var express = require('express');

var fs = require('fs');

var app = express();

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

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

var Global\_values = {

ing: 0,

random\_ing: 0,

random\_check: 0,

eventSave: null,

isIntervalPlaying: 0,

head\_location: "0.000",

tail\_count: -1,

check: 0,

name\_count: 2,

move: 0,

SleepTIme: 100,

bomb\_count: 0,

fast\_count: 0,

slow\_count: 0,

long\_count: 0

};

//ing변수의 경우 Interval 제어(정지)를 위해 생성

//random\_ing변수 경우 Random\_Interval 제어(정지)를 위해 생성

//random\_check변수 경우 랜덤 쓰래드가 한번이상 실행되는걸 막기위해 생성

//eventSave변수의 경우 클릭시, event객체를 잡아두기위해 생성

//isIntervalPlaying 경우 실행도중 다른 방향키를 입력받았을시 interval제어를 위해 생성

//head\_location 경우 머리의 위치 표시

//tail\_location 경우 꼬리의 위치 표시 -> 시작은 머리와 같음

//check의 경우 보드판 감추기에 실행읠 막기위해 생성

//name\_count의 경우 속성name으로 꼬리 조작 -> 버그방지를위해 3부터 시작

app.get("/",function(req, res){

res.sendfile("term\_client.html");

});

//서버단에서 사진 가져오기 blank

app.get("/blank",function(req, res){

fs.readFile('blank.png', function (error,data) {

res.writeHead(200,{'Content-Type':'text/html'});

res.end(data);

});

});

//boom

app.get("/boom",function(req, res){

fs.readFile('boom.png', function (error,data) {

res.writeHead(200,{'Content-Type':'text/html'});

res.end(data);

});

});

//fast

app.get("/fast",function(req, res){

fs.readFile('fast.png', function (error,data) {

res.writeHead(200,{'Content-Type':'text/html'});

res.end(data);

});

});

//head

app.get("/head",function(req, res){

fs.readFile('head.png', function (error,data) {

res.writeHead(200,{'Content-Type':'text/html'});

res.end(data);

});

});

//long\_item

app.get("/long\_item",function(req, res){

fs.readFile('long\_item.png', function (error,data) {

res.writeHead(200,{'Content-Type':'text/html'});

res.end(data);

});

});

//slow

app.get("/slow",function(req, res){

fs.readFile('slow.png', function (error,data) {

res.writeHead(200,{'Content-Type':'text/html'});

res.end(data);

});

});

//tail

app.get("/tail",function(req, res){

fs.readFile('tail.png', function (error,data) {

res.writeHead(200,{'Content-Type':'text/html'});

res.end(data);

});

});

//접속한 클라이언트수

var count = 0;

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

count++;

console.log('user connected: ', socket.id);

//var name = "user" + count++;

//io.to(socket.id).emit('change name',name);

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

count--;

console.log('user disconnected: ', socket.id);

});

socket.on('map size', function(map\_size){

console.log("mapsize:",map\_size);

});

socket.on('crt table', function (cr\_tab, input) {

console.log("cyka:",cr\_tab);

io.emit('created table',cr\_tab, input);

});

socket.on('move loc', function (head\_location, imge) {

io.emit('moved loc', head\_location, imge);

});

soket.on('tail move', function (tail\_point) {

io.emit('tail moved', tail\_point);

});

});

http.listen('3000', function(){

console.log("server on!");

});