นายเจนกิจ ประกิจถาวรกุล 6210110051
นายสมาบัติ ชำนาญเวทย์ 6210110356

Socket Assignment - PART 2

เกมเป่ายิ้งฉุบ

กติกาเกมเป่ายิ้งฉุบ :

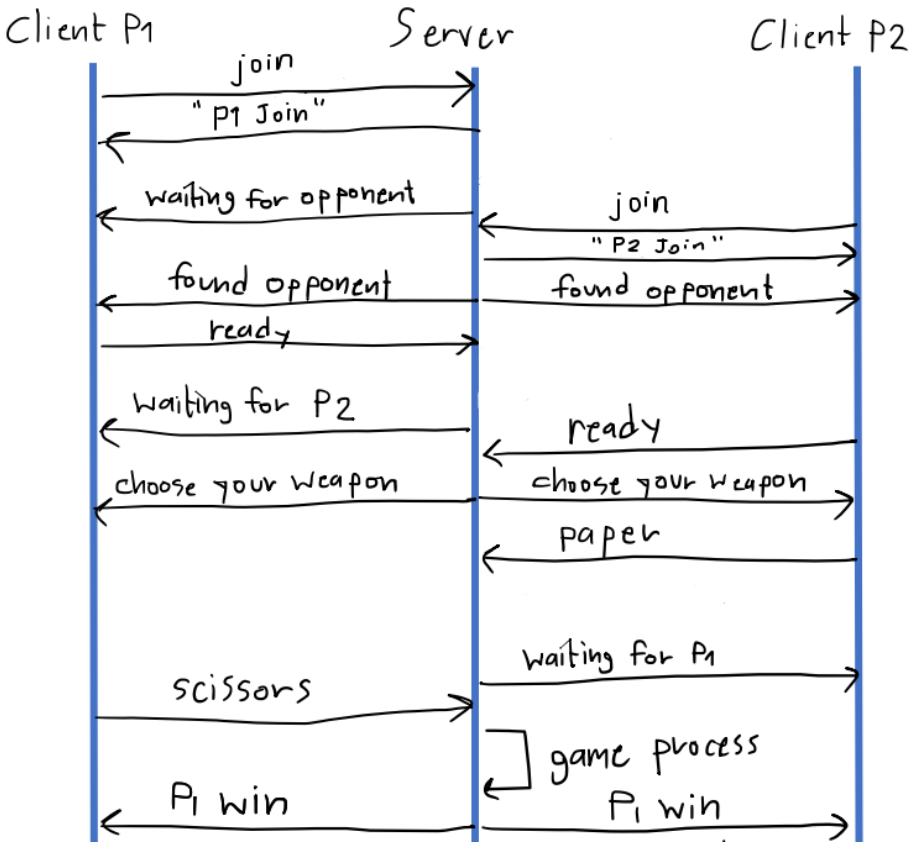
เกมเป่ายิ้งฉุบจะมีผู้เล่นทั้งหมดสองคน โดยผู้เล่นแต่ละคนจะต้องเลือกใช้ Rock, Paper หรือ Scissor หลังนั้นจึงนำมาเปรียบเทียบกันโดนจะมีกติกาดังนี้

Rock ชนะ Scissor Rock แพ้ Paper Rock เสมอ Rock

Paper ชนะ Rock Paper แพ้ Scissor Paper เสมอ Paper

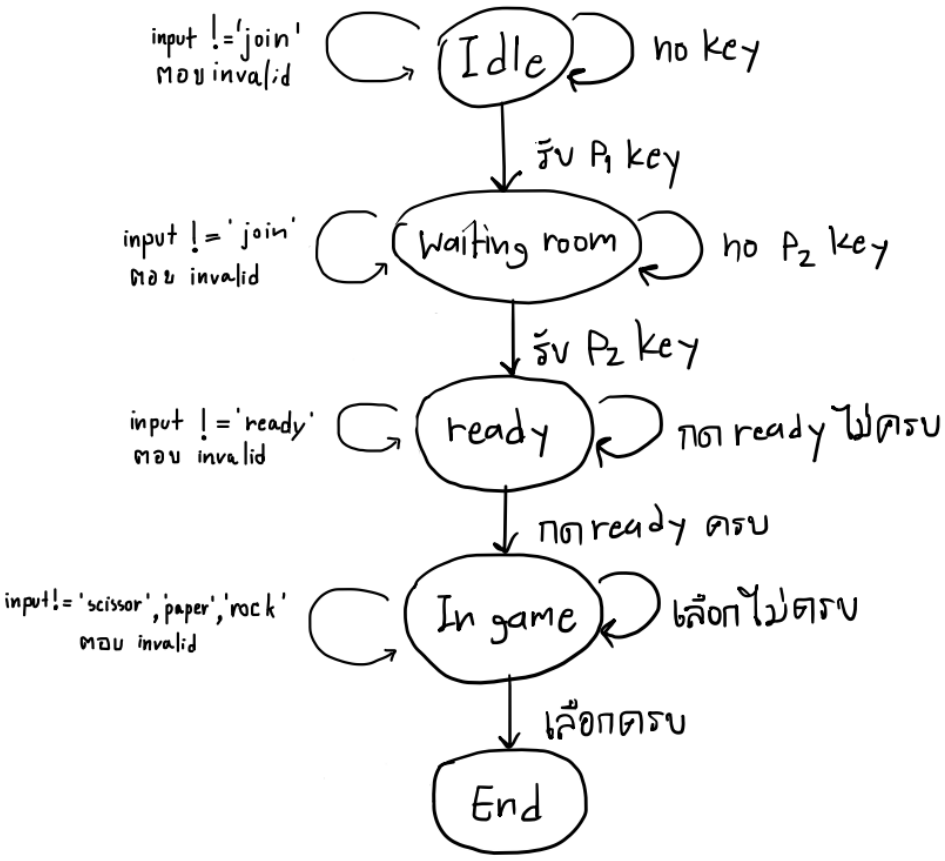
Scissor ชนะ Paper Scissor แพ้ Rock Scissor เสมอ Scissor

Sequence Diagram



นายเจนกิจ ประกิจถาวรกุล 6210110051
นายสมาบัติ ชำนาญเวทย์ 6210110356

State Diagram



Server.js

```
const net = require('net')

var players = []
var gameState = 0

net.createServer((socket) => {
```

```
}).listen(6969, '127.0.0.1')
console.log('Server listening on 127.0.0.1:6969')
```

เริ่มจากทำการสร้าง array players สำหรับเก็บข้อมูลของแต่ละ client และสร้างตัวแปร gameState สำหรับการแบ่ง state ของเกม จากนั้นจะทำการสร้าง server ขึ้นมาโดยกำหนด port = 6969

```
socket.on('data', (data) => {
    var msg = data.toString()
    var player = findPlayer(socket.remoteAddress, socket.remotePort)
```

จากนั้น server จะทำการเชื่อมต่อกับ socket เพื่อรับข้อมูลของแต่ละ client โดยข้อมูลที่รับเข้ามาจะถูกแปลงเป็น string และกำหนดให้ = msg และทำการเช็คหมายเลข address และหมายเลข port ของ client เพื่อเช็คหาผู้เล่น

```
if(gameState == 0)
{
    if(msg == 'join' && player == null && players.length == 0)
    {
        players.push({
            addr: socket.remoteAddress,
            port: socket.remotePort,
            name: `P1`,
            attack: null,
            client: socket
        })
        socket.write('P1(You) have joined a server. \ntype <ready> When you are ready.')

        console.log(`P1 joined.`)
        return
    }

    if(msg == 'join' && player == null && players.length == 1)
    {
        players.push({
            addr: socket.remoteAddress,
            port: socket.remotePort,
            name: `P2`,
            attack: null,
            client: socket
        })
        socket.write('P2(You) have joined a server. \ntype <ready> When you are ready.')

        console.log(`P2 joined.`)
        return
    }
```

ในส่วนของ gameState = 0 หรือ state ที่รอผู้เล่น join เข้าเกม

โดยจะทำการแบ่งเงื่อนไขเป็น 2 กรณี

1. กรณีที่ยังไม่มีผู้เล่นเข้าร่วม เมื่อมีผู้เล่นคนแรกพิมพ์ข้อความ join โปรแกรมจะทำการ push ค่า address,port และกำหนดให้เป็น P1
2. กรณีที่มีผู้เล่นคนแรกเข้ามาแล้ว เมื่อผู้เล่นคนที่สองพิมพ์ข้อความ ‘join’ โปรแกรมจะทำการ push ค่า address,port และจะกำหนดให้เป็น P2

```
    if(msg == 'ready' && player != null)     // ready
    {
        player['ready'] = 'ready'
        console.log(`${player.name}(${player.addr}:${player.port}) is ready.`)
        if(getReadyPlayer() == 2)
        {
            console.log('All players are ready.')
            gameState = 1

            announce('Type rock, paper or scissor')
        }
        return
    }

    else{
        socket.write('Invalid')
    }
}
```

```
function getReadyPlayer()
{
    var num = 0
    for(var player of players)
    {
        if(player['ready'] == 'ready') num += 1
    }
    return num
}
```

```
function announce(msg)
{
    for(var player of players)
    {
        player.client.write(msg)
    }
}
```

ถัดจากนั้น เมื่อผู้เล่นทำการ join เข้ามาแล้วเมื่อผู้เล่นพิมพ์ว่า ready จะทำการกำหนดให้ค่าในตัวแปรของ player เป็น ready เพื่อที่จะนำมาใช้ในฟังก์ชั่น getReadyPlayer() ที่จะตรวจสอบว่าผู้เล่น ready ทั้งหมดหรือไม่ ถ้า ready ครบทั้งสองคนแล้ว gameState จะถูกเซ็ตเป็น 1 และจะเข้าฟังก์ชั่น announce() ที่จะทำการส่งข้อความไปยัง Client ทุกคน

```
else{
    socket.write('Invalid')
}
```

ถ้าใส่ค่า input นอกเหนือจากในเงื่อนไขข้างบน Server จะส่งข้อความ ‘Invalid’ หา Client

นายเจนกิจ ประกิจถาวรกุล 6210110051
นายสมาบัติ ชำนาญเวทย์ 6210110356

```
    if(gameState == 1)
    {
        if(msg == 'rock' || msg == 'paper' || msg == 'scissor'){
            player['attack'] = msg
            if(getAnsPlayer() == 2){
                console.log('Calulating')
                fight(players[0],players[1])
            }
        }

        else{
            socket.write('invalid')
        }
    }
})

socket.on('close', function(){
    console.log(`${socket.remoteAddress}:${socket.remotePort} disconnected.`)
})

socket.on('error', function(err){

})
```

```
function getAnsPlayer()
{
    var num = 0
    for(var player of players)
    {
        if(player['attack'] == 'rock' || player['attack'] == 'paper' || player['attack'] == 'scissor')
            num += 1
    }
    return num
}
```

ในส่วนของ gameState = 1 หรือ state เริ่มเกม

เมื่อผู้เล่นทำการพิมพ์ตัวเลือกเรียบร้อยแล้วจะถูกนำค่าไปเก็บใน player ของแต่ละคน จากนั้นจะใช้ฟังก์ชั่น getAnsPlayer เพื่อที่จะเช็คว่าผู้เล่นทั้งคู่เลือกครบแล้วหรือไม่ (หลักการเดียวกับ ฟังก์ชั่น getReadyPlayer())

หลังจากผู้เล่นทั้งคู่เลือกครบแล้วจะรันฟังก์ชั่น fight() ซึ่งพารามิเตอร์ประกอบไปด้วย players[0] (P1) และ players[0] (P2)

นายเจนกิจ ประกิจถาวรกุล 6210110051
นายสมาบัติ ชำนาญเวทย์ 6210110356

```
function fight(P1,P2){
    if(P1['attack'] == 'rock' && P2['attack'] == 'paper'){
        announce('P2 win.')
    }
    else if(P1['attack'] == 'rock' && P2['attack'] == 'scissor'){
        announce('P1 win')
    }
    else if(P1['attack'] == 'paper' && P2['attack'] == 'rock'){
        announce('P1 win')
    }
    else if(P1['attack'] == 'paper' && P2['attack'] == 'scissor'){
        announce('P2 win')
    }
    else if(P1['attack'] == 'scissor' && P2['attack'] == 'paper'){
        announce('P1 win')
    }
    else if(P1['attack'] == 'scissor' && P2['attack'] == 'rock'){
        announce('P2 win')
    }
    else if(P1['attack'] == P2['attack']){
        announce('DRAW.')
    }
}
```

ในฟังก์ชั่น fight() จะทำการเช็คเงื่อนไขระหว่าง P1[‘attack’] กับ P2[‘attack’] ที่รับมาก่อนหน้าโดยใช้ logic ตามกฎของเกมเป่ายิงฉุบที่ได้อธิบายไปข้างต้น โดยจะมีการแจ้งข้อความแพ้ชนะให้กับผู้เล่นทั้ง 2 ผ่านฟังก์ชั่น announce()

นายเจนกิจ ประกิจถาวรกุล 6210110051
นายสมาบัติ ชำนาญเวทย์ 6210110356

Client.js

```
const readline = require('readline').createInterface({
    input: process.stdin,
    output: process.stdout
});
```

- เรียกใช้ readline เพื่อใช้รับค่า Input จาก stdin และแสดง Output ผ่าน stdout

```
client.connect(PORT, HOST, function(){
    console.log('CONNECTED TO: ' + HOST + ':' + PORT);
    readline.question('\n', input => {
        client.write(input);
    })
});
```

- แสดงข้อมูลของ HOST และ PORT จากนั้นจึงรับค่า Input แล้วส่งข้อความไปยัง Server

```
client.on('data', function(data){
    readline.question(data + '\n', input => {
        client.write(input);
    })
});
```

- รับค่า Input แล้วส่งข้อความไปยัง Server หลังจาก client.connect