

PING ME

네트워크 프로그래밍 프로젝트

이름	학번
서진교	20122394
이성화	20122430
한상훈	20122499

목차

1. 프로젝트 개요

2. 역할분담

3. 구성

4. 핵심코드 설명

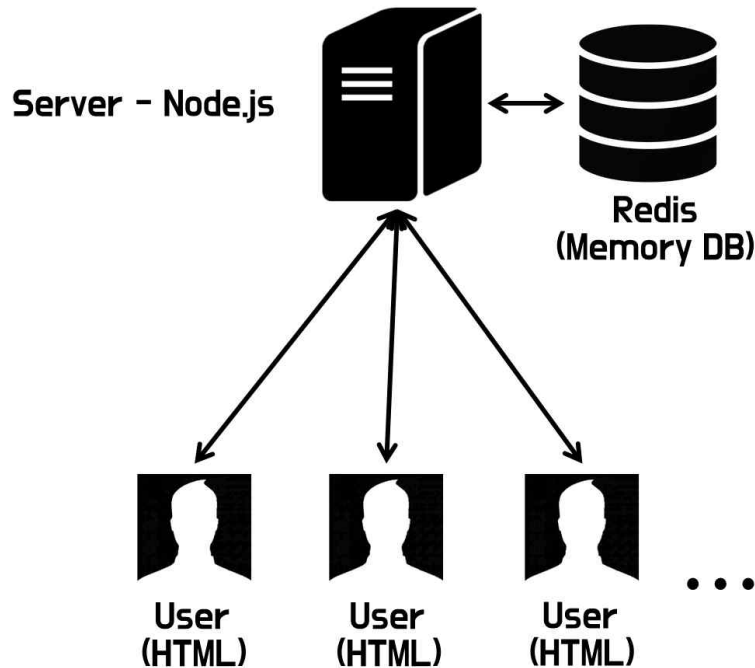
5. 기능 설명

6. 결론

1. 프로젝트 개요

우리 팀은 네트워크 프로그래밍 수업을 들으면서 실제로 네트워크가 유익하게 쓰일 방법이 없을까 고민했다. 그래서 실제로 사용할 수 있는 어플리케이션을 만들면 좋지 않을까 생각하게 되었고 구현하는 방법으로는 ‘웹’을 선택하였다. 웹을 선택한 이유는 팀원들이 모두 웹에 관심이 있었고 또 웹으로 만들면 시연하기도 쉬울뿐더러 항상 켜놓고 사용할 수 있기 때문이다. 그렇다면 실제로 사용할 수 있는 어플리케이션으로 무엇이 있을까 고민하던 중 우리가 평소에 가장 많이 부딪치는 팀 프로젝트에서 영감을 얻었다. 팀 프로젝트를 하면서 서로의 코드를 많이 볼 일이 생기는데 그럴 때 카카오톡으로 공유 한다던지 구글 독스를 이용한다던지 하는 불편함이 있었다. 그래서 우리 조는 ‘실시간 다중 알람 코드 편집기’를 주제로 정하여 이런 팀 프로젝트를 진행할 때의 불편함을 해결하려고 한다. 또한 기존의 실시간 코드 편집기와 차별화 되는 점은 바로 자신의 의사를 실시간으로 표시할 수 있다는 점이다. 우리 팀은 게임에서 사용하는 ‘핑’이라는 기능을 차용해 효과적으로 의사를 전달하는 기능을 추가하기로 하였다. 핑은 마우스를 드래그 하거나 특정 버튼을 눌러서 소리와 알람 표시가 함께 나타나는 기능이다. 프로젝트 명은 지금 내가 하고 있는 부분을 핑한다는 뜻에서 ‘PING ME’ 라고 정하였다.

시나리오는 <그림1>에서 볼 수 있듯이 여러 사용자가 웹 브라우저를 통해 서버에 접속하고 코드를 편집하고 핑을 송수신 할 수 있다. 입력한 코드들은 redis를 통해 저장이 된다.



<그림1 : 개요도>

Code Editor	Realtime	Multiple	Ping
Node.js	Socket.io	Javascript	Library

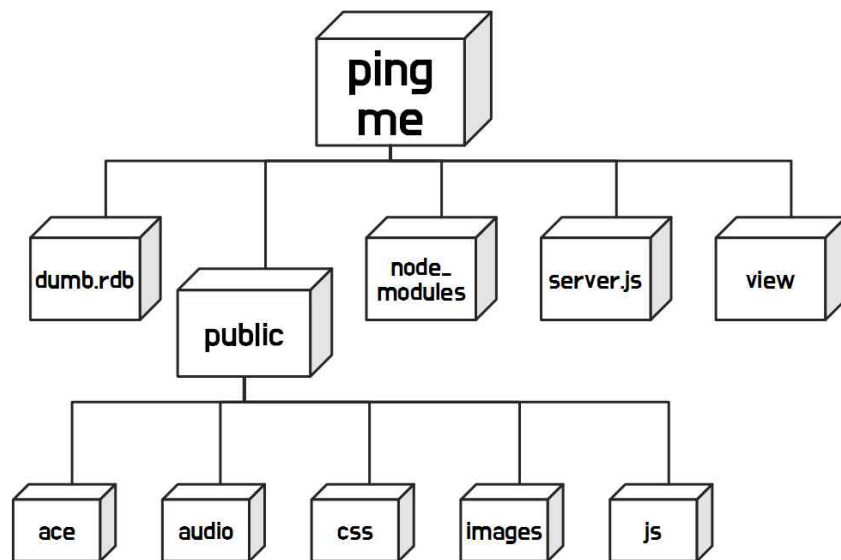
〈표1 : 키워드〉

2. 역할분담

서진교	이성화	한상훈
기획, 디자인, UI, 프로젝트 일정관리	Client Programming	Server Programming

〈표2 : 역할분담〉

3. 구성



〈그림2 : 프로젝트 구성도〉

프로젝트의 구성은 〈그림2〉와 같다. 최상위 디렉토리인 pingme디렉토리 밑에 dumb.rdb파일은 redis가 데이터를 저장하는 곳이다. node_modules디렉토리는 node.js에서 사용하는 모듈들을 모아둔 곳이다. public디렉토리는 직접 만들어 사용한 자원들이 들어있는 곳으로 음성파일, css 파일, js파일, image파일들이 담겨있다. view디렉토리에는 서버에서 사용하는 view파일들이 들

어있는데 view 엔진으로 ejs를 사용하기 때문에 ejs확장자파일이 담겨있다. server.js는 실제로 node.js에서 돌아가는 server파일로 routing, 모듈 정의 및 사용 event처리를 담당한다.

4. 핵심코드 설명

● 사용 모듈 및 라우팅

```
var express = require('express');
var app = express();
var events = require('events');
var socketio = require('socket.io');

// set the view engine to ejs
app.set('view engine', 'ejs');

// public folder to store assets
app.use(express.static(__dirname + '/public'));

// routes for app
app.get('/', function(req, res) {
  res.render('home');
});
app.get('/:id', function(req, res) {
  res.render('pad');
});
```

server.js에서 핵심역할을 하는 모듈로 express, events, socket.io을 사용하였다. 각각 라우팅, 이벤트제어, 소켓통신의 용도로 사용되었다. express모듈을 사용하여 id값(url의 경로 이름)을 바탕으로 하여 에디터를 룸별로 생성하고, 룸별로 통신 할 수 있게 하였다.

● 실시간 텍스트 편집

```
// get sharejs dependencies
var sharejs = require('share');
require('redis');

// options for sharejs
var options = {
  db: {type: 'redis'},
};

// attach the express server to sharejs
sharejs.server.attach(app, options);
```

server.js에서 share모듈을 사용하여 option을 redis옵션을 주어 redis DB를 사용하여 실시간으로 텍스트 편집이 가능하도록 하였다.

● Ping 기능 socket 통신

```
var socket = io.connect();
socket.on('connect', function(data){
    socket.emit('join',"no"+document.location.pathname.substring(1))

    socket.on('ping',function(data){
        triggerping(data.xPos,data.yPos,data.number);
    })
})
```

script.js안의 코드이다. client에서 작동하는 코드로 서버에서 작동하고 있는 socket을 이용하여 소켓과 connect이벤트가 발생하면 join이벤트를 발생시키고 ping이벤트를 받으면 triggerping 함수를 호출한다.

```
function callEmit(x,y,flag){
    socket.emit('ping',{
        xPos : x,
        yPos : y,
        number : flag,
    });
}
```

script.js안의 코드이다. ping.js에서 move함수가 작동하면 최종적으로 위의 함수를 호출하게 되는데 단순히 socket서버에 xPos, yPos, number값을 가지면서 ping이라는 이름을 가진 이벤트를 발생시키는 역할을 한다.

● 핑의 작동

```
<div id="main" draggable="true" ondragend="move(event)">
```

웹 페이지에서 우리가 핑이 작동하고자 하는 부분에 draggable 속성을 true로 만든다. 이렇게 하면 이 태그 안에서 드래그하는 움직임을 감지할 수 있다. 그리고 ondragend는 드래그가 끝났을 때를 뜻하는데, 드래그가 끝났을 때의 방향으로 핑의 종류가 결정되므로 드래그가 끝났을 때 move()를 호출한다.

● 핑의 구분

```
if((startX - endX) == 0){
    if(startY < endY){
        flag = 3;
    }else{
        flag = 1;
    }
}else{
    slide = (startY - endY) / (startX - endX);
    if(slide > -0.5 && startX > endX){
        flag = 1;
    }else if(slide < 0.5 && startX < endX){
        flag = 2;
    }else if((slide < -0.5 || slide > 0.5) && startY < endY){
        flag = 3;
    }
}
```

핑을 드래그해서 움직임이 끝났을 때 move() 안에서 핑의 위치를 구분하는 부분이다. 우리 팀의 핑은 총 3가지 종류로 구성되는데, 이것을 위해 화면을 120도 씩 3부분으로 나누어서 핑의 방향을 판단한다. flag 값에 따라서 핑의 종류가 결정된다. slide는 기울기를 결정하는 변수인데, 시작하는 점과 끝나는 점을 이용해서 각도를 판정한다.

● 핑의 방향 확인

```
ev = ev || window.event;
var x = ev.pageX,
    y = ev.pageY;
endX = x;
endY = y;
```

실시간으로 드래그 되고 있는 점을 x,y,변수에 저장하는 부분이다.

```
var draw = document.getElementById("pingcursor");
if(flag == 1){
    draw.style.backgroundImage="url('ping1.png')";
}else if(flag == 2){
    draw.style.backgroundImage="url('ping2.png')";
}else if(flag == 3){
    draw.style.backgroundImage="url('ping3.png')";
}
draw.style.zIndex = 100;
draw.style.display = "inline-block";
draw.style.position = "absolute";
draw.style.top = (startY-75)+"px";
draw.style.left = (startX-75)+"px";
```

드래그를 하는 중 내가 어떤 핑을 선택하고 싶은지 확인할 방법이 필요하므로 실시간으로 내가 선택하고 있는 핑을 보여주어야 한다. 실시간으로 위의 핑을 구분하는 function을 통해 핑의 방향을 확인한 다음 실시간으로 backgroundImage를 수정하여 내가 현재 어떤 핑을 선택하고 있는지 나타낸다.

● 핑 소리 재생

```
if (flag == 1) {
    ping.style.top = (startY-20)+"px";
    ping.style.left = (startX-35)+"px";
    myAudio.play();
    color = "url('finger_icon.png')";
} else if (flag == 2) {
    myAudio2.play();
    ping.style.top = (startY-85)+"px";
    ping.style.left = (startX-25)+"px";
    color = "url('flag_icon.png')";
} else if(flag == 3){
    myAudio3.play();
    ping.style.top = (startY-60)+"px";
    ping.style.left = (startX-50)+"px";
    color = "url('exclamation_icon.png')";
}
```

이전에 flag의 값을 구해놓았으므로 이 flag값에 따라 미리 로딩해 놓은 핑의 소리를 재생한다. myAudio1,2,3은 각각의 소리를 미리 로딩해 놓은 변수이다.

- 핑을 서버에 전송해 다른 사용자에게 Broadcasting

서버에 json 형식으로 핑의 x,y좌표 그리고 ping의 종류를 전송

5. 기능 설명

- Room 생성 및 입장

웹 브라우저에서 x.x.x.x:8000/{room_name}으로 접속을 하면 room_name을 이름으로 하는 Room이 생성되며 그 안에서 socket통신으로 실시간으로 서로의 코드를 작성할 수 있고 핑을 주고 받을 수 있다.

- Syntax 모드, theme 설정



각 편집기마다 Syntax모드와 theme를 기호에 따라 select태그에서 선택하여 설정할 수 있다.

- 핑의 작동



드래그를 하면 다음과 같은 드래그 UI가 드래그를 시작한 지점에 나타난다. 마우스를 해당 방향으로 가리키면 다음과 같은 3가지 방향의 드래그 아이콘이 생성된다.



다음과 같은 아이콘이 내가 핑을 시작한 위치에 표시되고 모든 사용자들에게 broadcast된다. 첫 번째 아이콘은 ‘여기야!’ : 여기를 봐주세요 라는 의미가 있다. 그리고 ‘도와주세요!’ : 코딩하다 어떤 부분을 모르겠다 싶을 때 쓰는 기능이다. 그리고 ‘응 아니야!’ 는 그 부분이 잘못되었다는 뜻이다.

6. 결론

처음 프로젝트를 시작하려고 할 때는 사실 어느 것부터 구현해야 할지 전혀 감이 잡히지 않았다. 그렇지만 node.js를 이용해 서버를 구현하는 예제를 따라하다 보니 조금씩 감이 왔고 수업 시간에 배운 Socket통신을 node.js가 훨씬 더 쉽게 구현해주고 있다는 사실을 알게 되었다. 우리는 share.js라는 모듈과 Ace Editor를 사용하였다. 이 두 모듈을 결합하는 과정에서도 상당히 힘들었다.

다음으로 구현해야 할 문제는 실시간으로 사용자가 텍스트를 입력할 때 그것을 어디에 저장해야 하는가 하는 것이었다. 이것은 Redis를 통해 세션에 저장하는 방법을 사용하면 직접 서버에 DB를 구축하지 않고도 값을 유지할 수 있다는 것을 알게되었다. 클라이언트는 html, 자바스크립트와 CSS를 활용해서 구현하였다. 자바스크립트를 활용해 사용자가 마우스를 드래그 하였다는 것을 인지하고 이것을 서버에 JSON 형식으로 핑의 X,Y좌표 그리고 ping의 종류를 전송한다. 이것을 서버는 받아서 다시 다른 사용자들에게 broadcasting해 준다. 이 부분에서 시간이 오래 걸렸는데, 기본적인 node.js 서버에서는 역시 구현되어있지 않는 부분이라서 이 부분을 구현하였다.

프로젝트를 시작하기 전에는 완성할 수 있을까? 라는 의문도 들었다. 처음 시연할 때 동영상까지 보여주면서 설명하다 보니까 그 정도로 구현을 해야 한다는 압박감도 있었다. 그렇지만 작은 기능부터 하나씩 구현해 나가다 보니까 완성을 할 수 있었다. 구현을 할 때 소켓 통신을 하는 부분이 힘들었다.

이번 프로젝트를 통해 웹 환경에서 소켓통신을 통해 데이터를 전송하는 방법에 대해서 알 수 있었고 클라이언트에서 자바스크립트를 통해 웹페이지를 구성하는 법을 배울 수 있었다.