设计研究与应用

基于 Django 和 Thrift 框架的在线联机游戏的设计与实现

牛鸿伟
（重庆师范大学，重庆 401331）

摘 要：随着互联网技术的不断发展，全球各大网络游戏纷纷抢进国内市场。项目是基于 Django 和 Thrift 框架搭建了一个 Web 在线联机网页游戏平台，具有各种特色游戏功能，以满足客户群体的需求。本文主要探讨了系统涉及的关键技术、系统功能的设计、主要功能的实现方法及原理，最后探讨了本项目使用的部署方式的优点，并总结了在生产环境中进行项目部署上线的步骤。以此来为游戏开发者提供不同的游戏设计思路和方法。

关键词：Django；Thrift；联机；游戏
中图分类号：G898.2;TP311.5 **文献标识码：**A **DOI：**10.3969/j.issn.1003-6970.2022.04.051
本文著录格式：牛鸿伟.基于Django和Thrift框架的在线联机游戏的设计与实现[J].软件,2022,43(04):177-180

Design and Implementation of Online Game Based on Django and Thrift Framework

NIU Hongwei
(Chongqing Normal University, Chongqing 401331)

【Abstract】：With the continuous development of Internet technology, the world's major online games have entered the domestic market. The project is based on Django and Thrift framework to build a Web online Web game, with various features of the game to meet the needs of customer groups. This paper mainly discusses the key technologies involved in the system, the design of system functions, the realization methods and principles of the main functions, and finally discusses the advantages of the deployment mode used in the project, and summarizes the steps of the project deployment online in the production environment. In order to provide game developers with different game design ideas and methods.

【Key words】：Django;Thrift;online;game

0 引言

现阶段我国网络游戏用户规模呈现波动上升态势，使用率保持在 50% 以上，且网络游戏市场实际销售收入规模呈现逐年增长趋势。互联网在线游戏的最主要特点是要求客户端程序能与服务器端程序及时交互，实时性很强，且要求信息的准确无误、按先后次序传递，从而达到信息的及时更新和显示。Django 框架自带强大的数据库和后台功能，配合前端的 Ajax 技术能够在客户端实时渲染数据。

基于以上社会热点，本文设计了一个基于 Django 和 Thrift 框架搭建的 Web 在线联机网页游戏平台，旨在为游戏开发者提供新颖的游戏设计思路。

1 关键技术

1.1 Django 搭建 Web 服务器

Django 框架采用了 MTV 的软件设计模式，即模型（Model），视图（View）和模板（Template），它借用了 MVC 的设计模式。使用 Django 自带 SQLite3 数据库和后台管理系统，框架目前生态完善，包含许多功能强大的第三方插件，可极大提高开发效率。

1.2 Thrift 搭建匹配系统

Thrift 作为远程过程调用（RPC）框架可以实现服务端和客户端使用不同语言进行通信，匹配系统服务器作为服务端，游戏服务器作为客户端。游戏服务器调用匹配系统服务器提供的 API 进行玩家匹配，并获取匹配

作者简介：牛鸿伟（2000—），男，四川泸州人，本科。

结果。

1.3 HTML+JavaScript+CSS+jQuery+Bootstrap 开发游戏界面

项目游戏主要通过 JavaScript 开发，包括游戏技能、对象逻辑的实现、数据渲染等；CSS 编写样式代码；利用 jQuery 简化获取 Dom 对象的方法；界面部分样式使用了 Bootstrap 提供的组件；Canvas 画布是主要绘制游戏各种图形。

1.4 Git 版本控制

使用 Git 进行版本控制可以保证有记录版本的可回溯性，其本地化版本库的特性及强大的分支能力可以有效把控软件开发项目的质量，提高管理效率，同时项目也提交至 GitHub 代码托管平台进行开源使用。

1.5 Docker 容器隔离运行环境

在 Ubuntu Server 20.04 上使用 Docker 技术创建镜像并通过镜像生成容器，每个镜像可生成多个容器，每个容器相当于一台独立的服务器，且 Docker 会在空间上进行一定程度优化，即相同镜像或容器只会存一份，以便复用。

1.6 WSS 广播同步消息

WebSocket（WS）是一种双向通信协议，在 WebSocket API 中，浏览器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。WSS 即 WebSocket 加密版本，使用 HTTPS 采用的安全机制保证 HTTP 连接的安全。当一个客户端向服务器发送消息，服务器会向所有连接的客户端发送一个广播告知客户端同步该消息。项目中（通过 django_channels）主要用于多人在线进行游戏引擎同步，包括创建玩家、位置坐标、技能、聊天等。

1.7 Redis 存储房间和玩家列表

Redis 支持数据的持久化，数据存放在内存中使得读取性能极高，适合用来存储小型数据。项目配置并使用 django_redis 进行缓存数据，通过键值对的方式存储房间号和玩家列表，将匹配成功的玩家放在同一个房间中以进行在线对战。

2 系统功能设计

系统由前后端分离实现，采用 B/S 架构模式，主要功能如图 1 所示。

（1）注册 / 登录：用户注册时系统将使用 django.contrib.auth 模块进行一定的校验，校验通过后将用户名和加密后的密码存储到数据库中，自动登录。

（2）菜单界面：包括个人模式、多人模式、排行榜、设置。用户选择模式后会跳转至选择。

（3）皮肤界面：选择皮肤后用户可进行游玩。

（4）对战界面：界面上方会显示对战时间、击败数量、毒圈扩散剩余时间。游戏地图存在障碍物和毒圈，增加了游戏的趣味性、新颖性。玩家可通过鼠标和键盘控制移动、释放技能，在多人模式下还可进行实时聊天交流，还可按下 ESC 键显示游戏小菜单。

（5）留言板：可查看按照最新时间排序，近期 30 d 内的用户留言内容。

（6）排行榜：可查看用户得分情况，排名通过分数降序排序，使用 django.core.paginator 模块进行数据分页统计。

（7）设置：可控制音量开关、修改用户资料、注销账号。

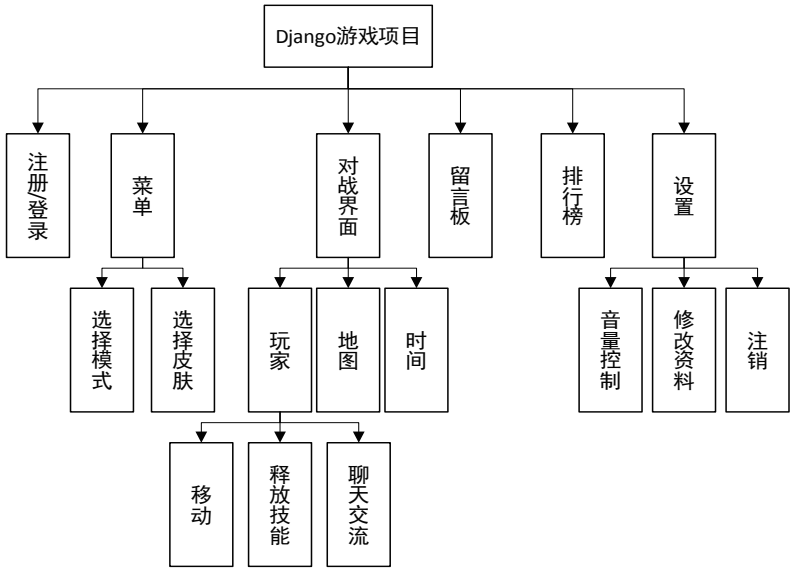

图 1 功能结构图

Fig.1 Functional structure diagram

3 主要功能实现

3.1 多人模式

通过 Thrift 框架实现了 match-client 游戏服务器、match-server 匹配系统、save-client 数据存储服务器。定义添加和删除用户的方法和参数格式：

```
namespace cpp match_service
service Match{
i32 add_player(1: i32 score, 2: string uuid, 3: string username, 4: string photo, 5: double px, 6: double py, 7: i32 total, 8: string channel_name),
i32 remove_player(1: i32 score, 2: string uuid, 3: string username, 4: string photo, 5: double px, 6: double py, 7: string channel_name)}
```

通过 thrift -r --gen py ../thrift/match.thrift 命令生成匹配系统服务器，接着在 ./main.py 文件中编写匹配池 Pool 中的方法，开启多线程执行生产者与消费者模型，每隔 1s 从匹配池中将玩家列表取出按照分数匹配算法对玩家进行匹配。其中分数匹配算法即 MATCH 方法是通过不断增加用户等待时长从而扩大分数匹配范围，每 1s 增大 50 分值，以保证每个用户都能匹配到玩家，不会导致阻塞等待。分数匹配算法的具体实现：

```
def check_match(self, t):  # 检查是否能够匹配
dt = abs(t[0].score - t[1].score)
a_max_dif = t[0].waiting_time * 50
b_max_dif = t[1].waiting_time * 50
return dt <= a_max_dif and dt <= b_max_dif
def match(self):
global player_total
while (len(self.players) >= player_total):
self.players = sorted(self.players, key=lambda p: p.score) # 分数从小到大排序
flag = False  # 标记是否匹配
check_flag = False  # 标记检查匹配
x = 0
for i in range(len(self.players) - player_total + 1): # 检查三个玩家是否匹配
for j in combinations(self.players, 2): # 利用组合来判断能否匹配
if self.check_match(j):
x += 1
if x == math.comb(player_total, 2): # 如果都能满足匹配
check_flag = True
if check_flag:
self.match_success(self.players)
self.players = self.players[:i] + self.players[i + player_total:] # 删除已匹配玩家
flag = True
break
if not flag: # 没发生匹配直接退出
break
self.increase_waiting_time() # 每个玩家等待时长 +1s
```

3.2 设置

游戏音量控制是通过 jQuery 获取音频对象并修改其 Muted 属性布尔值实现的；修改玩家资料中的头像上传是通过引入 OSS2 模块，使用 AccessKey ID 和 AccessKey Secret 申请外网访问权限，利用 SDK 方法调用 Bucket 对象提供的 put_object 方法将图片上传至 OSS 中并在数据库中更新当前图片 URL，前端调用 Getinfo 方法向后端发送 GET 请求，将返回的数据渲染到页面上；注销账号是先判断用户是否是登录状态，若是则调用利用 django.contrib.auth 模块中的 Logout 方法直接进行账号注销。

3.3 排行榜

排行榜通过分数高低进行降序排序，引入 django.core.paginator 模块中的 Paginator、PageNotAnInteger、InvalidPage，Paginator 用来实现分页功能，参数是 QuerySet 对象和每页数量，PageNotAnInteger 和 InvalidPage 是用来进行异常处理的，当页数不是整数和页数不存在或不合法时将进行异常处理，后续代码会继续执行。

3.4 留言板

前端使用 jQuery 封装的 Ajax 技术提交 POST 请求将留言数据提交给 Django 后台进行处理，同时通过 GET 请求向后台请求数据，达到实时渲染的效果。其中日期格式需要和后台管理系统相对应，在 settings.py 配置中需修改时区 TIME_ZONE = 'Asia/Shanghai'，后端使用 TimeZone 对象提供的 Now 和 LocalTime 方法获取当前时间和格式化时间。

4 部署上线

使用 Nginx 和 uWSGI 在 Ubuntu Server 20.04 的 Docker 容器中部署 Django 项目。浏览器将 HTTP 请求发送给 Nginx，Nginx 具备优秀的静态内容处理能力，将静态文件直接返回给浏览器，仅将动态内容转发给 uWSGI 服务器，uWSGI 服务器将动态内容转换为 WSGI 协议接受的格式，并发送给 WSGI，WSGI 根据请求调用程序中的

函数，最后将处理结果依次转换并返回给 WSGI—Nginx—浏览器，实现高并发，提升系统性能，如图 2 所示。

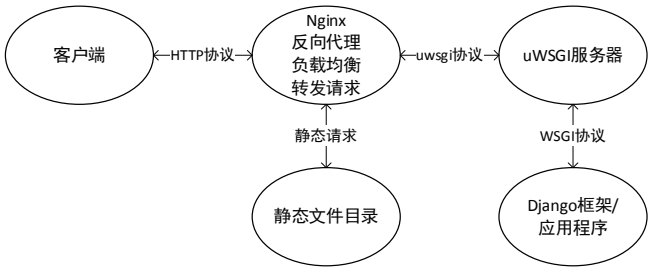

图 2 部署流程图

Fig.2 Deployment flowchart

部署步骤：

（1）增加容器的映射端口 80 与 443：docker run -p 20000:22 -p 8000:8000 -p 80:80 -p 443:443 –name CONTAINER_NAME -itd django_lesson:1.1

（2）修改 nginx.conf 配置文件并启动：sudo /etc/init.d/nginx start

（3）修改 uwsgi.ini 配置文件并启动：uwsgi --ini scripts/uwsgi.ini

（4）启动 django_channels（WebSocket 通信）：daphne -b 0.0.0.0 -p 5015 acapp.asgi:application

（5）启动匹配服务：./main.py

5 结语

游戏开发可利用很多技术实现，本文以开发网页游戏作为例子，介绍了基于 Django 和 Thrift 框架的在线联机游戏的设计与实现。其中详细介绍了关键技术和主要功能实现两大板块，主要功能实现中的多人模式匹配系统是在线联机游戏的核心，涉及了多线程、消息队列、生产者与消费者模型等知识进行开发。且该匹配系统是对现有匹配系统的重构和创新，具有一定的实用价值，将不断利用所学技术为开源社区作出代码贡献。

参考文献

[1] 杨华.互联网在线游戏的研究与实现[D].沈阳:沈阳工业大学,2002.

[2] 邱红丽,张舒雅.基于Django框架的Web项目开发研究[J].科学技术创新,2021(27):97-98.

[3] 王真.版本控制工具在软件开发项目管理中的应用:以GIT为例[J].项目管理技术,2020,18(6):131-134.

[4] 范文星.基于Django的网络运维管理系统的设计与实现[J].计算机科学,2012,39(S2):175-177.

[5] 杨小娇.轻量级高并发Web服务器的研究与实现[D].南京:南京邮电大学,2014.

[6] 卓武扬.网络游戏产业研究[J].江西财经大学学报,2004(1):51-55.

[7] 田翠珍.基于Thrift框架RPC的研究与实现[J].信息与电脑(理论版),2016(1):83-84.

[8] 曾超宇,李金香.Redis在高速缓存系统中的应用[J].微型机与应用,2013,32(12):11-13.

…… 上接第93页

利用新时代互联网新技术实现校园教育教学和管理服务的创新。积极利用云计算、大数据等新技术，创新构建教学资源平台和服务管理平台，为学校“双一流”建设提供强有力的技术支撑。

参考文献

[1] 韩锡斌,崔依冉,程建钢.以系统化思维推进信息化教学改革[J].中国教育网络,2021(7):24-26.

[2] 赵慧臣,彭梦甜.高校教师实施混合式教学问题与对策的质性研究[J].数字教育,2022,8(1):32-39.

[3] 刘娜.“互联网+”背景下线上线下混合式教学探讨[J].工业和信息化教育,2022(2):45-50.

[4] 于潇.数字时代高校混合式教学改革的SWOT分析[J].吉林省教育学院学报,2022,38(1):107-112.

[5] 贾云翔.基于超星学习通平台的混合式教学模式的构建与应用[J].山西警察学院学报,2022,30(1):104-109.

[6] 熊苗,赵幸子.高职院校线上线下混合式教学策略[J].华东纸业,2022,52(1):31-34.

[7] 孙琪,王鹏程.基于混合式教学模式的多维动态考核评价体系研究[J].浙江万里学院学报,2022,35(1):101-106.

[8] 姜华,姜锐.高职院校教师混合式教学能力的现状分析与提升路径[J].现代商贸工业,2022,43(5):147-149.

[9] 韩锡斌,周潜.技术赋能,推动职业教育体系重构[J].中国教育网络,2020(8):31-34.

[10] 贾云翔.基于超星学习通平台的混合式教学模式的构建与应用[J].山西警察学院学报,2022,30(1):104-109.