# 软件设计说明书

**1.引言**

贪吃蛇游戏是一款经典的休闲游戏，最初于20世纪70年代诞生于早期的个人计算机上。这款游戏以简单的操作和富有挑战性的玩法深受玩家喜爱，成为了计算机游戏史上的经典之作。贪吃蛇游戏的玩法简单直观，玩家通过操控一条蛇吃食物来不断增长蛇的长度，同时要避开蛇头碰到身体或游戏边界，直到蛇无法移动为止。尽管游戏规则简单，但是它的可玩性极高，容易上手但难以精通，因此备受广大玩家的欢迎。

**1.1编写目的**

随着计算机科技的发展和普及，贪吃蛇游戏已经成为了计算机编程学习和实践的经典案例之一。通过编写贪吃蛇游戏，学习者可以掌握诸如游戏逻辑设计、用户界面开发、碰撞检测等方面的基本编程技能，加深对编程语言和算法的理解和应用。同时，贪吃蛇游戏也是一个优秀的项目实践机会，通过完成一个完整的贪吃蛇游戏项目，学习者能够提升项目管理和团队合作能力，培养解决问题的思维和创新意识。

**1.2项目背景**

项目负责人：陶丽兰

硬件设计工程师：杨玥

软件测试员：杨玥、陶丽兰、段昕怡、王小丫

系统分析员：王小丫

文档编写员：段昕怡

**1.3定义**

用户日志：游戏用户表用以记录用户名和密码，游戏用户日志用以记录用户的ID和用户名、每次游戏的开始时间、持续时长和得分

**2.业务需求分析**

玩家通过方向键操控一条蛇吃食物来不断增长蛇的长度，同时要避开蛇头碰到身体或游戏边界，直到蛇无法移动为止；每吃到一次食物可以加十分；用户进入游戏前需要先注册注册成功后登录才能进入游戏。用户进入游戏选择注册或登录，首次进入游戏需注册账号完成注册后登录账号进入游戏，游戏时界面显示游戏得分、游戏用户名

**2.1功能性需求分析**

1.游戏启动和结束：

游戏启动时，显示欢迎界面。

游戏结束时，显示游戏结束信息。

2.控制：

使用键盘箭头键控制贪吃蛇的移动方向。

贪吃蛇沿着所选方向移动，直到碰到边界或自身。

3.贪吃蛇功能：

贪吃蛇初始长度为5个单位。

贪吃蛇每吃到一个食物，长度增加一个单位，得分加10分。

贪吃蛇不能穿墙，不能穿过自己。

4.食物：

随机生成食物，出现在游戏界面上的随机位置。

贪吃蛇吃到食物后，食物消失，贪吃蛇长度增加，出现新食物。

5.分数计算：

每次吃到食物，得分增加10分。

显示当前得分和当前玩家姓名。

6.用户界面：

显示游戏界面，包括用户注册、登录界面、贪吃蛇、食物和得分信息。

提供暂停和继续按钮。

提供查看用户日志按钮

**2.2非功能性需求分析**

游戏界面的美观性：

优化了登录注册以及游戏界面。

游戏的性能要求：

不卡顿蛇身操作丝滑

安全性需求：

能够保护用户的账号密码安全以及用户游戏数据

**3系统设计**

整体架构如 图1

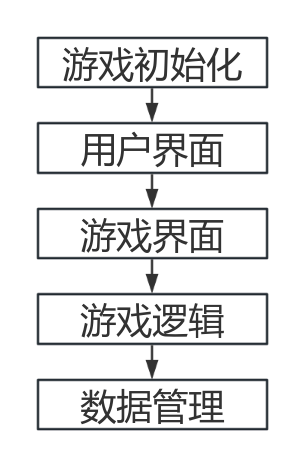


图1

各模块的数据结构：

1.蛇的控制模块：使用了链表数据结构来管理蛇的身体部分，通过动态内存分配和节点操作实现蛇的移动和增长。

2.食物生成模块：食物的位置被存储在一个snake类型的结构体中，这里snake结构体被用来临时存储食物的位置，但并未形成链表或其他更复杂的数据结构。食物只占用一个节点，与蛇身体的链表结构相区别。初始化随机数种子：使用srand((unsigned)time(NULL))初始化随机数生成器，确保每次程序运行时产生的随机数序列不同。使用的是基本的动态内存分配来单独处理一个节点的数据，食物位置被视为一个独立的实体，只需要单一节点来表示。

3.蛇的身体模块：使用了链表数据结构来存储贪吃蛇的身体，通过链表的方式连接蛇身的各个部分，并通过循环输出蛇身的位置信息、游戏地图等。

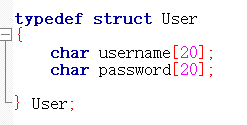
4.地图模块：主要是一个地图的绘制函数，通过控制台输出来呈现游戏的地图边界

**3.1系统实现**

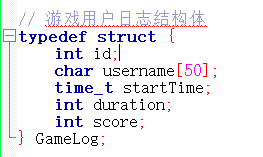
用C语言代码实现了贪吃蛇游戏，主要用到了结构体、链表、指针等数据结构实现。

代码片段：

定义用户表、游戏用户日志表；



定义一个User结构体来表示用户表，其中包含了用户名和密码字段。该结构体用来存储用户的信息。



定义一个GameLog结构体来表示游戏用户日志表，其中包含ID、用户名、开始时间、持续时长和得分字段。该结构体用来存储游戏用户的日志信息。

使用user.txt文件来存储用户信息，每行包含一个用户名和密码，用空格分隔；使用game\_logs.txt文件来存储游戏用户日志信息，每条日志记录占据一行，字段之间用制表符分隔。

**3.2测试计划与验证**

每个测试员分别注册5个账号，分别进行游戏，在不同得分时撞墙或撞自身或一直吃食物。

单元测试：

|  |  |  |  |
| --- | --- | --- | --- |
| ID | 密码 | 得分 | 死法 |
| dxy | 111111 | 30 | 撞墙 |
| zz | 222222 | 20 | 撞墙 |
| sh | 333333 | 40 | 撞蛇身 |
| hx | 444444 | 50 | 撞墙 |
| zh | 555555 | 30 | 撞蛇身 |

游戏体验：在用户查看用户日志后无法返回游戏只能撞墙结束游戏。

验证过程：尽量尝试各种用户可能出现的情况

**4用户界面设计**

游戏界面是一个很简洁的布局，有登录和注册两个选择，在界面周围是一圈黄色的星星。交互方式为直接在控制台上交互。

**5部署与维护**

发布到PM的GitHub仓库，下载到Windows系统就能运行，注册登录就能正常游戏。

安装指南和配置说明：有C语言编译器就能运行，不需要别的配置。

游戏的维护策略和周期：

1.定期更新和修复bug：

定期检查游戏代码，修复可能存在的bug，以确保游戏的稳定性和可靠性。

根据用户反馈和测试结果，解决游戏中出现的问题。

2.优化性能和改进功能：

分析游戏的性能瓶颈，进行必要的优化，以提高游戏的运行效率和流畅度。

根据用户需求和市场趋势，添加新功能或改进现有功能。

3.数据备份和恢复：

定期备份游戏数据和相关文件，以防止意外数据丢失或损坏。

开发恢复机制，确保在发生故障或意外情况下能够快速恢复游戏数据和状态。

4.保持安全性：

实施安全措施，防止游戏受到恶意攻击或非法访问。

定期更新游戏代码和依赖库，以修复已知的安全漏洞和脆弱性。

**5.1风险管理**

1.内存管理问题：C语言需要手动管理内存，容易出现内存泄漏或者非法内存访问的问题，特别是在处理动态分配内存时（比如处理蛇的身体长度变化、食物的生成等）容易出现问题。

2.边界检查不足：在处理用户输入或者数组访问时，可能存在边界检查不足的问题，导致数组越界或者其他意外情况。

3.逻辑错误：游戏逻辑设计不当可能导致游戏无法正常进行或者出现意料之外的行为，比如蛇的移动、食物生成等。

4.跨平台兼容性：C语言的一些特性在不同平台上的表现可能会有所差异，需要注意确保游戏在不同平台上的兼容性。

5.性能问题：C语言的性能高，但是如果算法设计不当或者代码实现不够高效，可能会影响游戏的性能表现。

6.代码可读性和可维护性：C语言相比其他高级语言，代码可读性和可维护性较差，需要注意代码的结构和注释，以便后续的维护和修改。

7.安全性问题：C语言常常被用于开发系统级的程序，如果对用户输入或者外部数据处理不当，可能会存在安全漏洞，比如缓冲区溢出等。

8.测试不足：测试在游戏开发中非常重要，但是由于C语言的特性，测试涵盖的范围可能不够全面，容易出现一些隐藏的问题。

分析可能出现的风险和问题，如游戏性能问题、兼容性问题等。

提供风险评估和应对措施。

**6参考资料**

1. [贪吃蛇游戏需求分析专项说明书 - 百度文库 (baidu.com)](https://wenku.baidu.com/view/69dfa7552d3f5727a5e9856a561252d381eb2052.html?_wkts_=1715013014221&bdQuery=%E8%B4%AA%E5%90%83%E8%9B%87%E8%BD%AF%E4%BB%B6%E5%88%86%E6%9E%90%E4%B8%8E%E8%AE%BE%E8%AE%A1%E8%AF%B4%E6%98%8E%E4%B9%A6)
2. [02软件设计说明书（五篇范例） - 写写帮文库 (xiexiebang.com)](https://www.xiexiebang.com/a14/201905150/f77778824e79631f.html)
3. [软件产品需求分析与设计说明书.docx-原创力文档 (book118.com)](https://max.book118.com/html/2022/0207/5121034110004141.shtm)
4. [软件设计说明书（5篇范例） - 百度文库 (baidu.com)](https://wenku.baidu.com/view/3c65ca725a0102020740be1e650e52ea5518cea6.html?_wkts_=1714993242465&bdQuery=%E8%BD%AF%E4%BB%B6%E5%88%86%E6%9E%90%E4%B8%8E%E8%AE%BE%E8%AE%A1%E8%AF%B4%E6%98%8E%E4%B9%A6)