



你画我猜【乞丐版】

vuex + socket-io



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



目录

Contents

- ◆ 你画我猜 - 项目准备
- ◆ 你画我猜 - 登录模块
- ◆ 你画我猜 - 进入房间
- ◆ 你画我猜 - 游戏主页展示
- ◆ 你画我猜 - 游戏控制

学习目标

Learning Objectives

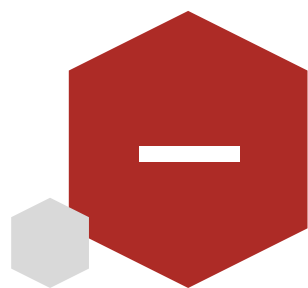
1. 能够掌握 **vuex** 于项目中的应用
2. 能够掌握 **socket-io** 于项目中的应用



目录

Contents

- ◆ 你画我猜 - 准备工作
- ◆ 你画我猜 - 登录模块
- ◆ 你画我猜 - 进入房间
- ◆ 你画我猜 - 游戏主页展示
- ◆ 你画我猜 - 游戏控制



你画我猜 - 准备工作

1. 你画我猜 小游戏 效果是什么样的? => 效果演示

是一个实时画图, 猜答案的房间小游戏

2. 我们能学到什么? => 技术栈

Vue + VueRouter + Vuex(核心) => 多组件通信

Socket.IO (核心) => 服务端和客户端双向通信

Konva (基于canvas的绘图库) => 辅助开发, 实现绘图



小结

本节课演示了游戏效果，说明了本项目的技术栈

1. 这个项目 主要练的核心技术栈 是哪两个？

Vuex + Socket-io

2. 他们两个的作用是什么

Vuex => 多组件通信

Socket-io => 服务器和浏览器双向通信

1. 准备基本静态结构

说明: 这边为了节约时间, 老师已经准备好了基础代码, 同学们可以直接去 gitee 上下载模板即可

地址: <https://gitee.com/jepsonpp/draw-and-guess>

2. 安装依赖 (draw-and-guess-template)

```
yarn
```

3. 启动模板

```
yarn serve
```





思考

因为要和服务器端 **即时双向通信**，会要求服务器能够
主动向客户端推消息，所以要用到哪个技术呢？

websocket => Socket.IO

目标：构建Socket.io模块

基本语法：

1. 创建连接: `const socket = io(地址)`
2. 发送消息: `socket.emit(消息type类型, 消息内容, 接收到消息的回调函数)`
3. 监听消息: `socket.on(事件type类型, 接收到消息的回调函数)`

为什么需要这个模块：

1. 用 on 监听事件， 接收服务器消息
2. 用 emit 发送消息到服务器

构建步骤：

1. 安装依赖包 `yarn add socket.io-client`
2. 创建 `src/socket/index.js` 模块文件，编写模块内容
3. `main.js` 中导入
4. `vue.config.js` 配置网络代理
5. 启动后台server， 测试接口



小结

我们封装这个Socket.IO模块能帮我们做什么？

on 监听事件， 接收服务器消息

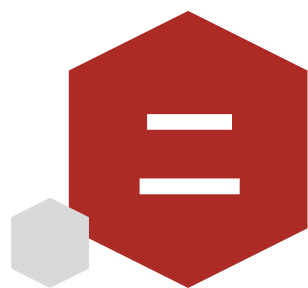
emit发送消息， 发送消息到服务器



目录

Contents

- ◆ 你画我猜 - 项目准备
- ◆ 你画我猜 - 登录模块
- ◆ 你画我猜 - 进入房间
- ◆ 你画我猜 - 游戏主页展示
- ◆ 你画我猜 - 游戏控制



你画我猜 - 登录模块

需求分析:

1. 点击进入游戏按钮，表单验证
2. 验证通过，发送昵称到服务器端验证 (action)
3. 若未被占用， 则跳转到游戏主页



The image shows a login form with a text input field and a button. The input field has a placeholder text "请输入您的昵称" (Please enter your nickname) and a red border. Below the input field, there is a red error message "请输入您的昵称" (Please enter your nickname). To the right of the input field is a blue button with the text "进入游戏" (Enter the game).



小结

本节课，主要完成了什么功能呢？

封装了action，进行了用户昵称是否占用的校验

如果没被占用，就跳转到 home 主页



目录

Contents

- ◆ 你画我猜 - 项目准备
- ◆ 你画我猜 - 登录模块
- ◆ 你画我猜 - 进入房间
- ◆ 你画我猜 - 游戏主页展示
- ◆ 你画我猜 - 游戏控制



你画我猜 - 进入房间

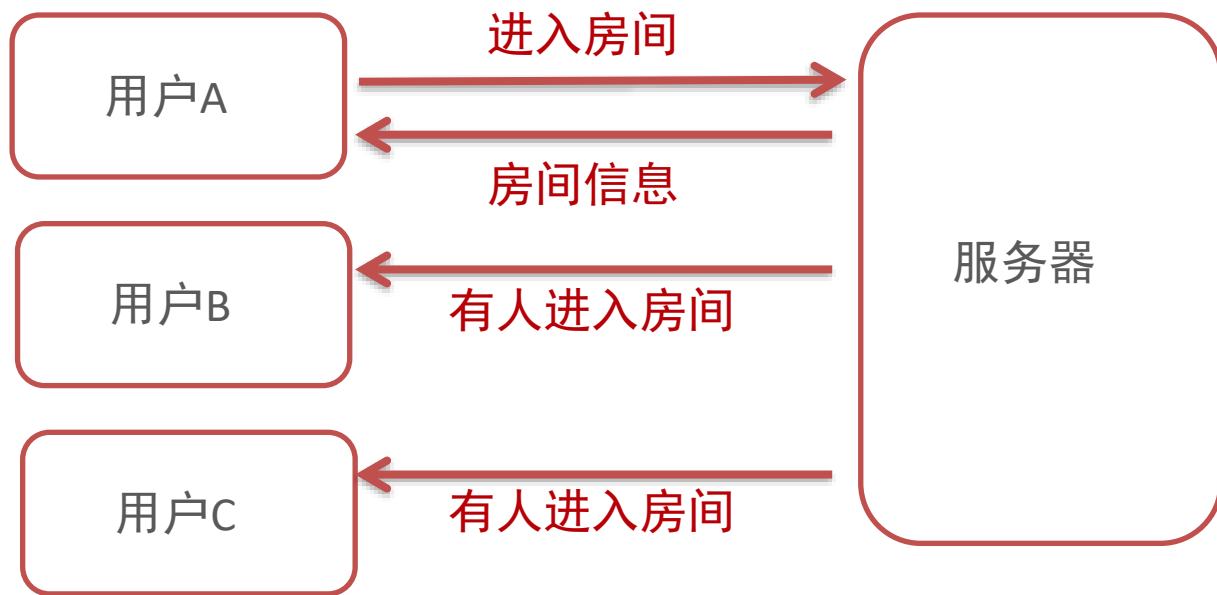
思考：跳转到首页，就是进入房间，进入房间要做哪些事情呢？

需求分析：

1. 用户进入房间，需要告诉服务器，你进来了
2. 服务器会通知返回 **房间信息** 如：房间**用户列表**，**主持人**，**绘图信息**等
3. 服务器会 **广播通知其他所有用户**，有新用户进入房间

进入房间 步骤分析：

1. 进入房间 - 请求服务器
2. 处理 room_info 事件
3. 处理 user_enter 事件

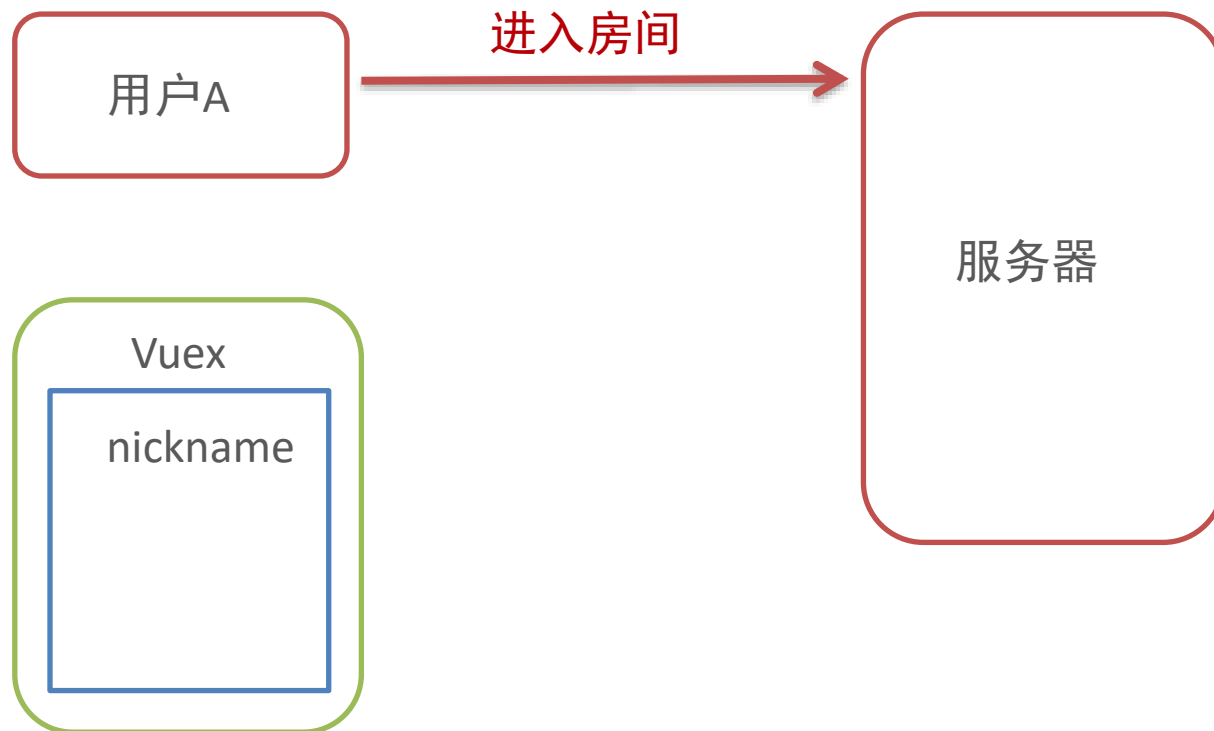


05- 进入房间 - 请求服务器

需求1：进入房间 - 请求服务器（用户进入房间，需要告诉服务器，你进来了）

操作步骤：

1. vuex声明相关state（备用）
(自己昵称, 用户列表, 主持人, 绘图信息)
2. vuex提供进入房间的 action 和 mutation
action: 发消息到服务器, 你来了
mutation: 更新昵称到 vuex 中
3. 主页中, 调用 action 即可

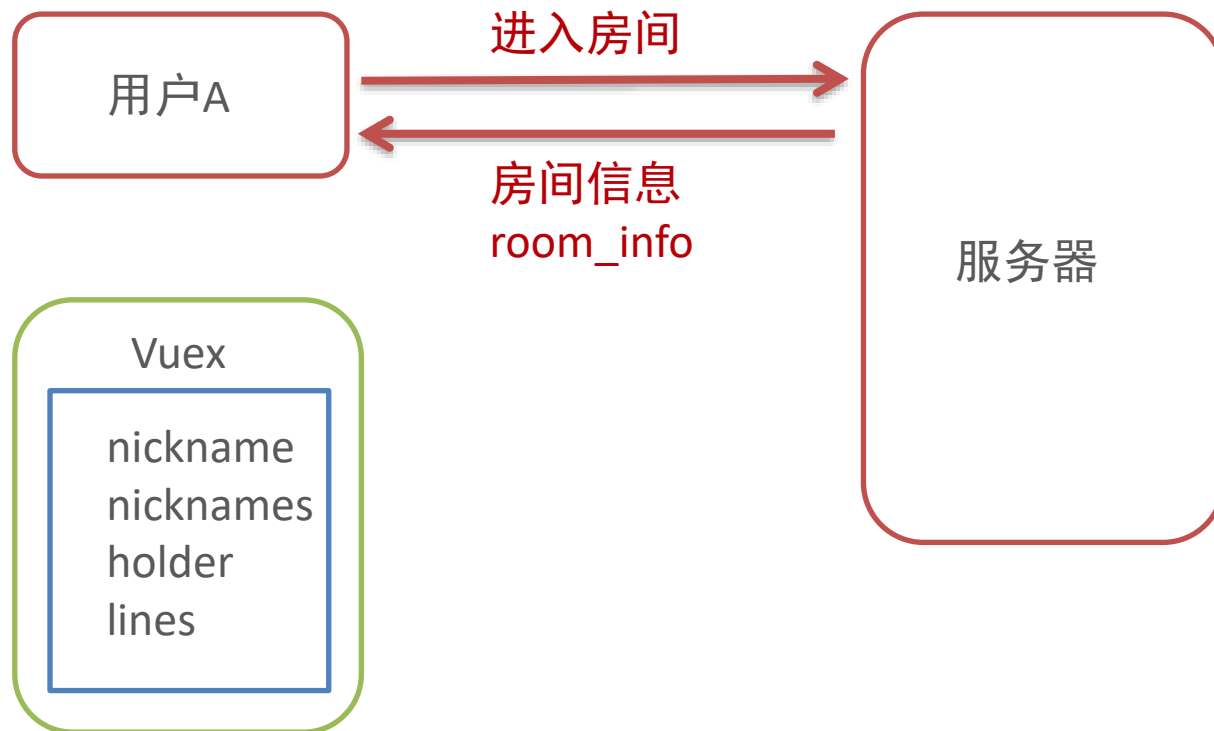


06- 进入房间 - 处理 room_info 事件

需求2：处理 room_info 事件（服务器收到消息，通过room_info事件响应房间信息）

操作步骤：

1. room_info 事件监听, 获取房间消息
(用户列表, 主持人, 绘图信息)
2. 提供对应的更新mutation,
调用后将获取的消息存到vuex中



07- 进入房间 - 处理 user_enter 事件

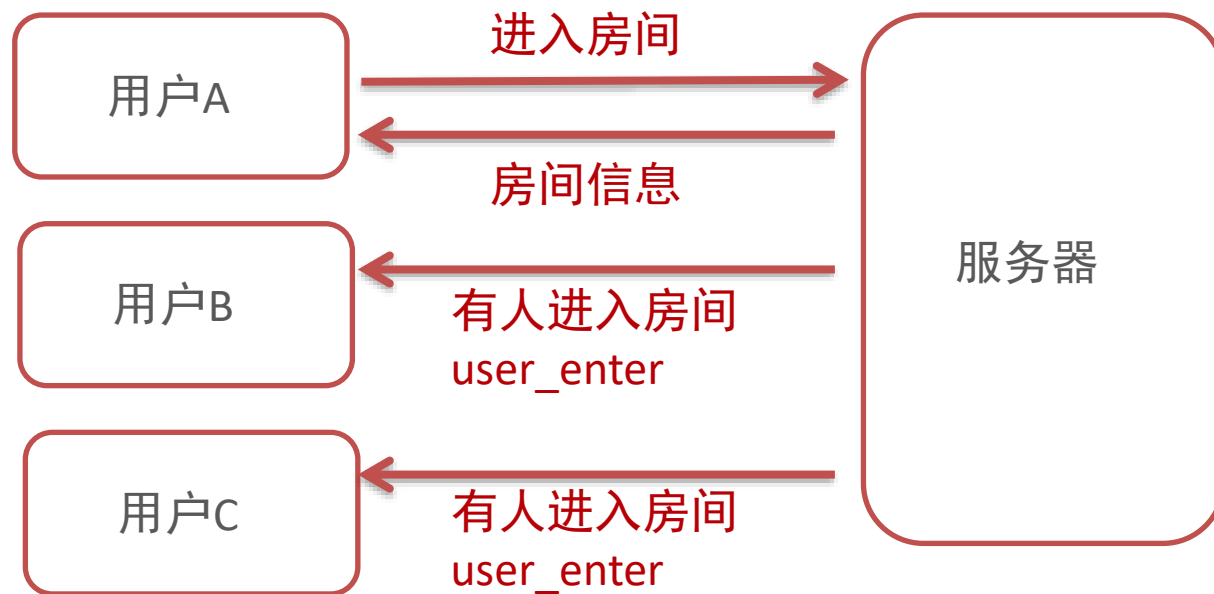
需求3：处理 user_enter 事件（服务器收到消息，通过user_enter事件广播通知其他用户）

操作步骤：

1. 监听 user_enter 事件，处理广播通知

2. 提供 mutation，更新 nicknames 列表

注意：用户不存在于列表中才追加





小结

我们进入房间一共做了哪三件事？

- (1) 向服务器发送了 **进入房间** 的申请请求
- (2) 处理 **room_info** 事件，拿到信息，存vuex
- (3) 处理 **user_enter** 事件，更新nicknames列表



目录

Contents

- ◆ 你画我猜 - 项目准备
- ◆ 你画我猜 - 登录模块
- ◆ 你画我猜 - 进入房间
- ◆ 你画我猜 - 游戏主页展示
- ◆ 你画我猜 - 游戏控制



你画我猜 - 主页展示

思考

1. 我们 vuex 中存了哪些信息呢?
2. 能用于首页状态栏和玩家列表的展示了么?

Vuex

nickname
nicknames
holder
lines

服务已连接

游戏进行中

头部信息

玩家列表信息

- xxx (我) 主持
- xxx
- xxx

主持游戏

退出游戏

08- 主页展示 - 头部状态信息展示

需求：完成 头部状态信息 的展示

分析：

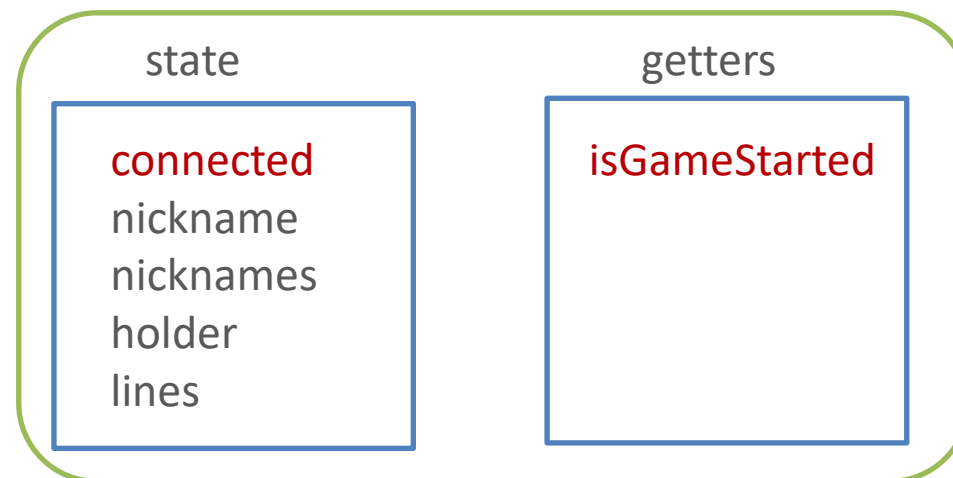
- 头部组件中展示：服务已连接、未连接，游戏进行中、未开始
- 都需要基于状态来判断

步骤：

1. 准备标记连接状态的 state => **connected**,
游戏是否开始的 getters => **isGameStarted**
2. 监听连接 和 断开事件，动态更新 **connected**
3. 界面处理 v-if 控制显示



Vuex



09- 主页展示 - 边栏玩家列表

需求：完成 边栏玩家列表 的展示

分析：

- vuex中直接就有玩家列表信息 nicknames

步骤：

1. mapState 引入 vuex 状态
2. v-for 遍历, 视图展示控制

服务已连接

游戏进行中

玩家列表信息

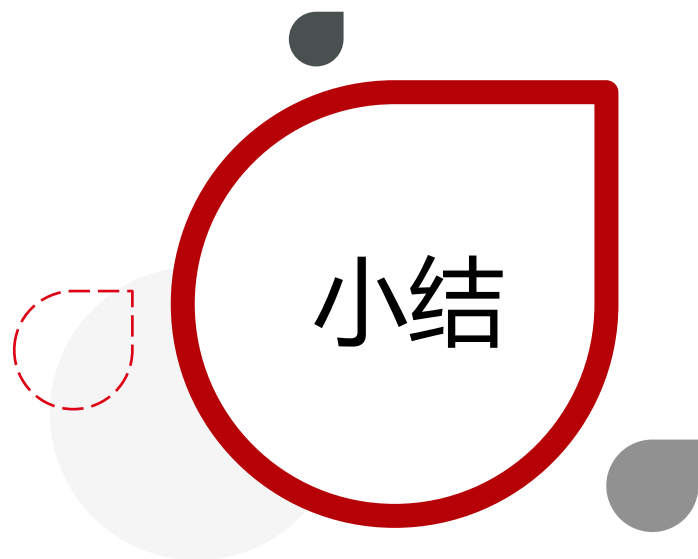
• XXX (我) 主持

• XXX

• XXX

主持游戏

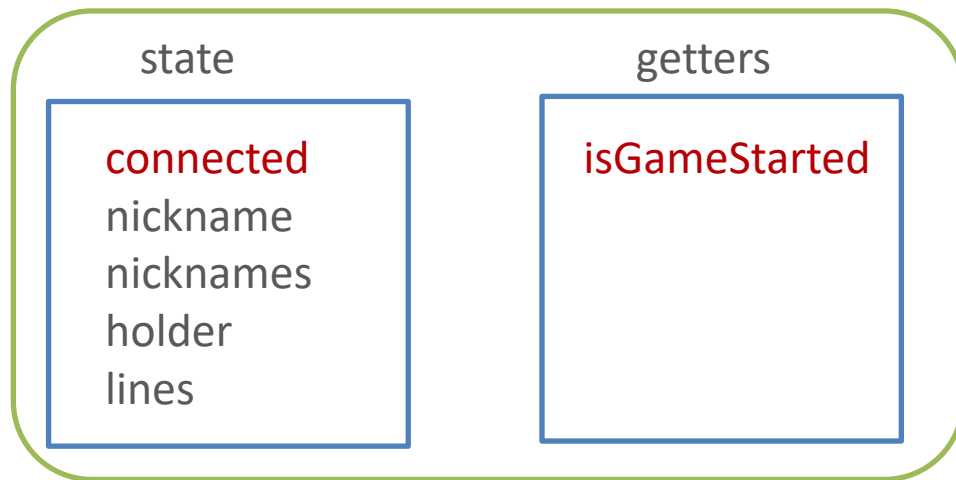
退出游戏



本章小结：本章节基于 vuex 中存的信息，进行了主页的基本展示

那么现在vuex里面存了哪些数据了？

Vuex

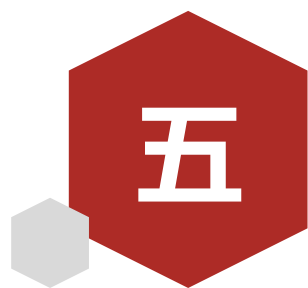




目录

Contents

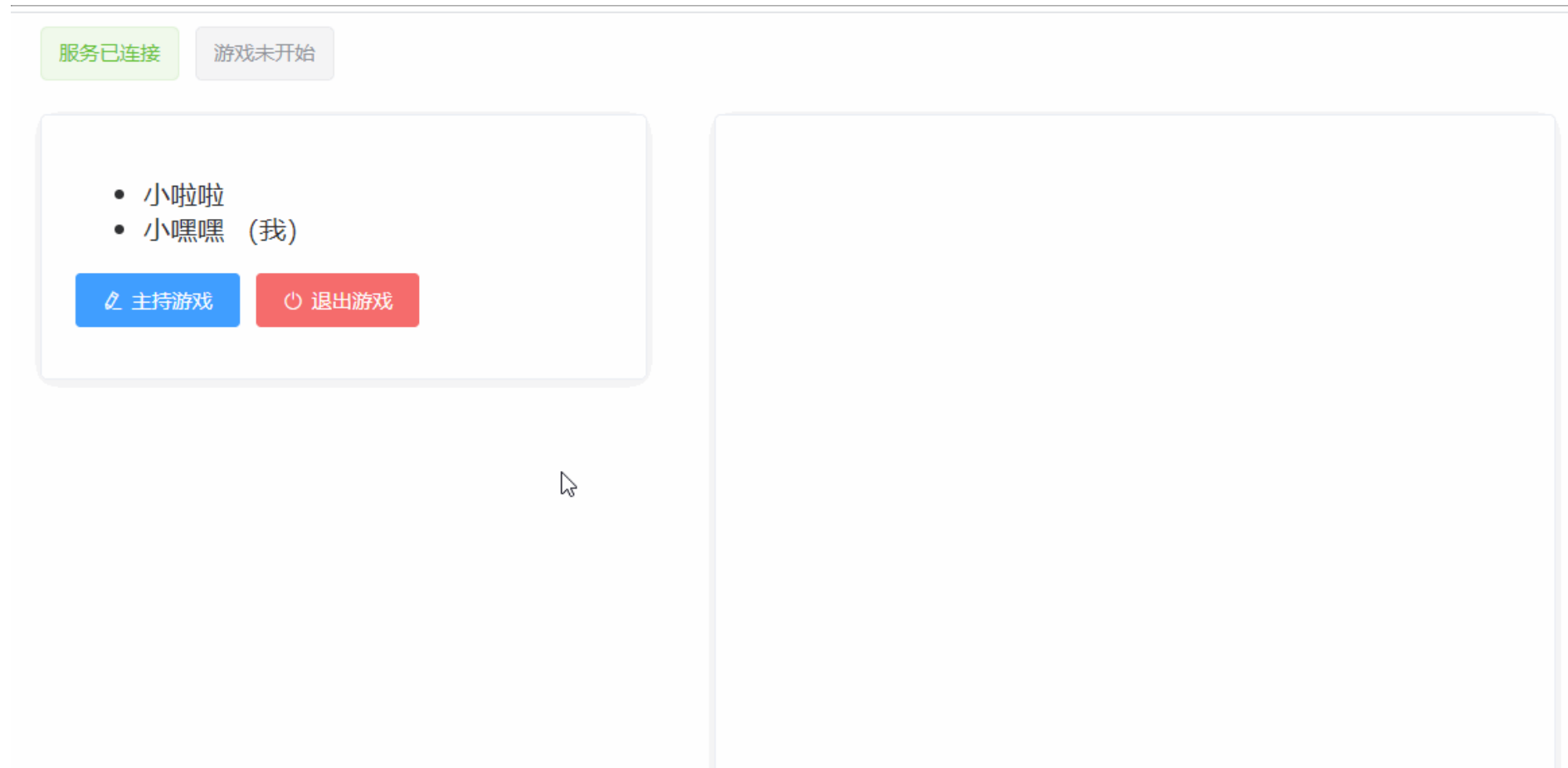
- ◆ 你画我猜 - 项目准备
- ◆ 你画我猜 - 登录模块
- ◆ 你画我猜 - 进入房间
- ◆ 你画我猜 - 游戏主页展示
- ◆ 你画我猜 - 游戏控制



你画我猜 - 游戏控制

10- 主持游戏

思考：点击主持游戏，会有什么效果呢？



10- 主持游戏

需求：主持游戏功能

需求要点：

1. 点击主持按钮，弹出对话框，填写答案
2. 点击确定，向服务器发起申请，开始主持游戏

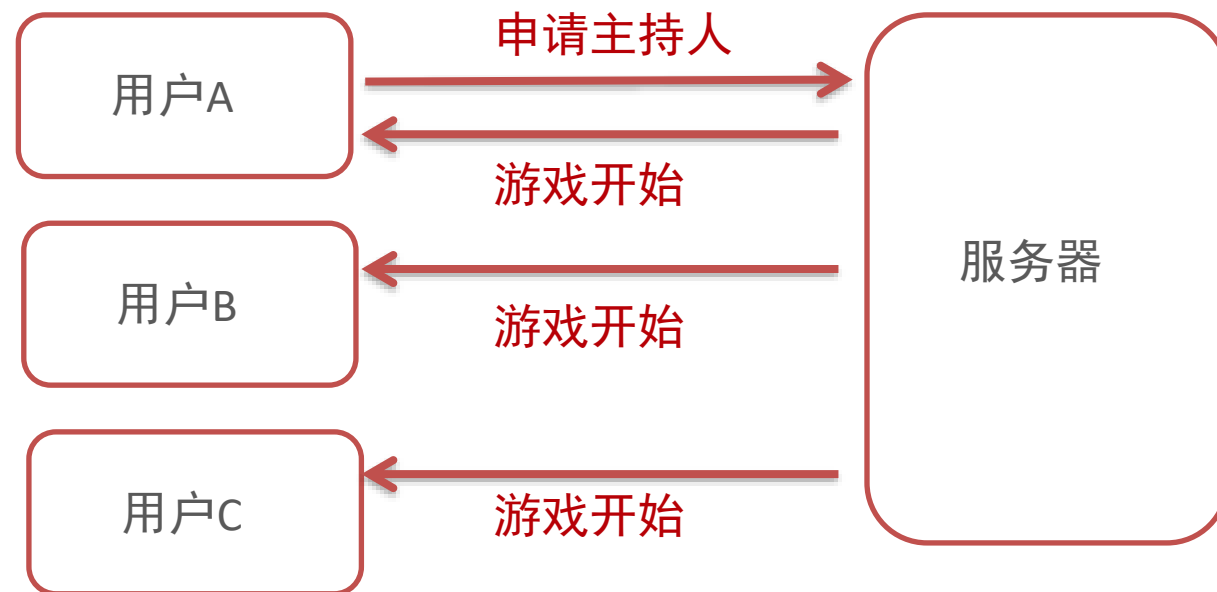
(1) 成功，成为主持人，服务器广播 game_started 事件

(2) 失败，有人已抢先当主持人，服务器通知你 already_started 事件



实现步骤：

1. 控制主持按钮的显示（只有游戏未开始显示）
2. 注册点击事件，点击主持按钮，弹出框
3. 输入内容，确定，发送请求申请
4. 处理服务器的事件（成功+失败）





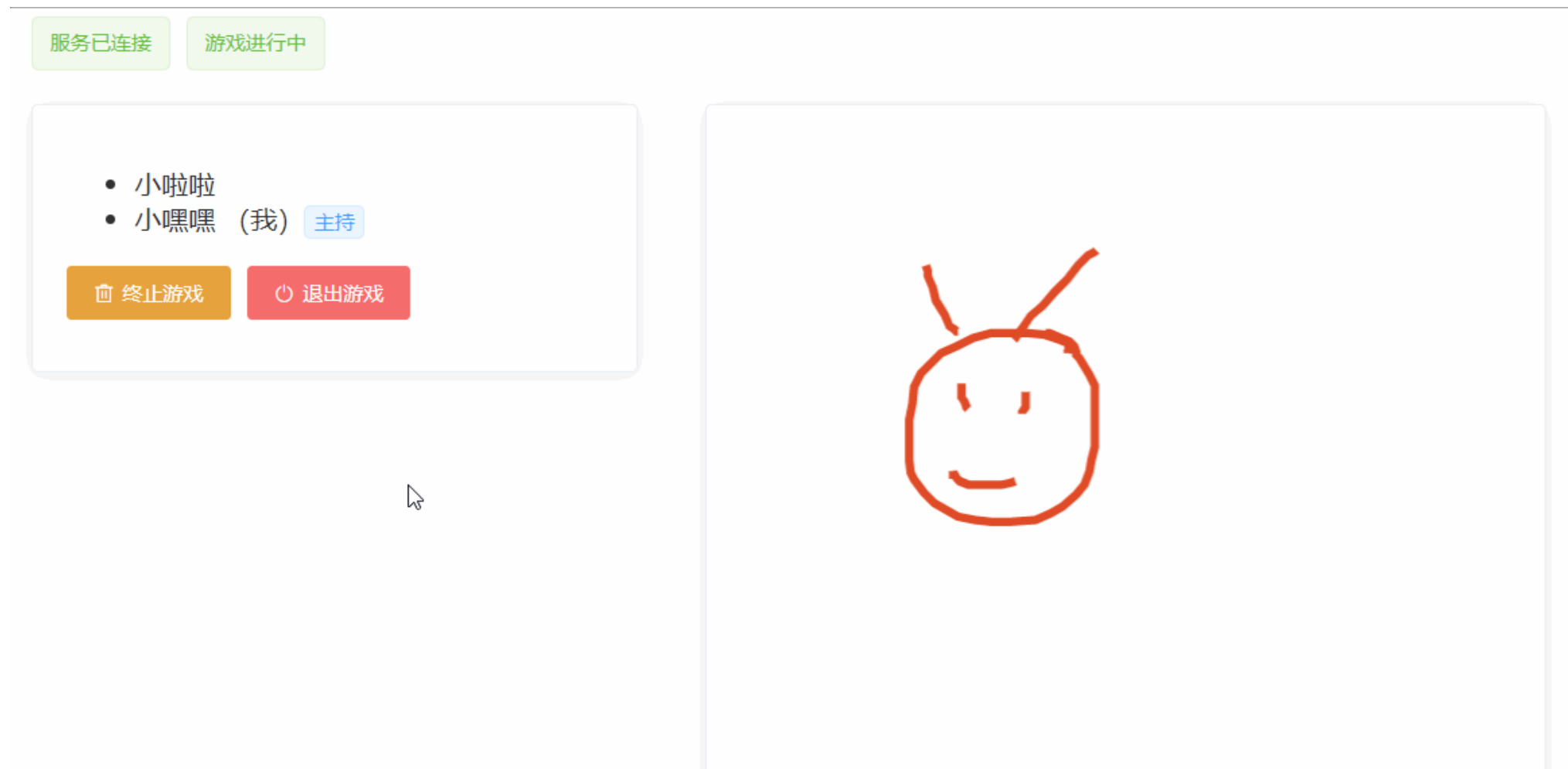
小结

本小节完成了什么功能?

主持游戏功能 (控制游戏的开始)

11- 终止游戏

思考：点击终止游戏，会有什么效果呢？



11- 主持游戏

需求：终止游戏功能

需求要点：

1.游戏开始，显示 **终止游戏** 或 **猜答案**

主持人终止游戏，其他人猜答案

2.点击终止游戏，向服务器发送 **终止游戏** 的消息

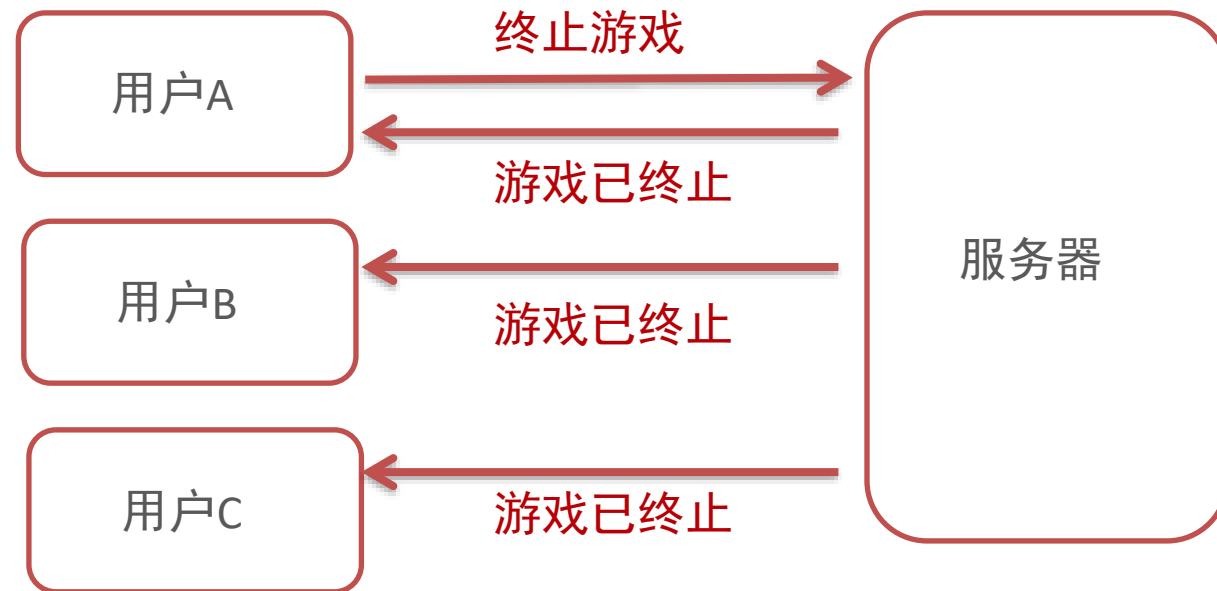
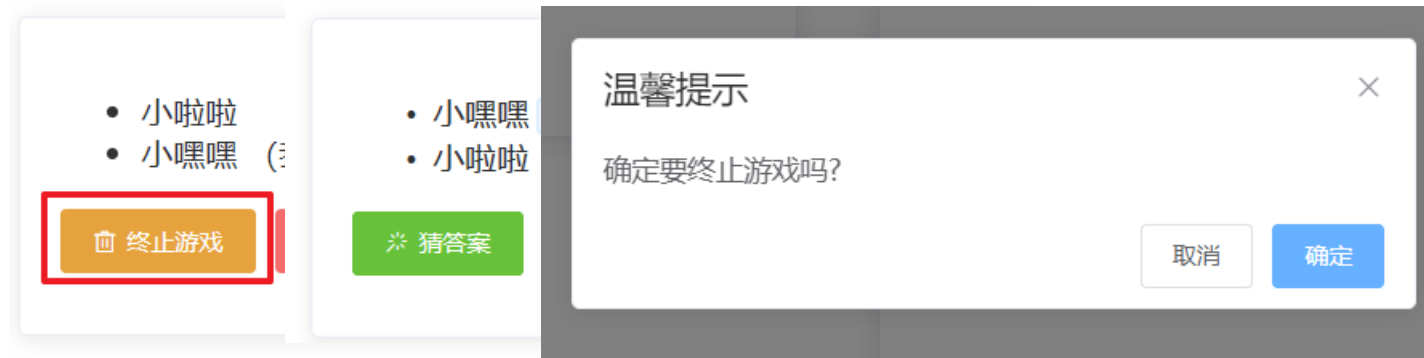
3.服务器向所有人**广播**通知 **游戏终止**，可以有人重新主持游戏

实现步骤：

1.控制终止按钮和猜答案的显示

2.注册点击事件，点击终止按钮，发送申请

3.处理服务器游戏终止的事件响应





小结

本小节完成了什么功能?

终止游戏功能 (控制游戏的关闭)



思考

目前为止，我们已经完成了游戏的主持和终止，

已经可以控制游戏的开始和关闭了！

但是开始游戏之后，我们应该要画图呀？如何鼠标绘图呢？

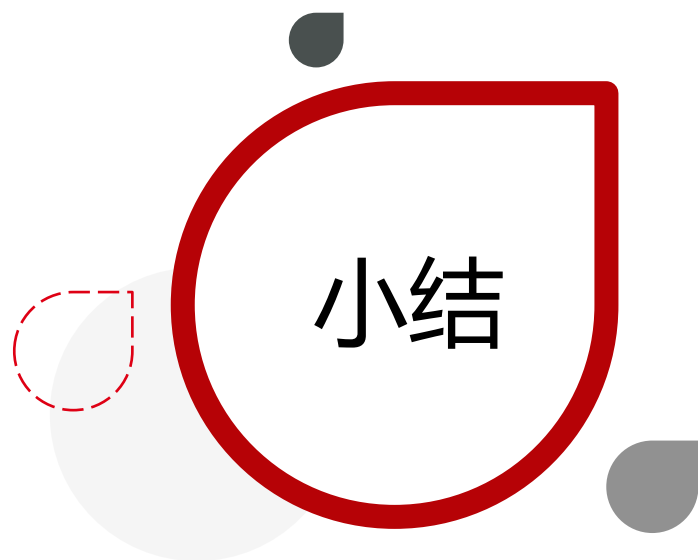
1. 说明：绘图需要用到插件 Vue-Konva

使用插件 **Vue-Konva**（基于Canvas的绘图库）可以帮助我们完成绘图功能

2. 怎么用呢？

使用步骤：

- (1) 安装依赖 `yarn add vue-konva konva`
- (2) main.js 中导入注册
- (3) 准备 v-stage 舞台，注册事件（快速）
- (4) 准备 v-layer 弹层，先写死数据，模拟绘制的线条



本小节完成了做了什么事情?

导入注册了vue-konva绘图插件



思考

1. 刚才绘制的的线是写死的，合适么？
2. 要基于鼠标动态控制，且多个地方可能会控制线，所以线的数据，应该存哪呢？

13- 鼠标绘图 - 基本绘制

明确: lines 应该是放在 vuex 中的 (便于控制), 需要结合鼠标事件, 完成功能

基本步骤:

1. 绘图中, line 线条的数据, 要从 vuex 里面取
2. 鼠标按下, 开启绘画状态, 创建一个新线条, 存到 vuex 中
3. 鼠标滑动, 往 lines 最后一项 (新线条) points 中补充绘图点
4. 鼠标弹起, 结束画线





本小节完成了什么功能?

基本绘图功能



思考

1. 这里游戏还没开始，就能绘图了，对么？

不对，应该是**游戏开始**且是**主持人**，才能绘图

2. 这边本地绘制的图有同步给其他玩家么？

需要发送给服务器 - 进行同步操作

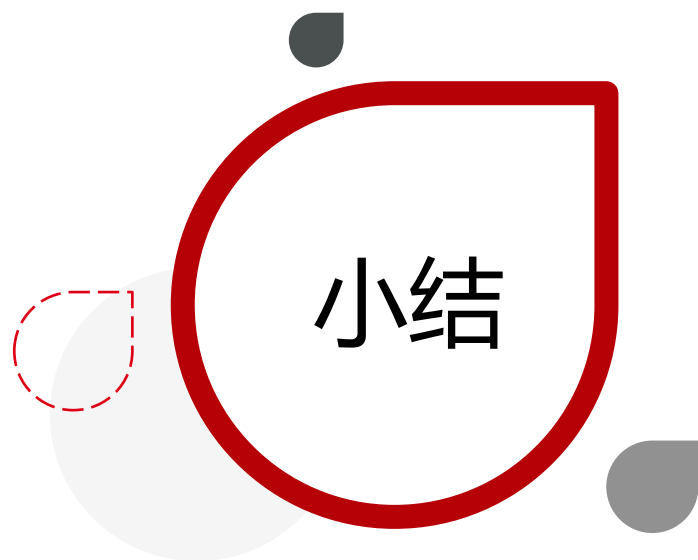
需求：完善功能，同步绘图显示

分析：

1. 要游戏开始, 且只有主持, 才可以进行绘图操作
2. 开始画新的线 `starting_line`, 还是 `updating_line` 更新线, 都要同步到服务器, 服务器会向其他用户广播
3. 要处理服务器同步的广播消息, 才能看到主持的绘图显示

基本步骤：

1. 给鼠标按下，添加判断限制，不可以直接绘图
2. 画新线和更新线， 同步到服务器 （服务器会自动广播）
3. 监听处理服务器的广播消息， 同步绘图

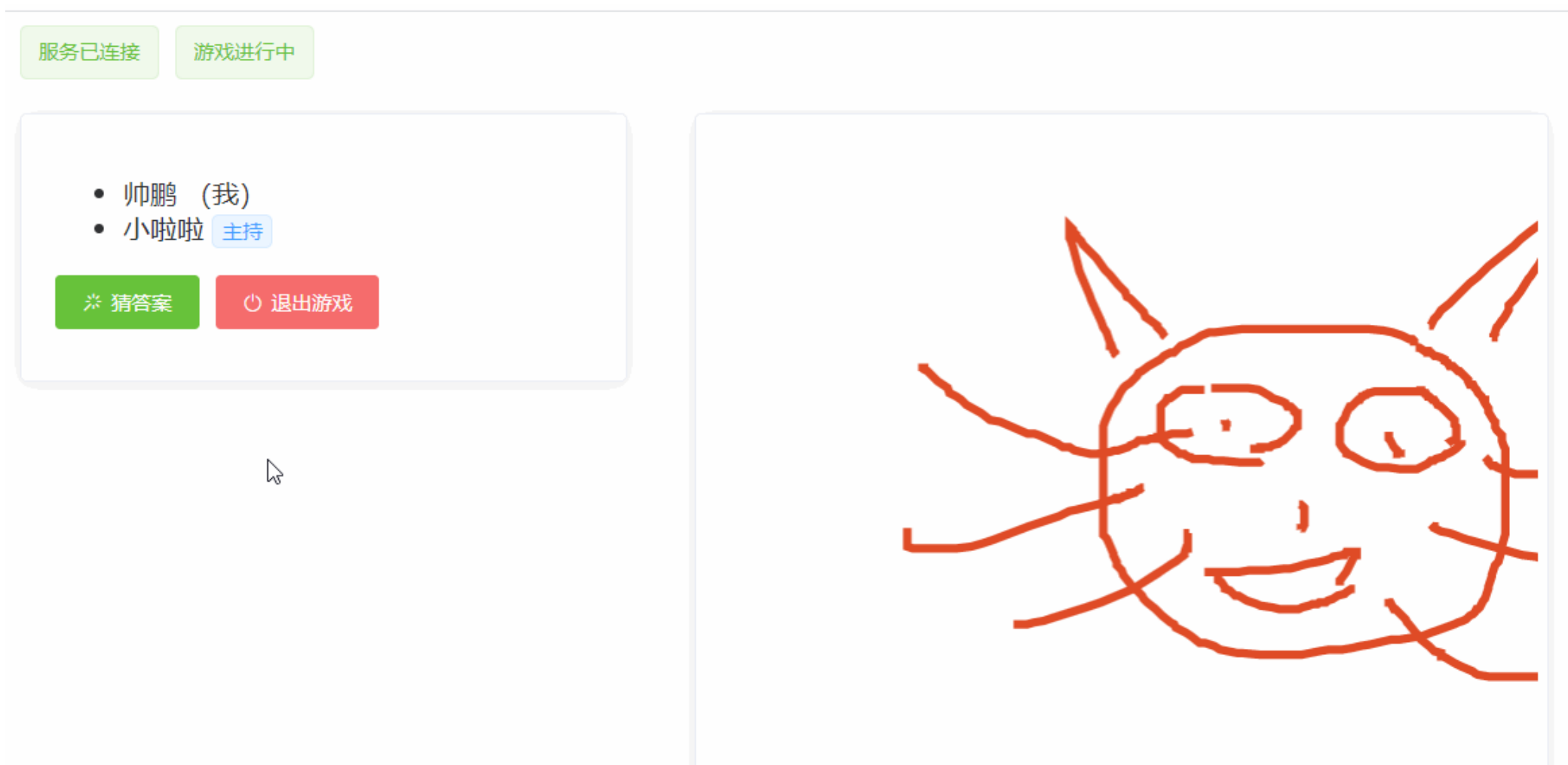


小结

至此, 我们已完成鼠标绘图的功能, 主持绘图时, 其他玩家可以看到效果

15- 玩家猜答案

思考：点击猜答案？应该有什么效果呢？



15- 玩家猜答案

需求：玩家猜答案功能

分析：

1. 点击按钮，弹出对话框，填写答案
2. 点击确定按钮，发送答案到服务器
3. 服务器会进行反馈 并 广播，所有用户都会收到 game_answered 事件

基本步骤：

1. 给按钮注册点击事件， 显示弹框
2. 点击弹框确定，调用action发送答案
3. 处理game_answered 事件





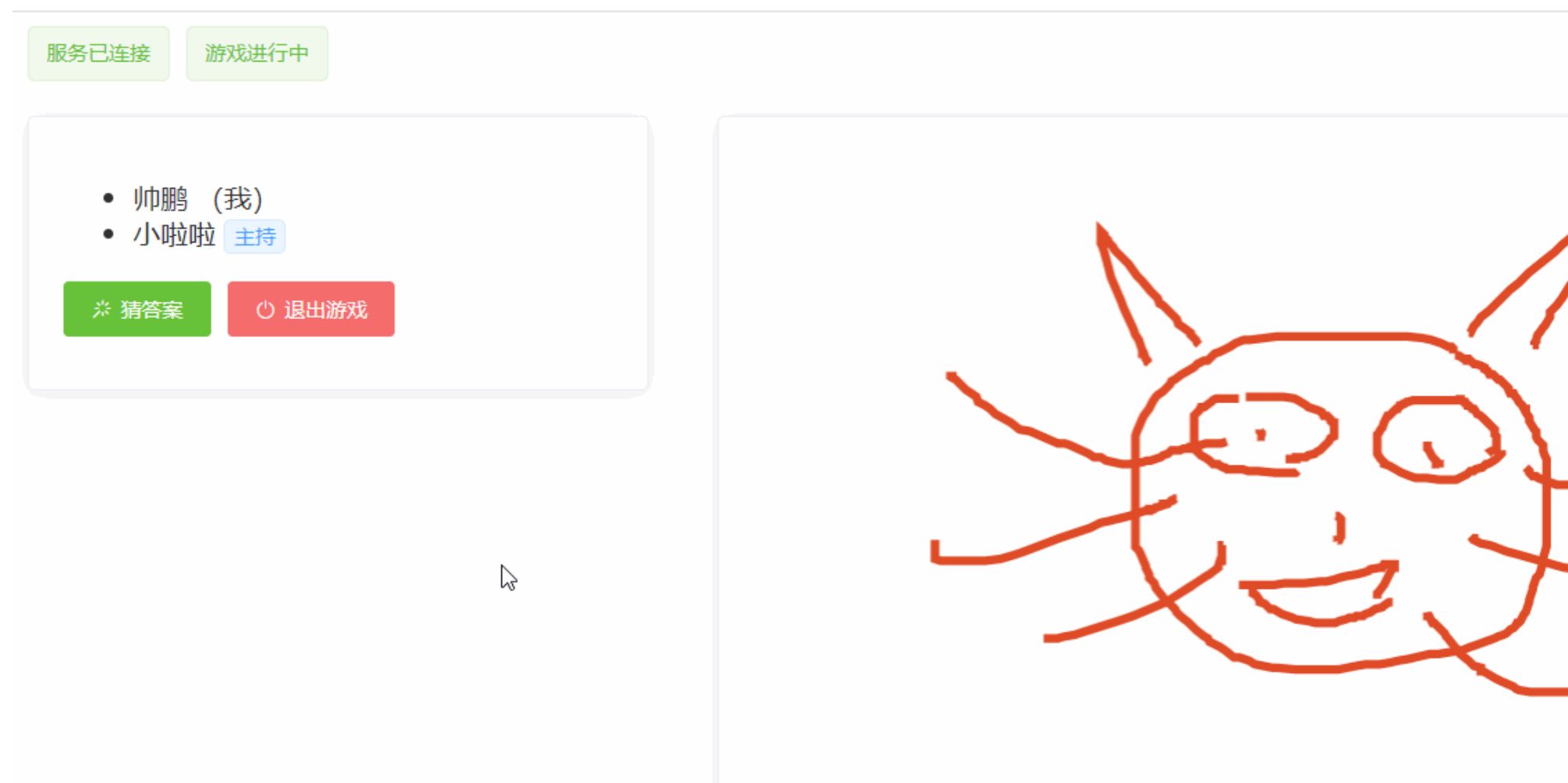
小结

本小节完成了什么功能?

玩家猜答案功能

16- 退出游戏

思考：点击退出游戏，什么效果呢？



16- 退出游戏

需求：退出游戏功能

分析：

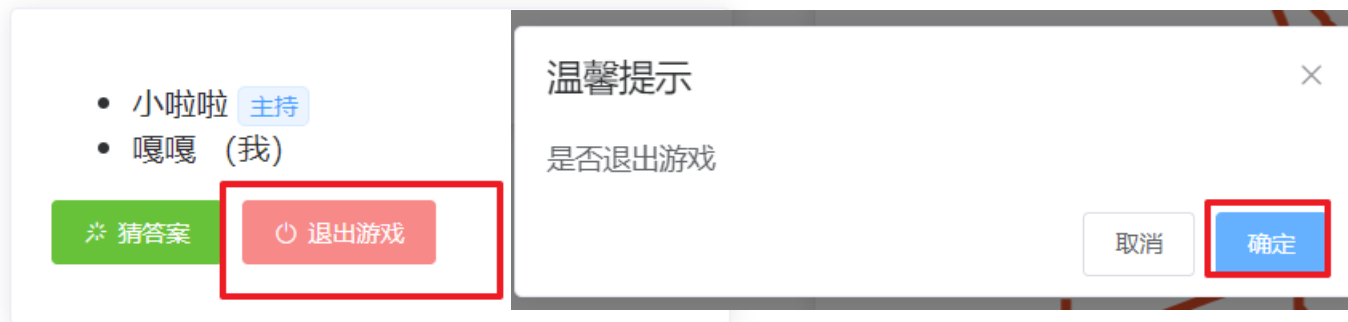
1. 点击退出游戏按钮，发送退出应用到服务器
2. 清空本地信息后跳转到登录页
3. 服务器广播用户退出事件给其他用户
4. 其他用户收到事件，将退出用户从用户列表移出（如果退出的是主持人，还要清除主持人和画线信息）

基本步骤：

1. 给退出按钮注册点击事件，点击时发出退出申请，并跳登录
2. 处理服务器广播的退出事件

补充（登录拦截）：

1. 如果有昵称，有账号，说明登录了，还访问登录页，直接引导到首页
2. 如果是未登录，直接访问 /home, 应该要拦截





总结

这个项目完成了一个很有意思的小游戏

那么本项目，主要练习的核心技术栈是哪两个？

socket.io

vuex



传智教育旗下高端IT教育品牌