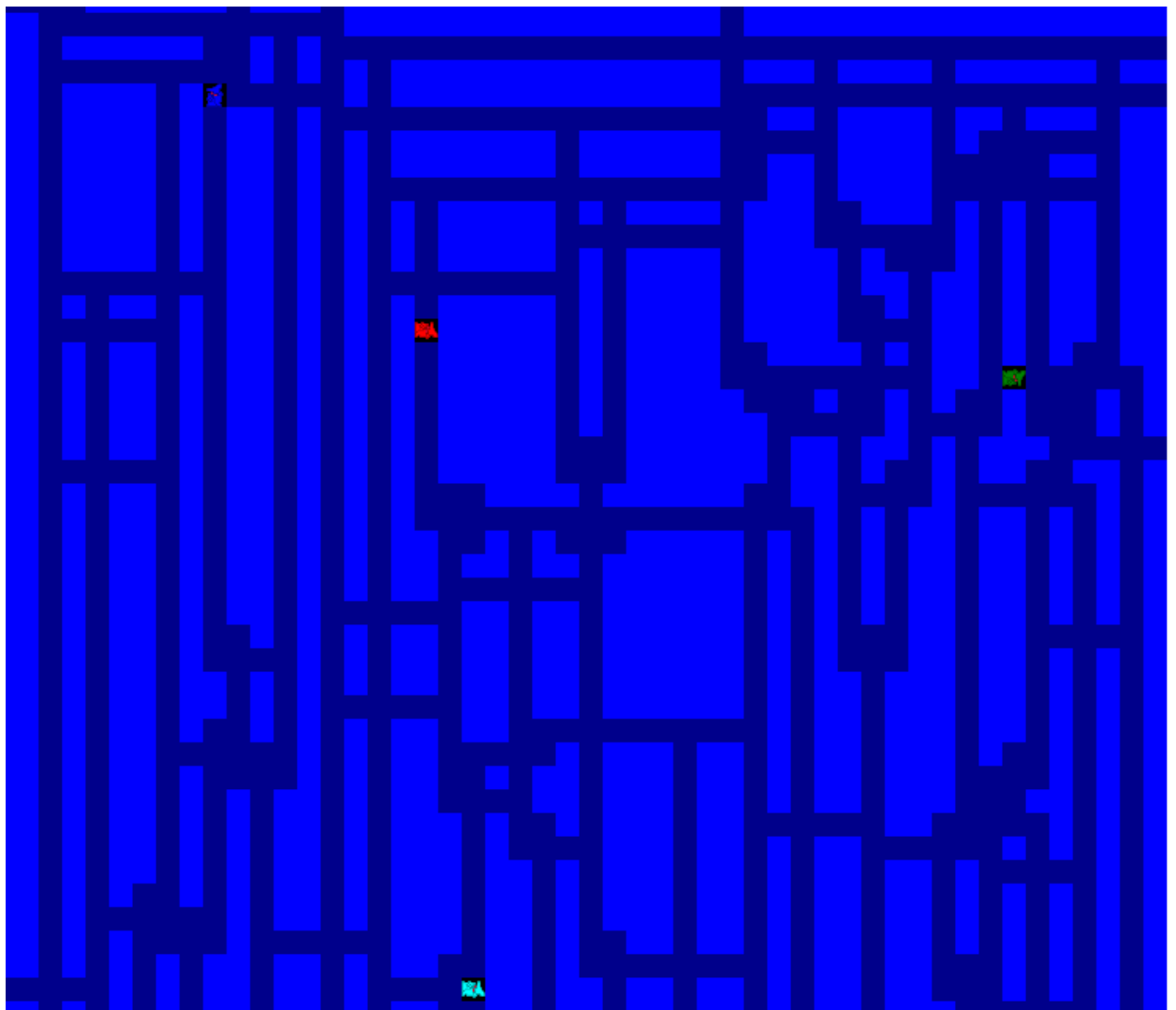
Lets start all of them, press start in control panel on the right:

And start button now disabled:



```
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406]
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406]
2015-04-03 10:54:07 [info] [client 127.0.0.1:45406]
Alice is login now
Bob is login now
Carol is login now
Oscar is login now
Alice is now starting the game
Bob is now starting the game
Carol is now starting the game
Oscar is now starting the game
```

And now they all in the same maze now

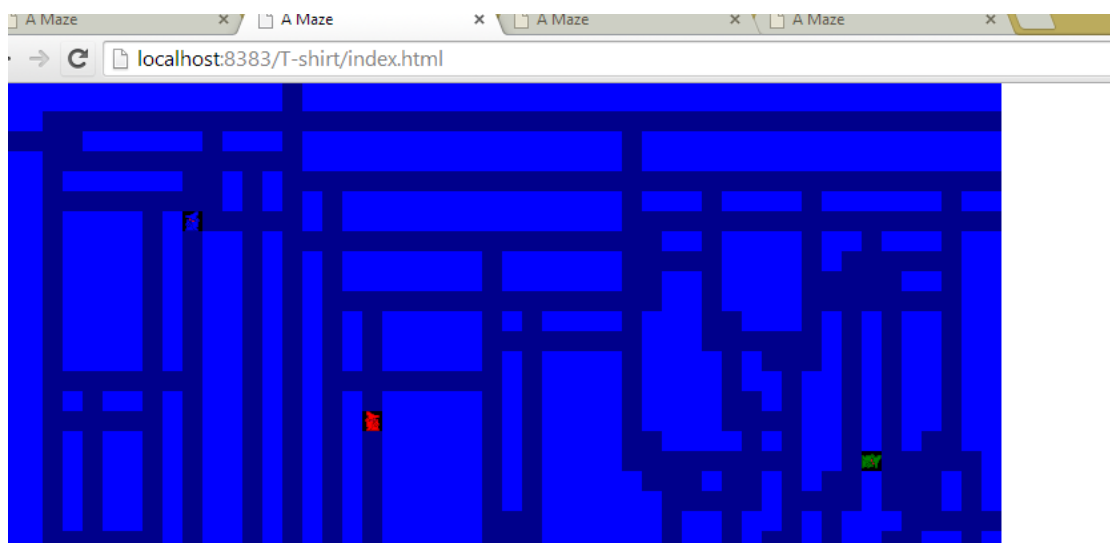
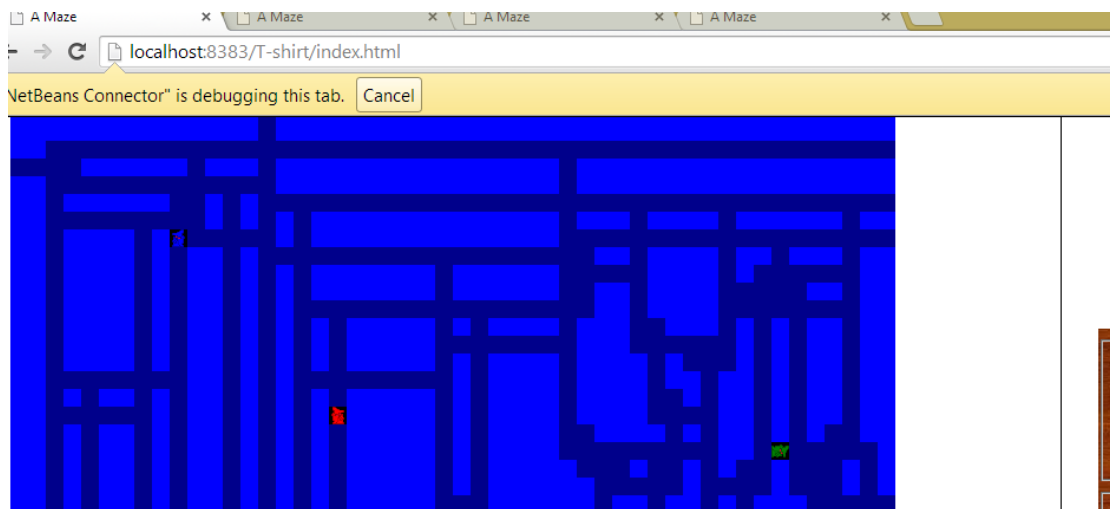


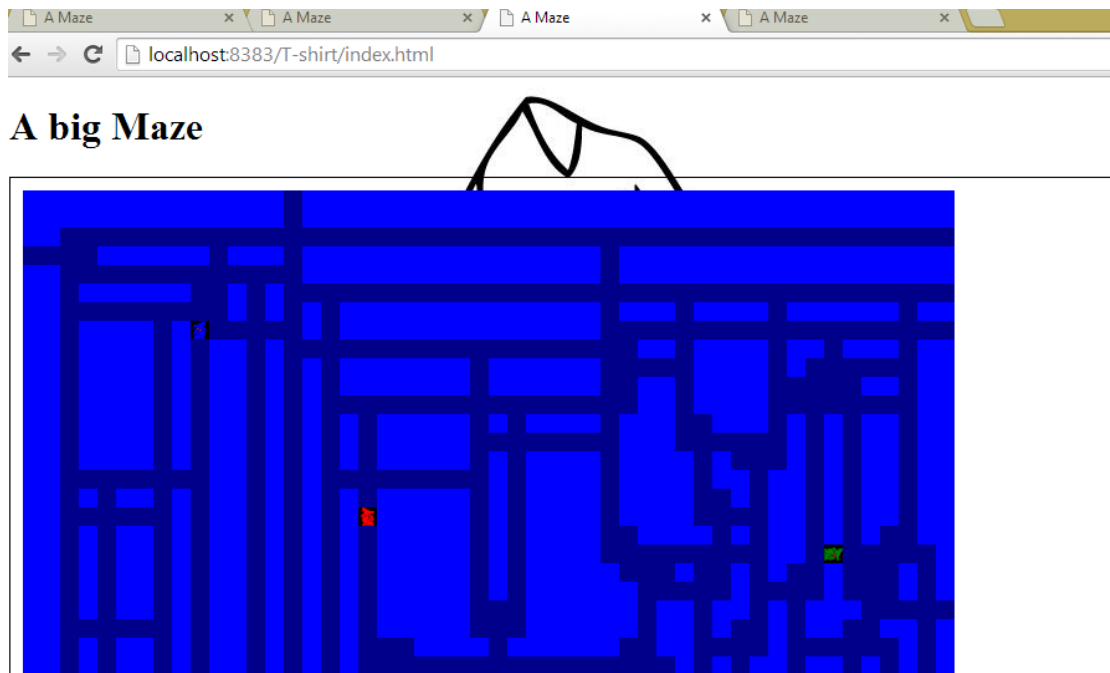
Let's start turning and moving the red client which is Alice

She's now facing up, et's make her turn

```
Alice is login now  
Bob is login now  
Carol is login now  
Oscar is login now  
Alice is now starting the game  
Bob is now starting the game  
Carol is now starting the game  
Oscar is now starting the game  
Alice turn right
```

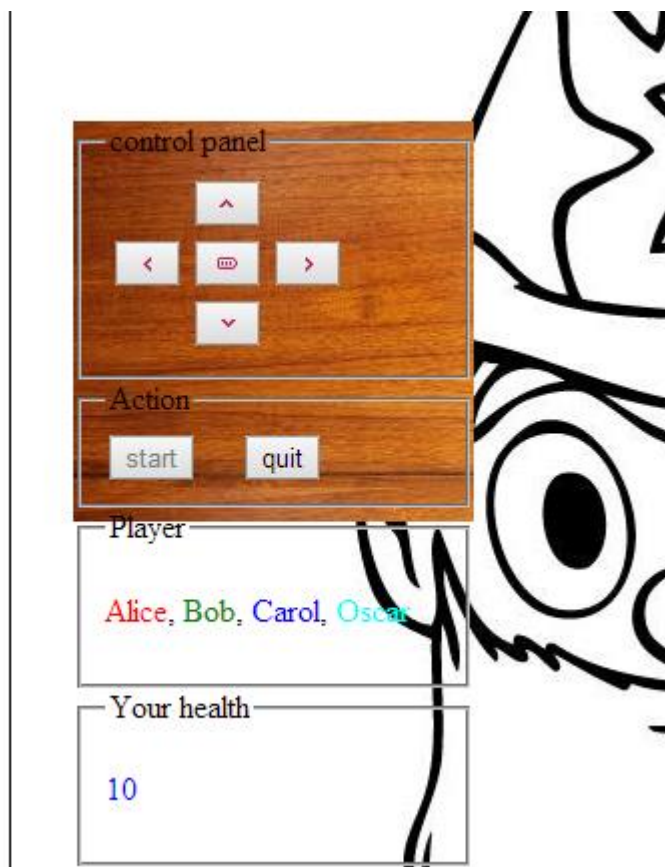
You can see other client also see her “turn right”





Alright and now let move her

Press the direction which now enable now after press start

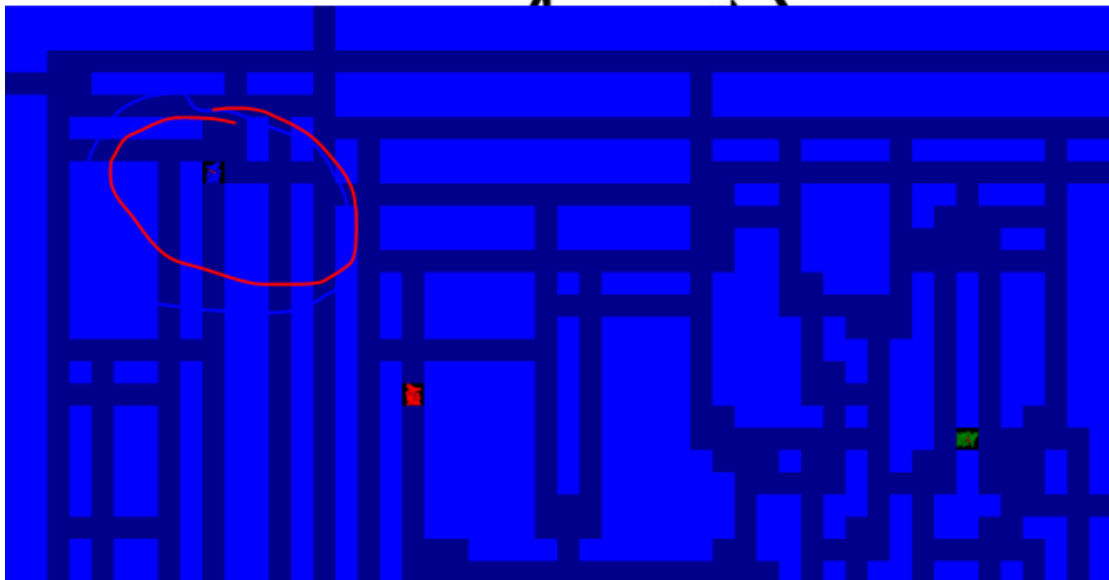


I will move Carol this time which is client three

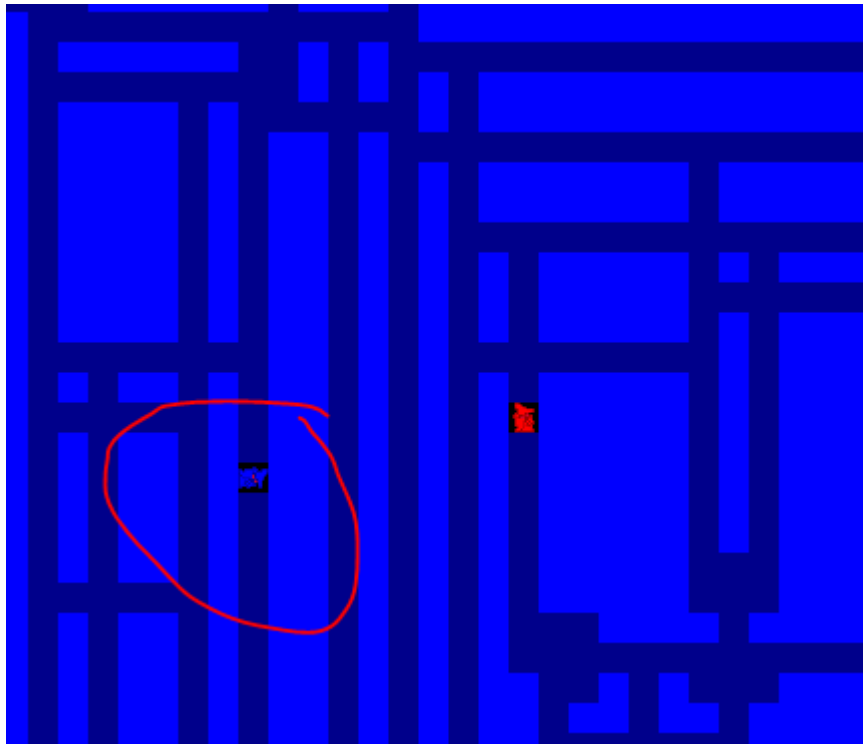
```
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down  
Carol move forward 1 step to down
```

Yep I move a lot of step

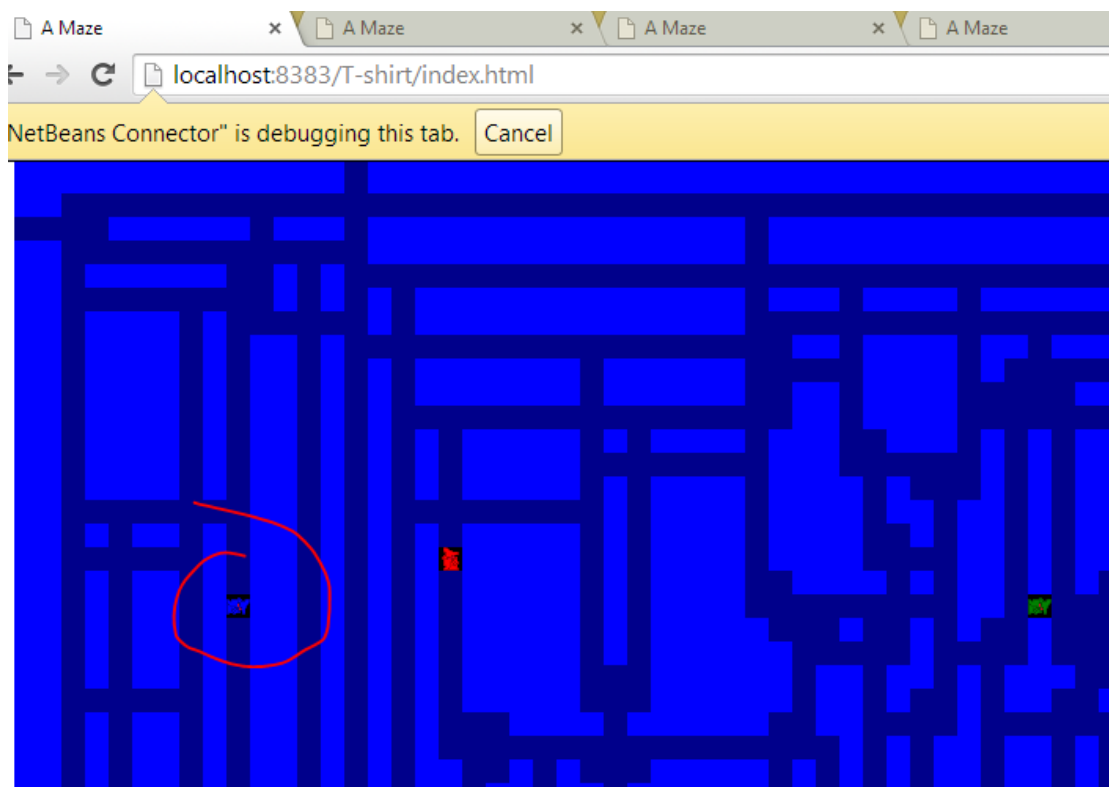
From here in red circle:

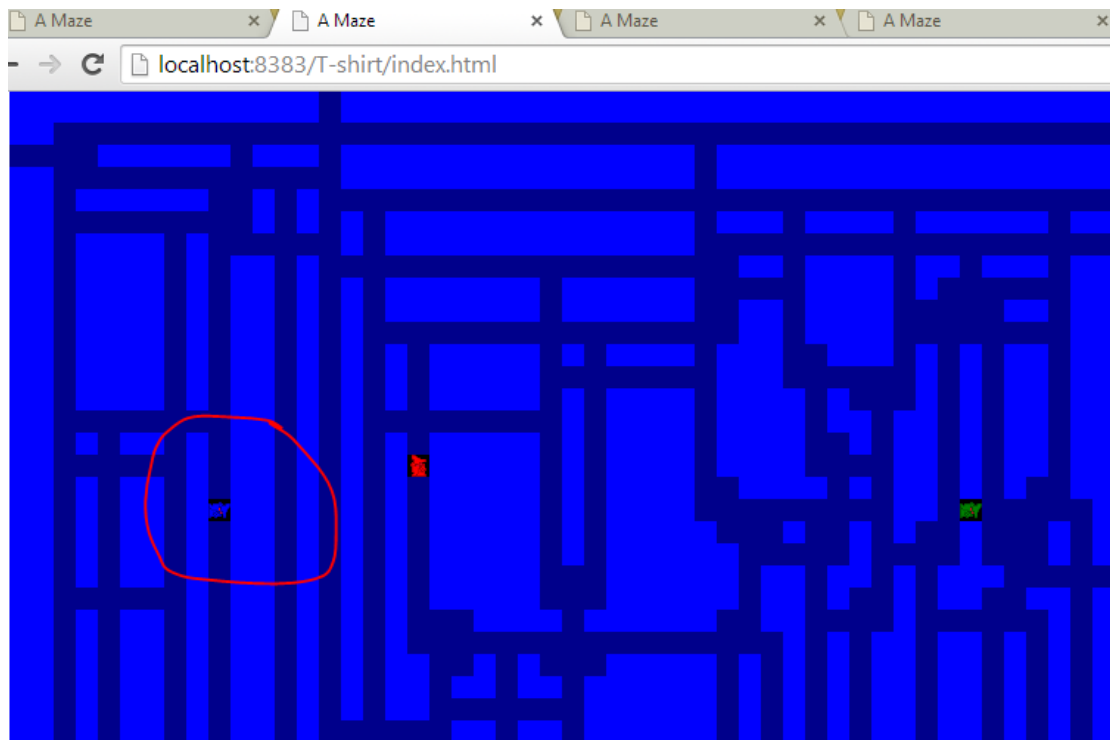


To here



And same each client see him move, still notice I change the browser tab which stands for each client:





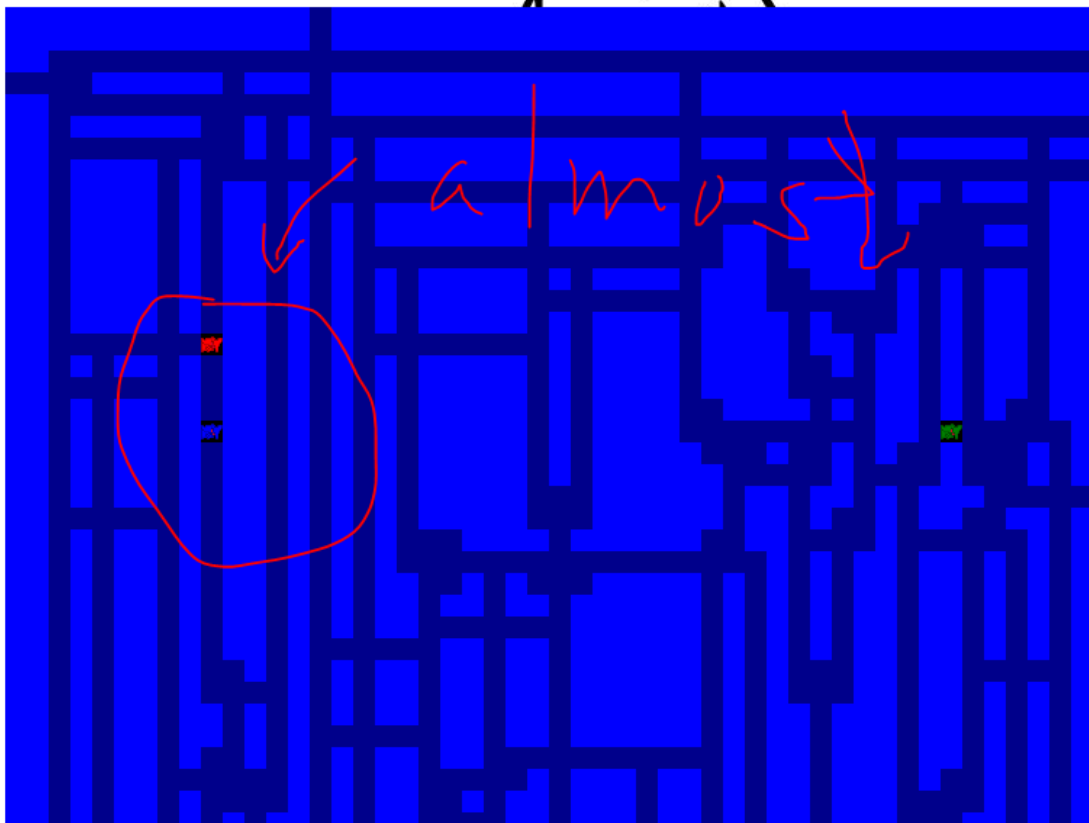
And now let's do some harm to others lol:

First show collision with other player

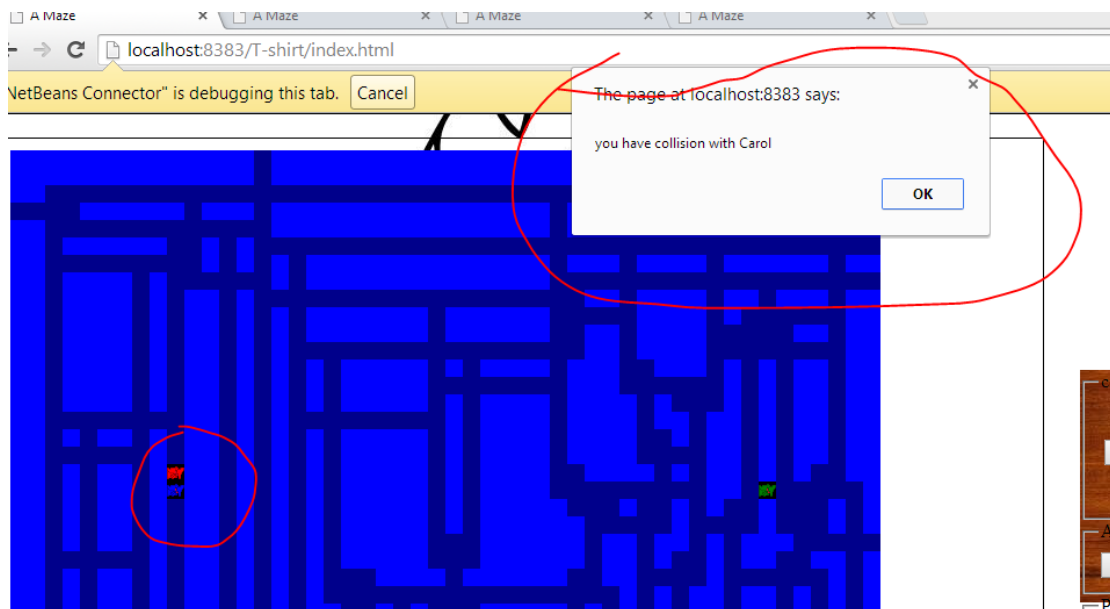
```
Alice move forward 1 step to left  
Alice turn up  
Alice move forward 1 step to up  
Alice turn left  
Alice move forward 1 step to left  
Alice move forward 1 step to left  
Alice move forward 1 step to left  
Alice move forward 1 step to left  
Alice move forward 1 step to left  
Alice turn down  
Alice move forward 1 step to down  
Alice move forward 1 step to down  
Alice move forward 1 step to down  
Alice move forward 1 step to down  
Alice move forward 1 step to down  
Alice move forward 1 step to down  
Alice move forward 1 step to down
```

Move a lot of step, close and still closing to the target:



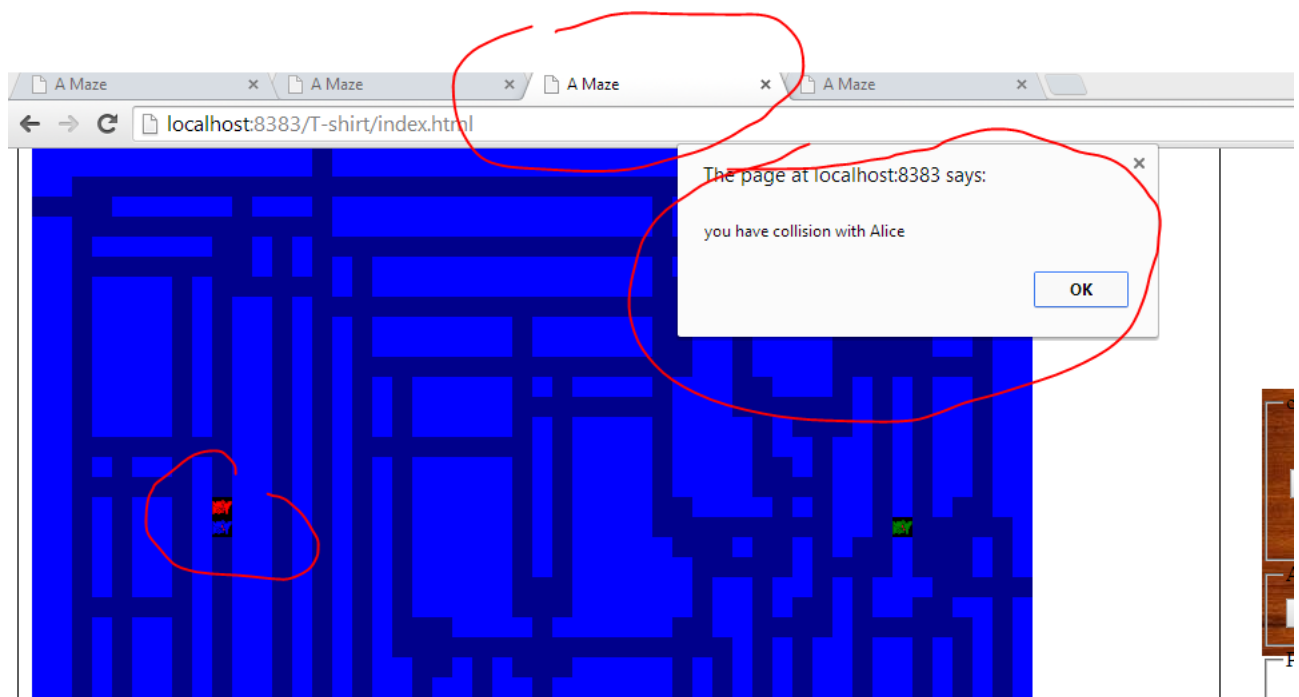


Bang!!!



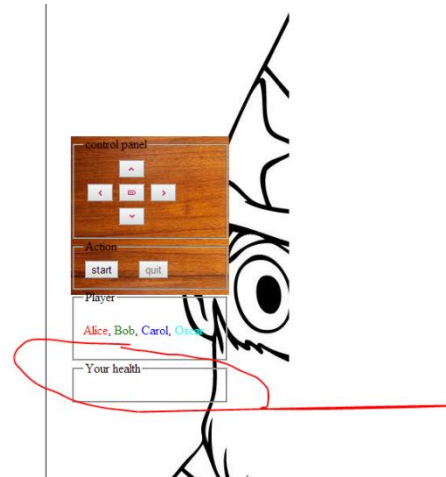
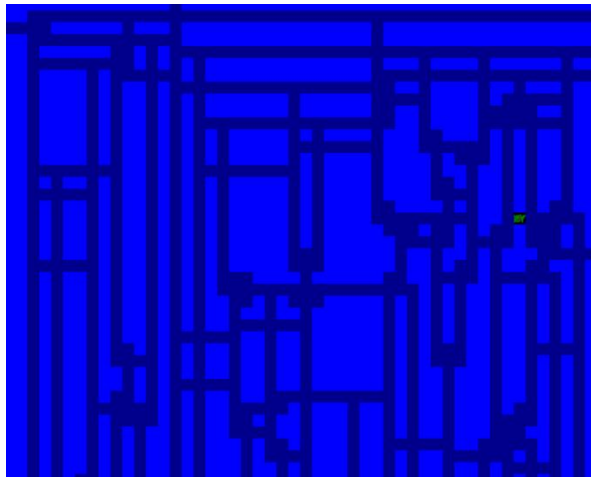
Now this client collision with Carol, let see how Carol going:

Bang!!

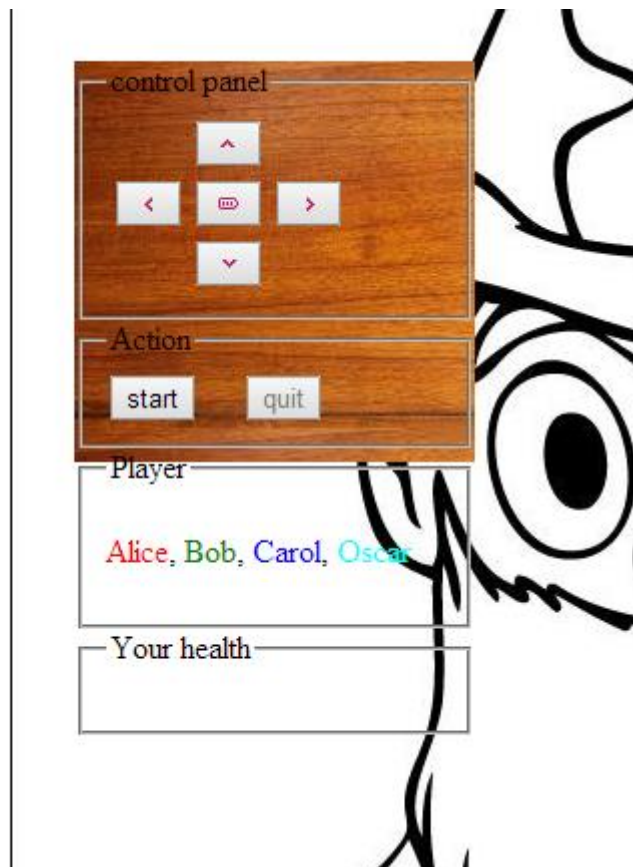


Both alice and carol out of game:

```
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice is out of the game
Carol is out of the game
```



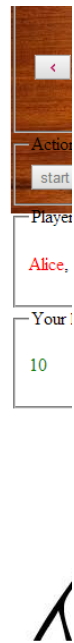
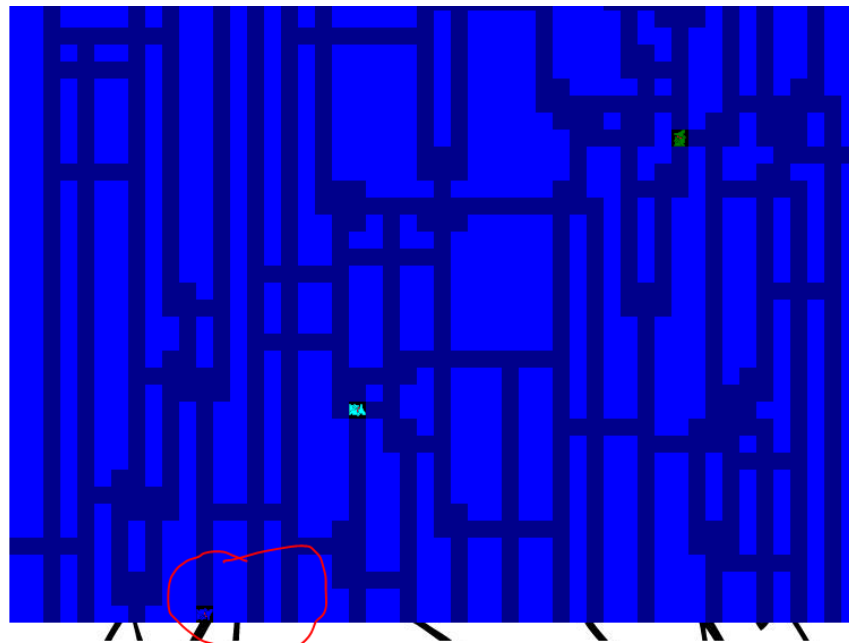
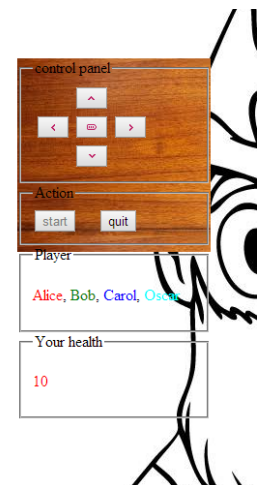
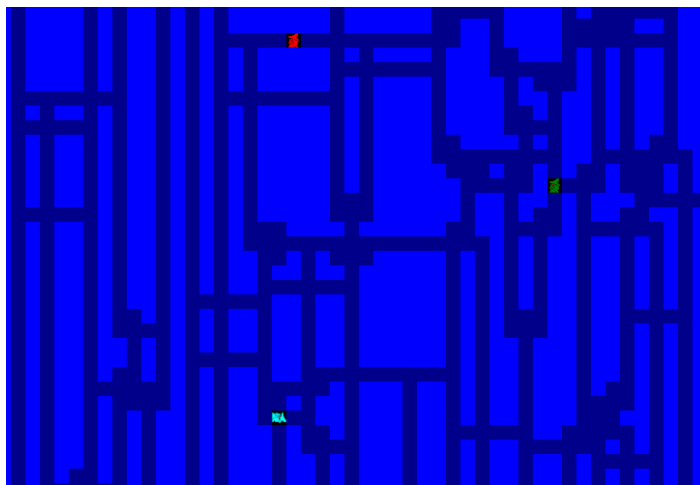
Well they all disappear in the maze and health is now empty,  
But u can see start button is able again which means u can start the  
game any time u want!!!



Let press start, to give them a chance:

```
Bob move forward 1 step to up  
Oscar move forward 1 step to up  
Oscar move forward 1 step to up  
Oscar move forward 1 step to up  
Carol is now starting the game  
Alice is now starting the game
```

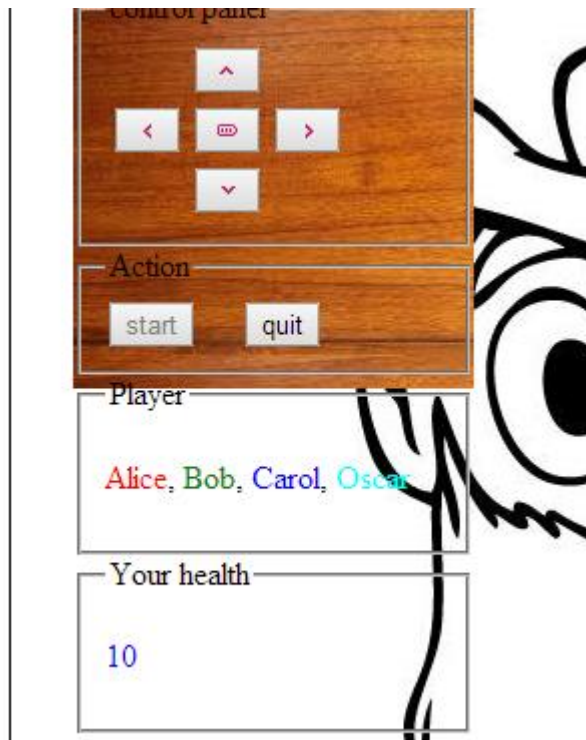
Health is back again:



Well this blue one is not so obvious in this blue maze, sorry about that, and the funny thing is random bring back to life at the edge of

maze so I'm thinking it's best to press quit and start again to revive him in some good position:

Press quit button:

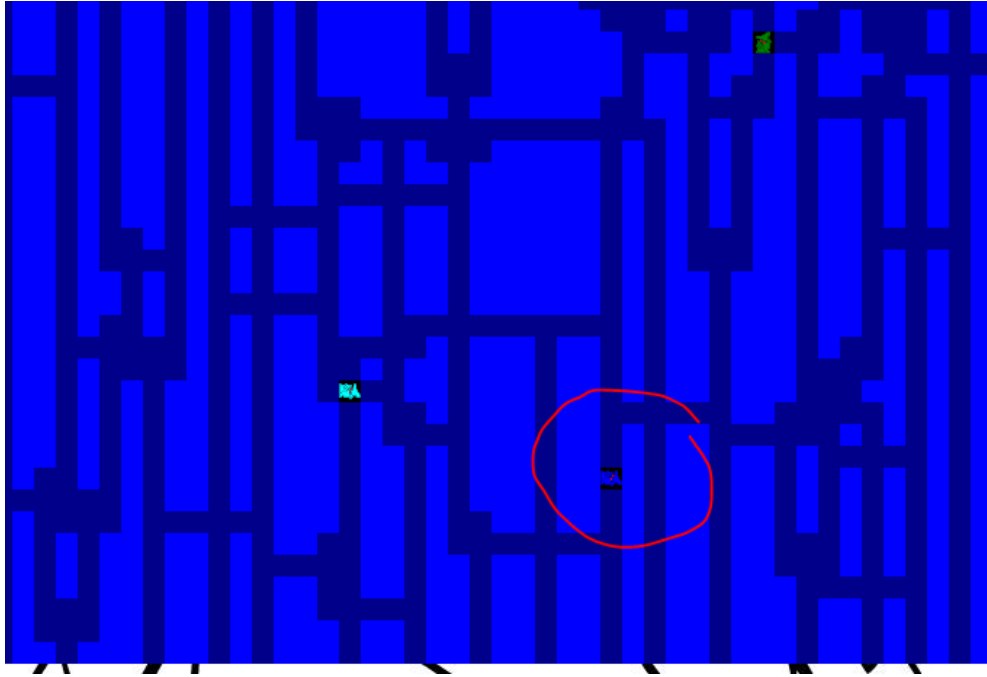


```
Oscar move forward 1 step to up
Oscar move forward 1 step to up
Oscar move forward 1 step to up
Carol is now starting the game
Alice is now starting the game
Carol is out of the game
```

And start again

```
Oscar move forward 1 step to up
Oscar move forward 1 step to up
Oscar move forward 1 step to up
Oscar move forward 1 step to up
Carol is now starting the game
Alice is now starting the game
Carol is out of the game
Carol is now starting the game
```

Now that's what I'm talking about~~



Next harm~~let's do some shoot~~~

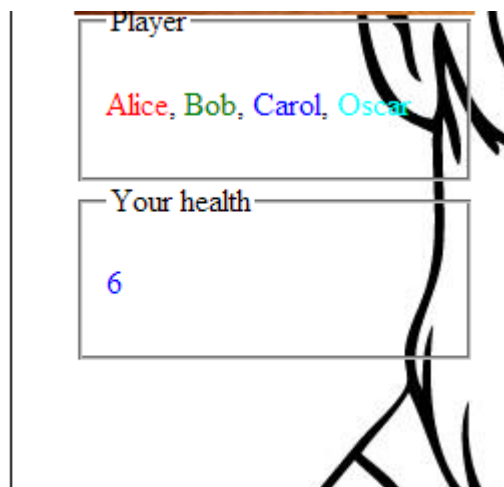
(1) three blocks away only 4 damages:



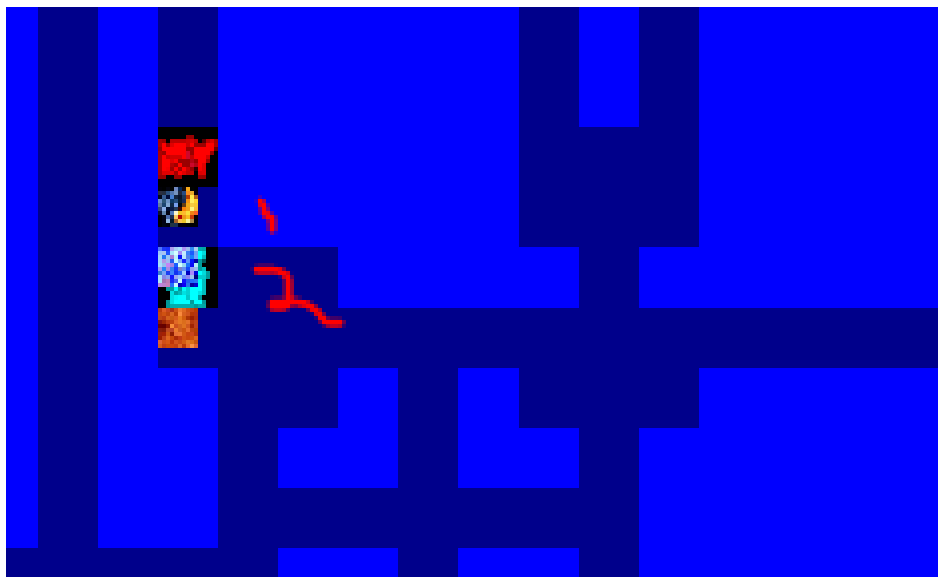
Green shoot blue:

```
Bob turn right
Bob move forward 1 step to right
Bob turn down
Bob move forward 1 step to down
Bob is firing down
Bob cause Carol 4 damages
Carol lost 4 health
```

Now blue should only have 6 now



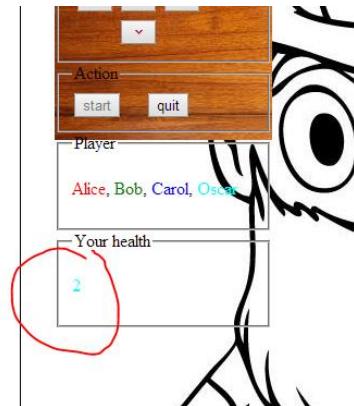
(2)two block away,cause 8 damage



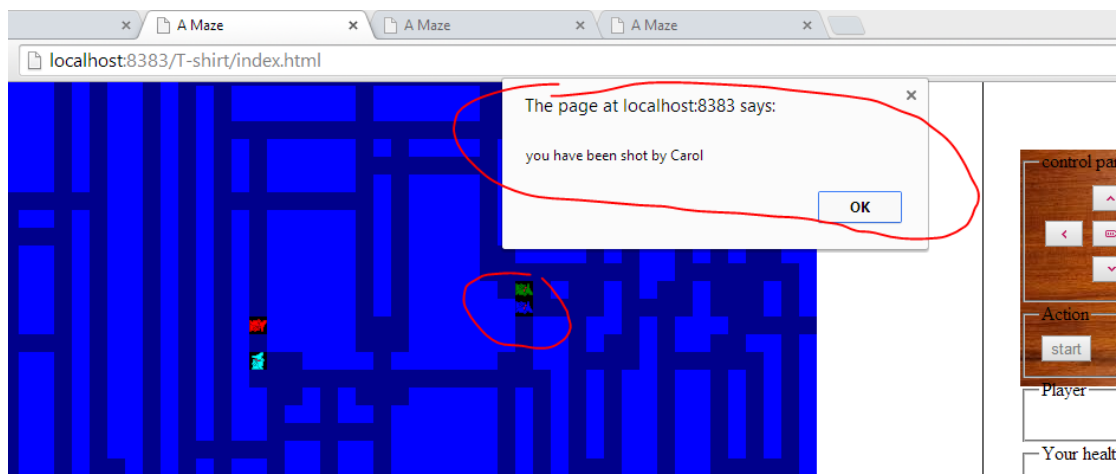
Red shot light blue,

```
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice move forward 1 step to down
Alice is firing down
Alice cause Oscar 8 damages
Oscar lost 8 health
```

And now the light blue which is oscar now only have 2 health, almost die,



(3)only one block away 16 damage:



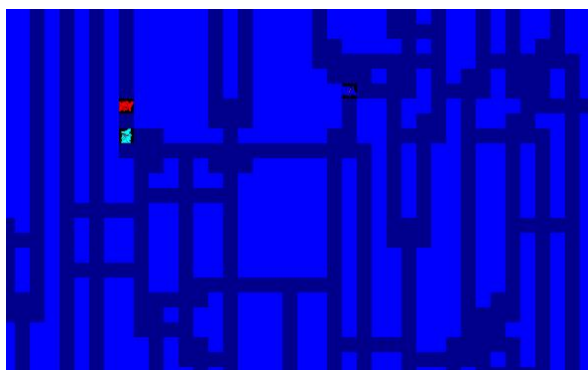


The damage is too big, well Bob is killed by carol

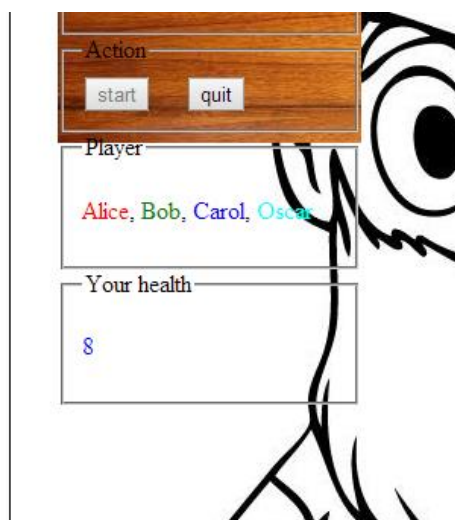
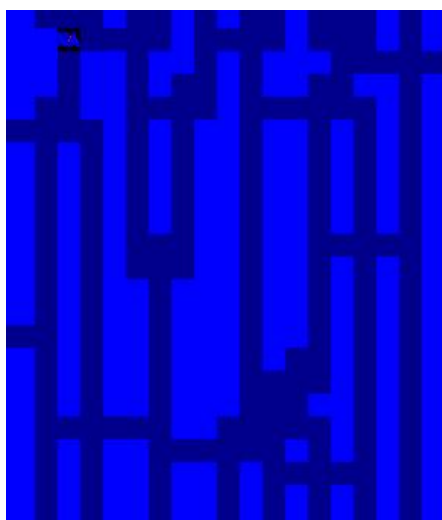
```
Alice is firing down  
Alice cause Oscar 8 damages  
Oscar lost 8 health  
Carol turn down  
Carol turn up  
Carol move forward 1 step to up  
Carol move forward 1 step to up  
Carol is firing up  
Carol killed Bob and get two health
```

```
Alice cause Oscar 8 damages  
Oscar lost 8 health  
Carol turn down  
Carol turn up  
Carol move forward 1 step to up  
Carol move forward 1 step to up  
Carol is firing up  
Carol killed Bob and get two health  
Bob is out of the game
```

I have my rules here every u killed some one u win two health:



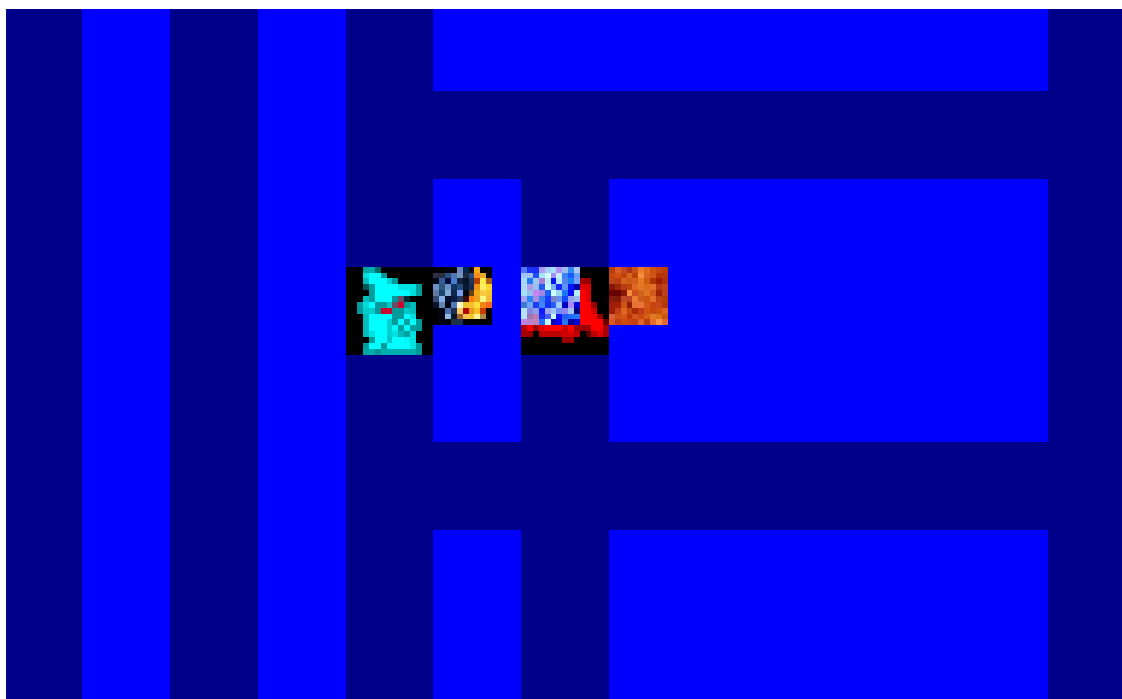
Bob health back to empty, and Carol win 2 health:



Because carol was shot by bob last three blocks away he's health now from 6 to 8

(4) damage lessened by wall:

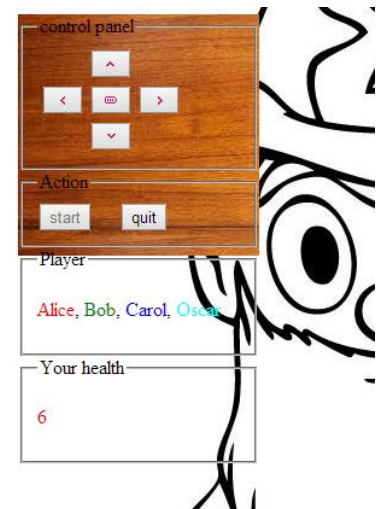
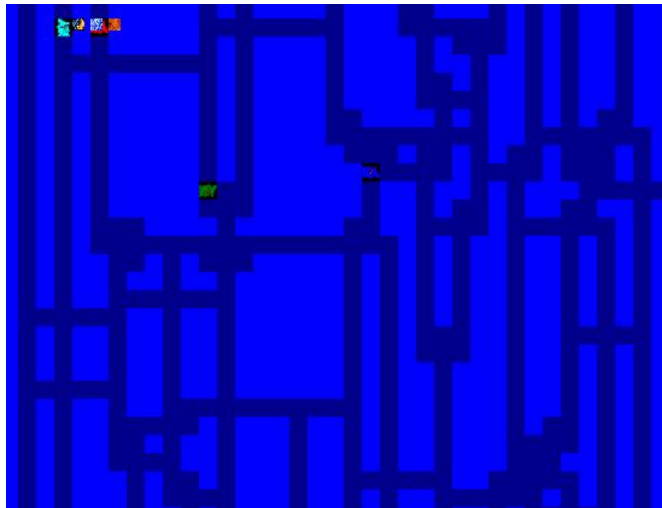
Well if target is behind the wall, it will less  $16/4 = 4$ , so



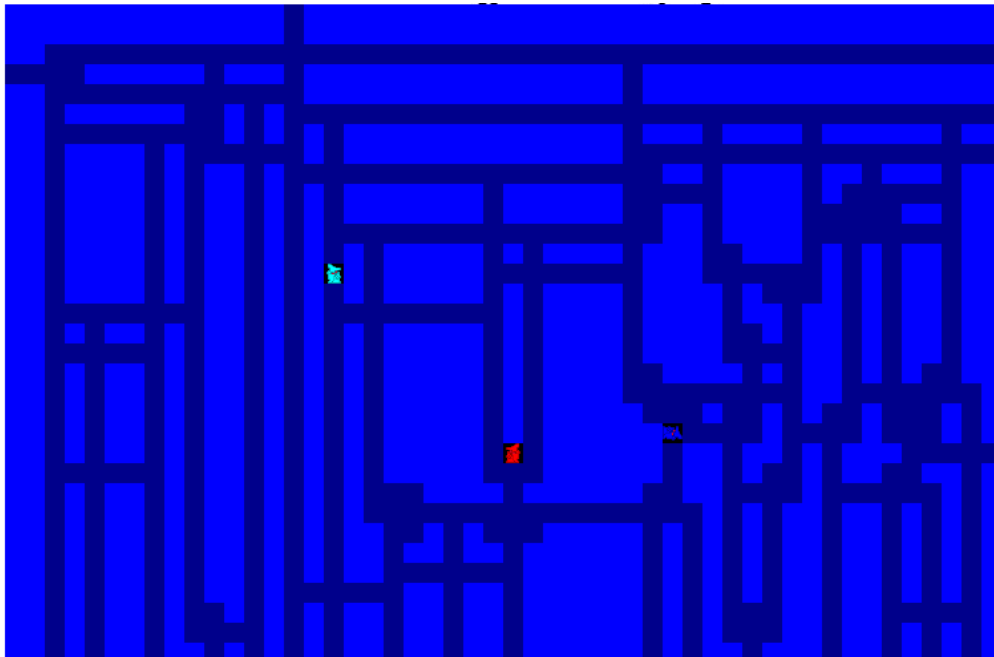
Yep and here it goes

```
Oscar turn up  
Oscar move forward 1 step to up  
Oscar move forward 1 step to up  
Oscar turn right  
Oscar is firing right  
Oscar cause Alice 4 damages  
Alice lost 4 health
```

Health is now 6, and by way the if u see bob here, mainly because I revive him, and if u don't like I will kill him again.



No Bob any more



```
Alice move forward 1 step to up  
Alice move forward 1 step to up  
Alice turn left  
Alice is firing left  
Alice killed Bob and get two health  
Bob is out of the game
```

Ok now end of game, time to study,lol~~~all disconnected

```
Alice move forward 1 step to up  
Alice move forward 1 step to up  
Alice turn left  
Alice is firing left  
Alice killed Bob and get two health  
Bob is out of the game  
2015-04-03 12:08:16 [info] [client 127.0.0.1:45379] Disconnected  
2015-04-03 12:08:16 [info] [client 127.0.0.1:45388] Disconnected  
2015-04-03 12:08:16 [info] [client 127.0.0.1:45397] Disconnected  
2015-04-03 12:08:16 [info] [client 127.0.0.1:45406] Disconnected
```

## Part 4 Display and explanation of code

Basically in this assignment, I use a lot of “//” to explain some line of my code, u will see

### MazeClient/index.html

```
<html>
  <head>
    <title>A Maze</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel='stylesheet' type="text/css" href="Mycss.css">
    <link rel='stylesheet' type="text/css" href=".jquery-ui-1.11.2.custom/jquery-ui.css">
    <style>
      canvas {border: 1px solid black;}
    </style>
  </head>
  <body>
    <h1>A big Maze</h1>

    <div id="loginarea">
      <fieldset>
        <legend><font size="5" color="white">My info login</font></legend>
        <p><font size="2" color="white">Name :</font> <input type="text" id="uname" size="20" /></p>

        <div id="buttons">
          <input type="button" value="Login" onclick="doLogin()" id="Loginbutton"/>
        </div>
      </fieldset>
    </div>

    <div id="mzarea">
      <canvas id="lookwhatIdrew" width="900" height="800"></canvas>
    </div>

    <div id="cPanearea">
      <fieldset>
        <legend>control panel</legend>
        <table>
          <tr>
            <td></td>
            <td><button class="conbu" onclick="up()" disabled><span class="ui-icon ui-icon-carat-1-
```

```

n"></span></button></td>
        <td></td>
    </tr>
    <tr>
        <td><button class="conbu" onclick="left()" disabled><span class="ui-icon ui-icon-carat-1-
w"></span></button></td>
        <td><button class="conbu" onclick="shoot()" disabled><span class="ui-icon ui-icon-battery-
3"></span></button></td>
        <td><button class="conbu" onclick="right()" disabled><span class="ui-icon ui-icon-carat-1-
e"></span></button></td>
    </tr>
    <tr>
        <td></td>
        <td><button class="conbu" onclick="down()" disabled><span class="ui-icon ui-icon-carat-1-
s"></span></button></td>
        <td></td>
    </tr>
</table>
</fieldset>
<fieldset>
    <legend>Action</legend>
    <div id="st">
        <input type="button" value="start" onclick="dostart()" id="start" disabled>
    </div>
    <div id="qu">
        <input type="button" value="quit" onclick="doquit()" id="quit" disabled>
    </div>
</fieldset>

<fieldset>
    <legend>Player</legend>
    <p><span id="persons"></span></p>
</fieldset>

<fieldset>
    <legend>Your health</legend>
    <p><span id="health"></span></p>
</fieldset>
</div>
<script type="text/javascript" src="jquery-2.1.3.min.js"></script>
<script type="text/javascript" src="Myscript.js"></script>
</body>

```

</html>

## MazeClient/Myscript.js

```
var socket = null;
var url = 'ws://localhost:8000/mazerunner';
var canvas,context;

var peopleup,
    peopledown,
    peopleleft,
    peopleright;

peopleup = new Image();
peopleup.src = "image/up.png";

peopledown = new Image();
peopledown.src = "image/down.png";

peopleleft = new Image();
peopleleft.src = "image/left.png";

peopleright = new Image();
peopleright.src = "image/right.png";

var startx = 10,
    starty = 10;//maze first point start to draw

var colourmap = new Object();
colourmap.red = "#ff0000";
colourmap.yellow = "#ffff00";
colourmap.blue="#0000ff";
colourmap.lime="#00ff00";
colourmap.aqua="#00ffff";
colourmap.fuschia="#ff00ff";
colourmap.maroon="#800000";
colourmap.green="#008000";
colourmap.navy="#000080";

var pi,
```

```

    pj;//this user position
var facing;//this user facing
var username;//this user name
var alive = false;//this user live or unactive
var mazematrix = [];

var facedir = ["up","down","left","right"];
var health;//the user health
var login = false;//this is for checking if user have login
var username = [];//user name list for all login
var usersi = 0;

var fireball = [];
fireball[0] = new Image();
fireball[0].src = "image/fire1.PNG";

fireball[1] = new Image();
fireball[1].src = "image/fire2.PNG";

fireball[2] = new Image();
fireball[2].src = "image/fire3.PNG";

window.onload = function(){
    canvas = document.getElementById("lookwhatIdrew");
    context = canvas.getContext("2d");
    doSetupSocket();
};

function doSetupSocket()
{
    // Get a WebSocket - browser dependent!
    socket = ("MozWebSocket" in window ? new MozWebSocket(url) : new WebSocket(url));
    console.log("socket created");
    socket.onclose = function(msg) {
        alert("Connection closed");
    }
    socket.onmessage = handleUpdate;
}

function handleUpdate(msg)
{
    var firei,firej;
    var firesig = false;
    var firedir;//I handle all the fire datas as some special datas

```



```

console.log("handle update");
var info = JSON.parse(msg.data);

var infodata = info['data'];

var i;
var persons = document.getElementById('persons');
var currentHealth = document.getElementById('health');
persons.innerHTML="";
var str = "";
var maze;
for(i in infodata){
    var clientdata = infodata[i];
    if(clientdata['login'] == true){//you have to login to see the maze
        var cname = clientdata['uname'];
        maze = clientdata['mazedt'];
        var colorname = clientdata['colour'];
        var paintcolor = colourmap[colorname];

        if(cname == username){
            login = clientdata['login'];
            alive = clientdata['alive'];
            facing = clientdata['facing'];
            pi = clientdata['pi'];
            pj = clientdata['pj'];
            if(alive == true){
                health = clientdata['health'];
                currentHealth.innerHTML = "<font style='color:" + paintcolor + ">" +
health + "</font> ";
                if(clientdata['diemsg'] == "collision"){
                    alert("you have collision with " + clientdata['maker']);
                    currentHealth.innerHTML = "";
                    alive = false;
                    doquit();
                }
                if(clientdata['diemsg'] == "shot"){
                    alert("you have been shot by " + clientdata['maker']);
                    currentHealth.innerHTML = "";
                    alive = false;
                    doquit();
                }
            }
        }
    }
}

```

```

    }

    if(clientdata['alive'] == true){//only people alive should draw in the maze
        var facing = clientdata['facing'];
        var ppi = clientdata['pi'];
        var ppj = clientdata['pj'];
        LoadPeople(ppi,ppj,facing,paintcolor);
    }

    firesig = clientdata['fith'];
    //if(firesig == true)
        //alert("get firing signal");
    firei = clientdata['fi'];
    firej = clientdata['fj'];
    firedir = clientdata['firedir'];

    str = str + "<font style='color:" + paintcolor + ">" + cname + "</font>, ";
}

if(username.indexOf(clientdata['uname']) == -1){
    username[usersi++] = clientdata['uname'];
}
}

if(login == true){
    persons.innerHTML = str;
    mazematrix = LoadMaze(maze);
    if(firesig == true){
        //alert("get firing signal");
        drawmagicfire(firei,firej,firedir);
    }
    $("#loginarea").hide();
}

}

function doLogin()
{
    username = document.getElementById('uname').value;
    if(username.indexOf(username) != -1){
        alert(username + " is already exist, pleasy try another");
    }
    else{
        var startb = document.getElementById("start");

```

```

        startb.disabled = false;

        console.log("you are login");

        var data = new Object();
        data.uname = username;
        var command = new Object();
        command.action = "login";
        command.data = data;

        var stringversion = JSON.stringify(command);
        socket.send(stringversion);
    }
}

function dostart(){
    var startb = document.getElementById("start");
    startb.disabled = true;

    $('.conbu').removeAttr("disabled");
    $('#quit').removeAttr("disabled");

    var command = new Object();
    command.action = "start";
    command.data = "";

    var stringversion = JSON.stringify(command);
    socket.send(stringversion);
}

function doquit(){
    $('.conbu').attr('disabled',"true");
    $('#quit').attr('disabled',"true");
    $('#start').removeAttr("disabled");

    var data = new Object();
    data.pi = pi;
    data.pj = pj;

    var command = new Object();
    command.action = "quit";
    command.data = data;

```

```

    var stringversion = JSON.stringify(command);
    socket.send(stringversion);
}

function up(){
    var command = new Object();
    var stringversion;
    var data = new Object();
    data.dir = "up";
    if(facing == "up"){
        if(pi - 1 >= 0){//alert(azematrix[pi - 1][pj]);
            if(mazematrix[pi - 1][pj] != 1){
                command.action = "move";
                command.data = data;
                stringversion = JSON.stringify(command);
                socket.send(stringversion);
            }
        }
    }
    else{
        facing = "up";
        command.action = "changedir";
        command.data = data;
        stringversion = JSON.stringify(command);
        socket.send(stringversion);
    }
}

function down(){
    var command = new Object();
    var stringversion;
    var data = new Object();
    data.dir = "down";
    if(facing == "down"){
        if(pi + 1 <= 49){//alert(mazematrix[pi + 1][pj]);
            if(mazematrix[pi + 1][pj] != 1){
                command.action = "move";
                command.data = data;
                stringversion = JSON.stringify(command);
                socket.send(stringversion);
            }
        }
    }
    else{

```

```

        facing = "down";
        command.action = "changedir";
        command.data = data;
        stringversion = JSON.stringify(command);
        socket.send(stringversion);
    }
}

function left(){
    var command = new Object();
    var stringversion;
    var data = new Object();
    data.dir = "left";
    if(facing == "left"){
        if(pj - 1 >= 0){//alert(mazematrix[pi][pj - 1]);
            if(mazematrix[pi][pj - 1] != 1){
                command.action = "move";
                command.data = data;
                stringversion = JSON.stringify(command);
                socket.send(stringversion);
            }
        }
    }
    else{
        facing = "left";
        command.action = "changedir";
        command.data = data;
        stringversion = JSON.stringify(command);
        socket.send(stringversion);
    }
}

```

```

function right(){
    var command = new Object();
    var stringversion;
    var data = new Object();
    data.dir = "right";
    if(facing == "right"){
        if(pj + 1 <= 49){//alert(mazematrix[pi][pj + 1]);
            if(mazematrix[pi][pj + 1] != 1){
                command.action = "move";
                command.data = data;
                stringversion = JSON.stringify(command);
                socket.send(stringversion);
            }
        }
    }
}

```

```

        }
    }
}
else{
    facing = "right";
    command.action = "changedir";
    command.data = data;
    stringversion = JSON.stringify(command);
    socket.send(stringversion);
}
}

function shoot(){
    var command = new Object();

    command.action = "fire";
    command.data = ""; //In my view this can be send as empty, because facing is already store in
the server side
    stringversion = JSON.stringify(command);
    socket.send(stringversion);
}

function LoadMaze(mdatas)
{
    var maze = [];

    for(var i = 0; i < mdatas.length; i++)
    {
        maze[i] = mdatas[i].split("");
    }

    context.fillStyle="#0000ff";
    for(var i = 0; i < maze.length; i++) //load walls
    {
        for(var j = 0; j < maze[i].length; j++)
        {
            if(maze[i][j] == 1){
                context.fillRect(startx, starty, 15, 15);
            }
            startx += 15;
        }
        startx = 10;
        starty += 15;
    }
}

```

```

startx = 10;
starty = 10;
context.fillStyle="#00008B";
for(var i = 0;i < maze.length;i++){//load road
    for(var j = 0;j < maze[i].length;j++){
        if(maze[i][j] == 0){
            context.fillRect(startx,starty,15,15);
        }
        startx += 15;
    }
    startx = 10;
    starty += 15;
}
startx = 10;
starty = 10;
return maze;
}

function LoadPeople(i,j,facing,col){
    var px = startx + 15 * j;
    var py = starty + 15 * i;

    context.fillStyle = col;

    switch(facing){
        case 'left':{
            context.fillRect(px,py,15,15);
            context.drawImage(peopleleft,px,py,15,15);}
            break;
        case 'right':{
            context.fillRect(px,py,15,15);
            context.drawImage(peopleright,px,py,15,15);}
            break;
        case 'up':{
            context.fillRect(px,py,15,15);
            context.drawImage(peopleup,px,py,15,15);}
            break;
        case 'down':{
            context.fillRect(px,py,15,15);
            context.drawImage(peopledown,px,py,15,15);}
            break;
    }
}

```

```

}

function drawmagicfire(firei,firej,firedir){
    var fx = startx + 15 * firej;
    var fy = starty + 15 * firei;

    if(firedir == "up"){
        fy -= 15;
        for(var i = 0;i < 3;i ++){
            if(firei - i - 1 >= 0){
                context.drawImage(fireball[i],fx,fy,10,10);
                fy -= 15;
            }
        }
    }
    else if(firedir == "down"){
        fy += 15;
        for(var i = 0;i < 3;i ++){
            if(firei + i + 1 < 50){
                context.drawImage(fireball[i],fx,fy,10,10);
                fy += 15;
            }
        }
    }
    else if(firedir == "left"){
        fx -= 15;
        for(var i = 0;i < 3;i ++){
            if(firej - i - 1 >=0){
                context.drawImage(fireball[i],fx,fy,10,10);
                fx -= 15;
            }
        }
    }
    else if(firedir == "right"){
        fx += 15;
        for(var i = 0;i < 3;i ++){
            if(firej + i + 1 < 50){
                context.drawImage(fireball[i],fx,fy,10,10);
                fx += 15;
            }
        }
    }
}

```



## MazeClient/Mycss.css

```
body{
    background-image: url(image/people.jpg);
}

#loginarea{
    background-image: url(image/welcomegame.png);
    height: 200px;
}

root {
    display: block;
}

div#left {
    float:left;
    width:480px;
}

div#buttons {
    clear:both;
}

#mzarea{
    width: 920px;
    height: 810px;
    float: left;
}

#cPanearea{
    background-image: url(image/panelboard.jpg);
    width: 200px;
    height: 200px;
    float: right;
    margin-top: 200px;
    margin-right: 200px;
}
```

```

#st{
    width: 50px;
    float: left;
}

#qu{
    width: 50px;
    float: right;
    margin-right: 50px
}

#pple{
    background-image: url(image/people.jpg);
}

```

Due to the fact the server side I only modify the server.php and also create a new class which extends Application class in Application.php so I only show stuff I write:

### MazeServer/server.php

```

ini_set('display_errors', 1);
error_reporting(E_ALL);

require(__DIR__ . '/lib/SplClassLoader.php');

$classLoader = new SplClassLoader('WebSocket', __DIR__ . '/lib');
$classLoader->register();

$server = new \WebSocket\Server('127.0.0.1', 8000, false);

$server->setMaxClients(100);

$server->setCheckOrigin(false);

$server->setMaxConnectionsPerIp(100);
$server->setMaxRequestsPerMinute(2000);

$server->registerApplication('mazerunner',
\WebSocket\Application\MazeApplication::getInstance());

```

```
$server->run();
```

## MazeServer/MazeApplicaton.php

```
<?php
namespace WebSocket\Application;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of MazeApplication
 *
 * @author Victor
 */
class MazeApplication extends Application{
    private static $colourNamesArray = array('red', 'green', 'blue', 'aqua',
        'fuschia', 'lime', 'maroon', 'navy', 'yellow');
    private static $colourcounter = 0;

    private static $facedir = array('up','down','left','right');
    private static $facecounter = 0;

    private $_clients = array();
    private static $block = [];//a block of maze raw datas
    private static $jpblock = [];//a block of maze data ready to convert to json
    private static $jblock = [];//a block of maze been jason stringfied

    private static $harmtype = "";//what kind of harm
    private static $target = "";
    private static $maker = "";//username who cause damage, or death
    private static $damage = array();//record who get damage and how bad

    private static $bulldet;//detect if there is fire in the maze every should be warned
    private static $fpos = [];//the i and j for bullet
    private static $firedir;

    function __construct() {
```

```

MazeApplication::$block = MazeApplication::inidataformaze(MazeApplication::$block);
MazeApplication::$jpbblock = MazeApplication::generator(MazeApplication::$block);
}

public function onConnect($client) {
    $id = $client->getClientId();
    $this->_clients[$id] = $client;

    $info = array();
    $info['colour'] = MazeApplication::$colourNamesArray[MazeApplication::$colourcounter];

    MazeApplication::$colourcounter++;
    if (MazeApplication::$colourcounter == 9)
        MazeApplication::$colourcounter = 0;

    $client->setClientInfo($info);
}

public function onData($data, $client) {
    $decodedData = $this->_decodeData($data);

    $action = $decodedData['action'];
    $info = $decodedData['data'];
    $this->_process($client, $action, $info);
}

private function _process($client, $action, $info){
    $infoarray = $client->getClientInfo();

    if($action == "start"){
        $posij = MazeApplication::inipos();
        $infoarray['alive'] = true;//you are live when u connect
        $infoarray['facing'] = MazeApplication::$facedir[MazeApplication::$facecounter];//initialize facing
        $infoarray['health'] = 10;//ini health
        $infoarray['pi'] = $posij[0]; //ini position i
        $infoarray['pj'] = $posij[1]; //ini position j
        $infoarray['diemsg'] = "";//ini die message
        MazeApplication::$jpbblock[$posij[0]][$posij[1]] = 2;

        MazeApplication::$facecounter++;
        if(MazeApplication::$facecounter == 3)
            MazeApplication::$facecounter = 0;

        echo $infoarray['uname']. " is now starting the game\n";
    }
}

```

```

}

if($action == "quit"){
    $infoarray['alive'] = false;
    $infoarray['diemsg'] = "";
    MazeApplication::$sharmtype = "";
    MazeApplication::$jpbblock[$info['pi']][$info['pj']] = 0;

    echo $infoarray['uname']. " is out of the game\n";
}

if($action == "login"){
    $infoarray['uname'] = $info['uname'];
    $infoarray['login'] = true;
    echo $infoarray['uname']. " is login now\n";
}

if($action == "changedir"){
    switch ($info['dir']){
        case 'up': $infoarray['facing'] = "up"; break;
        case 'down': $infoarray['facing'] = "down"; break;
        case 'left': $infoarray['facing'] = "left"; break;
        case 'right': $infoarray['facing'] = "right"; break;
    }
    echo $infoarray['uname']. " turn ". $infoarray['facing']. "\n";
}

$targeti = -1;
$targetj = -1; //target i and j

if($action == "move"){
    switch ($info['dir']){
        case 'up':{
            if(MazeApplication::$jpbblock[$infoarray['pi'] - 1][$infoarray['pj']] == 2){
                MazeApplication::$sharmtype = "collision";
                MazeApplication::$maker = $infoarray['uname'];
                $targeti = $infoarray['pi'] - 1;
                $targetj = $infoarray['pj'];
            }
            else if(MazeApplication::$jpbblock[$infoarray['pi'] - 1][$infoarray['pj']] == 0){
                MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 0;
                $infoarray['pi'] -= 1;
                MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 2;
                echo $infoarray['uname']. " move forward 1 step to ". $infoarray['facing']. "\n";
            }
        }
    }
}

```

```

    }
}break;
case'down':{
    if(MazeApplication::$jpbblock[$infoarray['pi'] + 1][$infoarray['pj']] == 2){
        MazeApplication::$sharmtype = "collision";
        MazeApplication::$maker = $infoarray['uname'];
        $targeti = $infoarray['pi'] + 1;
        $targetj = $infoarray['pj'];
    }
    else if(MazeApplication::$jpbblock[$infoarray['pi'] + 1][$infoarray['pj']] == 0){
        MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 0;
        $infoarray['pi'] += 1;
        MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 2;
        echo $infoarray['uname']. " move forward 1 step to ".$infoarray['facing']."\n";
    }
}break;
case'left':{
    if(MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj'] - 1] == 2){
        MazeApplication::$sharmtype = "collision";
        MazeApplication::$maker = $infoarray['uname'];
        $targeti = $infoarray['pi'];
        $targetj = $infoarray['pj'] - 1;
    }
    else if(MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj'] - 1] == 0){
        MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 0;
        $infoarray['pj'] -= 1;
        MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 2;
        echo $infoarray['uname']. " move forward 1 step to ".$infoarray['facing']."\n";
    }
}break;
case'right':{
    if(MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj'] + 1] == 2){
        MazeApplication::$sharmtype = "collision";
        MazeApplication::$maker = $infoarray['uname'];
        $targeti = $infoarray['pi'];
        $targetj = $infoarray['pj'] + 1;
    }
    else if(MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj'] + 1] == 0){
        MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 0;
        $infoarray['pj'] += 1;
        MazeApplication::$jpbblock[$infoarray['pi']][$infoarray['pj']] = 2;
        echo $infoarray['uname']. " move forward 1 step to ".$infoarray['facing']."\n";
    }
}break;

```

```

    }
}

if($action == "fire"){
    $maxdamage = 16;

    MazeApplication::$maker = $infoarray['uname'];
    MazeApplication::$bulldet = true;
    MazeApplication::$fpos[0] = $infoarray['pi'];
    MazeApplication::$fpos[1] = $infoarray['pj'];
    MazeApplication::$firedir = $infoarray['facing'];
    echo $infoarray['uname']. " is firing ". $infoarray['facing']. "\n";

    $ti = $infoarray['pi'];
    $tj = $infoarray['pj'];
    switch (MazeApplication::$firedir){
        case 'up':{
            for($i = 0;$i < 3;$i ++){
                if($ti - $i - 1 >= 0){
                    $ti -= 1;
                    if(MazeApplication::$jpbblock[$ti][$tj] == 1){
                        $maxdamage /= 4;
                    }
                    else if(MazeApplication::$jpbblock[$ti][$tj] == 2){
                        MazeApplication::$harmtype = "damage";
                        foreach ($this->_clients as $tclient){//tclient means target client
                            $tinfo = $tclient->getClientInfo();
                            if($ti == $tinfo['pi']&&$tj == $tinfo['pj']){
                                MazeApplication::$damage[$tinfo['uname']] = $maxdamage;
                                if($tinfo['health'] - $maxdamage <=0){
                                    $infoarray['health'] += 2;
                                    echo $infoarray['uname']. " killed ". $tinfo['uname']. " and get
two health\n";
                                }
                            }
                            else{
                                echo $infoarray['uname']. " cause ". $tinfo['uname']. "
". $maxdamage. " damages\n";
                            }
                        }
                        break;
                    }
                }
            }
            $maxdamage /= 2;
        }
        else{

```

```

        $maxdamage /= 2;
    }
}
}break;
case 'down':{
    for($i = 0;$i < 3;$i ++){
        if($ti + $i + 1 >= 0){
            $ti += 1;
            if(MazeApplication::$jpbblock[$ti][$tj] == 1){
                $maxdamage /= 4;
            }
            else if(MazeApplication::$jpbblock[$ti][$tj] == 2){
                MazeApplication::$sharmtype = "damage";
                foreach ($this->_clients as $tclient){//tclient means target client
                    $tinfo = $tclient->getClientInfo();
                    if($ti == $tinfo['pi']&&$tj == $tinfo['pj']){
                        MazeApplication::$damage[$tinfo['uname']] = $maxdamage;
                        if($tinfo['health'] - $maxdamage <=0){
                            $infoarray['health'] += 2;
                            echo $infoarray['uname']." killed ".$tinfo['uname']." and get
two health\n";
                        }
                    }
                    else{
                        echo $infoarray['uname']." cause ".$tinfo['uname']."
".$maxdamage." damages\n";
                    }
                }
                break;
            }
        }
        $maxdamage /= 2;
    }
    else{
        $maxdamage /= 2;
    }
}
}break;
case 'left':{
    for($i = 0;$i < 3;$i ++){
        if($tj - $i - 1 >= 0){
            $tj -= 1;
            if(MazeApplication::$jpbblock[$ti][$tj] == 1){
                $maxdamage /= 4;

```



```

    }
    else if(MazeApplication::$jpbblock[$ti][$tj] == 2){
        MazeApplication::$sharmtype = "damage";
        foreach ($this->_clients as $tclient){//tclient means target client
            $tinfo = $tclient->getClientInfo();
            if($ti == $tinfo['pi']&&$tj == $tinfo['pj']){
                MazeApplication::$damage[$tinfo['uname']] = $maxdamage;
                if($tinfo['health'] - $maxdamage <=0){
                    $infoarray['health'] += 2;
                    echo $infoarray['uname']. " killed ".$tinfo['uname']. " and get
two health\n";

                }
            }
            else{
                echo $infoarray['uname']. " cause ".$tinfo['uname']. "
".$maxdamage." damages\n";

            }
            break;
        }
    }
    $maxdamage /= 2;
}
else{
    $maxdamage /= 2;
}
}
}break;
case 'right':{
    for($i = 0;$i < 3;$i ++){
        if($tj + $i + 1 >= 0){
            $tj += 1;
            if(MazeApplication::$jpbblock[$ti][$tj] == 1){
                $maxdamage /= 4;
            }
            else if(MazeApplication::$jpbblock[$ti][$tj] == 2){
                MazeApplication::$sharmtype = "damage";
                foreach ($this->_clients as $tclient){//tclient means target client
                    $tinfo = $tclient->getClientInfo();
                    if($ti == $tinfo['pi']&&$tj == $tinfo['pj']){
                        MazeApplication::$damage[$tinfo['uname']] = $maxdamage;
                        if($tinfo['health'] - $maxdamage <=0){
                            $infoarray['health'] += 2;
                            echo $infoarray['uname']. " killed ".$tinfo['uname']. " and get
two health\n";

```

```

        }
        else{
            echo $infoarray['uname']." cause ".$tinfo['uname'].
"$.$maxdamage." damages\n";
        }
        break;
    }
}
$maxdamage /= 2;
}
else{
    $maxdamage /= 2;
}
}
}break;
}
}
}

```

```

if(MazeApplication::$harmtype == "collision"){
    $infoarray['diemsg'] = "collision";
    foreach ($this->_clients as $sclient){
        $ainfo = $sclient->getClientInfo();
        if($targeti == $ainfo['pi']&&$targetj == $ainfo['pj']){
            $infoarray['maker'] = $ainfo['uname'];
            MazeApplication::$target = $ainfo['uname'];
            break;
        }
    }
}
}

```

```

MazeApplication::$jblock = MazeApplication::JstringMaze(MazeApplication::$jblock);//convert to
string

```

```

$client->setClientInfo($infoarray);

```

```

$updateData = $this->_composeUpdateMessage();

```

```

/****reset all these after used****/

```

```

MazeApplication::$harmtype = "";

```

```

MazeApplication::$maker = "";

```

```

MazeApplication::$target = "";

```

```

MazeApplication::$damage = array();

$encodedUpdate = $this->_encodeData('update', $updateData);

foreach ($this->_clients as $sendto) {
    $sendto->send($encodedUpdate);
}
}

private function _composeUpdateMessage() { //each client get all client message
    $msgdata = array();
    foreach ($this->_clients as $sclient) {
        $info = $sclient->getClientInfo();

        if(array_key_exists('uname', $info)){
            if($info['uname'] != MazeApplication::$maker){
                if(MazeApplication::$sharmtype == "collision"){
                    if($info['uname'] == MazeApplication::$target){
                        $info['diemsg'] = "collision";
                        $info['maker'] = MazeApplication::$maker;
                    }
                }
            }

            if(MazeApplication::$sharmtype == "damage"){
                if(array_key_exists($info['uname'], MazeApplication::$damage)){
                    $thealth = $info['health'] - MazeApplication::$damage[$info['uname']];
                    if($thealth <= 0){
                        $info['diemsg'] = "shot";
                        $info['maker'] = MazeApplication::$maker;
                    }
                } else{
                    $info['health'] = $thealth;
                    echo $info['uname']. " lost ".MazeApplication::$damage[$info['uname']]. "
health\n";
                }
            }
        }
    }
}

if(MazeApplication::$bulldet == true){
    $info['firedir'] = MazeApplication::$firedir;
    $info['fi'] = MazeApplication::$fpos[0];
}

```

```

        $info['fj'] = MazeApplication::$fpos[1];
        $info['fith'] = true;//fire in the hole
    }
    else{
        $info['fith'] = false;
    }

    $info['mazedt'] = MazeApplication::$jblock;//each client should updata there maze
    $msgdata[] = $info;
    $aclient->setClientInfo($info);
}
MazeApplication::$bulldet = false;    //reset,otherwise fire everywhere
return $msgdata;
}

public function onDisconnect($client) {
    $id = $client->getClientId();
    unset($this->_clients[$id]);
}

public function inidataformaze($block){
    for($i = 0;$i < count($block);$i ++){
        {
            $block[$i] = [];
        }

        for($i = 0;$i < 50;$i ++)//ini maze with 1
        {
            for($j = 0;$j < 50;$j++){
                {
                    $block[$i][$j] = 1;
                }
            }
        }
        return $block;
    }
}

public function generator($a){//I use shot gun algorithm to ramdom generator maze, which I invent it and
name it

    $countr = 0;
    while($countr < 70)//random spread 50 holes on the wall
    {
        $random = range(0, 49);
        shuffle($random);
        $x = $random[0];

```

```

$y = $random[1];
if($x != 0&&$x != 49&&$y != 0&&$y != 49){
    $a[$y][$x] = 0;
    $countr++;
}
}

```

```

$statistic = [];
for($i = 0;$i <= 49;$i ++){
{
    for($j = 0;$j <= 49;$j ++){
    {
        if($a[$i][$j] == 0){
            $cell = [$i,$j];
            $statistic[] = $cell;
        }
    }
}
}

```

for(\$i = 0;\$i < count(\$statistic);\$i ++)//make a holes spread like 'cross' pattern from 4 direction,  
which reach end or another road stop push down the wall

```

{
    $tcell = $statistic[$i];
    $curri = $tcell[0];
    $currj = $tcell[1];

    //dig to up
    $curri--;
    while($curri >= 0){
        if($a[$curri][$currj] == 0)
            break;
        if($a[$curri][$currj - 1] == 0 || $a[$curri][$currj + 1] == 0){//left or right,
            $a[$curri][$currj] = 0;
            break;
        }
        $a[$curri][$currj] = 0;
        $curri--;
    }
}

```

```

$curri = $tcell[0];
$currj = $tcell[1];

```

```

//dig to down
$curri++;

```

```

while($curri <= 49){
    if($a[$curri][$currj] == 0)
        break;
    if($a[$curri][$currj - 1] == 0 || $a[$curri][$currj + 1] == 0){
        $a[$curri][$currj] = 0;
        break;
    }
    $a[$curri][$currj] = 0;
    $curri++;
}

$curri = $tcell[0];
$currj = $tcell[1];

//dig to left
$currj--;
while($currj >= 0){
    if($a[$curri][$currj] == 0)
        break;
    if($a[$curri - 1][$currj] == 0 || $a[$curri + 1][$currj] == 0){
        $a[$curri][$currj] = 0;
        break;
    }
    $a[$curri][$currj] = 0;
    $currj--;
}

$curri = $tcell[0];
$currj = $tcell[1];

//dig to right
$currj++;
while($currj <= 49){
    if($a[$curri][$currj] == 0)
        break;
    if($a[$curri - 1][$currj] == 0 || $a[$curri + 1][$currj] == 0){
        $a[$curri][$currj] = 0;
        break;
    }
    $a[$curri][$currj] = 0;
    $currj++;
}
}
return $a;

```

```

    }

    public function JstringMaze($a){//this is for generate stringfy json datas
        for($i = 0;$i < 50;$i++)
        {
            $sstr = "";
            for($j = 0;$j < 50;$j++)
            {
                $sstr = $sstr.$a[$i][$j];
            }
            $jsonformat[] = $sstr;
        }
        return $jsonformat;
    }

    public function inipos(){
        $posarr;
        $random = range(0,49);
        while(true){
            shuffle($random);
            $i = $random[0];
            shuffle($random);
            $j = $random[0];
            if(MazeApplication::$jpbblock[$i][$j] == 0){
                $posarr[0] = $i;
                $posarr[1] = $j;
                break;
            }
        }
        return $posarr;
    }

}

```