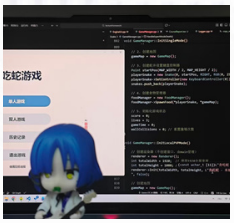


高级语言程序设计（进阶）
2025-2026 学年第一学期
项目作业展示

贪吃蛇游戏

姓名：李天成

学号：2451367

邮箱：2650611142@qq.com

项目简介

本项目旨在开发一款基于 EasyX 图形库的综合性贪吃蛇游戏系统。该系统不仅复现了经典的基础玩法，更通过引入多元化的游戏模式极大地丰富了交互体验，具体涵盖了从入门到高级的单人挑战、本地双人同屏竞技、基于 TCP 协议的网络联机对战以及人机对抗（PvE）等七种模式。项目构建了完整的图形用户界面（UI），集成了主菜单、实时游戏场景及具备筛选与分页功能的历史记录查看模块，并实现了基于 CSV 格式的数据持久化存储，确保了游戏数据的完整性与可追溯性。

设计思路

在系统架构设计层面，本项目严格遵循 Model-View-Controller (MVC) 架构模式，以实现业务逻辑、数据状态与图形渲染的有效分离。模型层（Model）由 Snake、GameMap 和 FoodManager 等核心类构成，负责维护游戏内的实体状态与规则逻辑；视图层（View）通过 Renderer 类封装 EasyX 绘图接口，专注于视觉呈现；控制层（Controller）则负责处理输入指令与决策生成。项目在控制逻辑中深入应用了策略模式，通过定义 IController 纯虚接口，蛇实体对象能够与具体的控制策略解耦。这种设计使得系统能够根据当前的游戏模式，利用工厂思想动态实例化并组合相应的控制器，从而在不修改核心代码的前提下，灵活支持多种对战形式。

此外，项目引入了有限状态机（FSM）来管理应用程序的生命周期，确保系统在菜单、游戏进行、暂停及结算等状态间流转的逻辑严密性。在人工智能模块的实现中，采用了广度优先搜索（BFS）算法进行路径规划，使电脑对手具备寻找最短路径及规避障碍的智能决策能力。整体设计充分体现了面向对象编程中高内聚、低耦合的原则，为后续的功能扩展与维护奠定了坚实基础。

实现难点

本项目的核心技术挑战主要集中在三个维度：首先是控制系统的抽象与解耦，即如何设计统一接口以兼容键盘、网络及 AI 等多种异构输入源，并确保多态环境下的资源安全；其次是 AI 算法的鲁棒性；以及如何设计一个五色莫兰迪配色的UI，并且设计出胶囊式可变按钮的同时大幅降低渲染开销。最后是网络通信的同步机制，需解决数据序列化问题，并消除网络延迟对跨客户端逻辑状态一致性的影响。

解决方式

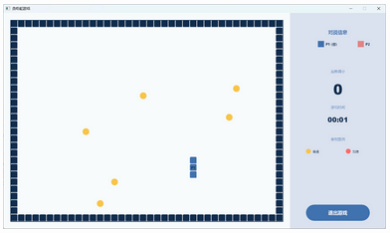
1.基于策略模式的控制解耦：为解决多源输入问题，系统定义了 IController 纯虚基类作为统一的决策接口，将具体的控制逻辑封装于 KeyboardController、AIController 等派生类中。同时在 GameManager 中引入工厂思想，根据游戏模式（GameMode）动态装配相应的控制器组合，实现了“开闭原则”，在不修改核心逻辑的前提下灵活扩展新的控制方式。

2.莫兰迪风格的 UI 重构与交互优化：为提升视觉体验与交互一致性，项目从底层重构了 UI 系统。采用五色莫兰迪配色方案建立清晰的视觉层级，并通过“拼图法”实现了 EasyX 原生不支持的胶囊按钮组件。在渲染层面，将墙体绘制算法从 O(n) 优化至 O(1)，大幅降低绘图开销。同时，重新设计了侧边栏布局与双人计分板，确保了多模式下的信息对称与操作反馈的统一性。

3.应用层协议与逻辑帧同步：网络模块中，自定义了轻量级的应用层协议结构体 GamePacket，用于封装方向指令与状态信息，解决了数据的序列化传输。为了保证公平性与一致性，采用了基于锁步思想的同步机制。在主循环中，本地客户端在发送自身操作后，必须等待接收到对手的指令包（或确认信号）后才推进下一帧的逻辑更新。这种机制确保了双方客户端在任何时刻的游戏状态（保持严格一致，有效规避了因渲染帧率不同步导致的逻辑分叉。

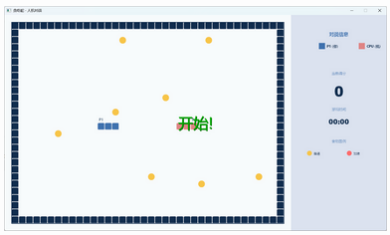
实现成果展示

首页（背景贪吃蛇会随机移动）



单机游戏界面

历史记录



双人游戏界面

房间选择界面

多人游戏准备

心得体会

通过本次贪吃蛇项目的开发，我深刻体会到了架构设计与交互美学并重的重要性。在代码层面，MVC 架构的实践让我明白了“关注点分离”的威力，将业务逻辑与视图渲染解耦，不仅让代码结构清晰，更使得后续的多模式扩展变得游刃有余。而在 UI 设计上，我摒弃了“边写边画”的随意性，转而在编码前就确立了莫兰迪五色方案与扁平化设计语言。这种前置的规划极大地提升了开发效率与成品质量。统一的胶囊按钮组件与克制的配色，不仅解决了视觉杂乱的问题，更通过一致的交互反馈增强了用户体验。这次经历让我认识到，优秀的软件不仅需要健壮的内核，更需要统一且富有美感的设计语言来支撑。