

田 忠

18992692056 | 4 2 2 3 4 2 1 5 8 @ q q . c o m

专业技能

- 熟练掌握 Python、Java 编程语言，有着优秀的编码习惯并具备相关语言调优能力。
- 熟悉理解 HTTP 请求、响应的全流程。熟悉常见Web框架的使用，如 Flask、Django、Spring。基于 Socket 实现过 原生 HTTP Server 和 MVC Web 框架。
- 熟悉理解 操作系统、计算机体系结构 等通用知识，如 进程、线程、协程 等，理解 垃圾回收 机制。
- 熟练掌握 MySQL、MongoDB 等数据库的使用。理解 数据库事务、索引。掌握 ORM (对象关系映射) 并理解生成 Sql 性能局限性，能够利用性能分析工具生成性能占比图定位性能瓶颈进行调优。了解读写分离、分库分表等，具备相关优化能力。
- 熟练掌握 Redis 理解其 基本数据结构 和 应用场景 等，在多个项目使用其进行 数据缓存、跨进程数据共享 进而提高性能。
- 熟悉理解 消息队列 的 常见用途，熟悉 RabbitMQ、Kafka 等 常见消息中间件 的使用方式和应用场景。
- 熟悉理解 网络编程 相关知识，如 HTTP/HTTP(S)、TCP/UDP 等。熟悉常见的 网络安全问题，如 XSRF(跨站请求伪造)、XSS(跨站脚本攻击)、SQL 注入 等并能针对此类问题做出正确的防范。
- 熟练掌握 Linux/Mac 平台开发维护、Git 版本控制、Docker 虚拟化技术，能完成服务器开发环境搭建、持续集成、持续部署等加快项目迭代。
- 熟练掌握 JavaScript、HTML、Canvas 等前端领域技术，基于 原生JS 自制过游戏引擎 在此基础上实现过飞机大战、贪吃蛇、FlappyBird 等游戏。
- 熟练掌握常用 数据结构 如 链表、栈、队列 等并通过学习原理实现过 HashTable。
- 自我学习能力强，有强烈的责任心，具备良好的沟通能力和团队合作能力。

工作经历

北京悉之智能科技有限公司 | 后端开发工程师

2020.12 - 2023.01

海外版App TutorEva

技术栈: Python、Django、Go、Gin、MySQL、Redis、RabbitMQ、Celery、Docker、Pydantic、AWS

- 参与了公司核心产品后端从 0 到 1 的架构设计和建设过程。主要负责后端项目的模块业务设计、开发、联调、上线、服务维护。作为开发主力独立负责了 App 后端用户模块、活动模块、消息中心等中枢模块。
- 参与设计开发了后端开发工具集。主要包括中间件(鉴权、日志、异常)、入参和出参校验方式、API 开发模板代码的生成、YAPI 接口文档自动上传。开发实现了相关基础服务主要包括 AWS 服务(邮件、Kinesis 流数据处理、S3 对象存储)、用户 Session 和 验证码服务、接口限流服务等，供后端团队使用。
- 独立负责了用户模块，除了实现常见用户功能以外承担并接入了第三方登录 Apple 和 Google，设计开发了游客模式。在此基础上实现了用户 Ticket 和 邀请码的创建、发放、分配、使用、排行榜和定制化奖励方案等，帮助平台带来上百万用户。
- 独立负责了消息中心模块，负责底层接入了 Google FCM(Firebase Cloud Messaging)，在 App 内部完成了模板消息的配置、封装、订阅、推送，支持自定义多种消息类型。并在开发流程中协同原生 iOS 和 Android 开发设计交互方案，统一了 App 内部跳转规则支持原生页内部和外部 H5 页面。快速接手 App 后台项目并重构了后台消息的逻辑，从代码逻辑和数据库查询层面进行优化提升了 20 倍响应速度。
- 独立负责了活动模块，在 App 后台实现了活动的创建、修改、方案配置、发布、下架，在 App 内部实现了活动信息列表、活动奖励逻辑，使得产品运营能自定义配置活动方案内容，提高了 300% 的活动迭代效率。优化了 App 首页加载逻辑，删除了不必要的请求数据，将 App 通用数据统一入口预先加载，使 App 首页加载速度提高了 150%。
- 参与开发了商城模块，在 App 后台实现虚拟钻石、皮肤、体验卡、Ticket包等商品配置管理，包括套餐的商品名、价格、折扣、有效期等。在 App 内部参与实现了类王者荣耀游戏皮肤商城系统，包括钻石充值，皮肤、Ticket包、体验卡等购买、使用、虚拟人物换肤、限时过期等逻辑。在后台区分了内部测试和实际生产数据，并在 App 人物换肤上兼容多个版本逻辑保证了用户使用体验。

K12App 悉塔助教

技术栈: Python、Django、MongoDB、Docker、Celery、Docker、DRF、Redis

- 项目介绍：学生使用 App 做题，根据过往做题数据分析进行个性化推题，智能答疑。针对学生情况进行定制化AI自主学习、AI任务学、AI讲解进行互动式的讲解，通过自主学习利用智能错题本和数据学习周报，帮助学生提高学习效率。
- 参与项目优化过程，针对原有项目结构，框架进行二次开发，简化数据交互层流程，重构数据校验层方式，将不同功能层逻辑隔离、通用工具模块抽象、大量重复代码封装复用，提升项目迭代效率和可维护性。
- 参与 App 多版本的开发迭代，包括学习计划、假期计划、单题讲解、AI课、AI讲解、AI伴学计划、个性化学案、智能错题本、教辅答疑、学习记录、学习报告等，确保数据稳定可靠，功能模块稳定迭代。设计实现了 App 后端与其他多个端事件注册异步回调机制，保证了多端数据交互稳定性和准确性。
- 根据业务需求，设计接口测试用例，采用 Locust 压测并对测试数据分析、生成压测报告、定位性能瓶颈、提出解决方案。

技术栈: Java、SpringMVC、MyBatis、Mysql、Redis

- 入职 一周 便可熟悉并 快速上手 公司内部项目框架代码完成日常所需开发任务。熟悉项目开发流程后, 能够 独立负责开发项目, 完成后台的管理功能的 设计、开发、测试、部署。
- 使用 SpringMVC 设计实现 RESTful API, 协调生成相关 API 接口文档, 配合前端调优完成 JSON 数据格式对接, 加强了接口的复用性和易用性。

个人项目 | Github

HTTP Server + MVC Web

- 底层基于 **Socket API** 通信实现 **原生 HTTP 服务器**。
- 通过 **解析/拼接 HTTP** 报文信息, 接收客户端请求返回相应的响应。
- 利用 **多线程** 并发处理请求, 通过封装接口并能支持 **WSGI协议**。
- **(M)** Model 框架层基于 MySQL 实现了 **自制 ORM** (对象关系映射) 具有 **Create (增加)、Read (读取)、Update (更新) 和 Delete (删除)** 的功能。
- **(V)** View 视图层利用 **单例模式** 生成加载器通过 **Jinja2** 来渲染模板。
- **(C)** Controller 控制层利用 **高阶函数** 和 **字典 路由注册、分发**。

2D 游戏引擎

- 采用 **OOP思想** 使代码的高内聚低耦合, 实现了游戏对象管理 **场景抽象、静态对象、动态对象、游戏动画等**。
- 在游戏内部实现资源 **预加载** 提高游戏运行时的性能, 在开发过程中实现了 **事件注册机制** 优化了开发体验并加速开发效率。
- 以 **CodeGen** 的方式使游戏可 **自动配置**。在 Debug 过程中动态更改游戏内部参数, 如在运行过程中, 修改游戏FPS, 实时暂停游戏解决了开发期手动反复调参效率低下的问题。

FC 模拟器

- 采用 **TDD** 方式, 实现了 **FC** 所使用的 **CPU6502** 的整套指令集的模拟。
- 解析处理了 **NES**, 模拟了 **CPU/PPU** 在运行时处理了图案表将虚拟显存映射到虚拟显示屏。
- 模拟 **轮转任务** 实现按扫描线渲染, 采用 **键盘输入** 模拟手柄输入控制。

教育经历

延安大学

2017.09 - 2021.06

数学与计算机科学学院 物联网工程 本科