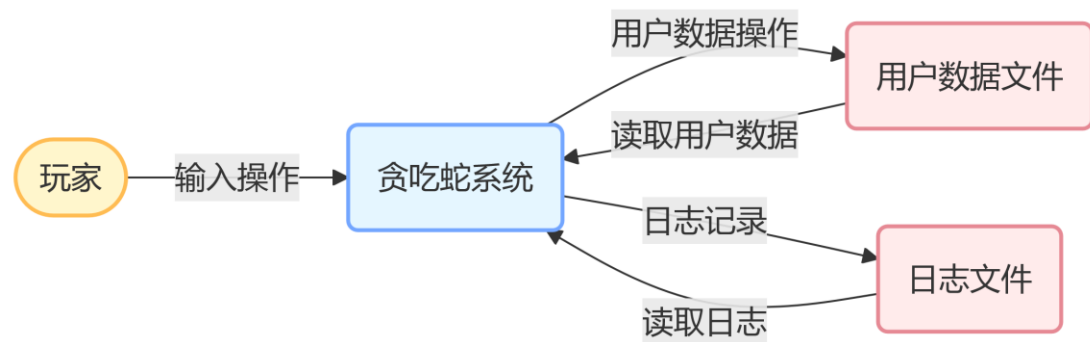


西北师范大学计算机科学与工程学院学生实验报告

学 号	202231607232	姓 名	牛波			
专 业	计算机科学与技术（卓越）	班 级	一班			
课程名称	软件工程	课程类型	必修			
实验名称	结构化软件需求分析与建模					
实验目的： 自主学习软件建模工具； 自主学习软件项目 WBS 方法： 练习结构化软件需求建模技术。						
实验内容： ProcessOn 软件功能简介： <p>ProcessOn 是一款基于云端的专业绘图工具。其功能丰富，支持绘制思维导图、流程图、UML 图等多种图形。提供海量模板，能激发灵感，提升效率，且支持个性化定制，可自由设计风格样式。支持多人实时在线协作，多端同步，历史版本可回溯。文件安全有保障，采用数据加密 + 分布式存储。还能一键转化为大纲、Word、PPT 等，支持多种格式导入导出。此外，AI 功能可一键生成图表，降低创作门槛。</p> 贪吃蛇整个数据流图及建模： 数据流图： 顶层数据流图：						



解释：

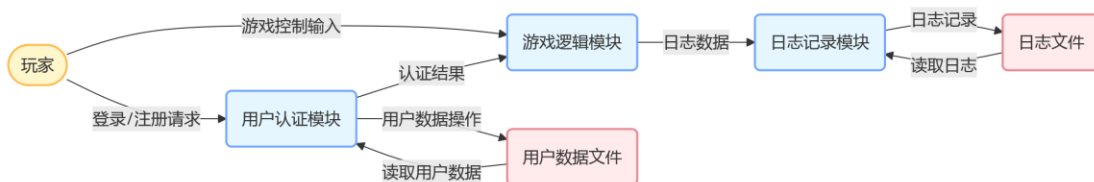
玩家：作为外部实体，向贪吃蛇系统输入操作，如登录、注册、游戏控制等。

贪吃蛇系统：核心处理模块，负责处理玩家的输入，与用户数据文件和日志文件进行交互。

用户数据文件：存储用户的基本信息，如用户名、密码等，系统可对其进行读写操作。

日志文件：记录游戏相关的日志，如游戏得分、时长等，系统会写入日志，也可能读取日志用于展示。

第一层数据流图：



解释：

玩家：发起登录 / 注册请求给用户认证模块，同时输入游戏控制指令给游戏逻辑模块。

用户认证模块：处理玩家的登录和注册请求，与用户数据文件交互，验证用户信息，并将认证结果传递给游戏逻辑模块。

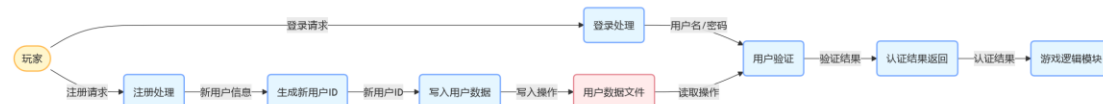
游戏逻辑模块：负责游戏的具体逻辑，如蛇的移动、碰撞检测等，接收用户认证模块的认证结果，将游戏相关的日志数据传递给日志记录模块。

日志记录模块：接收游戏逻辑模块的日志数据，将其写入日志文件，也可读取日志文件用于展示。

用户数据文件：存储用户信息，供用户认证模块读写。

日志文件：存储游戏日志，由日志记录模块读写。

第二层数据流图 - 用户认证模块细化：



解释：

玩家：发起登录和注册请求。

登录处理：接收玩家的登录请求，提取用户名和密码传递给用户验证模块。

注册处理：接收玩家的注册请求，将新用户信息传递给生成新用户 ID 模块。

用户验证：从用户数据文件读取用户信息，验证用户名和密码是否匹配，将验证结果传递给认证结果返回模块。

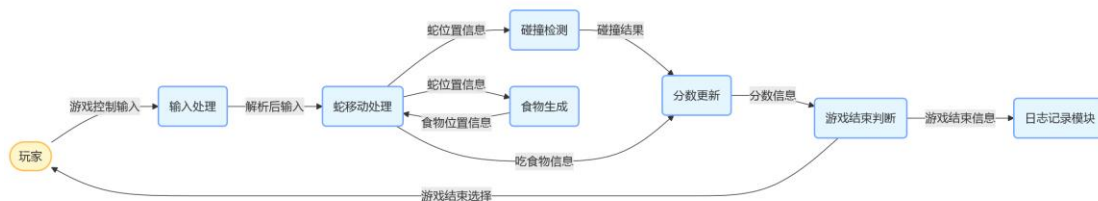
生成新用户 ID：根据用户数据文件的现有信息生成新用户的 ID。

认证结果返回：将用户认证的结果返回给游戏逻辑模块。

写入用户数据：将新用户的信息写入用户数据文件。

用户数据文件：存储用户信息，供用户验证和写入用户数据模块操作。

第二层数据流图 - 游戏逻辑模块细化：



解释：

玩家：输入游戏控制指令。

输入处理：解析玩家的输入，将解析后的输入传递给蛇移动处理模块。

蛇移动处理：根据输入更新蛇的位置，与食物生成模块交互获取食物位置，将蛇的位置信息传递给碰撞检测模块。

碰撞检测：检测蛇是否发生碰撞，将碰撞结果传递给分数更新模块。

分数更新：根据碰撞结果和吃食物信息更新分数，将分数信息传递给游戏结束判断模块。

游戏结束判断：根据分数和碰撞结果判断游戏是否结束，将游戏结束信息传递给日志记录模块，同时将游戏结束选择反馈给玩家。

食物生成：随机生成食物的位置，将食物位置信息传递给蛇移动处理模块。

日志记录模块：接收游戏结束信息，记录游戏日志。

数据字典：

数据存储：

用户信息表

名称：用户信息表

组成：id + username[32] + password[32]

存储方式：二进制文件（users.dat）

访问频率：登录 / 注册时读写

安全要求：仅用户注册、登录相关操作访问

游戏日志

名称：游戏日志

组成：id + username + 开始时间 + 结束时间 + 时长(s) + 得分

存储方式：文本文件（userlog.txt）

访问频率：游戏结束时写入，查看日志时读取

安全要求：追加写入，查看时读取

蛇数据

名称：蛇节点

组成：x + y + next（指针）

存储方式：动态内存分配

访问频率：游戏运行时频繁读写

安全要求：程序运行时管理，结束后释放内存

处理逻辑：

登录验证

处理名：登录验证

输入：用户名、密码

输出：验证结果（成功 / 失败）

逻辑：读取 `users.dat`，检查用户名与密码是否匹配

触发条件：用户选择登录操作

注册用户

处理名：注册用户

输入：用户名、密码

输出：无（仅注册成功提示）

逻辑：检查用户名唯一性，若唯一则将 `id`、用户名、密码写入 `users.dat`

触发条件：用户选择注册操作

生成食物

处理名：生成食物

输入：无

输出：食物坐标 (x, y)

逻辑：随机生成坐标，检查不与蛇身重叠，若重叠则重新生成

触发条件：游戏开始或蛇吃掉食物后

蛇移动

处理名：蛇移动

输入：移动方向 (U/D/L/R)

输出：碰撞结果 (0= 正常, 1= 撞墙, 2= 咬到自己)

逻辑：

根据方向更新蛇头坐标，检查边界（撞墙则返回 1）。

若蛇头坐标与食物重合，增加得分并重新生成食物；否则删除蛇尾。

检查蛇头是否与身体重合（咬到自己则返回 2）。

触发条件：定时触发（每隔 `sleeptime` 毫秒）

游戏结束

处理名：游戏结束

输入：碰撞原因 (1/2/3)、得分

输出：游戏状态（重新登录 / 再来一局 / 退出）

逻辑：

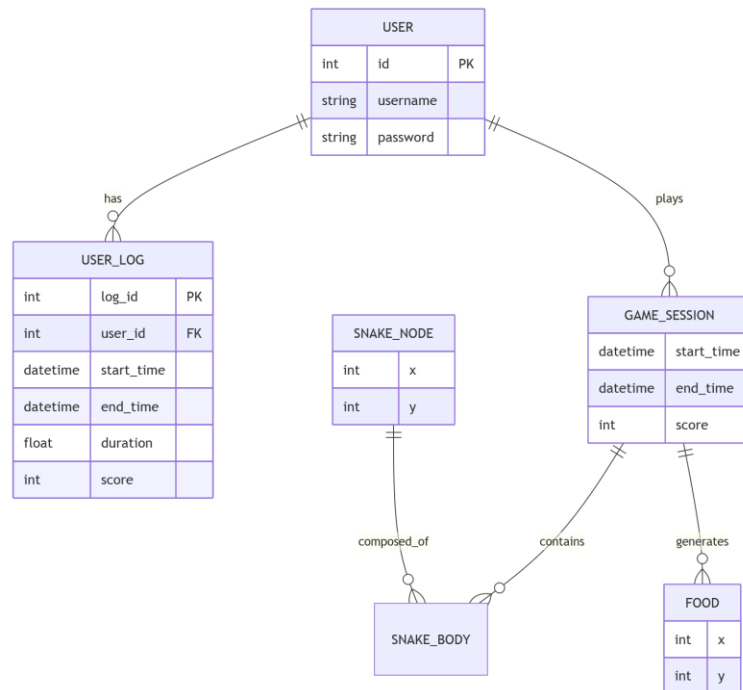
写入游戏日志 (`userlog.txt`)。

显示结束原因（撞墙、咬到自己、主动退出）和得分。

根据用户选择返回对应状态（MENU/PLAY/EXIT）。

触发条件：撞墙（原因 1）、咬到自己（原因 2）、主动退出（原因 3）

实体联系图：



解释：

USER（用户）

属性：用户 ID（主键）、用户名、密码

关系：与用户日志（**USER_LOG**）和游戏会话（**GAME_SESSION**）一对多关联

USER_LOG（用户日志）

属性：日志 ID（主键）、用户 ID（外键）、开始时间、结束时间、游戏时长、得分
作用：记录用户每次游戏的详细信息，通过 `user_id` 与用户关联

SNAKE_NODE（蛇节点）

属性：X 坐标、Y 坐标

结构：蛇身体由多个 **SNAKE_NODE** 通过链表组成（**SNAKE_BODY**）

FOOD（食物）

属性：X 坐标、Y 坐标

生成：每个游戏会话（**GAME_SESSION**）会生成一个食物实例

GAME_SESSION（游戏会话）

属性：开始时间、结束时间、得分

关系：包含蛇身体（**SNAKE_BODY**）和食物（**FOOD**），与用户（**USER**）关联

关系说明：

用户-日志关系：一个用户对应多条游戏日志

用户-游戏会话：一个用户可创建多个游戏会话

游戏会话组成：

每个会话包含一条蛇（由多个蛇节点组成）

每个会话生成一个食物实例

蛇结构：蛇身体由多个 SNAKE_NODE 节点通过链表连接

代码映射说明：

User 结构体对应 USER 实体

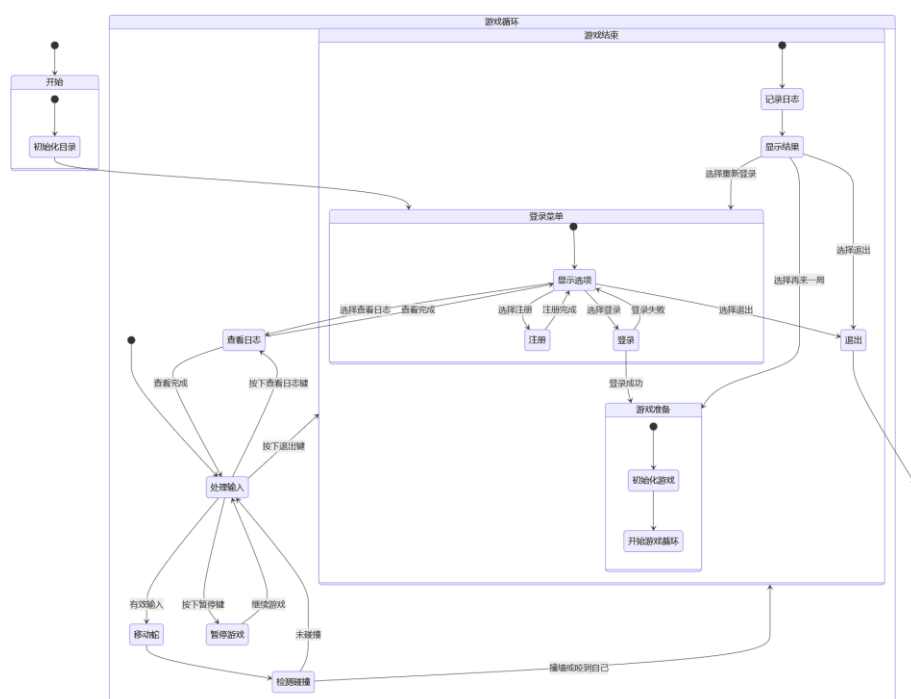
userlog.txt 中的日志条目对应 USER_LOG 实体

snake 结构体对应 SNAKE_NODE 实体

全局变量 game_start_time 和 score 对应 GAME_SESSION 属性

食物生成逻辑（createfood()）对应 FOOD 实体生成

状态图：



解释

开始阶段

系统从 开始 状态启动，首先进行 初始化目录 操作，之后进入 登录菜单。这对应代码里 main 函数中的 initWorkingDirectory() 函数调用，它的作用是设定工作目录。

登录菜单状态

登录菜单 状态会显示登录、注册、查看日志和退出的选项。

若选择登录，会进入 登录 环节，登录成功就进入 游戏准备 状态，失败则回到显示选项；若选择注册，完成注册后回到显示选项；若选择查看日志，查看结束后回到显示选项；若选择退出，系统就结束运行。

游戏准备状态

进入 游戏准备 状态后，会执行 初始化游戏 操作，包含创建地图、初始化蛇和食物等，之后进入 开始游戏循环。这和代码里的 `gamestart()` 函数相对应。

游戏循环状态

游戏循环 是游戏的核心部分。

首先处理玩家输入，若输入有效，蛇就会移动，接着检测是否发生碰撞。若发生撞墙或者咬到自己的情况，游戏就会结束；若未碰撞，继续处理输入。

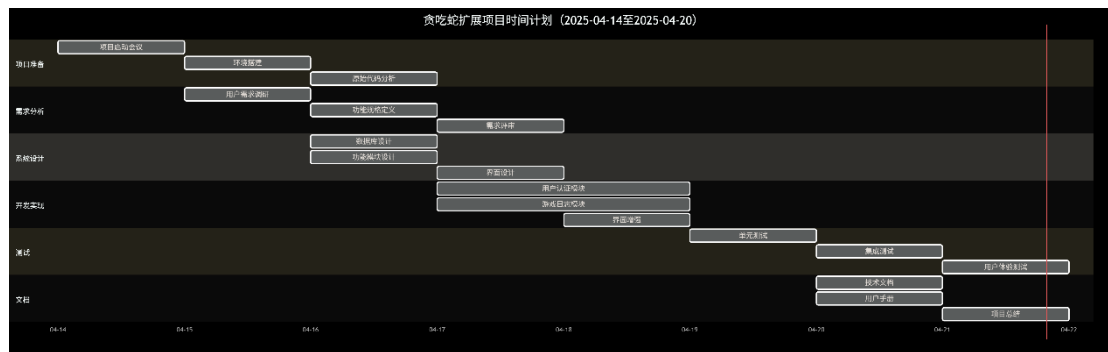
玩家按下暂停键，游戏进入 暂停游戏 状态，按下继续键则回到处理输入状态；按下查看日志键，进入查看日志状态，查看完成后回到处理输入状态；按下退出键，游戏结束。

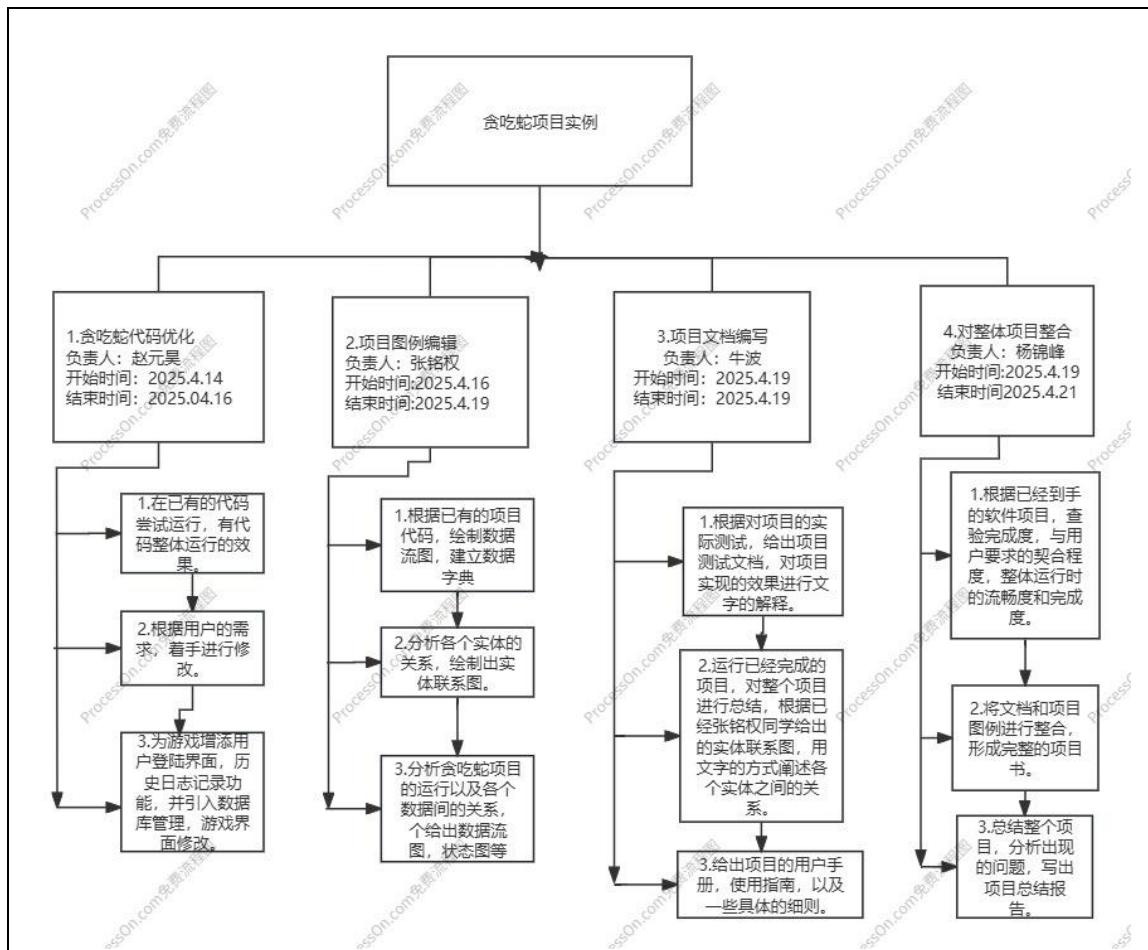
游戏结束状态

游戏结束后，会 记录日志，接着 显示结果。

玩家可以选择重新登录回到 登录菜单 状态，选择再来一局回到 游戏准备 状态，选择退出则结束系统运行。

本次实验的 WBS:





贪吃蛇项目代码	
负责人	赵元昊
时间安排	2025 年 4 月 14 日 - 2025 年 4 月 16 日
任务内容	1. 运行现有贪吃蛇代码, 测试整体功能效果。 2. 根据用户需求对代码进行优化和修改。 3. 新增功能: 用户登录界面 历史游戏日志记录功能 数据库管理 (存储用户信息及游戏记录) 游戏界面优化 (如显示当前玩家、操作提示等)

项目图例编辑	
负责人:	张铭权
时间安排:	2025 年 4 月 16 日 - 2025 年 4 月 19 日
任务内容:	1. 基于项目代码绘制以下图表: - **数据流图** (描述系统数据流动关系) - **数据字典** (定义系统中使用的数据项) 2. 分析实体关系, 绘制**实体联系图 (ER 图)**。 3. 结合数据流图、状态图等, 分析贪吃蛇游戏的运行逻辑及数据交互关系。

项目文档编写	
负责人:	牛波
时间安排:	2025 年 4 月 19 日
任务内容:	1. 编写测试文档: 记录测试结果, 说明项目实现的功能效果。 2. 撰写项目总结: <ul style="list-style-type: none"> - 运行最终项目, 验证功能完整性。 - 结合实体联系图, 用文字描述各实体关系 (如用户与游戏日志的关联)。 3. 编写用户手册: <ul style="list-style-type: none"> - 提供游戏使用指南 (如登录操作、日志查询快捷键 F5 等)。 - 列出功能细则和注意事项。

项目文档编写	
负责人:	杨锦峰
时间安排:	2025 年 4 月 20 日
任务内容:	1. 项目验收: <ul style="list-style-type: none"> - 检查代码功能与用户需求的匹配度。 - 测试整体运行的流畅性和完成度。 2. 文档整合: <ul style="list-style-type: none"> - 将图例 (数据流图、ER 图等) 与文档合并, 形成完整的项目书。 3. 项目总结报告: <ul style="list-style-type: none"> - 分析项目开发过程中的问题 (如代码兼容性、数据库设计难点等)。 - 提出改进建议或未来优化方向。

补充说明

- **协作关系**:

- 代码优化 (赵元昊) 需在项目图例编辑 (张铭权) 开始前完成核心功能。
- 文档编写 (牛波) 与项目整合 (杨锦峰) 部分任务并行进行 (4 月 19 日)。

- **交付成果**:

- 可运行的贪吃蛇游戏 (含登录、日志功能)。
- 全套项目图表 (数据流图、ER 图等)。
- 完整文档 (测试报告、用户手册、总结报告)。

实验总结:

此次项目文档编写让我系统掌握了从测试验证到用户说明的全流程文档规范。测试过程中, 对细节的反复校验培养了我的严谨态度; 梳理实体关系加深了对数据库逻辑的理解; 用户手册撰写则让我学会换位思考, 以用户视角优化功能指引。过程中也发现自身对复杂逻辑表述能力的不足, 后续将加强结构化表达训练, 提升文档撰写的专业性与实用性。

实验评语:

(该部分由老师填写, 宋体, 小四号, 两端对齐, 段落起首空两字距离, 行间距: 固定值 20 磅)

实验成绩		教师签名	