

CSGO猜选手系统

组长：20225943-徐越昊-计算机2207

组员：庞圣镒，高健，廖佳超，祝诗迪，张皓

引言

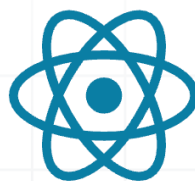
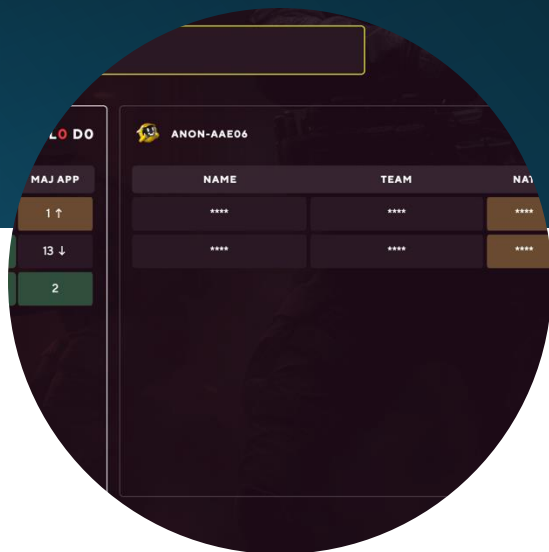
编写目的

- 本软件需求规格说明书旨在明确描述“CSGO猜选手系统”网页游戏的功能需求、性能要求、用户特征、系统约束及开发环境等内容，为后续的系统设计、开发、测试和维护提供依据。

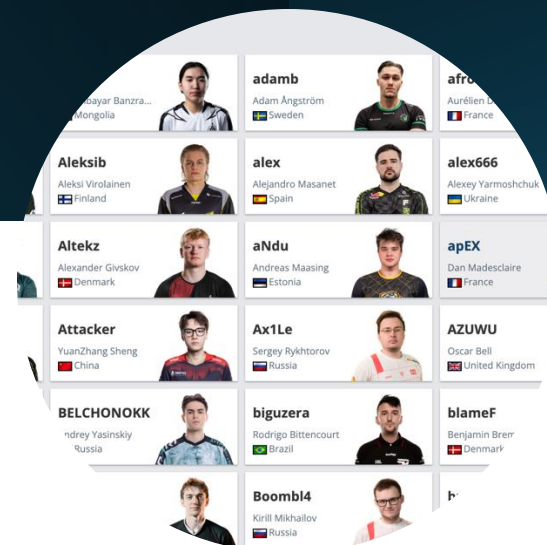
背景

- 本系统是一款结合了《Counter-Strike(CS)》游戏元素与字谜玩法的网页游戏，灵感来源于 BLAST.tv 推出的“Counter-Strike”游戏。玩家需要在限定次数内猜出一位《CS》职业选手的昵称，通过颜色提示逐步接近正确答案。该游戏旨在提升玩家对《CS》职业选手的了解，并提供娱乐性挑战

参考资料



React Native
Once, write any



- BLAST.tv 的 “Counter-Strikle” 游戏: <https://blast.tv/counter-strikle>
- React Native 框架 <https://reactnative.dev/>
- CS 选手数据集网站 <https://www.hltv.org/players>

任务概述

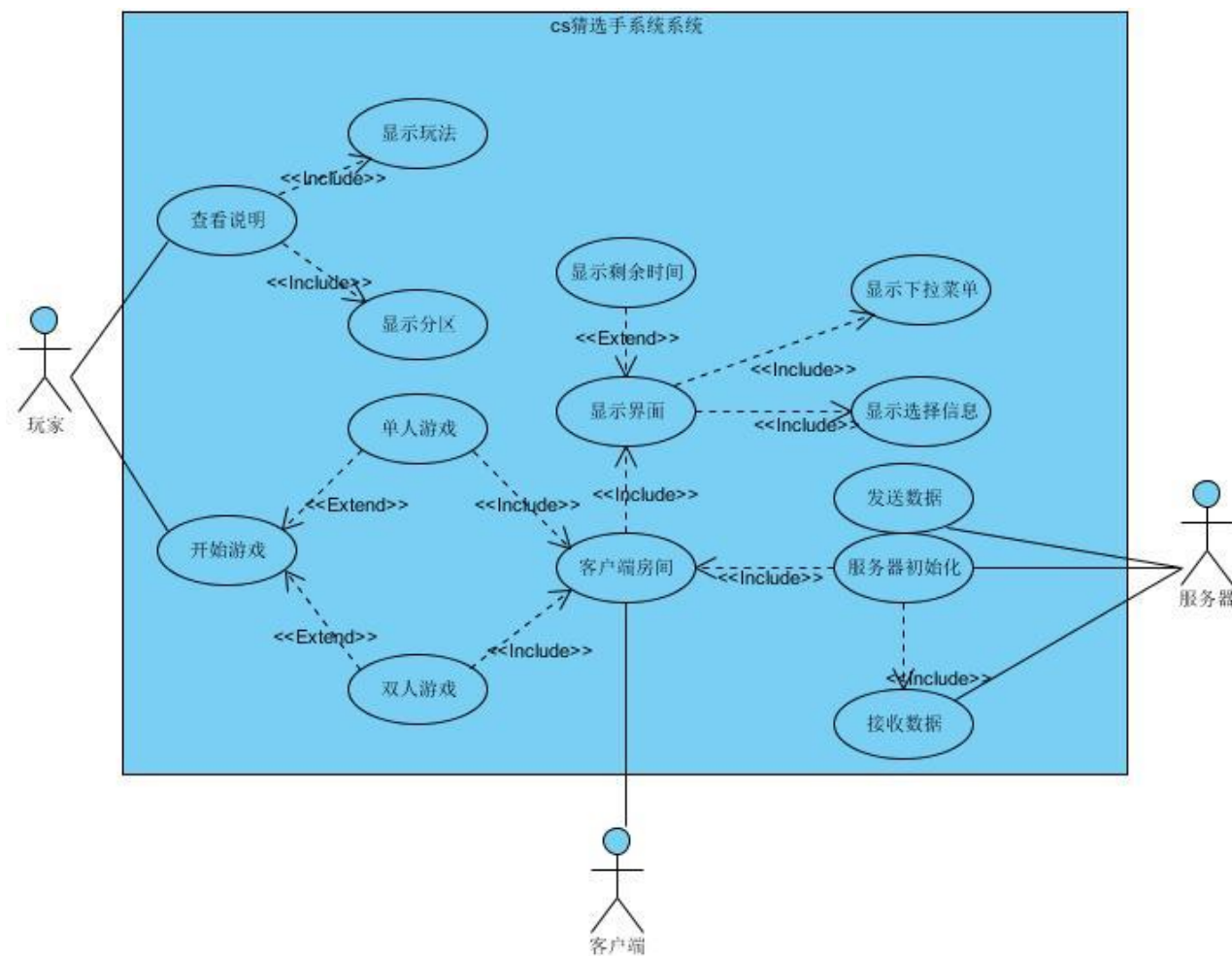
目标

- 开发一款基于网页的“CSGO猜选手”游戏，具备以下功能：
 - 可以无限制进行单人挑战，也可以进行双人对战，每次随机产生一个答案。
 - 提供颜色提示，帮助玩家逐步猜出正确答案。
 - 记录玩家的游戏数据，如猜测次数、连胜记录等。
 - 可以访问选手的hltv网页，让玩家能够更加了解该职业选手，让更多人了解 CS 比赛和职业选手。

用例分析


系统 Actor 分析及用例描述

- **玩家**：游玩游戏，查看游戏规则和分区规则，选择单人游戏或双人游戏。
- **客户端**：承载用户看到的界面，激活服务器，并传递用户与服务器之间的信息。
- **服务器**：承载客户端，由客户端请求，为联网双人对战功能提供基础，提供后端数据给客户端。




服务器用例图

(1) 服务器初始化

 初始化服务器，承载客户端，并且传递数据。用例说明如表 3.1 所示。

用例名称	服务器初始化
用例描述	初始化服务器，能够将服务器中的数据传送到客户端。
主执行者	服务器
触发条件	外部客户端请求
后置条件	承载客户端房间，接收数据
基本事件流	<div>1. 用户点击开始游戏，选择了游戏模式。</div> <div>2. 产生客户端，客户端请求服务器。</div> <div>3. 服务器承载客户端房间。</div>
异常事件流	网络条件不佳，无法开启服务器。

表 3.1 服务器初始化用例说明 

服务器用例图

(2) 接收数据

从客户端接收数据。用例说明如表 3.2 所示。

用例名称	接受数据
用例描述	服务器将接收来自承载的客户端发送的数据。
主执行者	服务器
触发条件	外部客户端发送数据
后置条件	无
基本事件流	1. 服务器开启。 2. 客户端向绑定的服务器发送数据。 3. 服务器接收数据。
异常事件流	数据未传输到服务器端。

表 3.2 服务器初始化用例说明

服务器用例图

(3) 发送数据

向客户端发送数据。用例说明如表 3.3 所示。

用例名称	发送数据
用例描述	服务器将向客户端发送数据。
主执行者	服务器
触发条件	服务器开启或者服务器接收数据并且处理之后。
后置条件	无
基本事件流	<div>1. 服务器开启或者服务器接收数据并且处理之后。</div> <div>2. 从文件中将要传递的数据取出。</div> <div>3. 向客户端传递数据。</div>
异常事件流	数据未传输到服务器端。

表 3.3 服务器初始化用例说明

玩家用例图

(1) 开始游戏

选择不同游戏模式。用例说明如表 3.4 所示。

	用例名称	开始游戏
	用例描述	开启游戏客户端
	主执行者	玩家
	触发条件	通过用户鼠标点击选择
	后置条件	单人游戏，双人游戏
	基本事件流	1. 用户在单人游戏和双人游戏两个选项中点击选择一个游戏模式。 2. 进入该游戏模式的客户端。 3. 客户端激活一个服务器，用来传递处理数据，承载客户端。
	异常事件流	客户端，服务器未正常开始。

表 3.4 开始游戏用例说明

玩家用例图

(2) 查看说明

查看游戏说明，地区分区说明。用例说明如表 3.5 所示。

用例名称	查看说明
用例描述	显示说明界面。
主执行者	玩家
触发条件	通过用户鼠标点击选择
后置条件	无。
基本事件流	1. 用户点击选择。 2. 通过数据文件，产生游戏玩法说明界面和分区说明界面。
异常事件流	说明文件未正常传输。

表 3.5 查看说明用例说明

客户端用例图

(1) 客户端房间

生成客户端房间。用例说明如表 3.6 所示。

用例名称	客户端房间
用例描述	生成客户端房间，激活服务器。
主执行者	客户端
触发条件	用户选择游戏模式。
后置条件	
基本事件流	<div>1. 用户选择游戏模式。</div> <div>2. 客户端通过识别不同的模式，开启不同的客户端房间。</div> <div>3. 不同的客户端房间，向服务器请求不同的服务器。</div>
异常事件流	无法请求服务器。

表 3.6 客户端房间用例说明

客户端用例图

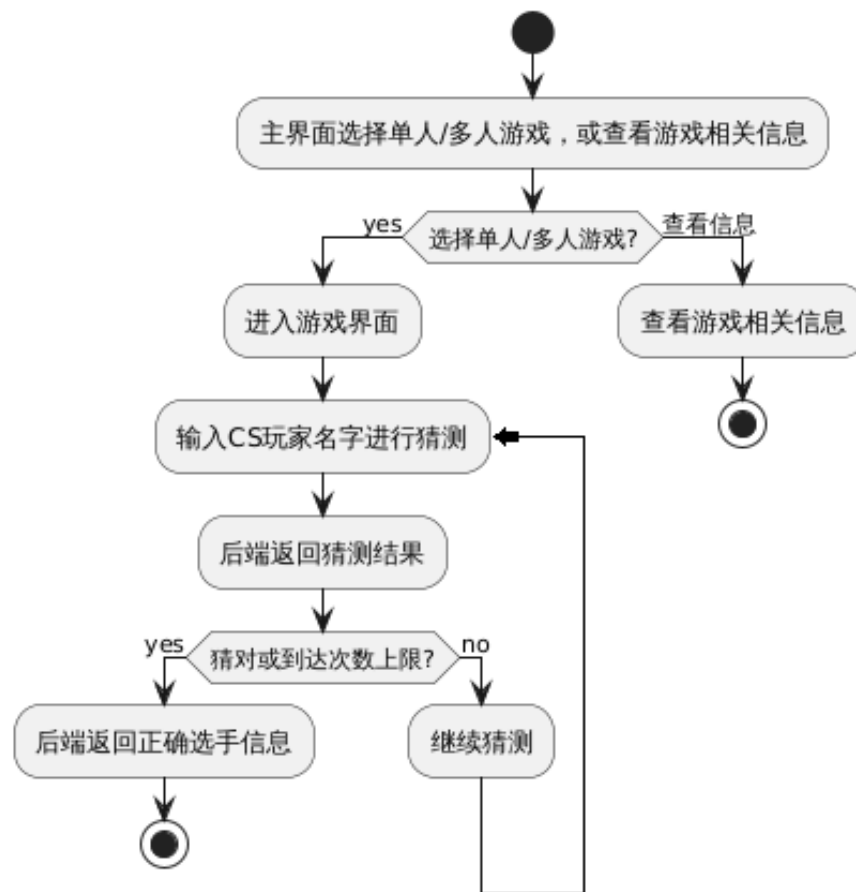
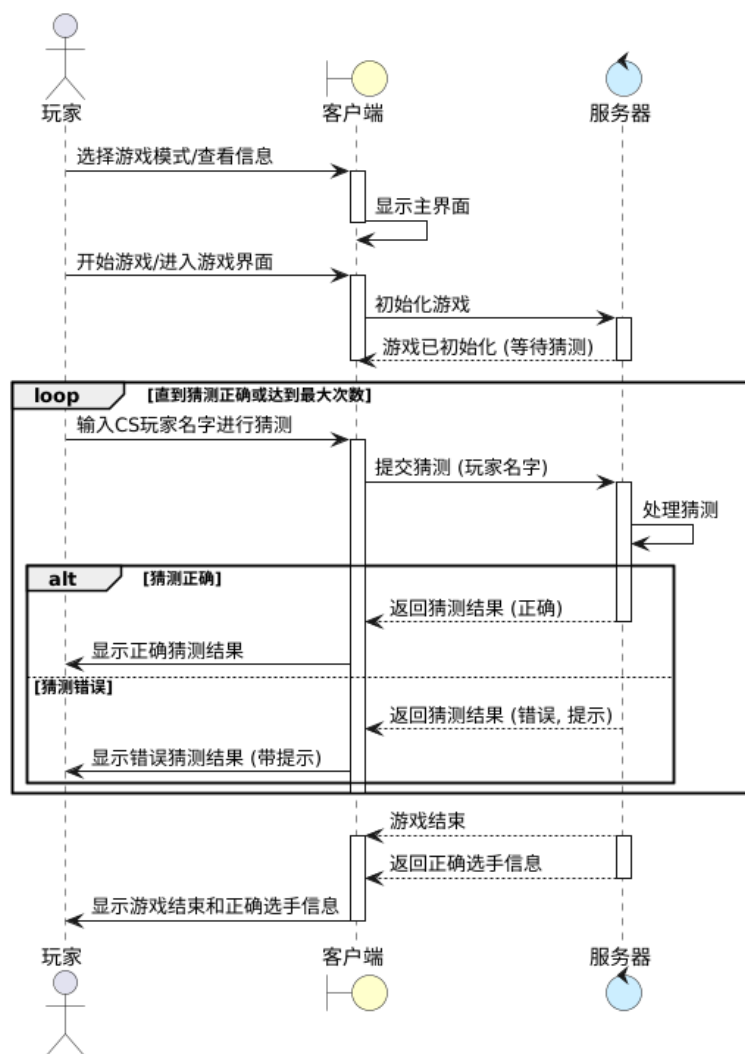
(2) 显示界面

显示前端界面。用例说明如表 3.7 所示。

用例名称	客户端房间
用例描述	生成客户端房间，激活服务器。
主执行者	客户端
触发条件	用户选择游戏成功，进入客户端游戏界面。
后置条件	无
基本事件流	<div>1. 客户端房间建立成功。</div> <div>2. 请求服务器成功。</div> <div>3. 显示下拉菜单，显示选择信息。如果是双人模式，还会显示对手模糊选择信息和剩余时间。</div>
异常事件流	<div>1. 服务器请求失败。</div> <div>2. 服务器与客户端之间的数据传输失败。</div>

表 3.7 显示界面用例说明

动态行为模型及系统流程分析



用户特点

1

核心用户：熟悉《Counter-Strike》职业选手的游戏玩家。

2

普通用户：对字谜游戏感兴趣的普通玩家。

3

技术专长：用户年龄范围广泛，具备基本的网页操作能力即可

4

预期使用频度：每天三次使用，每次十分钟至半小时。

假定和约束

游戏需在主流浏览器上正常运行。

后端使用 Node.js 处理游戏逻辑，Python 负责数据分析。

前端使用react架构实现用户界面。

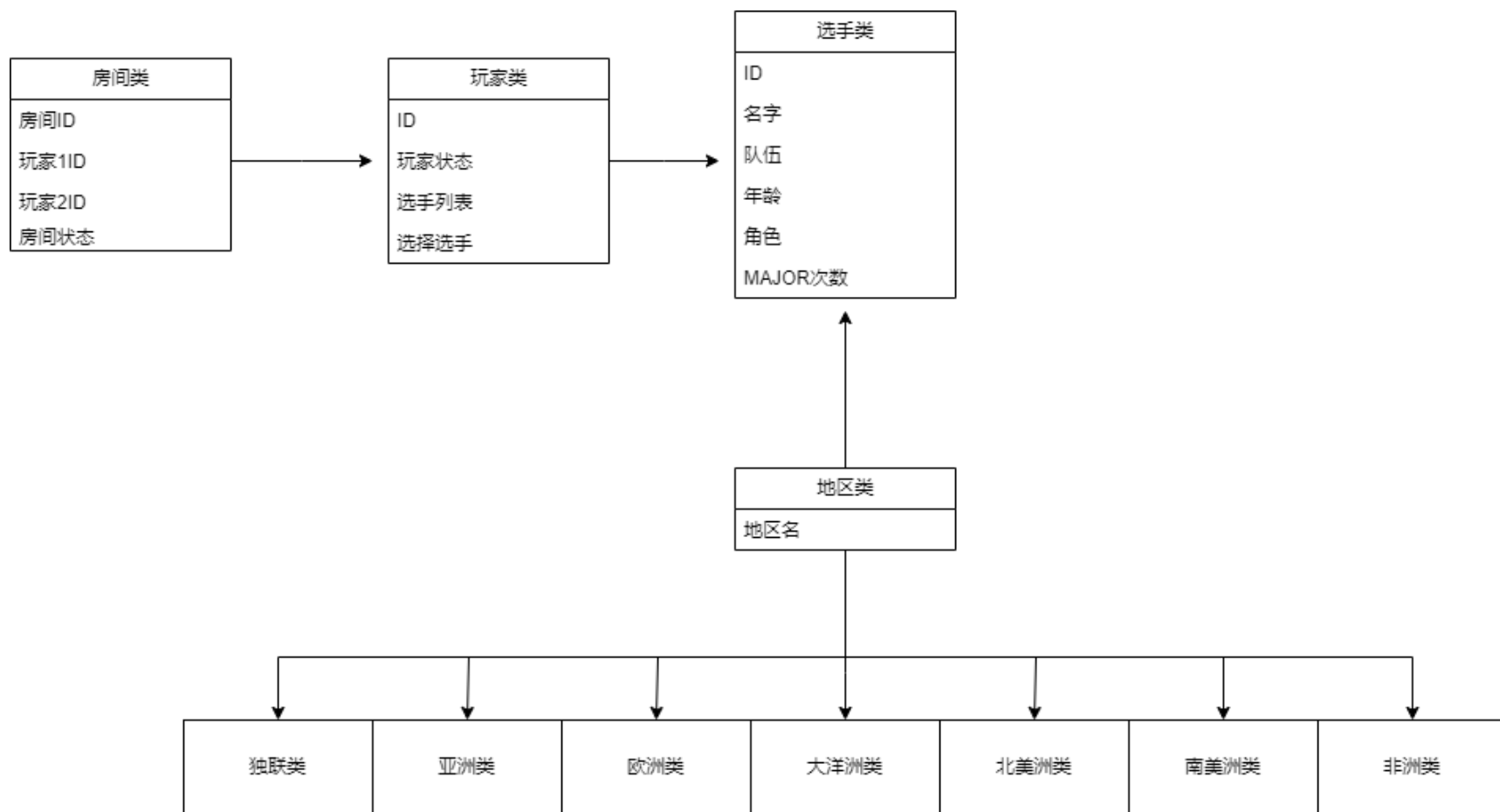
数据存储采用 JSON 文件

系统功能设计

功能模块设计

- **后端模块** 负责处理游戏的数据请求、管理游戏房间、处理玩家连接和游戏逻辑。
 - 核心服务模块
 - 房间管理模块
 - 玩家数据 API 模块
 - 实时通信模块
- **前端模块** 用于呈现给玩家双方当前信息，提交玩家数据请求，从后端接收数据并呈现。我们将使用React框架完成前端模块的编写。
 - 下拉菜单选择模块
 - 过程显示模块
 - 倒计时模块
 - 结束信息显示模块
 - 国家所属地区模块

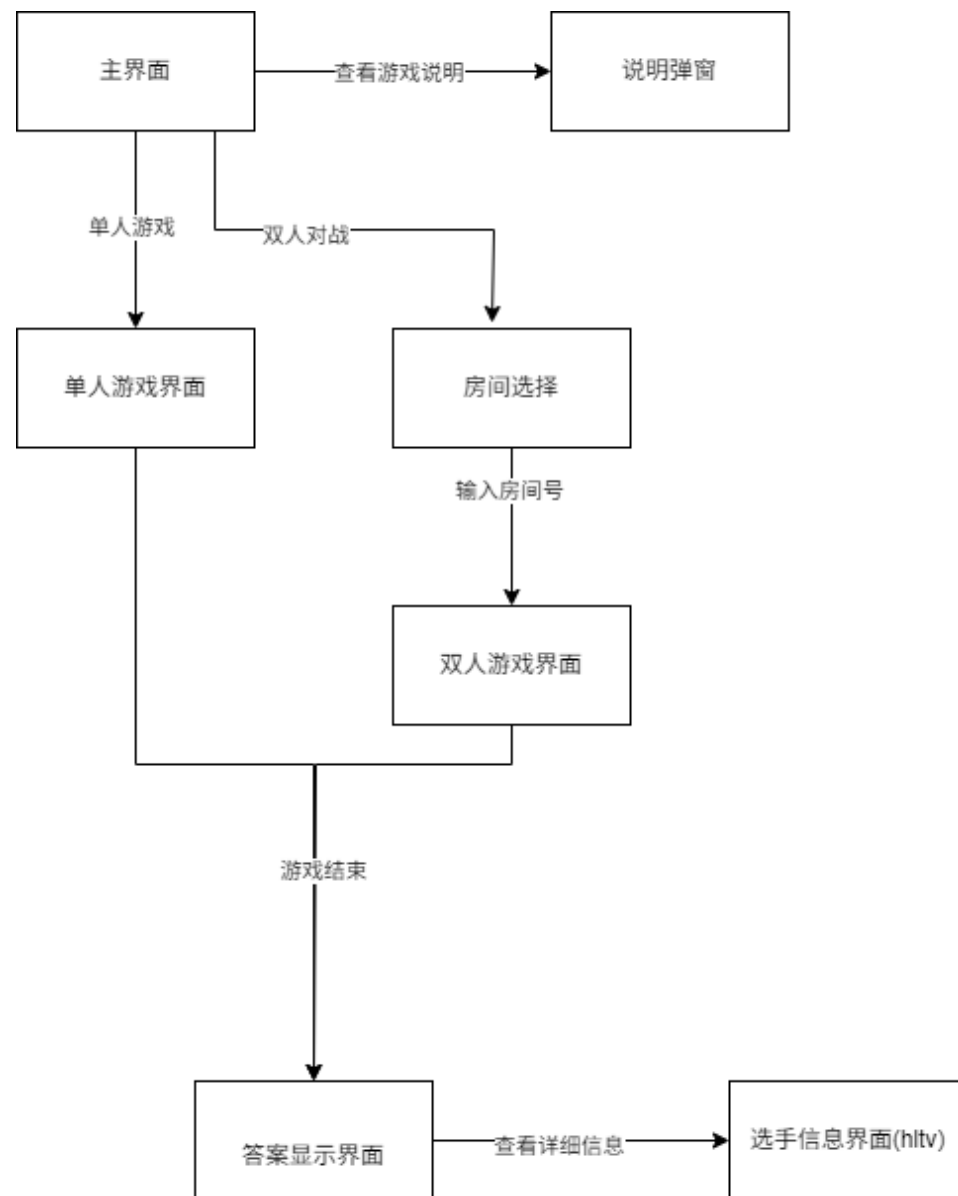
类设计



接口及过程设计

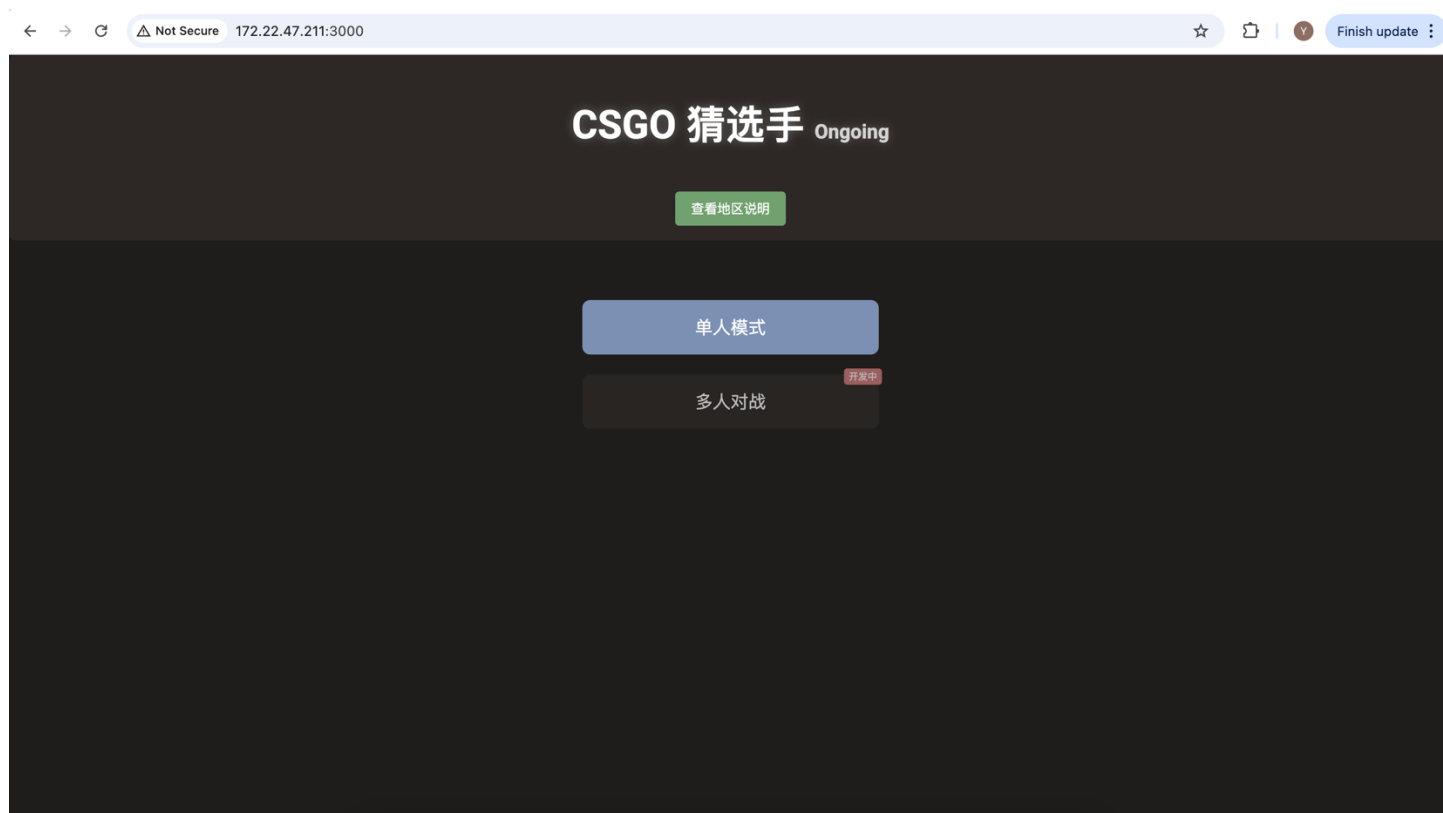
子模块名称	I(输入)	P(处理)	O(输出)
下拉菜单选择	用户输入字符串	将用户输入的字符串与所有选手名称进行比较。	返回若干对应的选手
过程显示模块	后端广播的数据	根据数据以及最终正确选手，处理相同或相似信息	返回 HTML 文件，在用户屏幕上显示
倒计时模块	房间开始信号	进行倒计时	返回倒计时时间
结束信息显示模块	游戏结束信号	根据正确选手 ID 显示相关信息	返回对应 HTML
国家所属地区模块	国家信息	寻找国家所属子类	返回国家所属地区

界面流程设计



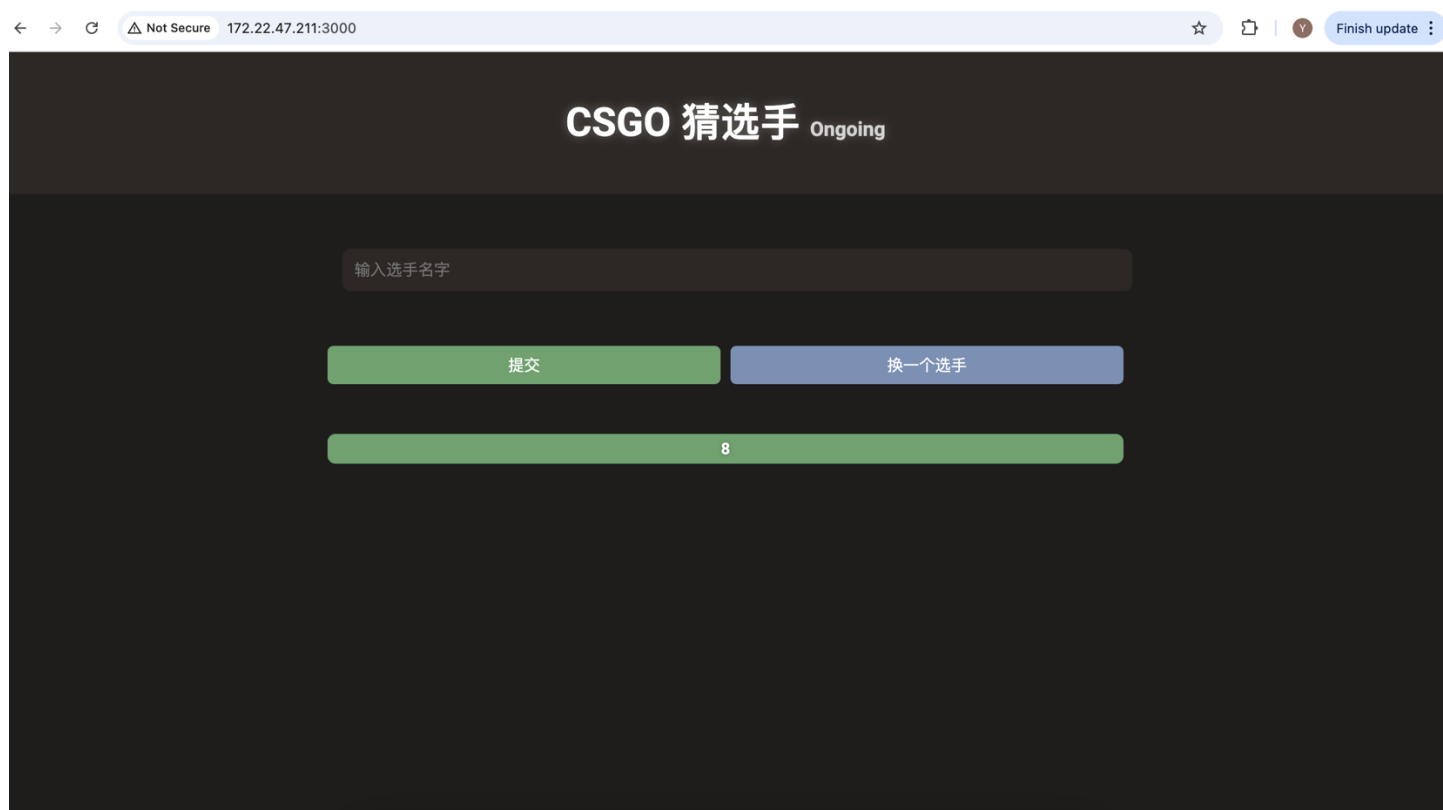
界面设计

主界面用户可以选择单人/多人游戏，或查看游戏相关信息。



界面设计

- 网页顶部有一个搜索框，提示用户可以输入 CS 玩家的名字进行搜索。双人对战的主体部分分为左右两个大的信息展示区域，每个区域展示一个玩家的猜得信息。若为单人游戏，主题部分仅有一个信息展示区域。



界面设计

- 在每个展示区域中，包含一个表格，表格有“名字”“队伍”“国家及地区”“年龄”“角色”“MAJOR次数”表头，用于展示选手的相关信息，表格中的内容以行的形式呈现不同的玩家数据。双人对战模式中将会在中间位置显示倒计时信息。

提交

换一个选手

名字	队伍	国家及地区	年龄	角色	MAJOR次数
donk	Spirit	俄罗斯	18 ↑	Rifler	2 ↓
ZywOo	Vitality	法国	24 ↓	Awper	8 ↓

界面设计

- 在搜索框中我们实现下拉菜单功能。当用户在搜索框中输入字符串时，它会显示一个与输入内容相关的玩家名称列表。每个条目包含职业选手的游戏昵称。这个功能可以帮助用户快速找到与输入内容匹配的 CS 选手姓名。



界面设计

- 游戏结束后，我们将会利用弹窗的形式将正确的选手展示给玩家。
- 下方的“查看选手资料”会将页面跳转到 hltv 官方选手信息界面。

游戏结束

选手名字:	HeavyGod
所属战队:	G2
国家及地区:	以色列
出生年份:	2003
游戏角色:	Rifler
Major次数:	1

[查看选手资料 →](#)

[下一个选手](#)[查看历史记录](#)



谢谢大家