**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**LÂM TIÊN ĐIỀN AN**

**PHẠM VŨ THIÊN BẢO**

**HUỲNH TIẾN ĐẠT**

**NGÔ XUÂN ĐỨC**

**ĐỒ ÁN LẬP TRÌNH NHÚNG CĂN BẢN**

**GAME BATTLESHIP**

**TP. HỒ CHÍ MINH, 2018**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**LÂM TIÊN ĐIỀN AN - 15520002**

**PHẠM VŨ THIÊN BẢO - 15520042**

**HUỲNH TIẾN ĐẠT - 15520101**

**NGÔ XUÂN ĐỨC - 15520124**

**ĐỒ ÁN MÔN HỌC**

**GAME BATTLESHIP**

**GIẢNG VIÊN HƯỚNG DẪN**

**TRẦN NGỌC ĐỨC**

**TP. HỒ CHÍ MINH, 2018**

MỤC LỤC

MỞ ĐẦU

# TỔNG QUAN

## Mục tiêu của đề tài

Tạo ra trò chơi battleship online.

### Tạo server hỗ trợ người chơi online

Sử dụng dịch vụ Amazon Web Services tạo máy chủ windows.

Hỗ trợ người chơi online trực tiếp qua mạng và lưu trữ dữ liệu người chơi.

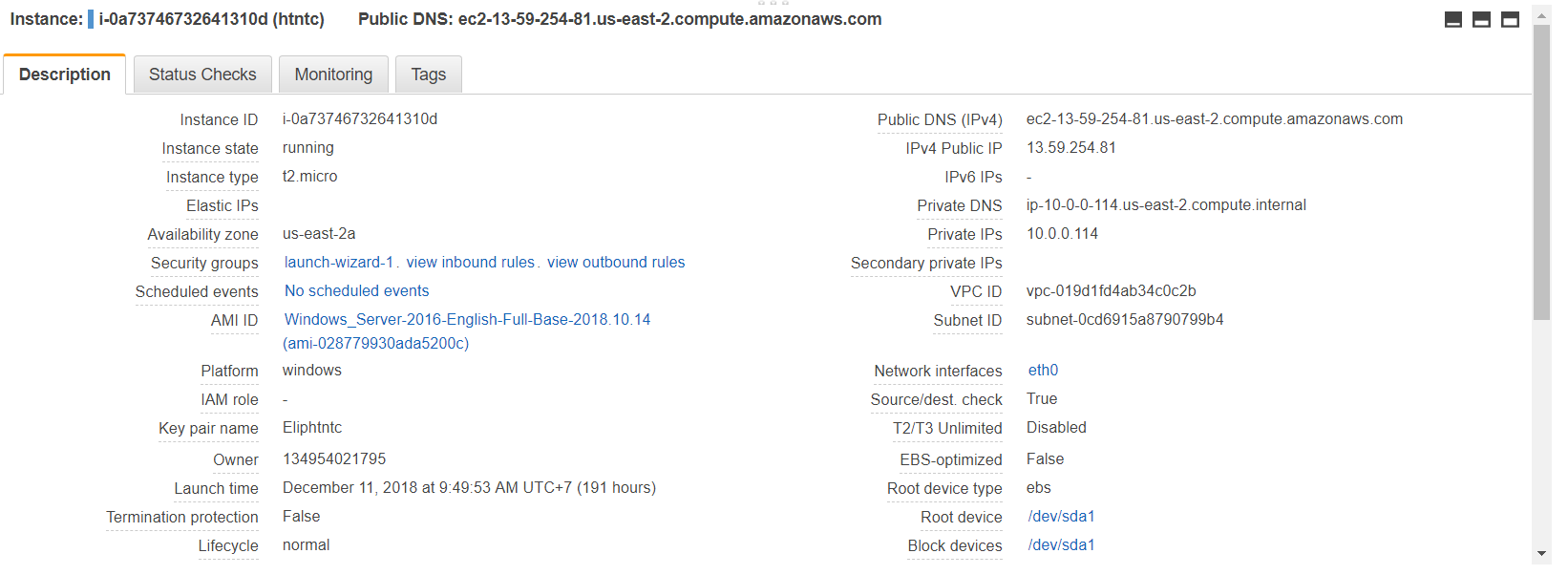
### Tạo gamepad kết nối qua wifi để điều khiển trò chơi

Gamepad không dây hỗ trợ điều khiển trò chơi, tạo điều kiện thuận lợi cho người chơi ngồi cách xa màn hình.

# CƠ SỞ LÝ THUYẾT

## Battleship

### Tạo windows server sử dụng dịch vụ amazon



Server chạy tại địa chỉ: 13.59.254.81

Domain: [thuyenchien.tk](http://thuyenchien.tk)

### Game battleship

#### Giao diện

Sử dụng XAMPP với PHP ver 5.6.38 chạy trên server.

Dùng <table> tạo ra bảng có kích thước 10x10 để đặt các tàu chiến.

Sử dụng js để random vị trí các tàu chiến trong bảng.

<table class="board second black">

<tr class="row top">

<td class="corner"></td

><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td class="corner"></td></tr>

<tr class="row">

<td class="side">A</td>

<td id="tgt-00" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-01" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-02" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-03" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-04" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-05" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-06" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-07" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-08" onclick="tdclick(this);">&nbsp;</td>

<td id="tgt-09" onclick="tdclick(this);">&nbsp;</td>

<td class="side">A</td>

</tr>

...

</table>

#### Xử lý

Sử dụng javascript tạo file *game.js* để xử lý các thao tác như di chuyển lên xuống qua lại và bắn trong bảng.

#### Kết nối với gamepad

function tdclick(tdc) { //khi một ô trong bảng được chọn sẽ gọi hàm này

if (tdc != null) {

start.focus();

start.style.backgroundColor = '';

tdc.focus();

tdc.style.backgroundColor = '#ff6699'; //đổi màu background

start = tdc;

}

if (my\_turn){

var $this = $(tdc);

var id = $this.attr('id').slice(4);

if ($this.hasClass('curshot')) { // removing square

var $shots = $('#shots');

var value = $shots.val( );

value = value.split(',');

value.splice(value.indexOf(id), 1);

value.join(',');

$shots.val(value);

++shots;

}

else { // adding square

var $shots = $('#shots');

var value = $shots.val( );

value = value.split(',');

value.push(id);

value.join(',');

$shots.val(value);

--shots;

}

// update the shot markers

update\_shots( );

.....

Dùng nodejs mở port 3000 chạy trên máy chủ để nhận thông tin từ các client.

Trang game battleship là một client nhận các lệnh lên xuống qua lại và bắn từ server nodejs.

Gamepad có ESP8266 là một client gửi các lệnh di chuyển và bắn lên server nodejs.

Server nodejs có nhiệm vụ nhận các lệnh từ gamepad và xác định đúng player kết nối với gamepad để gửi trả về.

Server nodejs:

//server.js

var app = require('express')();

var http = require('http').Server(app);

var io = require('socket.io')(http);

io.on('connection', function (socket){//Room

socket.on('esp1', function(message){

socket.join('room1');

console.log('ESP1 joined Room1');

esp1='on';

});

socket.on('esp2', function(message){

socket.join('room2');

console.log('ESP2 joined Room2');

});

socket.on('Player1', function(message){

socket.join('room1');

console.log('Player1 joined Room1');

});

socket.on('Player2', function(message){

socket.join('room2');

console.log('Player2 joined Room2');

});

socket.on('P1-UP', function(message){

console.log("client say UP");

io.to('room1').emit('UP', {command: 'P1-UP'});

});

socket.on('P1-DOWN', function(message){

console.log("client say DOWN");

io.to('room1').emit('DOWN', {command: 'P1-DOWN'});

});

....

-Client:

//client.js

var socket = io.connect('http://10.0.0.194:3000', {reconnect: true});

socket.on('connect', function (socket) {

console.log('Connected!');

});

socket.on('A', function(socket){

console.log('Server say A');

});

socket.on('UP', function(socket){

console.log('Server say UP'); //'38'

checkinput('38');

});

socket.on('DOWN', function(socket){

console.log('Server say DOWN'); //'40'

checkinput('40');

});

socket.on('LEFT', function(socket){

console.log('Server say LEFT'); //'37'

checkinput('37');

//tương ứng với phím mũi tên trái trên bàn phím

});

....

## Gamepad

* Sử dụng kit Nuvoton 131 kết hợp với ESP8266 để giao tiếp với server.

### MCU: Kit Nuvoton 131



* NuMaker uno là một công cụ phát triển cụ thể cho dòng NuMicro™ Cortex®-M0 mà người dùng có thể lập trình rõ ràng.
* Nuvoton cung cấp gói hộ trợ cho nuc131 tại <https://github.com/OpenNuvoton/NUC131BSP>
* BSP này bao gồm
  + Document: Driver reference manual and revision history.
  + Libraly : Driver header and source files.
  + SampleCode: Driver samplecode for Keil C/IDA/GCC.
* Đồ án sử dụng kit này làm trung gian giao tiếp giữa nút nhấn và ESP8266 thông qua interrupt và UART.

### Kết nối server thông qua wifi: ESP8266



* ESP8266 là một chip wifi giá thấp với đầy đủ TCP/IP stack và khả năng của một vi điều khiển.

### Giao tiếp giữa Kit nuvoton131 và esp8266

* Để giao tiếp giữa kít nuvoton 131 và esp8266 nhóm sử dung UART với baurd rate 9600.
* Kit nuvoton131.
  + Sử dụng ngắt io tại port A để nhận thông tin từ nút bấm với mỗi nút tương ướng kit nuvoton sẻ gởi 1 chuỗi lệnh đến esp8266.
  + Các PIN port A ứng với các hoạt động như sau:

|  |  |
| --- | --- |
| PIN | Hoạt động |
| PA0 | Gởi phím A |
| PA1 | Gởi phím left |
| PA2 | Gởi phím down |
| PA3 | Gởi phím up |
| PA4 | Gởi phím right |
| PA5 | Gởi phím fire |

* + Trong hàm ngắt chương trình kiểm tra cờ ngắt đẻ xác định ngắt của pin tương ứng để thực hiện lệnh tương ứng.
  + Trong BSP nuvoton131 cung cấp thư viện gpio.c dùng cấu hình cho các pin.
    - Hàm cấu hình I/O:

void GPIO\_SetMode(GPIO\_T \*port, uint32\_t u32PinMask, uint32\_t u32Mode);

//@param[in] port GPIO port. It could be PA, PB, PC, PD, PE or PF.

// @param[in] u32PinMask The single or multiple pins of specified GPIO port. \n

//@param[in] u32Mode Operation mode. It could be \n

// GPIO\_PMD\_INPUT, GPIO\_PMD\_OUTPUT, GPIO\_PMD\_OPEN\_DRAIN, GPIO\_PMD\_QUASI

* Hàm cấu hình ngắt:

void GPIO\_EnableInt(GPIO\_T \*port, uint32\_t u32Pin, uint32\_t u32IntAttribs);

//@param[in] port GPIO port. It could be PA, PB, PC, PD, PE or PF.

//@param[in] u32Pin The pin of specified GPIO port. \n

//@param[in] u32IntAttribs The interrupt attribute of specified GPIO pin. It could be \n

// GPIO\_INT\_RISING, GPIO\_INT\_FALLING, GPIO\_INT\_BOTH\_EDGE, GPIO\_INT\_HIGH, GPIO\_INT\_LOW.

* + Cấu hình ngắt IO kích cạnh xuống cho PortA.

void GPIO\_Inter(void)

{ // cau hinh IO

GPIO\_SetMode(PA,BIT0,GPIO\_PMD\_INPUT);

GPIO\_SetMode(PA,BIT1,GPIO\_PMD\_INPUT);

GPIO\_SetMode(PA,BIT2,GPIO\_PMD\_INPUT);

GPIO\_SetMode(PA,BIT3,GPIO\_PMD\_INPUT);

GPIO\_SetMode(PA,BIT4,GPIO\_PMD\_INPUT);

GPIO\_SetMode(PA,BIT5,GPIO\_PMD\_INPUT);

// cau hinh ngat

GPIO\_EnableInt(PA,0,GPIO\_INT\_FALLING);

GPIO\_EnableInt(PA,1,GPIO\_INT\_FALLING);

GPIO\_EnableInt(PA,2,GPIO\_INT\_FALLING);

GPIO\_EnableInt(PA,3,GPIO\_INT\_FALLING);

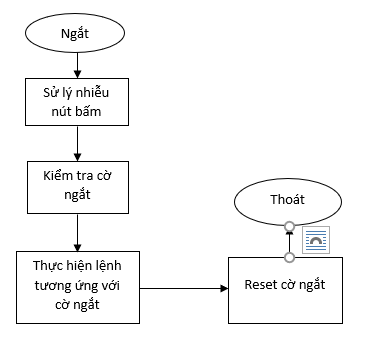
GPIO\_EnableInt(PA,4,GPIO\_INT\_FALLING);

GPIO\_EnableInt(PA,5,GPIO\_INT\_FALLING);

//

NVIC\_EnableIRQ(GPAB\_IRQn);

}

* + Lưu đồ giải thuật hàm ngắt
  + Để có thể gởi lệnh đếnh esp8266 ta gởi chuỗi lệnh thông qua UART0 của kit nuvoton131.
  + BSP nuvuton 131 củng cung cấp thư viện uart.c dùng phục vụ giao tiếp UART.

void UART0\_Init(void)

{

/\* Init UART \*/

/\* Reset UART0 module \*/

SYS\_ResetModule(UART0\_RST);

/\* Configure UART0 and set UART0 Baudrate \*/

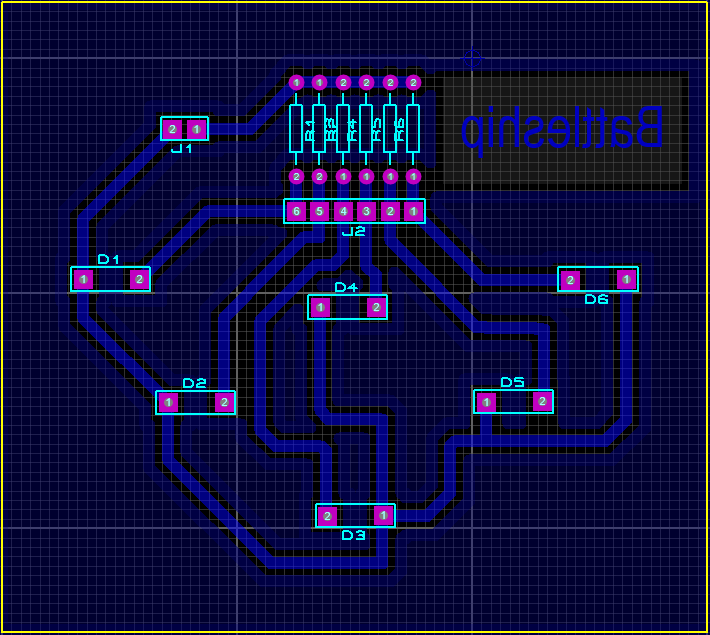
UART\_Open(UART0, 9600);

}

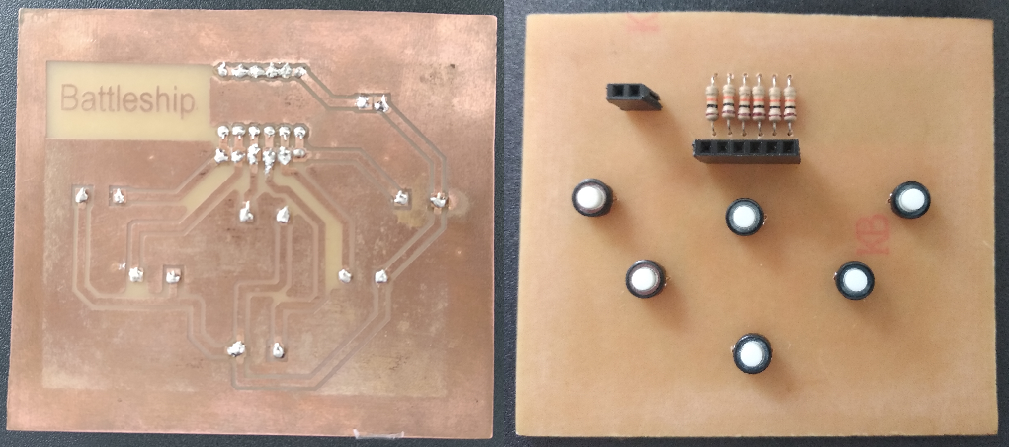
* Dùng lệnh printf(); để xuất chuỗi ra UART.
* ESP8266
  + Sử dụng thư viện SocketIOClient.h cung cấp bỏi github cùng trình Arduino IDE lập trình cho esp8266.
  + Kết nối socket đến server 13.59.254.81 port 3000.
  + Lắng nghe UART nhận lệnh từ kit nuvoton131 và emit sự kiện tương ứng đên server.

### Nút bấm

* Do cấu hình ngắt của kit nuvoton131 là ngắt io kích cạnh xuống cho nên các nút bấm trên gamepad sẻ nối điện trở kéo lên và đầu cong lại nối GND. Để tạo kích cạnh xuống khi nhấn nút.
* PCB layout (Proteus):

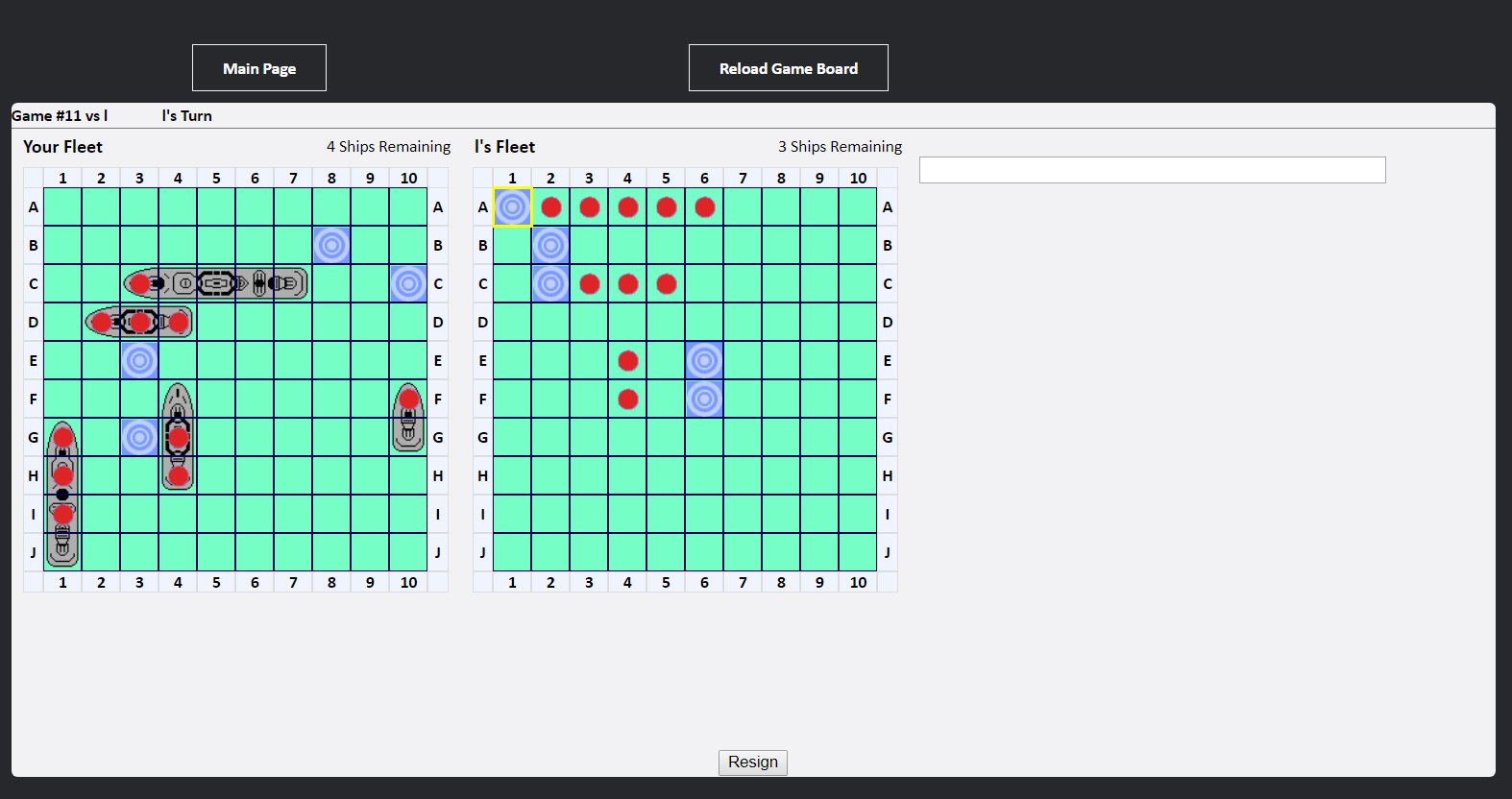


* Sản phẩm thực tế:



# KẾT QUẢ THỰC NGHIỆM

## Battleship

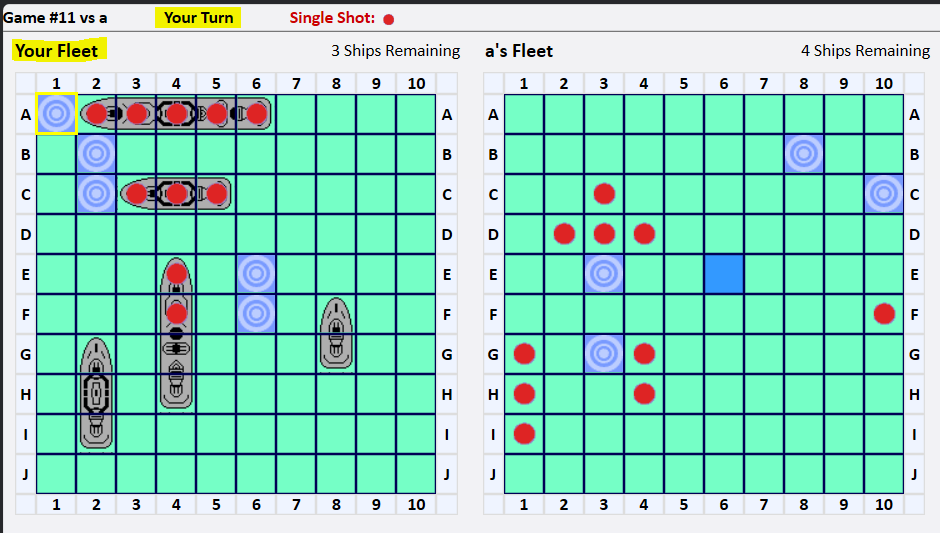


-Các tàu chiến được đặt trong bảng 10x10, có 4 loại kích thước: 2, 3, 4, 5 được random tự động.

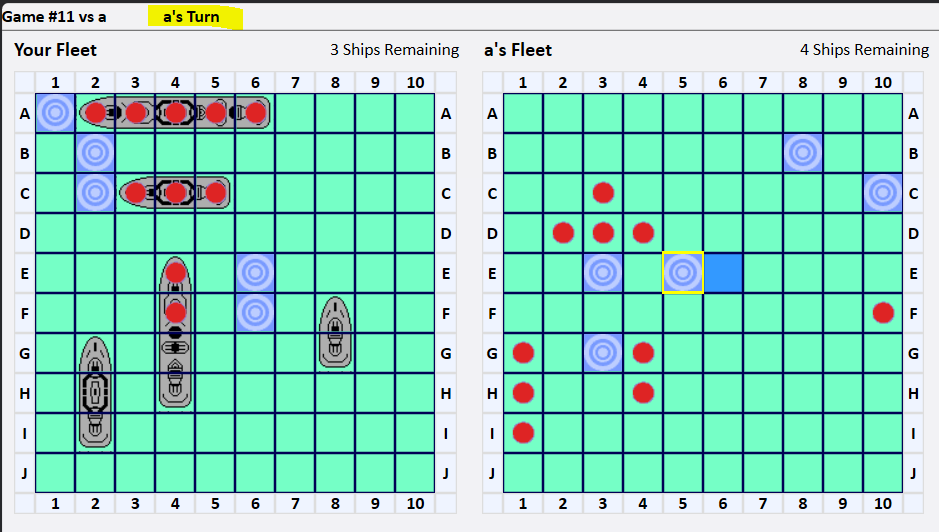
-Bảng bên trái là các chiến hạm của bên ta, bên phải là của địch.

-Title hiện id của game, lượt chơi, số tàu chiến còn lại chưa bị phá hủy của mỗi bên...

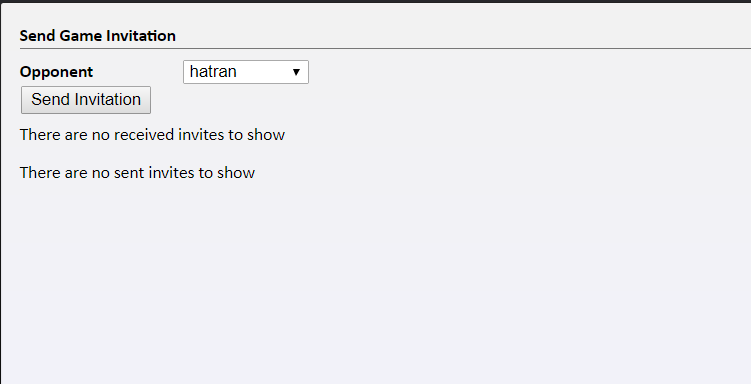
-Khi đến lượt bên ta bắn:



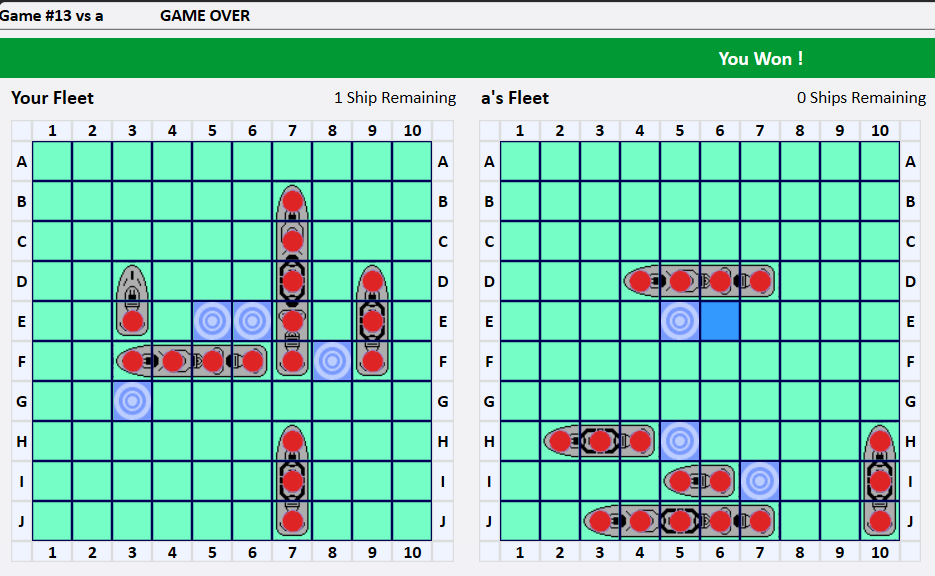
-Khi đến lượt địch bắn:



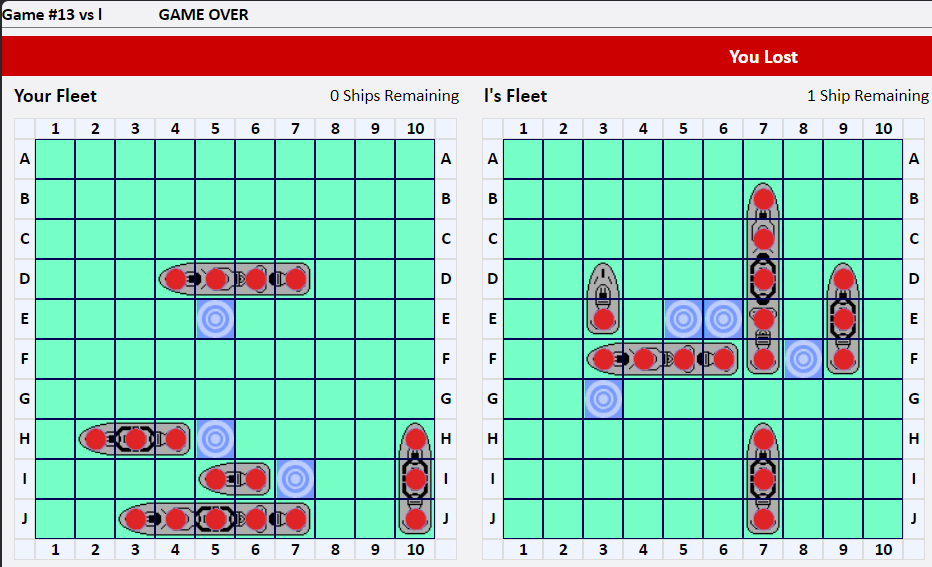
-Giao diện tạo phòng, mời người chơi:



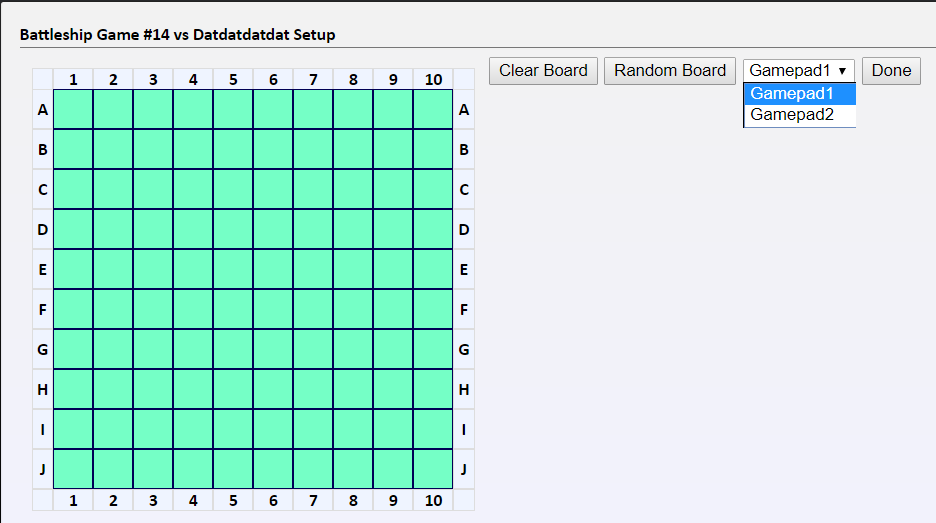
-Khi thắng:



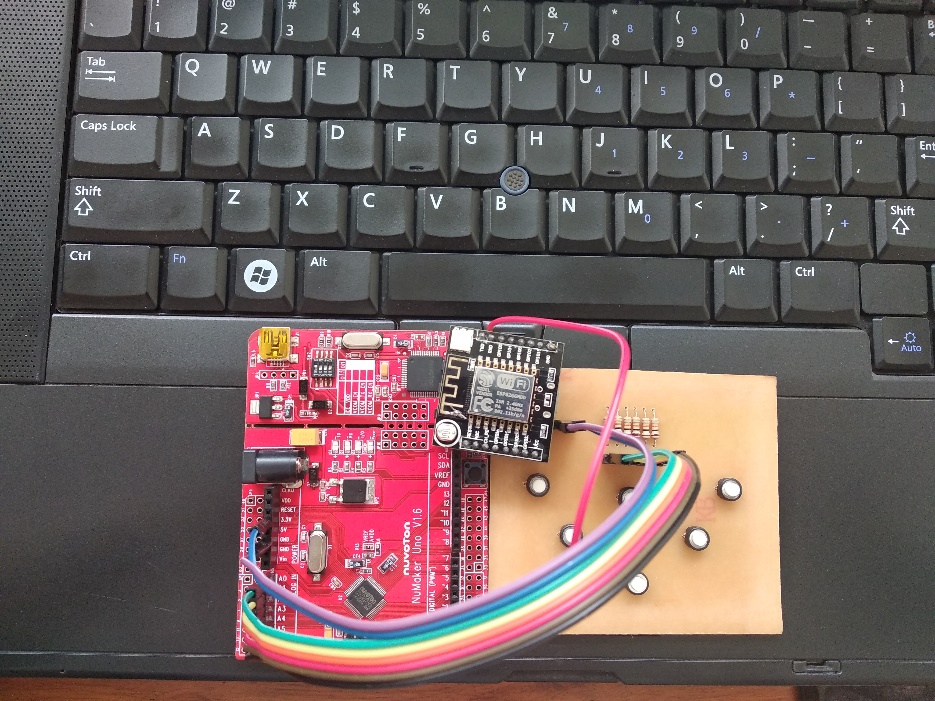
-Khi thua:

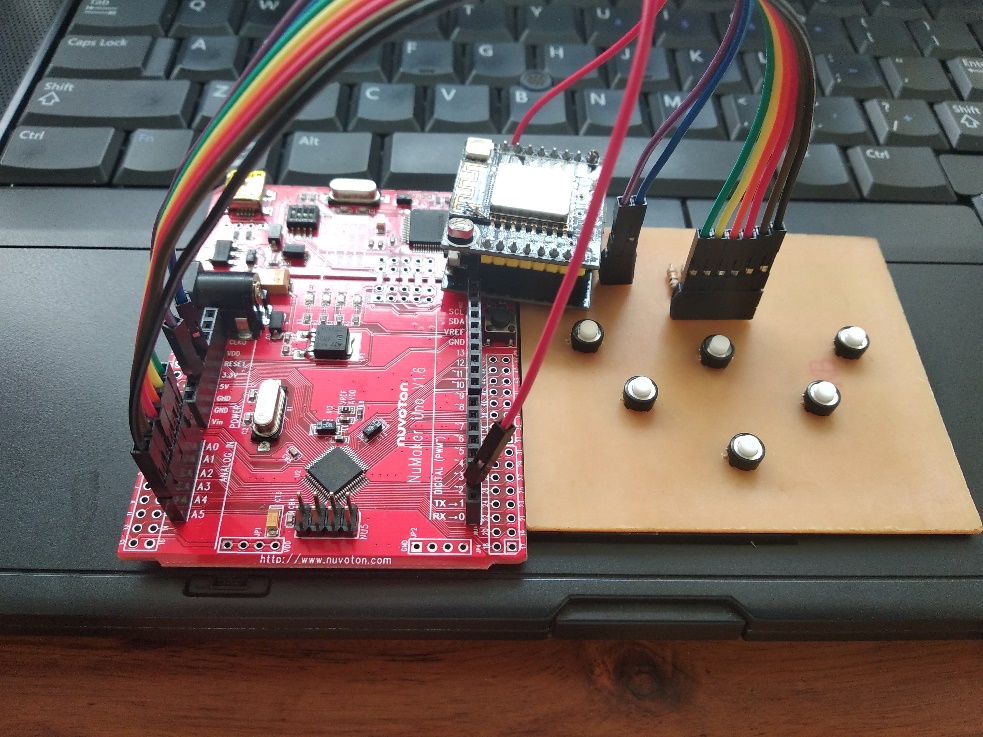


-Chọn gamepad:



## Gamepad





# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả

Đã đạt được mục tiêu đề ra ban đầu:

-Tạo ra trang web game battleship online.

-Tạo ra gamepad điều khiển chơi game thông qua wifi.

## Hướng phát triển

-Sử dụng một phiên bản khác của module ESP8266 để dễ dàng tích hợp vào gamepad, sử dụng antenna để khắc phục lỗi thường xuyên bị mất kết nối.

-Chỉnh sửa giao diện đẹp hơn để thu hút người dùng.

-Hỗ trợ nhập tọa độ để bắn từ gamepad thay vì di chuyển từng ô như hiện tại.

TÀI LIỆU THAM KHẢO

Game battleship:

<https://github.com/benjamw/battleship/>

Socket nodejs:

<https://viblo.asia/p/nodejs-va-socketio-can-ban-jlA7GKxdvKZQ>

<https://socket.io/docs/rooms-and-namespaces/>

Lập trình html và css:

<https://www.w3schools.com/>

Lập trình nuvoton 131:

<https://github.com/OpenNuvoton/NUC131BSP>