**贪吃蛇大作战项目用例描述说明书**

**作者：182056203 朱晓俊**

**182056211 王凯**

**182056223 王秀峰**

**182056231 王梓铭**

**文档变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 版本号 | 修订内容 | 修订 | 审核 |
| 2020-3-30 | V2.0 | 玩家注册、签到、登录处理 | 全体成员 | 朱晓俊 |

1. **引言**

编写目的：优化项目，使项目更具完整性

预期读者：组内成员

参考资料：CSDN、百度

1. **项目概述**

项目背景：以C# 为学习

\_目意义：熟悉简单游戏的制作过程及代码

项目现状：待开发

项目目标：完成一个游戏的基本功能，且没有BUG

项目范围：全体电脑用户

项目作用：开发、锻炼人的眼手并用能力

1. **玩家登录、注册、开始游戏用例描述**

**3.1 玩家登录**

**3.1.1正常处理**

**【用例名称】**

玩家登录

**【场景】**

Who ：玩家

Where：欢迎界面显示之后

When ：欢迎界面显示完成

**【用例描述】**

1. 游戏运行显示欢迎界面；
2. 玩家输入账号和密码并确认；
3. 系统进行账号和密码验证；
4. 验证通过，系统进入签到流程；验证失败，提示用户进行重新注册或重新验证或退出游戏；

**【用例价值】**

玩家登录后，可以查看玩家等级、拥有金币、游戏好友等个人信息，并且在游戏过程中可以累加金币、升级等级，并且可以查看游戏记录和历史最高。

**【约束和限制】**

1. 用户名和密码需对应；
2. 玩家必须成功登录游戏后才能进行游戏；
3. 玩家用户名和密码必须为英文字符或符号；
4. 玩家用户名不能重复；
5. 玩家的密码不能太简单（只为数字）

**3.1.2异常处理**

**【用例名称】**

玩家登录

**【场景】**

Who：玩家

Where：欢迎界面显示之后

When：欢迎界面显示完成

**【用例描述】**

1. 游戏运行显示欢迎界面；
2. 玩家输入账号和密码并确认；

2.1 玩家没有注册，则提示到注册界面，注册完成后方可登录

1. 系统进行账号和密码验证；

3.1 玩家输入的账号不存在，弹出注册窗口，提示玩家进行注册，

玩家可以选择重新登录、注册或退出游戏；

1. 验证通过，系统进入签到流程；验证失败，玩家选择重新验证或重

新注册或退出游戏；

**【用例价值】**

玩家登录后，可以查看玩家等级、拥有金币、游戏好友等个人信息，并且在游戏过程中可以累加金币、升级等级，并且可以查看游戏记录和历史最高。

**【约束和限制】**

1. 用户名和密码需对应；
2. 玩家必须成功登录游戏后才能进行游戏；
3. 玩家用户名和密码必须为英文字符或符号；
4. 玩家用户名不能重复；
5. 玩家的密码不能太简单；

**3.1.3替代处理**

可以使用第三方登录，如QQ、微信；

**3.2 玩家注册**

**3.2.1正常处理**

**【用例名称】**

玩家注册

**【场景】**

Who：玩家

Where：登录界面、登录验证用户名不存在、游戏主界面

When：玩家点击“注册”按钮

**【用例描述】**

1. 系统显示注册界面
2. 玩家输入注册用户名、密码及确认密码并提交；
3. 系统验证用户名是否存在；
4. 如果用户名不存在则系统保存用户名和密码并提示用户注册成功，否则提示用户名已经存在，要求重新输入用户名；

**【用例价值】**

玩家注册后获得登录游戏权限，正常登录后才能进行游戏。

**【约束和限制】**

1. 系统中玩家用户名不能重复；
2. 玩家用户名和密码必须为英文字符或符号；
3. 密码不能太简单（只为数字）

**3.2.2异常处理**

**【用例名称】**

玩家注册

**【场景】**

Who：玩家

Where：登录界面、登录验证用户名不存在、游戏主界面

When：玩家点击“注册”按钮

**【用例描述】**

1. 系统显示注册界面
2. 玩家输入注册用户名、密码及确认密码并提交；

2.1 玩家输入的注册用户名和密码为非英文字符，提示用户输入合法的英文字符；

1. 系统验证用户名是否存在；
2. 如果用户名不存在则系统保存用户名和密码并提示用户注册成功，否则提示用户名已经存在，要求重新输入用户名；

**【用例价值】**

玩家注册后获得登录游戏权限，正常登录后才能进行游戏。

**【约束和限制】**

1. 系统中玩家用户名不能重复；
2. 玩家用户名和密码必须为英文字符或符号；
3. 密码不能太简单（只为数字）

**3.2.3替代处理**

无

**3.3 开始游戏**

**3.3.1正常处理**

**【用例名称】**

开始游戏

**【场景】**

Who：玩家

Where：登录完成之后

When：玩家点击开始游戏之后

**【用例描述】**

1. 玩家点击开始游戏按钮；
2. 出现小蛇和随机食物，并且计分、计时面板开启；
3. 按下回车键，小蛇开始向右运动。玩家自主控制小蛇运动方向，吃到食物后，分数增加。一次食物加10分，累计50分后进入下一关。

下一关的食物数量相应减少、速度相应增加；

1. 小蛇碰到自身或墙壁则判定死亡，提示玩家游戏失败，并且玩家可以选择新游戏或退出游戏；
2. 游戏失败后记录此局的分数和时间，并与历史记录相比较，得出分数与时间排名。

**【用例价值】**

玩家开始游戏可以享受一局游戏的快乐

**【约束和限制】**

游戏失败后可以牺牲部分物资复活

**3.3.2异常处理**

**【用例名称】**

开始游戏

**【场景】**

Who：玩家

Where：登录完成之后

When：玩家点击开始游戏之后

**【用例描述】**

1. 玩家点击开始游戏按钮
2. 出现小蛇和随机食物，并且计分、计时面板开启；
3. 小蛇碰到自身或墙壁后游戏继续，如果发生此类情况，则提示用户游戏错误，需要修复游戏并重新启动；
4. 修复并重启游戏后则可正常游戏；

**【用例价值】**

玩家开始游戏可以享受一局游戏的快乐

**【约束和限制】**

游戏失败后不可复活

**3.4.3替代处理**

无