

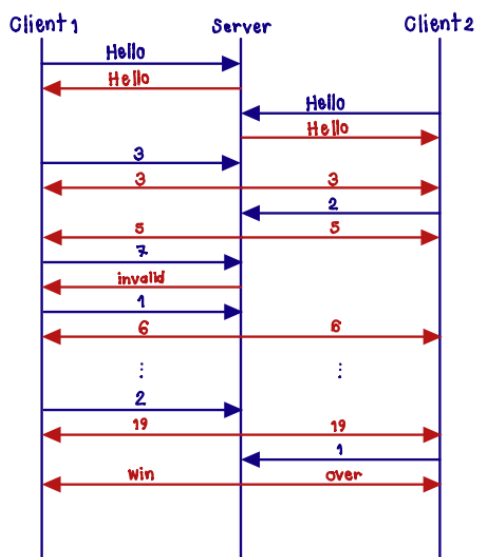
## Socket Assignment (Part 2)

### เกมนับถึง 20 แพ้!

#### กติกาเกม :

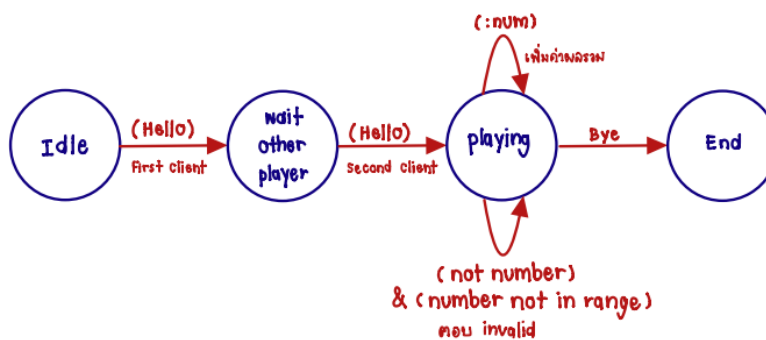
เกมจะเริ่มต้นที่ค่า 0 โดยให้ผู้เล่น 2 คนสลับกันเลือกว่าจะบวกเพิ่ม 1, 2 หรือ 3 ผู้ที่ทำให้ผลรวมมีค่ามากกว่าหรือเท่ากับ 20 จะทำให้แพ้ไป และเกมจบ

#### Sequence Diagram :



#### State Diagram :

- State Diagram ของ Server



- State Diagram ของ Client



## อธิบาย Code Server.js :

- สร้าง Server รวมถึงกำหนดค่า host และ port

```
1 const net = require('net');
2 const port = 1234;
3 const host = 'localhost';
4
5 const server = net.createServer();
6
7 server.listen(port, host, () => {
8   console.log('TCP server listening on' + host + port);
9 });
```

- สร้างตัวแปร ดังนี้

- state : เพื่อเก็บสถานะของ state
- count : เพื่อเก็บผลรวมของผลลัพธ์
- sockets : เป็น Array ไว้สำหรับสร้าง Socket ได้หลายตัว (เชื่อมต่อกับหลาย Clients ได้)

- ให้ Server เชื่อมต่อกับ Sockets และยังให้แสดง Address ของแต่ละ Socket ที่เชื่อมต่อด้วย

```
10 let state = 0;
11 let count = 0;
12 let sockets = [];
13 server.on('connection', (socket) => {
14   var clientAddress = socket.remoteAddress + ' : ' + socket.remotePort;
15   console.log('new client connected: ' + clientAddress);
16   sockets.push(socket);
17   socket.on('data', (data) => {
18     console.log('Client: ' + clientAddress + ' : ' + data);
19     sockets.forEach((sock) => {
20       sock.write(socket.remoteAddress + ':' + socket.remotePort + " said " + data + '\n');
21     });
22   });
23 });
```

- ใช้เงื่อนไขรูปแบบ Switch Case

- Case 0 มีเงื่อนไขว่า ถ้าข้อมูลที่ Client ส่งมาเท่ากับคำว่า “hello” จะให้ Server ตอบกลับไปด้วย Hello Player 1! Wait for another player! และทำการเปลี่ยนค่าของตัวแปร state = 1 เพื่อให้การทำงานของ Server ในครั้งหน้า จะเริ่มเช็คเงื่อนไขที่ Case ที่ 1
- กรณีที่ไม่เข้าเงื่อนไข(else) โดยให้ Server ส่งข้อความไปว่า INVALID : Enter "Hello"

```
23 switch (state) {
24   case 0:
25     if (data.toString().toLowerCase() == 'hello') {
26       socket.write('Hello Player 1!\n' + 'wait for another player!');
27       state = 1;
28       console.log('state=' + state);
29     }
30     else socket.write('INVALID : Enter "Hello"')
31     break;
```

- Case 1 มีเงื่อนไขว่า ถ้าข้อมูลที่ Client ส่งมาเท่ากับคำว่า “hello” อีกครั้ง จะให้ Server ตอบกลับไปด้วย Hello Player 2! Start Game! และเปลี่ยนค่าของตัวแปร state = 2
- กรณีที่ไม่เข้าเงื่อนไข(else) โดยให้ Server ส่งข้อความไปว่า INVALID

```

32     case 1:
33         if (data.toString().toLowerCase() == 'hello') {
34             socket.write('Hello Player 2!\n' + 'Start Game!')
35             state = 2;
36             console.log('state=' + state)
37         }
38         else socket.write('INVALID')
39         break;

```

- Case 2 เปลี่ยนค่าใน data ให้อยู่ในรูปแบบ int แล้วเก็บค่านั้นในตัวแปร num และเปรียบเทียบว่า ถ้า num มีค่าระหว่าง 1-3 ให้อัดค่าเพิ่มใน count แต่ถ้าไม่ใช่ จะให้ Server ตอบกลับไปว่า Invalid, please selects 1, 2 or 3
- และอีกเงื่อนไขคือ ถ้า count มีค่ามากกว่า 20 จะให้ Server ตอบกลับไปว่า Total over 20 !!! Game Over และเปลี่ยนค่าของตัวแปร state = 3

```

40     case 2:
41         try {
42             let num = parseInt(data || 0)
43
44             if (num > 0 && num <= 3) {
45                 count += num;
46                 console.log('count=' + count)
47                 socket.write('Total: ' + count);
48             }
49             else if (num < 1 || num > 3) sock.write('Invalide, please selects 1, 2 or 3');
50
51             if (count >= 20) {
52                 socket.write('\nTotal over 20 !!! \n' + 'Game Over');
53                 state = 3
54             }
55         } catch (e) {
56             socket.write('SERVER HAS EXCEPTION')
57         }
58         break;

```

- Case 3 ถ้าข้อมูลที่ Client ส่งมาเท่ากับคำว่า “bye” จะให้ Server ตอบกลับไปว่า END และจบการทำงานของ Socket นั้น

```

59     case 3:
60         if (data.toString().toLowerCase() == 'bye') {
61             socket.write('END')
62             console.log(state)
63             socket.end()
64         }
65         else socket.write('INVALID')
66         break;

```

- สร้าง event ให้กับ socket โดยประกอบด้วย closed (จบการทำงานของ socket) และ error (กรณี socket เกิดข้อผิดพลาดขึ้นมา)

```

71 socket.on('close', (data) => {
72   let index = sockets.findIndex((o) => {
73     return o.remoteAddress === socket.remoteAddress && o.remotePort === socket.remotePort;
74   })
75   if (index !== -1) sockets.splice(index, 1);
76   sockets.forEach((sock) => {
77     sock.write(`${clientAddress} disconnected\n`);
78   });
79   console.log(`connection closed: ${clientAddress}`);
80 });
81
82 socket.on('error', (err) => {
83   console.log(`Error occurred in ${clientAddress}: ${err.message}`);
84 });

```

### อธิบาย Code Client.js :

- มีการขอเรียกใช้ readline เพื่อใช้ในการรับค่าจาก Standard Input และแสดงผลบน Standard Output ได้

```

3 const readline = require('readline').createInterface({
4   input: process.stdin,
5   output: process.stdout
6 });

```

- Client เชื่อมกับ Server โดยจะแสดงชื่อ Host และ Port ใน console

```

13 client.connect(PORT, HOST, function () {
14   console.log('Client connected to: ' + HOST + ':' + PORT);
15   readline.question('\n', input => {
16     client.write(input);
17   })
18 });

```

- เมื่อมี data เข้ามา จะให้แสดงข้อมูลใน data ที่ console ของ Client
- สร้าง event โดยประกอบด้วย close (จบการทำงานของ Client) และ error (กรณี Client เกิดข้อผิดพลาดขึ้นมา)

```

26 client.on('data', function (data) {
27   console.log('Client received: ' + data);
28 });
29
30 client.on('close', function () {
31   console.log('Client closed');
32 });
33
34 client.on('error', function (err) {
35   console.error(err);
36 });

```

### ผลลัพธ์การทำงาน :

เรียงลำดับดังนี้ Server, Client 1 และ Client 2

- Clients ทั้ง 2 เชื่อมต่อกับ Server เรียบร้อย สังเกตจาก new client connected
  - Client 1 = 127.0.0.1 : 53789
  - Client 2 = 127.0.0.1 : 53791

PS C:\Users\asus\Desktop\3term2\311 distributed\03> node server.js TCP server listening onlocalhost1234 new client connected: 127.0.0.1 : 53789 new client connected: 127.0.0.1 : 53791 █	PS C:\Users\asus\Desktop\3term2\311 distributed\03> node client.js Client connected to: localhost:1234 █	PS C:\Users\asus\Desktop\3term2\311 distributed\03> node client.js Client connected to: localhost:1234 █
---	--	--

- Client 1 : Hello
- Server : Hello Player 1! Wait for another player!
- Client 2 : Hello
- Server : Hello Player 2! Start Game!

PS C:\Users\asus\Desktop\3term2\311 distributed\03> node server.js TCP server listening onlocalhost1234 new client connected: 127.0.0.1 : 53789 new client connected: 127.0.0.1 : 53791 Client: 127.0.0.1 : 53789 : Hello state=1 Client: 127.0.0.1 : 53791 : Hello state=2 █	PS C:\Users\asus\Desktop\3term2\311 distributed\03> node client.js Client connected to: localhost:1234  Hello  Client received: 127.0.0.1:53789 said Hello  Client received: Hello Player 1!  Client received: 127.0.0.1:53791 said Hello █	PS C:\Users\asus\Desktop\3term2\311 distributed\03> node client.js Client connected to: localhost:1234  Client received: 127.0.0.1:53789 said Hello Hello  Client received: 127.0.0.1:53791 said Hello Hello Client received: Hello Player 2! Start Game! █
---	---	---

- Client 1 : 1 (Client 2 จะเห็นข้อความนี้ด้วย)
- Server : Total = 1 (เนื่องด้วยมีการพิมพ์ตอบกลับทำให้บรรทัดนี้มองไม่เห็น)
- Client 2 : 2 (Client 1 จะเห็นข้อความนี้ด้วย)
- Server : Total = 3
- ...
- Client 2 : 3
- Server : Total = 21 Total over 20 !!! Game Over

<pre> PS C:\Users\asus\Desktop\3term2\311 dist ributed\03&gt; node server.js TCP server listening onlocalhost1234 new client connected: 127.0.0.1 : 53789 new client connected: 127.0.0.1 : 53791 Client: 127.0.0.1 : 53789 : Hello state=1 Client: 127.0.0.1 : 53791 : Hello state=2 Client: 127.0.0.1 : 53789 : 1 count=1 Client: 127.0.0.1 : 53791 : 2 count=3 Client: 127.0.0.1 : 53789 : 3 count=6 Client: 127.0.0.1 : 53791 : 3 count=9 Client: 127.0.0.1 : 53789 : 3 count=12 Client: 127.0.0.1 : 53791 : 3 count=15 Client: 127.0.0.1 : 53789 : 3 count=18 Client: 127.0.0.1 : 53791 : 3 count=21 </pre>	<pre> 1 Client received: 127.0.0.1:53789 said 1  Client received: 127.0.0.1:53791 said 2  3 Client received: 127.0.0.1:53789 said 3  Client received: 127.0.0.1:53791 said 3  3 Client received: 127.0.0.1:53789 said 3  3 Client received: 127.0.0.1:53791 said 3  3 Client received: 127.0.0.1:53789 said 3  3 Client received: 127.0.0.1:53791 said 3  Client received: 127.0.0.1:53789 said 3 </pre>	<pre> 2 Client received: 127.0.0.1:53791 said 2  Client received: 127.0.0.1:53789 said 3  3 Client received: 127.0.0.1:53791 said 3  Client received: 127.0.0.1:53789 said 3  3 Client received: 127.0.0.1:53791 said 3  Client received: 127.0.0.1:53789 said 3  3 Client received: 127.0.0.1:53791 said 3  Client received: Total: 21 Total over 20 !!! Game Over </pre>
--	--	--

- **Client 1** : bye (หยุดการเชื่อมต่อกับ Server ของ Client1 ซึ่ง Client2 เห็นการทำงานนี้ และมีข้อความว่า Client 1 disconnected ไปแล้ว)
- **Server** : END
- **Client 2** : bye (หยุดการเชื่อมต่อกับ Server ของ Client2)
- **Server** : END

<pre> Client: 127.0.0.1 : 61192 : 2 count=17 Client: 127.0.0.1 : 61193 : 2 count=19 Client: 127.0.0.1 : 61192 : 1 count=20 Client: 127.0.0.1 : 61192 : bye 3 connection closed: 127.0.0.1 : 61192 Client: 127.0.0.1 : 61193 : bye 3 connection closed: 127.0.0.1 : 61193 </pre>	<pre> Client received: 127.0.0.1:61192 said 1  Client received: Total: 20 Total over 20 !!! Game Over bye  Client received: 127.0.0.1:61192 said bye  Client received: END Client closed </pre>	<pre> Client received: 127.0.0.1:61192 said 1  Client received: 127.0.0.1:61192 said bye  Client received: 127.0.0.1 : 61192 disconnected bye  Client received: 127.0.0.1:61193 said bye  Client received: END Client closed </pre>
---	---	---

### สมาชิกในกลุ่ม

- |               |         |            |
|---------------|---------|------------|
| 1. นายภาสกร   | บัวแก้ว | 6210110266 |
| 2. นางสาวเมษา | เจริญไ  | 6210110291 |
| 3. นายวสุพจน์ | พองมณี  | 6210110322 |