

结构化软件分析与设计说明书

贪吃蛇小游戏

团队名称：技术泰坦

团队成员：张吴平 史欢欢 包光 高军辉

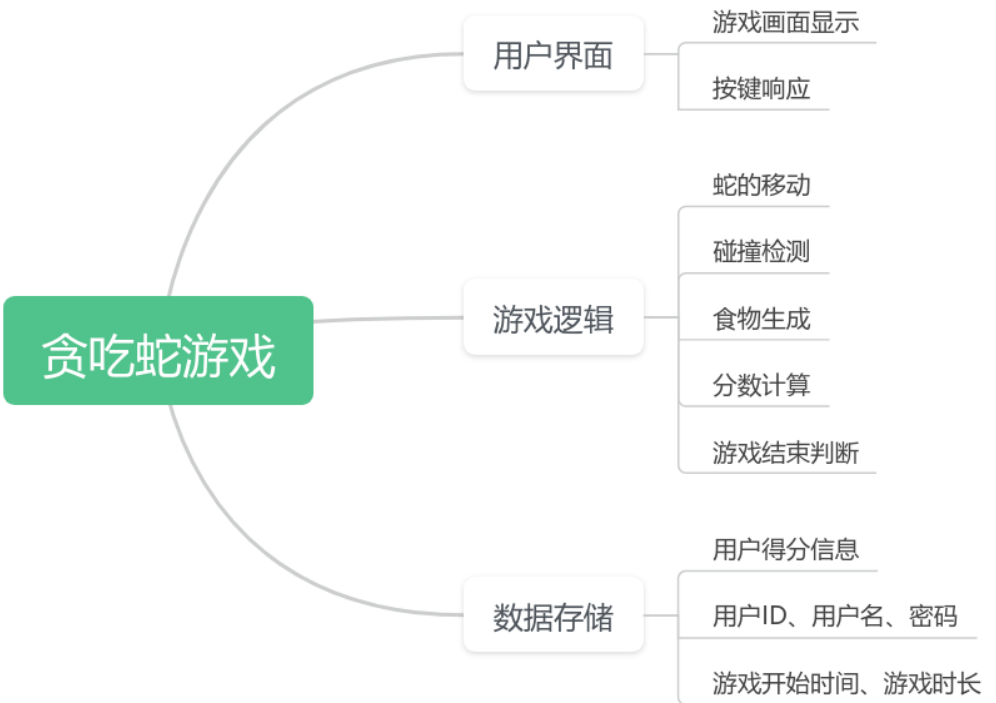
目录

1、需求分析阶段	3
1.1 功能需求	3
1.2 性能需求	4
1.3 界面需求	5
1.4 运行环境需求	5
2、概要设计	8
2.1 贪吃蛇游戏部分	8
2.2 用户管理部分	8
2.3 文件操作	8
3、详细设计	8
3.1 登录注册模块的设计	10
3.1.1 登录功能的设计	10
3.1.2 注册功能的设计	10
3.1.3 找回密码功能的设计	10
3.2 游戏模块的设计	11
3.2.1 游戏日志记录的设计	11
3.2.2 显示用户名正在游戏中的设计	11
3.2.3 按 F5 显示日志信息的设计	11
3.3 其他设计考虑因素	12
3.3.1 用户信息保存与安全	12
3.3.2 游戏界面设计	12
3.3.3 错误处理与日志记录	12
4. 编码与设计	13
4.1 编码规范	13
4.2 架构设计	13
4.3 模块设计	16
4.4 数据库设计	16
4.5 界面设计	18
4.6 安全性设计	19
4.7 性能优化	19
4.8 测试策略	19
4.9 版本控制	20

1、需求分析阶段

1.1 功能需求

游戏总体框架：



功能划分和描述如下：

1、用户界面：

- ① 游戏画面显示：负责绘制游戏界面，包括蛇、食物和墙等元素的渲染。
- ② 按键响应：监听用户的按键操作，根据用户输入来控制蛇的移动方向。

2、游戏逻辑：

- ① 蛇的移动：用户通过上下左右键控制蛇的移动方向和速度。
- ② 碰撞检测：检测蛇与食物、边界或者自身的碰撞情况，判断游戏是否结

束。

③ 食物生成：在地图上随机生成食物，供蛇吃掉并增加长度。

④ 分数计算：跟踪蛇吃掉的食物数量，并根据规则计算得分。

⑤ 游戏结束判断：根据碰撞检测结果判断游戏是否结束，如蛇撞到边界或者自身。

3、数据存储：

① 用户得分记录：保存每次游戏用户的得分记录，以便后续展示。

② 用户基本信息：记录用户 ID、用户名以及登录密码，便于用户登录及后台管理。

③ 游戏时间：记录用户开始游戏的时间和游戏时长，以便后续展出。

1.2 性能需求

1、时间要求：

① 响应时间：用户按下按键后，游戏应立即响应并改变蛇的移动方向。

② 数据转换和传送时间：在进行数据传输时，尽量减少延迟，确保数据的及时性和准确性。

2、适应性：

① 游戏界面适应性：游戏应具有响应式设计，能够适应不同屏幕尺寸和分辨率，确保在各种设备上都能正常显示游戏内容。

② 运行环境适应性：游戏应能在常见的操作系统上运行，如 Windows

③ 与其他软件接口适应性：考虑到可能的未来扩展和更新，游戏应具有良好的可扩展性和可维护性，以便在后续版本中添加新功能或改进现有功能。

3、安全性：

① 容错能力：游戏应具备容错能力，即使出现意外情况导致程序崩溃或异常退出，也能够正常恢复或提供友好的错误提示信息。

② 数据安全性：如果涉及到用户得分记录以及用户个人信息的数据存储，需要确保数据的安全性，包括加密传输、防止数据丢失和篡改等安全措施。

总结：贪吃蛇游戏的时间要求应保证流畅的游戏体验，响应迅速；适应性应考虑不同操作方式、运行环境和与其他软件的接口；安全性应保护用户数据和提供容错处理。具体的实现和调优还需要根据实际情况进行进一步的细化。

1.3 界面需求

1、界面要求：

- ① 游戏画面显示：贪吃蛇游戏界面应以图形方式展示，包括蛇身、食物和背景等元素，以提供直观的游戏体验。界面设计可以根据需求进行美化和个性化。
- ② 按键响应：游戏应能够捕获键盘输入并进行对应操作，例如控制蛇的移动方向。

2、数据输入/输出格式：

- ① 输入格式：用户通过键盘输入控制蛇的移动方向，在游戏中表现为按下特定按键。
- ② 输出格式：游戏会实时更新画面，显示蛇的位置、食物的位置、得分等信息。

3、数据存储介质和显示器分辨率要求：

- ① 数据存储介质：用户得分记录等数据存储在本地设备的文件系统中。
- ② 显示器分辨率要求：贪吃蛇游戏对显示器的分辨率没有具体要求，可以适应不同分辨率的显示器，但需要保证游戏画面的可见性和完整性。

总结：游戏主界面应清晰明了，包含游戏画面、分数、长度等信息。显示当前用户的游戏状态信息，如“张三正在游戏中”。提供用户友好的按键提示，如按 F5 显示游戏用户日志。用户注册和验证界面应简洁明了，指导用户完成相应操作。

1.4 运行环境需求

1、目标系统对于网络设置的要求：

贪吃蛇游戏不需要特定的网络设置。只有多人对战等在线功能时，可能需要与互联网连接。

2、目标系统对于硬件设备的要求：

- ① CPU：贪吃蛇游戏相对简单，一般要求较低的 CPU 性能即可。
- ② 内存：至少需要足够的内存来加载和运行游戏程序。具体要求取决于游戏的规模和复杂度。
- ③ 显卡：对于简单的 2D 贪吃蛇游戏而言，一般集成显卡即可满足需求。
- ④ 存储空间：贪吃蛇游戏所占用的存储空间相对较小，几十到几百兆字节之间。

3、目标系统对于温度和湿度的要求：

温度和湿度的要求通常与其他硬件设备的要求相同，如操作系统、CPU、内存、显卡等。

4、对操作系统、数据库和浏览器等软件配置的要求：

- ①操作系统：贪吃蛇游戏可以在多种操作系统上运行，如 Windows、macOS、Linux 等。具体适应的操作系统版本取决于游戏开发者的选择和用户的需求。
- ②数据库：如果涉及到用户得分记录和用户个人信息等数据存储，可以选择适合的数据库以进行数据的读写和管理。

总结：贪吃蛇游戏对硬件设备的要求较低，一般的计算机和移动设备都能够满足

要求。温度和湿度方面，需遵循硬件设备的相关要求。操作系统、数据库等软件配置的要求与相应的兼容性和功能需求有关。具体的实现和调优还需要根据实际情况进行进一步的细化。

状态转换图

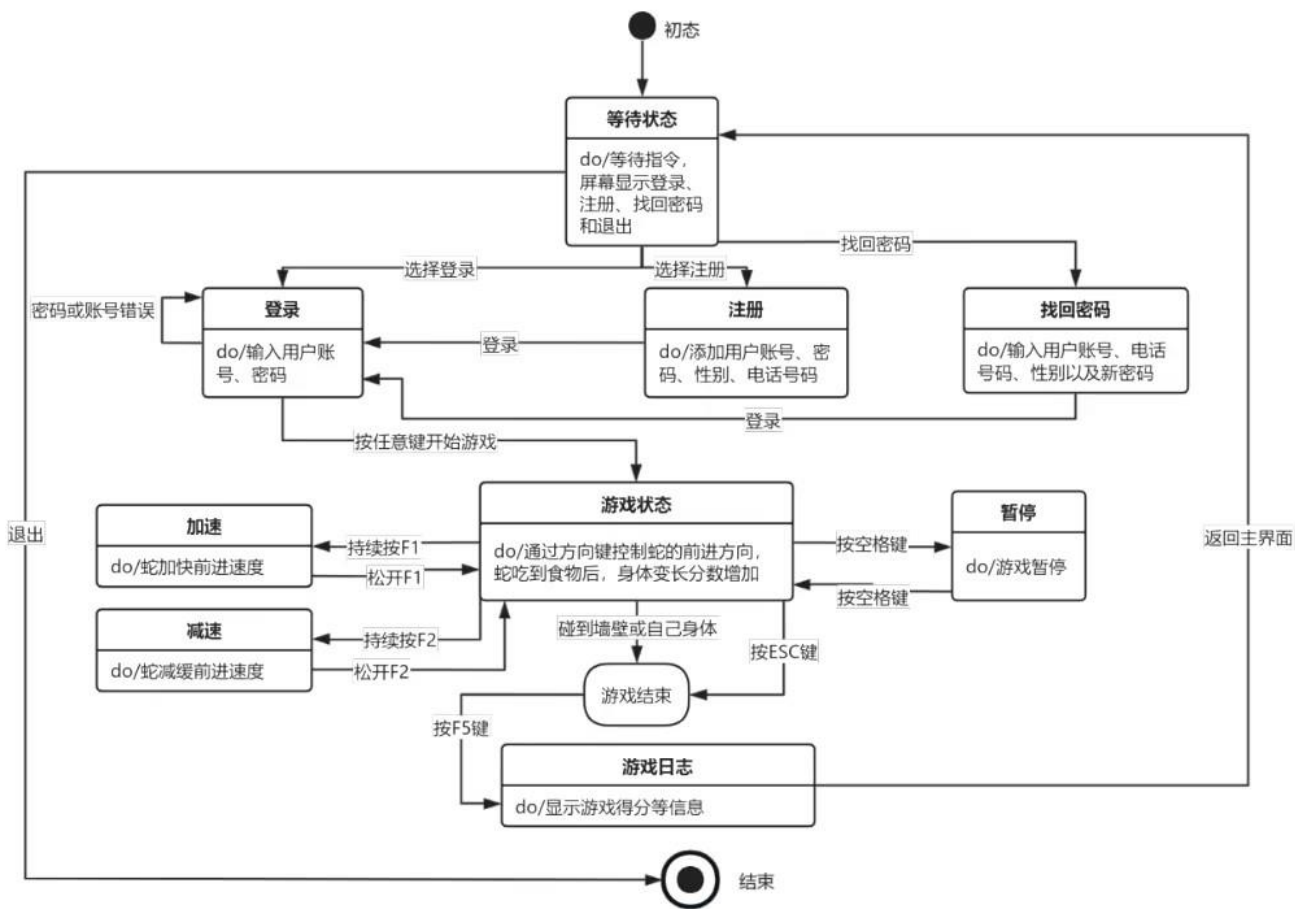


图 1 状态图

编译代码，程序进入等待状态，主界面显示注册、登录、找回密码和退出程序四个选项。选择注册，用户需要输入自己的用户账户、密码、性别、电话号码；选择登录，用户需要输入用户账户以及自己的密码，如果账号或密码错误，则重新输入；选择找回密码，用户需要输入自己的电话号码、性别以及新密码；选择退出，随即结束程序。登录成功按任意键进入游戏，通过方向键控制蛇的前进方向，通过 F1 键加速前进，F5 键减速前进，蛇碰到自己的身体或墙壁或按下 ESC 键将结束游戏。之后按下 F5 键打印游戏日志并返回到主界面。

2、概要设计

2.1 贪吃蛇游戏部分

实现了一个简单的贪吃蛇游戏，包括蛇的移动、吃食物、得分、游戏结束等功能。

使用控制台绘制游戏地图和蛇身，通过键盘控制蛇的移动方向。

包含了暂停游戏、加速减速、游戏结束等功能。

游戏日志会记录用户的游戏得分、开始时间和结束时间，并将日志保存到文件中。

2.2 用户管理部分

提供了用户注册、登录和找回密码的功能。

用户注册时需要输入账号、姓名、性别、电话号码和密码，密码输入时支持掩盖。

用户登录时需要输入账号和密码，验证成功后可以进入游戏。

找回密码功能需要输入账号、姓名、性别、电话号码，验证成功后可以修改密码。

2.3 文件操作

将用户信息和游戏日志保存在文件中，包括用户注册信息和游戏日志记录。

可以打印游戏日志到控制台，方便用户查看。

3、详细设计

程序的流程图设计如下：

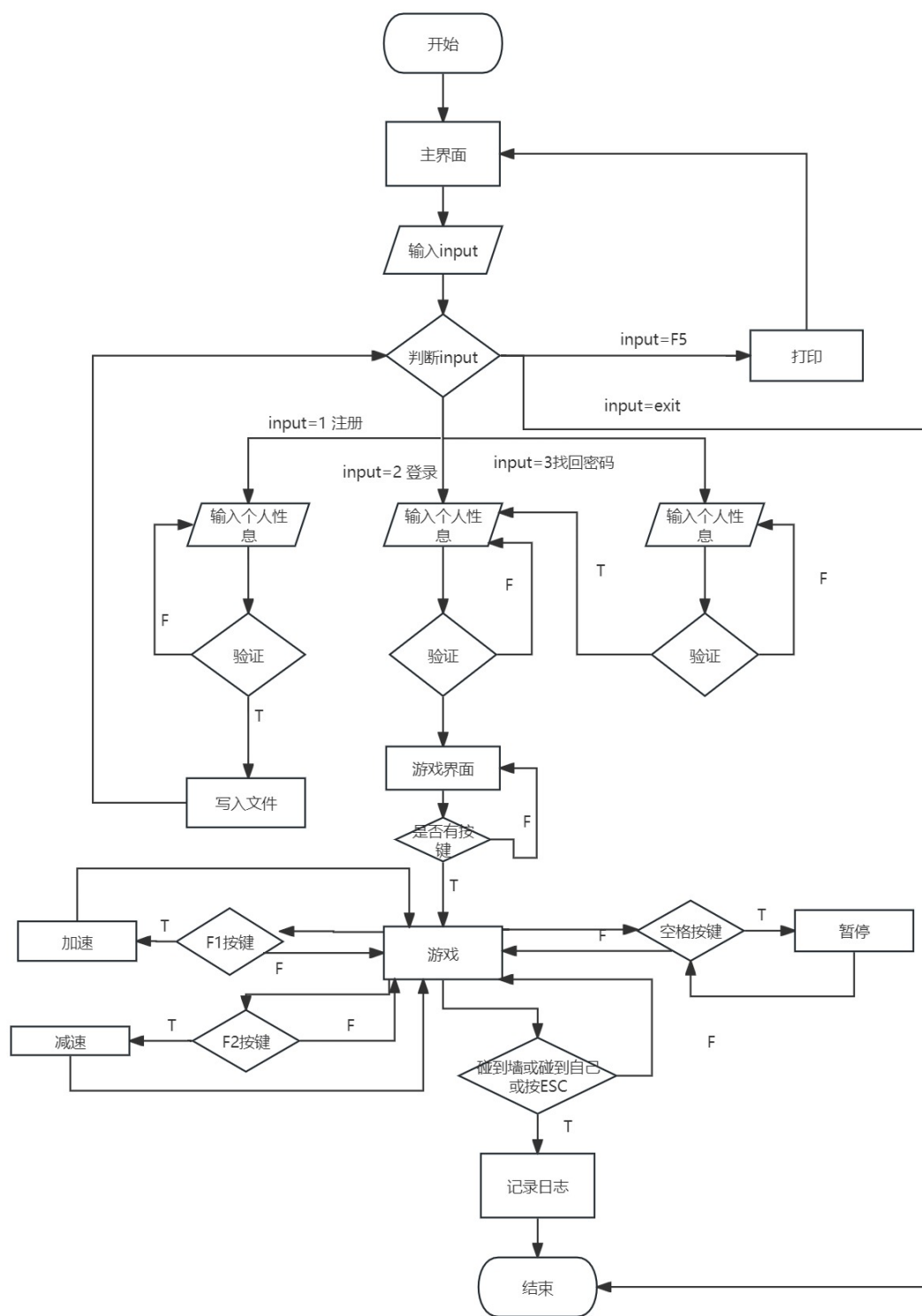


图 2 程序流程图

程序启动后会显示菜单，用户可以选择注册、登录、找回密码或退出。

用户注册成功后可以登录，登录成功后进入贪吃蛇游戏。

在游戏中，用户可以控制贪吃蛇移动，吃食物得分，游戏结束后记录得分并保存

日志

3.1. 登录注册模块的设计

3.1.1 登录功能的设计

- **输入界面：**设计一个用户登录界面，包括用户名和密码输入框、登录按钮和一个“找回密码”链接。
- **处理逻辑：**当用户点击登录按钮时，系统会验证用户输入的信息是否正确。如果信息正确，则允许用户进入游戏主界面；如果信息不正确，则提示用户重新输入或进行密码找回操作。
- **数据库访问：**在验证过程中，需要与数据库进行交互，查询用户名和对应的密码。如果数据库返回的结果与用户输入的不匹配，则表明登录失败。
- **安全性考虑：**确保密码输入采用加密方式，防止数据在传输过程中被窃取。同时，应该限制连续尝试错误的次数，以防暴力破解。

3.1.2 注册功能的设计

- **输入界面：**设计一个用户注册界面，包括用户名、密码、确认密码、性别等字段。
- **处理逻辑：**当用户提交注册信息时，系统会验证所填写的所有字段是否符合要求（如密码长度、电子邮件格式等）。如果一切正常，则将新用户的信息添加到数据库中。
- **数据库访问：**在注册过程中，需要与数据库进行交互，创建新的用户记录。

3.1.3 找回密码功能的设计

- **输入界面：**设计一个找回密码界面，包括用户名输入框和用户的其他注册信息的界面

- **处理逻辑：**当用户选择找回密码的功能的时候，需要输入注册时输入的信息，然后和数据库的信息进行比对，如果一样就允许用户重置密码。并且要同步更新数据库中保存的密码信息。
- **数据库访问：**在用户输入注册信息之后，再次和数据库进行交互，如果输入的信息和数据库中注册的信息一致就允许用户更改密码，与此同时需要同步更改数据库中的密码。

3.2 游戏模块的设计

3.2.1 游戏日志记录的设计

- **输入界面：**设计一个游戏日志记录界面，包括开始时间、结束时间和游戏结果等信息。
- **处理逻辑：**在游戏开始和结束时，系统应自动记录当前的时间和游戏的结果（如胜利、失败等）。
- **文件存储：**除了数据库外，还可以考虑将游戏日志以文本文件的形式保存，以便于后续的分析和管理。

3.2.2 显示用户名正在游戏中的设计

- **输入界面：**设计一个状态栏或小窗口，用于显示当前正在游戏的玩家名单。
- **处理逻辑：**每当有新玩家加入游戏或退出游戏时，系统应及时更新状态栏中的玩家名单。
- **数据库访问：**系统应定期从数据库中读取正在游戏的玩家信息，并更新状态栏中的列表。

3.2.3 按 F5 显示日志信息的设计

- **输入界面：**设计一个快捷键绑定，使玩家可以通过按下 F5 键来刷新日志信息。

- **处理逻辑：**当玩家按下 F5 键时，系统应自动从数据库中读取最新的游戏日志，并将其显示在屏幕上。
- **实时性要求：**为了确保日志信息的时效性，系统应尽量实现日志信息的实时更新。

3.3 其他设计考虑因素

3.3.1 用户信息保存与安全

- 设计一个用户资料管理模块，允许玩家保存自己的个人信息，如性别，电话号等信息。
- 确保用户资料的存储安全，采用加密技术保护用户隐私。用户在输入密码的时候不会直接显示密码而是采用*来保护密码。
- 提供定期备份用户数据的功能，以防数据丢失。

3.3.2 游戏界面设计

- 设计一个美观且易于使用的游戏界面，包括菜单栏、游戏状态栏等。
- 使用清晰易读的文字和图形元素，提高用户体验。
- 考虑不同分辨率下的适配性，确保在不同设备上都能良好运行。

3.3.3 错误处理与日志记录

- 设计完善的错误处理机制，及时发现并解决程序中的 bug。
- 记录错误日志

4. 编码与设计

4.1 编码规范

- 编程语言选择：C 语言。
- 编码标准：每个函数写明注释和对应的模块以及接口。

4.2 架构设计

- 系统架构：系统基于 Windows 10 及以上设计。系统总体分为两个模块，一个是注册登录模块，一个是游戏模块，注册登录模块主要包含注册登录以及找回密码等功能。游戏模块主要包括游戏的开始结束以及统计分数等功能。
- 数据流：根据数据流图的描述。

数据流图及描述如下：

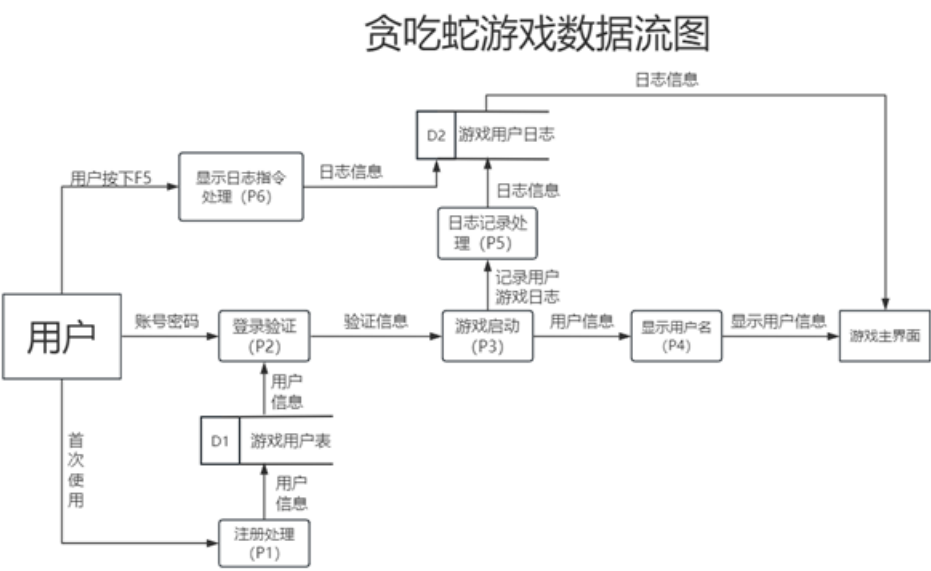


图 3 数据流图

数据流图说明

外部实体：

用户：提供用户名和密码进行注册和登录。

加工/处理：

注册处理（P1）：处理用户输入的用户名和密码，验证唯一性后存储到游戏用户表。

登录验证（P2）：验证用户输入的用户名和密码是否与游戏用户表匹配。

游戏启动（P3）：在验证成功后启动游戏，并显示用户信息。

显示用户名（P4）：在游戏主界面上显示用户正在游戏的状态，如“张三正在游戏中”。

日志记录处理（P5）：记录用户开始游戏的时间、游戏结束时间（计算持续时长）、得分到游戏用户日志。

显示日志指令处理（P6）：处理用户按下 F5 键的请求，展示游戏用户日志。

数据存储：

游戏用户表（D1）：存储用户名和密码。

游戏用户日志（D2）：记录用户的 ID、用户名、游戏开始时间、持续时长和得分。

数据流连接：

用户 -> P1：提供用户名和密码进行注册。

P1 -> D1：注册成功后，存储用户信息。

用户 -> P2：提供用户名和密码尝试登录。

D1 -> P2：验证登录信息。

P2 -> P3：验证成功后，启动游戏进程。

P3 -> P4：启动游戏后，提供用户信息以显示。

P4 -> 游戏主界面：显示“张三正在游戏中”。

P3 -> P5：游戏开始时启动记录日志流程。

P5 -> D2：记录游戏日志信息。

用户（通过键盘 F5）-> P6：请求查看游戏日志。

P6 -> D2：从日志数据库获取信息。

D2 -> 游戏主界面（或弹窗）：显示游戏用户日志信息。

数据字典如下：

贪吃蛇小游戏数据字典卡片

<p>名字: 用户信息表 别名: 信息表 描述: 记录系统中存储用户的账号和密码信息 定义: 用户信息表=用户名+密码 位置: 用户注册 用户登录 输出到数据库</p>	<p>名字: 用户名 别名: 账号 描述: 用户注册时提供的唯一标识 定义: 用户名={字母 数字} 位置: 用户信息表 信息表 用户注册 用户登录 找回密码</p>	<p>名字: 密码 别名: 描述: 用户设置的登录凭证, 存储时加密 定义: 密码={字母 数字} 位置: 用户信息表 信息表 用户注册 用户登录 找回密码</p>
<p>名字: 游戏日志表 别名: 日志表 描述: 记录用户何时开始玩游戏, 玩游戏时长以及游戏对局得分 定义: 游戏日志表=用户名+密码+开始时间+持续时间+得分 位置: 日志文件</p>	<p>名字: 得分 别名: 最终成绩 描述: 记录用户在该局游戏中的得分 定义: 得分={数字} 位置: 游戏日志表 日志表</p>	<p>名字: 开始时间 别名: 描述: 记录每局游戏开始的准确时间 定义: 开始时间={数字} 位置: 游戏日志表 日志表</p>
<p>名字: 持续时间 别名: 描述: 单位为秒, 记录每局游戏的持续时间 定义: 结束时间={数字} 位置: 游戏日志表 日志表</p>		

图 4 数据字典

贪吃蛇游戏的数据字典是一个重要的组成部分，它用于存储和管理游戏中的关键数据。数据字典中包含了两个主要的表：用户信息表和游戏日志表。

用户信息表用于存储用户的基本信息。其中，用户名是用户在注册游戏时提供的唯一标识。用户名用于用户身份识别和登录游戏。密码是用户设置的登录凭证，在存储时进行加密处理，以保证用户信息安全。

游戏日志表用于记录用户的游戏活动。其中，用户名与用户信息表关联，便于在游戏日志中识别具体用户。开始时间准确记录每局游戏开始的时间，以便于统计和分析用户的游戏行为。持续时长单位为秒，记录每局游戏的持续时间，用于计算用户的游戏成绩和排名。

通过这个数据字典，我们可以方便地管理和跟踪用户的信息以及游戏活动。例如，当用户首次使用游戏时，我们需要注册游戏用户名和密码，这些信息将被存储在用户信息表中。每次用户开始游戏时，我们需要验证用户名和密码，验证通过后用户将进入游戏。在游戏主界面，我们可以添加显示“张三正在游戏中”，以展示当前登录的用户名。此外，我们还可以在游戏

主界面添加显示“按 F5 显示游戏用户日志”，并提供查看用户日志的功能。游戏用户日志表将记录用户的 ID、用户名、每次游戏的开始时间、持续时长和得分。

4.3 模块设计

- **模块划分：**根据功能将系统划分为 7 个模块。

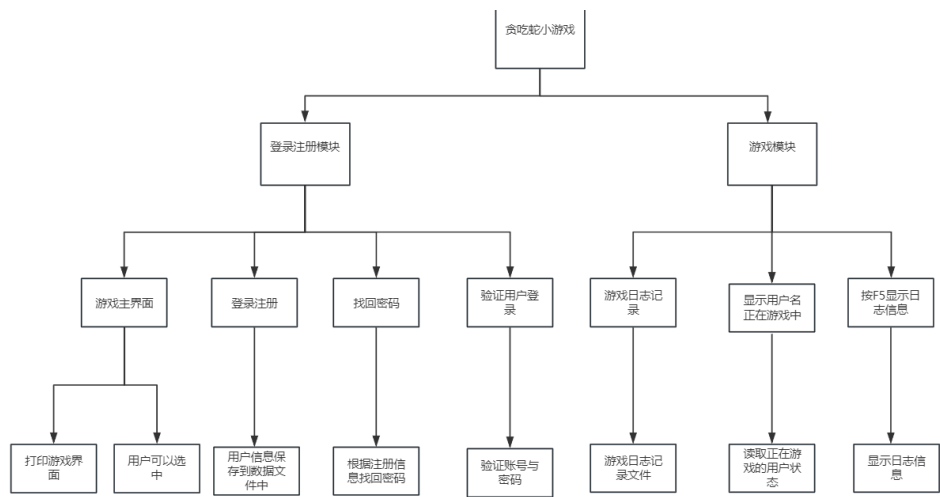


图 5 模块设计

- **模块接口：**定义模块之间的接口，包括输入、输出和可能的异常处理。
- **模块实现：**详细描述每个模块的实现细节，包括关键算法和数据结构。

4.4 数据库设计

- **数据库架构**

数据库的设计如下：

Users 表：

Field	Type	Null	Key	Default	Extra
id	VARCHAR(20)	NO	PRI	NULL	
pwd	VARCHAR(20)	YES		NULL	
name	VARCHAR(50)	YES		NULL	
sex	VARCHAR(10)	YES		NULL	
phone	VARCHAR(20)	YES		NULL	

表 1 用户表

UserGameLog 表:

Field	Type	Null	Key	Default	Extra
username	VARCHAR(50)	YES	MUL	NULL	
password	VARCHAR(50)	YES		NULL	
start_time	TIMESTAMP	YES		NULL	
end_time	TIMESTAMP	YES		NULL	
score	INT	YES		NULL	

表 2 日志表

实体联系图如下：

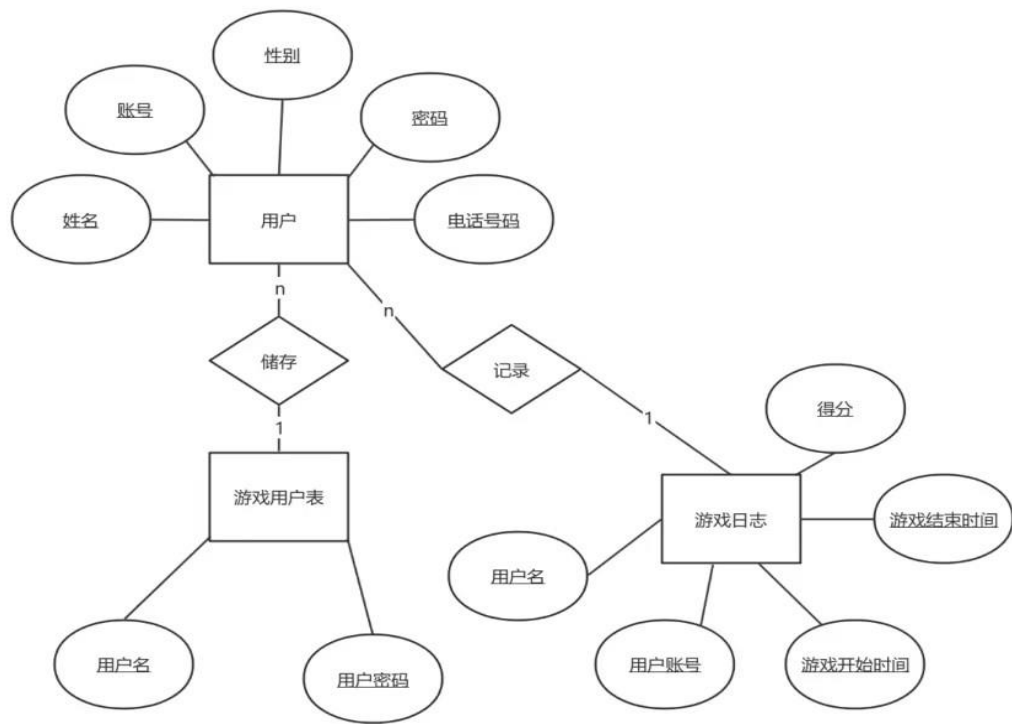


图 6 ER 图

4.5 界面设计

- 用户界面：
设计用户的界面如下：

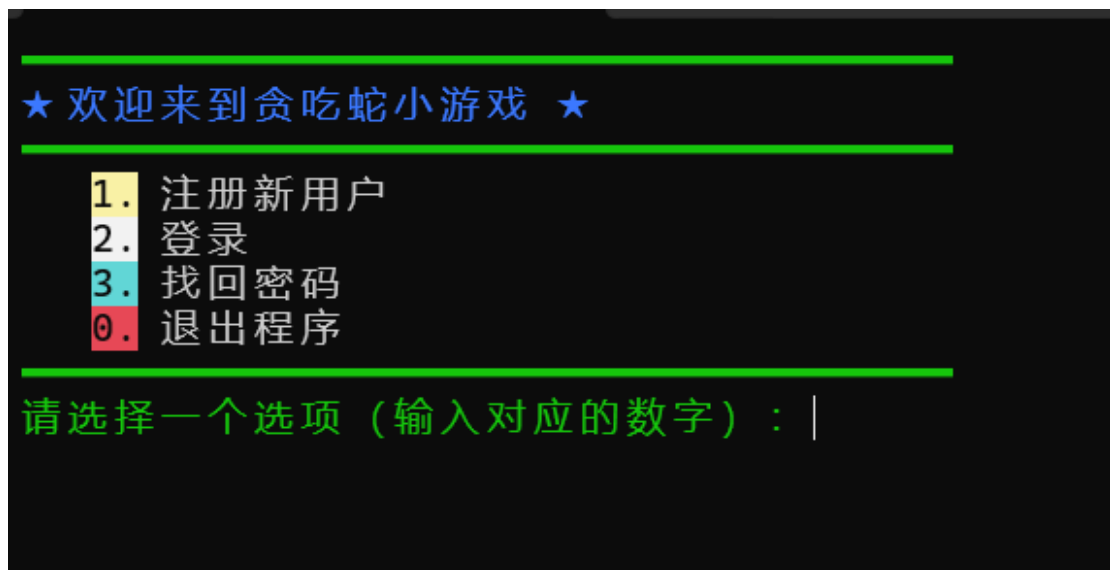


图 7 界面设计

4.6 安全性设计

- **数据保护：**当用户输入密码的时候不会显示密码显示的是*。
- **异常处理：**发生异常的时候结束整个程序的运行并返回错误原因。

4.7 性能优化

- **效率考虑：**通过改变贪吃蛇游戏的算法来提升游戏的性能。

4.8 测试策略

- **单元测试：**

测试登录注册模块的时候，需要测试用户输入的信息是否合理，如账号和密码是否为正常的，密码长度不能为 0，账号信息不能一致否则会导致问题等。

游戏模块需要对游戏进行测试，游戏是否可以正确加减分等。

- **集成测试：**

检查账号密码是否输入正确，如果输入正确才能进入游戏否则不能进入游戏。

4.9 版本控制

- **版本管理：**用版本控制工具 Git 来上传和管理。