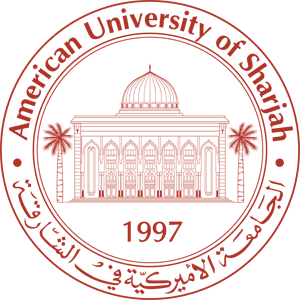
American University of Sharjah

College of Engineering



Internet and IoT Programming COE457

Fall 2020

Date due: 4th October 2020

Instructor: Imran Zualkernan

Ramy Gendy b00074249

Github link:

<https://github.com/Ramy-Gendy/COE457-HW2>

Code:

Mycompass.html:

<html>

<head>

<style>

body {

display: flex;

justify-content: center;

}

.container {

display: flex;

height: 40%;

width: 20%;

border-radius: 50%;

background-color: black;

}

.center {

margin: auto;

}

#arrow {

font-size: 200;

color: grey;

transform: rotate(0deg);

}

</style>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script>

window.onload = function() {

function getAngle(source, destination) {

console.log(current);

console.log(destination);

//calculating angle between two coordinates

var longtitudeDifference = destination[1] - source[1];

var angle = Math.atan2(Math.sin(longtitudeDifference) \* Math.cos(destination[0]), ((Math.cos(source[0]) \* Math.sin(destination[0])) - (Math.sin(source[0]) \* Math.cos(destination[0]) \* Math.cos(longtitudeDifference))));

return (angle \* 180 / Math.PI);

}

getLocation();

var updateLocationInterval = window.setInterval(getLocation, 1000);

function getLocation() {

$.ajax({

type: "GET",

url: "/map",

}).done(function(response) {

console.log("success");

console.log(response);

if (response != '{}') {

$('#arrow').css('color', 'white');

const parsedResp = $.parseJSON(response);

const source = [parseFloat(parsedResp.sourceLat), parseFloat(parsedResp.sourceLng)];

const destination = [parseFloat(parsedResp.destLat), parseFloat(parsedResp.destLng)];

angle = getAngle(source, destination);

console.log(angle);

$('#arrow').css('transform', 'rotate(' + (angle-90) + 'deg)');

}

});

}

}

</script>

<link rel="icon" href="data:,">

<title> My Compass </title>

</head>

<body>

<div class="container center">

<div class="center" id="arrow"> > </div>

</div>

</body>

</html>

Server.js:

const net = require('net');

var fs = require('fs');

var locations = {};

function parsePOSTRequest(req) {

const bodyIndex = r.indexOf("\n\r");

const reqBody = r.substring(bodyIndex + 3, r.length);

let indexOfKey = 0;

let indexOfEqual = 0;

let indexOfAmp = 0;

let done = false;

let resp = {};

while (!done) {

indexOfEqual = reqBody.indexOf('=', indexOfEqual + 1);

indexOfAmp = reqBody.indexOf('&', indexOfAmp + 1);

let key = reqBody.substring(indexOfKey, indexOfEqual);

let value;

if (indexOfAmp == -1) {

value = reqBody.substring(indexOfEqual + 1, req.length)

}

else {

key = reqBody.substring(indexOfKey, indexOfEqual)

indexOfKey = indexOfAmp + 1;

value = reqBody.substring(indexOfEqual + 1, indexOfAmp)

}

resp[key] = value;

}

return resp;

}

var server = net.createServer(function (socket) {

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

r = data.toString();

if (r.substring(0, 10) == "GET /arrow") {

socket.write("HTTP/1.1 200 OK\n");

fs.readFile('mycompass.html', 'utf8', function (err, contents) {

socket.write("Content-Length:" + contents.length);

socket.write("\n\n");

socket.write(contents);

})

}

else if (r.substring(0, 8) == "GET /map") {

socket.write("HTTP/1.1 200 OK\n");

let JSONContents = JSON.stringify(locations);

socket.write("Content-Length:" + JSONContents.length);

socket.write("\n\n");

socket.write(JSONContents);

console.log(JSONContents);

}

else if (r.substring(0, 3) == "GET") {

console.log('OK');

socket.write("HTTP/1.1 200 OK\n");

fs.readFile('mymap.html', 'utf8', function (err, contents) {

socket.write("Content-Length:" + contents.length);

socket.write("\n\n");

socket.write(contents);

})

}

else if (r.substring(0, 4) == "POST") {

console.log('sending');

locations = parsePOSTRequest(r);

console.log(locations);

socket.write('HTTP/1.1 200 OK\n');

socket.write("Content-Length:" + 0);

socket.write("\n\n");

}

});

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

console.log('Connection closed');

});

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

console.log('client disconnected');

});

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

console.log('client disconnected');

});

});

server.listen(8080, function () {

console.log('server is listening on port 8080');

});

mymap.html:

<html>

<head>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/leaflet.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/leaflet.js"></script>

<script src="https://www.mapquestapi.com/sdk/leaflet/v2.2/mq-map.js?key=IudPx1KhI7KYBiGbISQbTLJpGkv7AOn8"></script>

<script

src="https://www.mapquestapi.com/sdk/leaflet/v2.2/mq-routing.js?key=IudPx1KhI7KYBiGbISQbTLJpGkv7AOn8"></script>

<script type="text/javascript">

window.onload = function () {

var map;

var direction;

var clicked;

var location;

var clickedLocation = {};

var currentMark;

map = L.map('map', {

layers: MQ.mapLayer(),

center: [25.196099, 55.281245],

zoom: 10

});

direction = MQ.routing.directions();

var layer = MQ.routing.routeLayer({

directions: direction,

fitBounds: true

});

map.addLayer(layer);

map.locate({ setView: true, }).on('locationfound', (e) => getNewLocation(e));

map.on('click', function (e) {

if (clicked) {

map.removeLayer(layer)

direction = MQ.routing.directions();

layer = MQ.routing.routeLayer({

directions: direction,

fitBounds: true

});

map.addLayer(layer)

}

clickedLocation = [e.latlng.lat, e.latlng.lng];

clicked = true;

direction.route({

locations: [

location[0] + ',' + location[1],

clickedLocation[0] + ',' + clickedLocation[1]

]

});

sendToCompass(clickedLocation, location);

});

var sendInterval = window.setInterval(sendToCompass, 1000);

function getNewLocation(e) {

if (clicked) {

map.removeLayer(layer);

direction = MQ.routing.directions();

layer = MQ.routing.routeLayer({

directions: direction,

fitBounds: true

})

map.addLayer(layer);

}

location = [e.latitude, e.longitude];

if (clickedLocation == undefined)

clickedLocation = [(location[0] + 0.00005), location[1]];

direction.route({

locations: [

location[0] + ',' + location[1],

clickedLocation[0] + ',' + clickedLocation[1],

]

});

sendToCompass(clickedLocation, location);

}

}

function sendToCompass(clickedLocation, location) {

console.log(clickedLocation);

if (clickedLocation != undefined) {

var locationToSend = {

sourceLat: location[0],

sourceLng: location[1],

destLat: clickedLocation[0],

destLng: clickedLocation[1],

}

$.ajax({

type: "POST",

data: locationToSend

}).done(function (response) {

console.log("sent location");

});

}

}

</script>

<title> My Map </title>

</head>

<body>

<div id='map' style='width: 100%; height:100%;'></div>

</body>

</html>

Screenshots:

