|  |  |
| --- | --- |
| **项目编号** |  |
| **文档编号** |  |
| **密级** |  |

**概要设计说明书**

**广西民族大学**

目录

[1. 引言 5](#_Toc26735)

[1.1 目的 5](#_Toc15909)

[1.2 参考资料 5](#_Toc21892)

[1.3 版本更新信息 5](#_Toc21267)

[2. 项目需求简介 5](#_Toc10196)

[2.1. 场景设置 5](#_Toc14326)

[2.2. 人物移动 5](#_Toc20797)

[2.3. 骨骼动画 5](#_Toc20702)

[2.4. 碰撞检测 5](#_Toc4912)

[2.5. 联网设计 5](#_Toc26532)

[3. 体系结构设计 5](#_Toc19476)

[3.1. 表现层 6](#_Toc6084)

[3.2. 控制层 6](#_Toc3944)

[3.3. 业务逻辑层 7](#_Toc25636)

[4. 功能模块设计 8](#_Toc2704)

[5. 数据库设计 12](#_Toc32762)

[5.1. 数据库选择 13](#_Toc15168)

[5.2. 数据库逻辑结构 13](#_Toc13458)

[5.3. 物理结构设计 13](#_Toc3612)

[6. 界面设计 13](#_Toc7212)

1. 引言
   1. 目的

该文档目的在于明确系统的体系结构、功能模块、数据库、界面设计的具体细节，使得软件开发人员知道软件开发流程，软件测试时更有条理。

* 1. 参考资料

《UML面向对象分析与设计教程》——胡荷芬 高斐 著 清华大学出版社

《Android应用开发基础教程》——王卫红 著 机械工业出版社

* 1. 版本更新信息

本文档的更新记录如表1-1所示

表1-1版本更新信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容 | 评审人员 |
| 001 | 2021.6.1 | 1.0 | 目录及正文 | 初稿定制 | 黄芮潼 |
| 002 | 2021.6.15 | 2.0 | 目录 | 修改内容 | 黄芮潼 |
| 003 | 2021.6.24 | 3.0 | 正文 | 完善内容 | 黄芮潼 |

|  |  |  |  |
| --- | --- | --- | --- |
| 项目编号 | 1 | | |
| 项目名称 | 概要设计 | | |
| 审查时间 | 2021.6.14 | | |
| 存在问题 | 所在位置 | 所在页码 | 具体建议 |
| 运行环境Windows版本没具体 | 运行环境 | 第四页 | 运行环境修改为Windows7.0及以上 |
| 接口设计不明确 | 接口设计 | 第五页 | 细分为用户接口、外部接口、内部接口 |
| 逻辑结构设计不清晰 | 逻辑结构设计 | 第六页 | 可列出用到的数据库表直观展示 |
| SQL实际代码未写 | 数据结构与程序的关系 | 第六页 | 根据系统中相应代码补充 |
| 没有补救措施 | 补救措施 | 第七页 | 补充补救措施 |
| 文档模板难于理解 | 文档目录 | 第二页 | 及时更换能够理解的、合适的模板 |

|  |  |  |  |
| --- | --- | --- | --- |
| 项目编号 | 2 | | |
| 项目名称 | 概要设计 | | |
| 审查时间 | 2021.6.24 | | |
| 存在问题 | 所在位置 | 所在页码 | 具体建议 |
| 体系结构设计分得不够明确 | 体系结构设计 | 第四页 | 细分为表现层、控制层、业务逻辑层 |
| 功能模块设计中不需要主页面模块 | 功能模块设计 | 第五页 | 删除主页面模块 |
| 用户表和战绩结算表中角色个数多余 | 数据库逻辑结构 | 第八页 | 更新用户表和战绩结算表 |
| 界面设计中不符合实际 | 界面设计 | 第八页 | 更改为游戏界面截图 |

1. 项目需求简介

通过团队的规划和游戏功能分工，开发一个3D人物对战游戏，实现以下功能。

* 1. 场景设置

设置地面、天空，树木、石头、建筑等场景

* 1. 人物移动

人物可通过键盘控制左右、前后、上跳、蹲爬、滑铲、爬行、探头进行移动。

* 1. 骨骼动画

人物有相应动作的骨骼动画

* 1. 碰撞检测

攻击方式包括远程射击类和近身攻击方案，要求攻击有精确碰撞检测判定攻击的有效性

* 1. 联网设计

通过改进可延伸设计你为联网游戏。

1. 体系结构设计

本系统本着软件设计原则，采用MVC三层构架，即Model、View、Controller即模型、视图、控制器。View层是界面，Model层是业务逻辑，Controller层是控制，用来调度View层和Model层，将用户界面和业务逻辑合理的组织在一起，起粘合剂的效果。MVC将软件用户界面和业务逻辑分离以使代码可扩展性、可复用性、可维护性、灵活性加强。

* 1. 表现层

主要功能是显示数据和接受传输用户的数据，可以为网站的系统运行提供交互式操作界面，表现层的应用方式比较常见，例如Windows窗体和Web页面。

该游戏主要有7个模块，每个模块都有各自的表现层，具体如下：

* + 1. 登录

登录模块的表现层主要完成用户的登录功能，登录页面时要求用户输入账号密码的基本信息，确认后页面给出响应消息，提示登录成功或失败的提示。

* + 1. 主页面

主页面模块的表现层主要是显示出关于游戏的一些功能，用户可根据需要选择点击不同的功能，点击后页面给出响应信息，即跳转至相应功能的页面。

* + 1. 模式选择

模式选择模块的表现层主要完成对游戏模式（即人机对战或联机对战）和游戏场景的选择。确定模式之后便可开始游戏。

* + 1. 退出游戏

退出游戏的表现层主要是完成退出游戏界面。

* + 1. 游戏帮助

游戏帮助的表现层主要完成显示游戏帮助给用户。

* + 1. 关于游戏

关于游戏的表现层主要是完成向用户显示关于游戏的相关内容、信息的界面。

* + 1. 战绩结算

战绩提示的表现层主要是完成向用户提示本局游戏的输赢战况等信息。

* 1. 控制层

将业务流程、权限校验等工作进行处理。通过COM/DCOM通讯与逻辑层建立连接。

该游戏主要有7个模块，每个模块都有各自的控制层，具体如下：

* + 1. 登录

登录模块的控制层负责接受来自登录页的用户输入，同时调用登录模块的业务逻辑接口，将用户名与密码等用户关键信息传递到业务逻辑层进行判定。等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并显示主页面。

* + 1. 主页面

主页面的控制层负责接受用户的选择，同时调用主页面模块业务逻辑接口，将用户选择的关键信息传递给业务逻辑层进行处理，等到业务逻辑处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示的页面。

* + 1. 模式选择

模式选择的控制层负责接受来自模式选择页面的选择，同时调用模块选择模块的业务逻辑接口，将用户选择的关键信息传递到业务逻辑层。

* + 1. 退出游戏

退出游戏的控制层负责接受用户退出游戏的选择，同时调用退出游戏模块的业务逻辑接口，将退出游戏的信息传递给业务逻辑层，并由业务逻辑层处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。

* + 1. 游戏帮助

游戏帮助的控制层负责接受用户游戏帮助的选择，同时调用游戏帮助模块的业务逻辑接口，将选择游戏帮助的信息传递给业务逻辑层，并由业务逻辑层处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。

* + 1. 关于游戏

关于游戏的控制层负责接受用户关于游戏的选择，同时调用关于游戏模块的业务逻辑接口，将关于游戏的信息传递给业务逻辑层，并由业务逻辑层处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。

* + 1. 战绩结算

战绩提示的控制层负责在游戏结束后提示战绩，同时调用战绩提示模块的业务逻辑接口，将游戏提示的信息传递给业务逻辑层，并由业务逻辑层处理完成之后，将来自业务逻辑层的相应信息传到表现层，并决定显示页面。

* 1. 业务逻辑层

将用户的输入信息进行甄别处理，分别保存，建立新的数据存储方式，在存储过程中对数据进行读取，将“商业逻辑”描述代码进行包含。

该游戏主要有7个模块，每个模块都有各自的业务逻辑层，具体如下：

* + 1. 登录

登录模块的业务逻辑层主要完成对用户登录逻辑的判定，同时调用登录模块的业务逻辑接口。比如：用户到登录是输入的用户名是否存在、密码是否正确，同时在对用户的身份进行判定。

在登录模块的业务逻辑层是调用了公用的----接口同时实现接口。

* + 1. 主页面

主页面的业务逻辑层主要完成对用户所选择功能的判定，同时调用主页面的业务逻辑接口。

* + 1. 模式选择

模式选择的业务逻辑层主要完成对用户模式选择逻辑的判定，同时调用模式选择的业务逻辑接口。

* + 1. 退出游戏

退出游戏的业务逻辑层主要完成对用户退出游戏的选择进行下一步操作，同时调用退出游戏模块的业务逻辑接口。

在退出游戏模块的业务逻辑层是调用了公用的----接口，同时在实现该接口。

* + 1. 游戏帮助

游戏帮助的业务逻辑层主要完成对用户游戏帮助的选择进行下一步操作，同时调用游戏帮助模块的业务逻辑接口。

在游戏帮助模块的业务逻辑层是调用了公用的----接口，同时在实现该接口。

* + 1. 关于游戏

游戏帮助的业务逻辑层主要完成对用户游戏帮助的选择进行下一步操作，同时调用游戏帮助模块的业务逻辑接口。

在游戏帮助模块的业务逻辑层是调用了公用的----接口，同时在实现该接口。

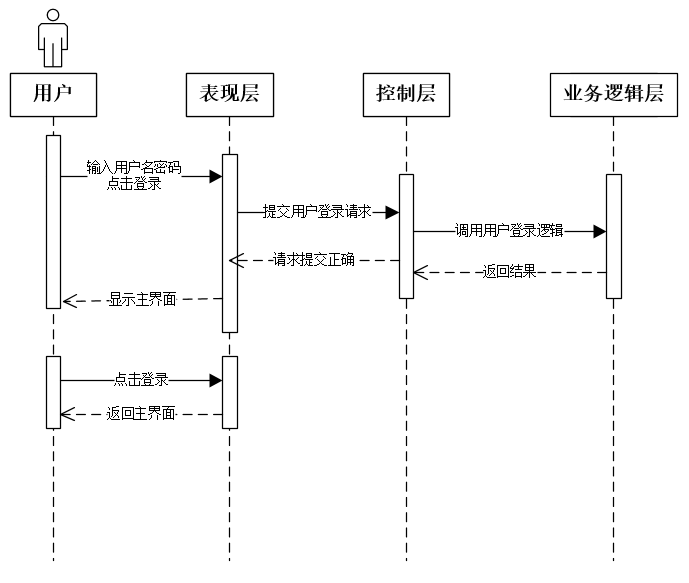
* + 1. 战绩结算

战绩提示的业务逻辑层主要完成提示用户战绩情况的操作，同时调用战绩提示模块的业务逻辑接口。

在战绩提示模块的业务逻辑层是调用了公用的----接口，同时在实现该接口。

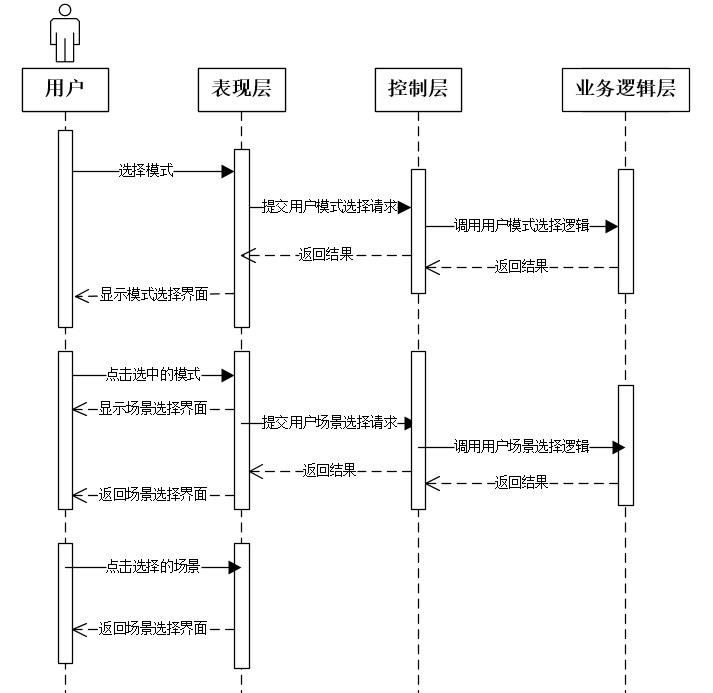
1. 功能模块设计
   1. 登录模块

登录模块的操作流程如图2-1所示



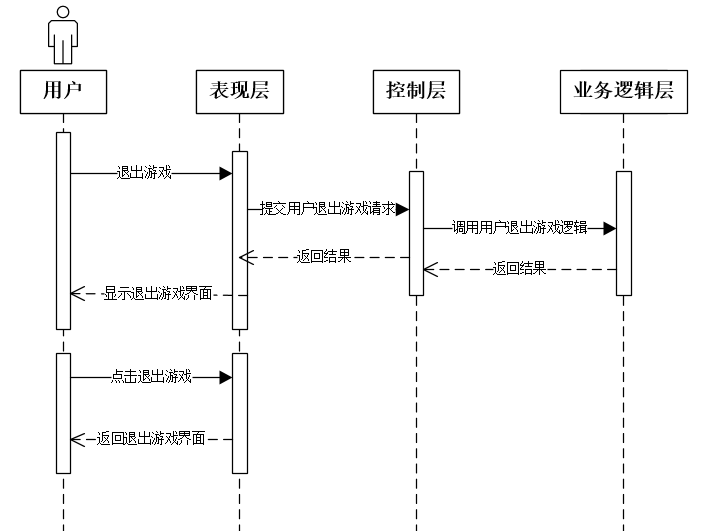
* 1. 模式选择

模式选择模块的操作流程如图2-2所示



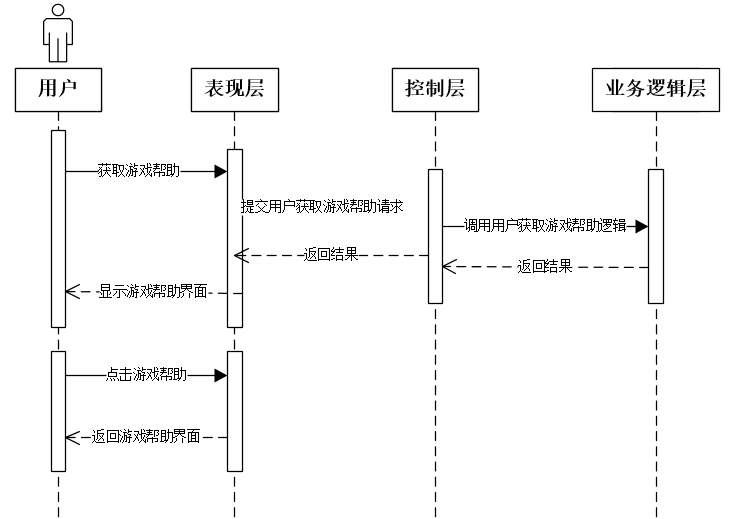
* 1. 退出游戏

退出游戏模块的操作流程如图2-3所示



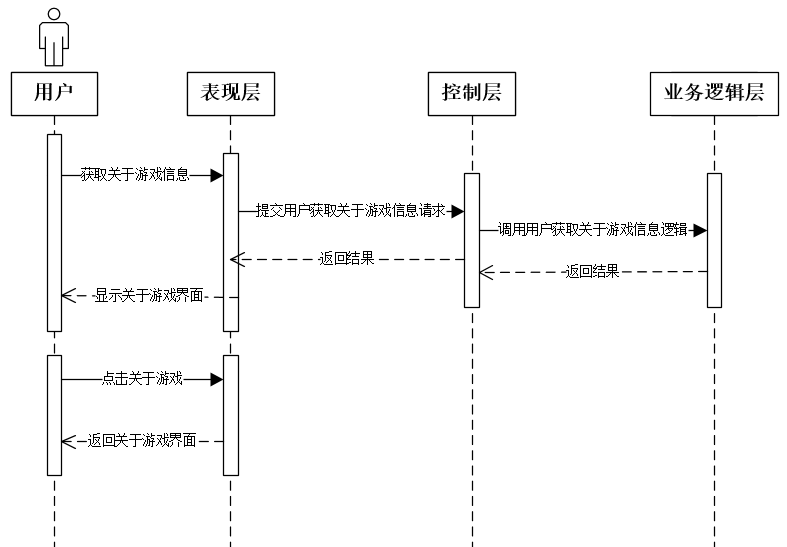
* 1. 游戏帮助

游戏帮助模块的操作流程如图2-4所示



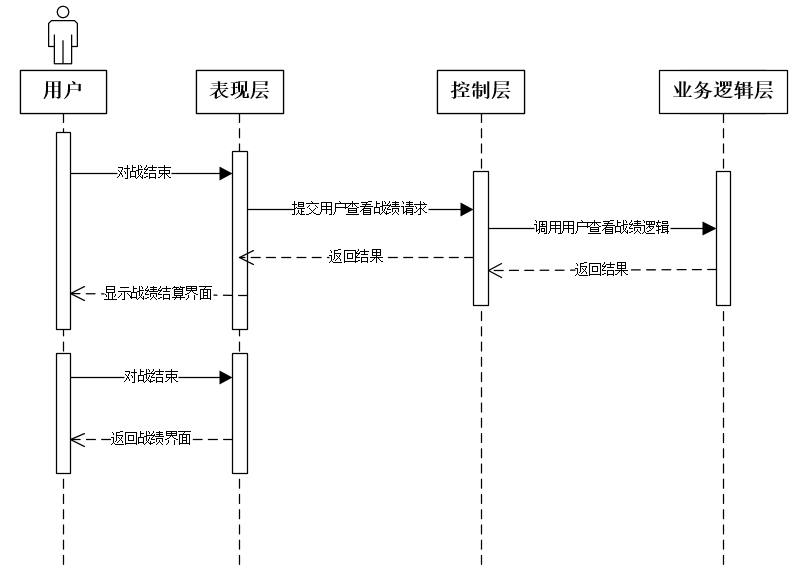
* 1. 关于游戏

关于游戏模块的操作流程如图2-5所示



* 1. 战绩结算模块

战绩结算模块的操作流程如图2-6所示

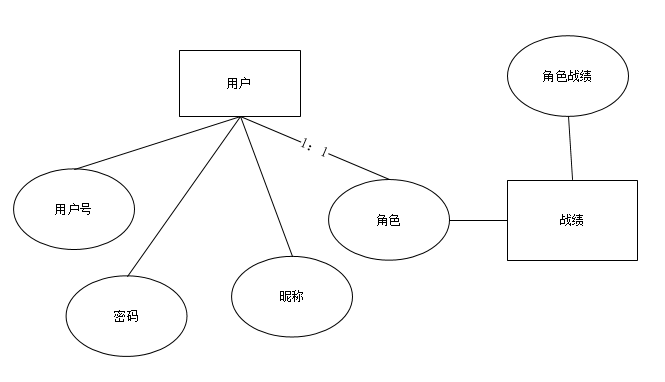


1. 数据库设计
   1. 数据库选择

SQL server 2019

* 1. 数据库逻辑结构

3D人物对战游戏研发的实体关系图（概念数据模型）如下图2-7所示所示



* 1. 物理结构设计
     1. 表1：user表

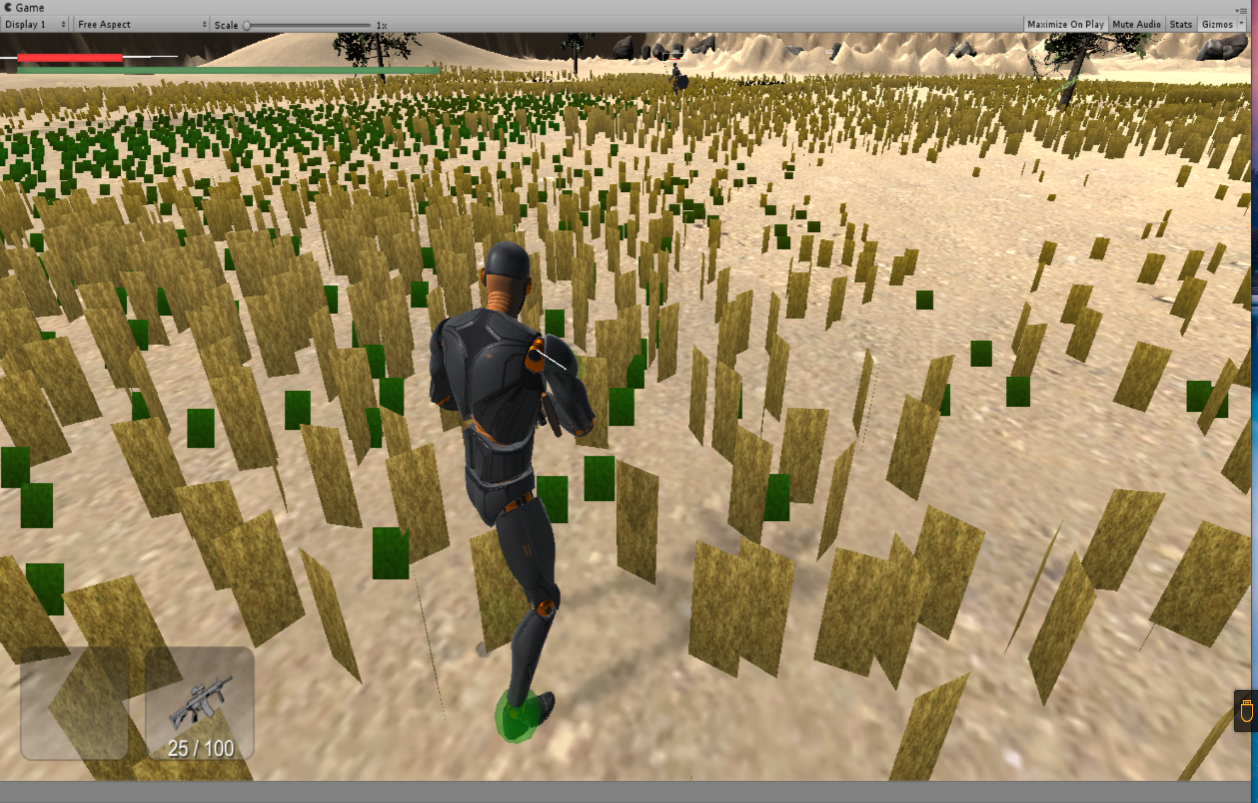
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 用户身份 | User\_ID | INT | N | 主键 |
| 用户昵称 | User\_Name | CHAR | N |  |
| 用户选用角色 | User\_Role1 | CHAR | Y |  |

* + 1. 表2：record表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 可否为空 | 备注 |
| 用户身份 | User\_ID | INT | N | 外键 |
| 用户选用角色 | User\_Role1 | CHAR | N |  |
| 选用角色战绩 | Role1\_Record | CHAR | Y |  |

1. 界面设计

游戏界面场景：



人物对战实景：

