# 基于Websocket的火拼俄罗斯

1. 基于Websocket的火拼俄罗斯



1. 先安装npm install nodejs-websocket，可参考 <https://github.com/sitegui/nodejs-websocket>

新建wsServer.js文件，需要加上错误处理事件conn.on('error', function(err){})，否则浏览器关闭页面后会抛出一个错误，

var ws = require("nodejs-websocket")

var PORT = 3000;

// Scream server example: "hi" -> "HI!!!"

var server = ws.createServer(function(conn) {

console.log("New connection")

conn.on("text", function(str) {

console.log("Received " + str)

conn.sendText(str.toUpperCase() + "!!!")

})

conn.on("close", function(code, reason) {

console.log("Connection closed")

})

conn.on("error", function(error) {

console.log("error")

console.log(error)

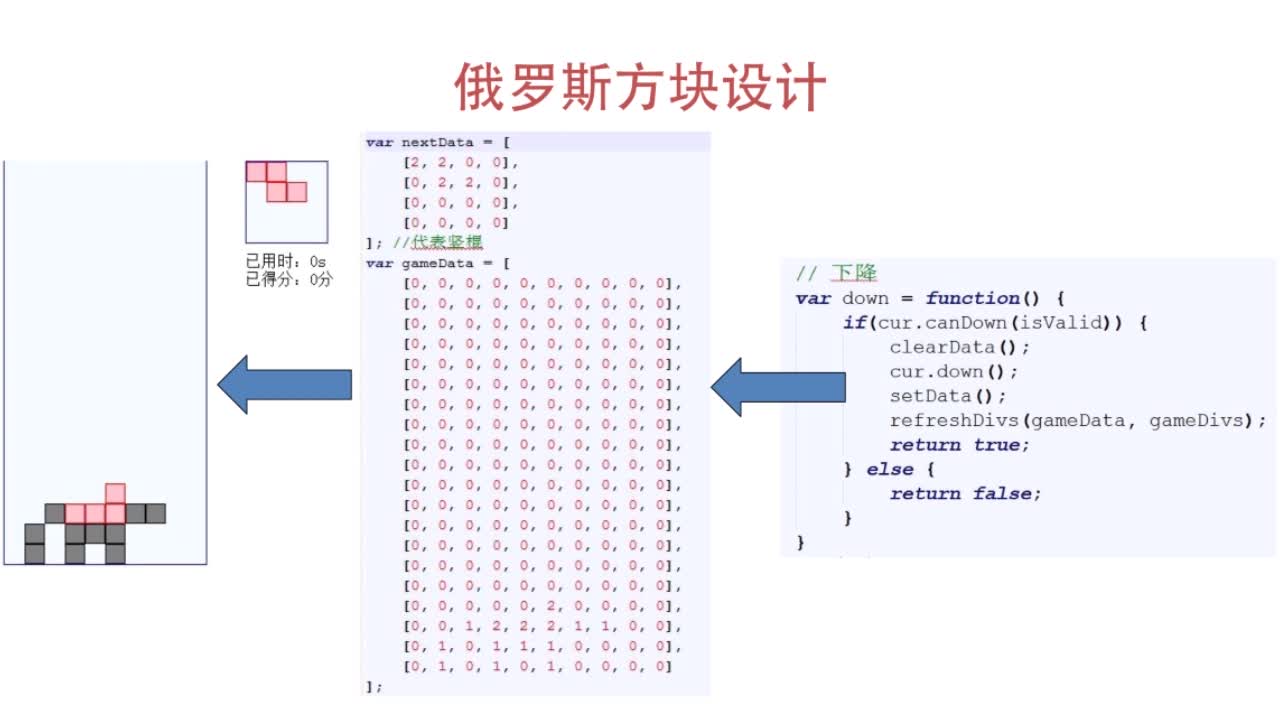
})

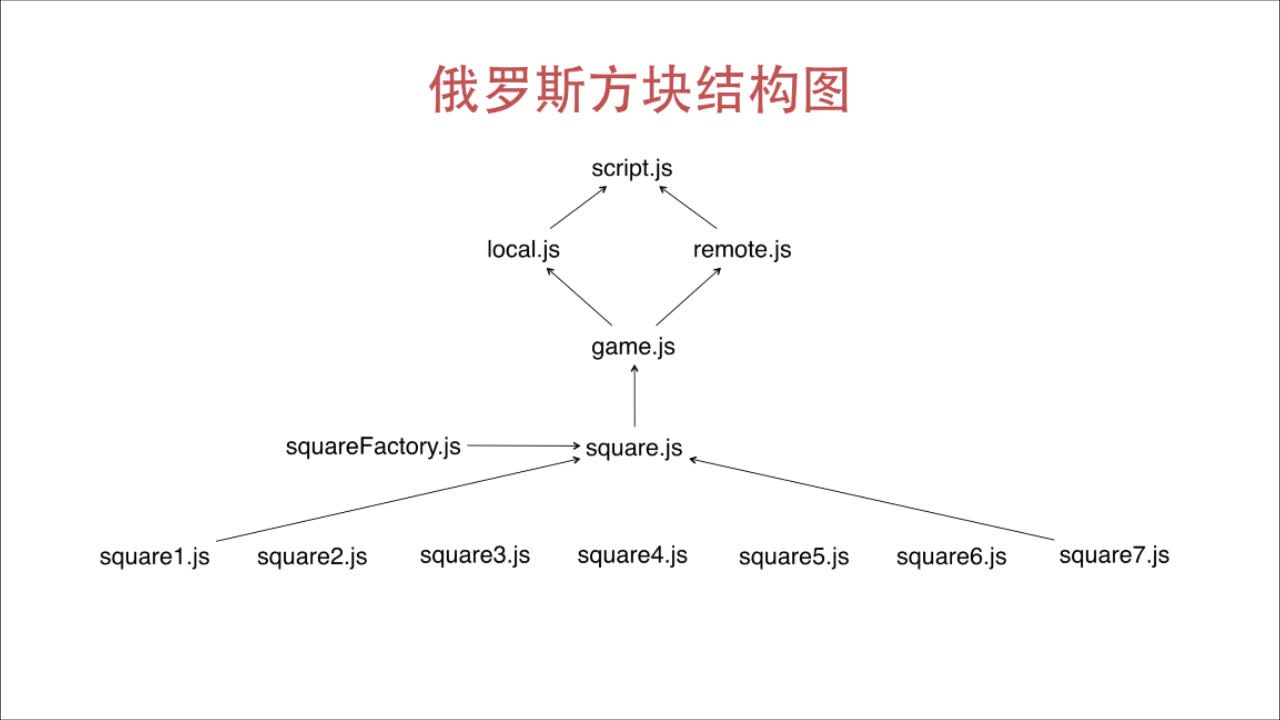
}).listen(PORT)

console.log('websocket server listening on port' + PORT);

启动任务：node wsServer.js

1. socket.io入门（发送数据时可以直接发送js对象，不需要自己格式化；可以方便自定义消息类型）
2. 作为程序员，强烈建议大家在本地安装一个linux环境
3. 服务器向浏览器发送数据的场景
4. 
5. 火拼俄罗斯游戏原理：视图view-模型modal-控制器control



1. 
2. 采用MVC的思想编写代码，参考《《基于Websocket的火拼俄罗斯》，代码逻辑很严谨，需学习》
3. 客户端1 emit 发送消息到服务器，服务器on接收消息，再emit转发消息到客户端2，客户端2 on 接受消息；这样客户端1和客户端2就能实时发送消息，进行通讯；
4. Websocket应用场景：社交聊天、弹幕、多玩家游戏、协同编辑、股票基金协同报价、体育实况更新、视频会议/聊天、基于位置的应用、在线教育、智能家居；适用于高实时性场景

# JavaScript的新语法新趋势

1. js日期对象的用法

# 前端实时可视化开发工具体验

1. 
2. 1
3. 1
4. 1
5. 1