

人工智能学院

课程设计说明书

|  |  |
| --- | --- |
| 题目 | 末日狂潮  ——概要设计说明书 |
| 专业 | 软件工程 |
| 班级 | 21软件工程01班 |
| 学生姓名 | 谢菁菁、谢恒艺、李世珍、叶羽霜 |
| 指导老师 | 高广尚 |
| 实验时间 | 2023/11/20——2023/12/24 |
| 实验地点 | 文综楼 |

**目录**

**[1.引言 3](#_Toc22697)**

**[1.1 编写目的 3](#_Toc4713)**

**[1.2 项目背景 3](#_Toc8097)**

**[1.3 参考资料 3](#_Toc32448)**

**[2. 系统概述 4](#_Toc27257)**

**[2.1 系统目标 4](#_Toc29628)**

**[2.2 系统架构 5](#_Toc31643)**

**[2.3 系统部署 5](#_Toc7735)**

**[3. 功能模块设计 6](#_Toc22567)**

**[3.1 功能模块划分 6](#_Toc5051)**

**[3.2 功能描述 6](#_Toc30915)**

**[3.2.1开始界面 6](#_Toc18064)**

**[3.2.2结束界面 7](#_Toc23015)**

**[3.2.3暂停界面 7](#_Toc3929)**

**[3.2.4怪物逻辑 7](#_Toc12175)**

**[3.2.5角色属性 7](#_Toc8316)**

**[3.2.6医疗物品属性 7](#_Toc24441)**

**[3.2.7武器属性 7](#_Toc6632)**

**[3.2.8战斗反馈 7](#_Toc15961)**

**[3.2.9拾取物品 8](#_Toc26619)**

**[3.2.10碰撞反馈 8](#_Toc12202)**

**[4. 数据结构设计 8](#_Toc11916)**

**[4.1 数据结构概述 8](#_Toc1284)**

**[4.2 数据实体定义 8](#_Toc18801)**

**[4.2.1角色实体 8](#_Toc8477)**

**[4.2.2装备实体 8](#_Toc30192)**

**[4.2.4医疗物品实体 9](#_Toc17378)**

**[4.2.5敌人实体 9](#_Toc5372)**

**[4.2.6场景实体 9](#_Toc20555)**

**[5. 技术选型与实现 9](#_Toc13367)**

**[5.1 技术框架 9](#_Toc8661)**

**[5.2 关键技术 10](#_Toc15365)**

**[6. 测试与维护 10](#_Toc30409)**

**[6.1 测试策略 10](#_Toc21302)**

**[6.2 维护策略 10](#_Toc23989)**

# 1.引言

## 1.1 编写目的

本设计书是《末日狂潮》3D对战游戏项目的研发概要设计，将项目开发进程中或者项目结束后提供给开发人员使用，同时也可以作为实施后期的维护人员使用。本计划书主要解决实现该游戏需求的程序模块设计问题，包括如何把该游戏划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及模块构造的设计等。

## 1.2 项目背景

近年来随着我国游戏产业链持续升级，包括电子竞技、游戏直播等新业态为游戏产业的发展提供新的增长点。现在是一个很好的时代，作为独立游戏开发者，在人们日益提高的娱乐需求之下，开发一款快节奏且操作简单有趣的游戏是尤为重要的，并且可以符合快节奏生活之下高压力人群的减压需求。

作为软件开发的前期文档,可以帮助程序设计人员和管理人员提供清晰的设计思路，在软件开发后期的维护阶段也起到至关重要的作用。

项目开发小组成员：谢菁菁、谢恒艺、李世珍、叶羽霜

软件的系统名称：《末日狂潮》

本项目的用户：喜欢对战游戏的青少年

## 1.3 参考资料

《末日狂潮》开发计划书

《末日狂潮》需求规格说明书

# 2. 系统概述

## 2.1 系统目标

本游戏确保游戏在满足玩家需求、具备良好用户体验的基础上，实现以下目标：游戏能简单快速易于操作，强化游戏教学和引导功能，帮助玩家更快地熟悉游戏玩法和规则，实现多人游戏功能，提供玩家间的互动和竞技体验。确保游戏在各种平台（如PC、移动设备等）上的兼容性和稳定性，提供一致的游戏体验。构建一个高效、可扩展的系统架构，以便于后续内容的扩展和优化。优化游戏性能，确保在各种硬件设备上均能实现流畅运行。

本游戏的功能需求：开始游戏、重新开始、读取存档、人物移动、游戏存档、切换模式、拾取物品、使用物品、进入战斗、战斗。

## 2.2 系统架构

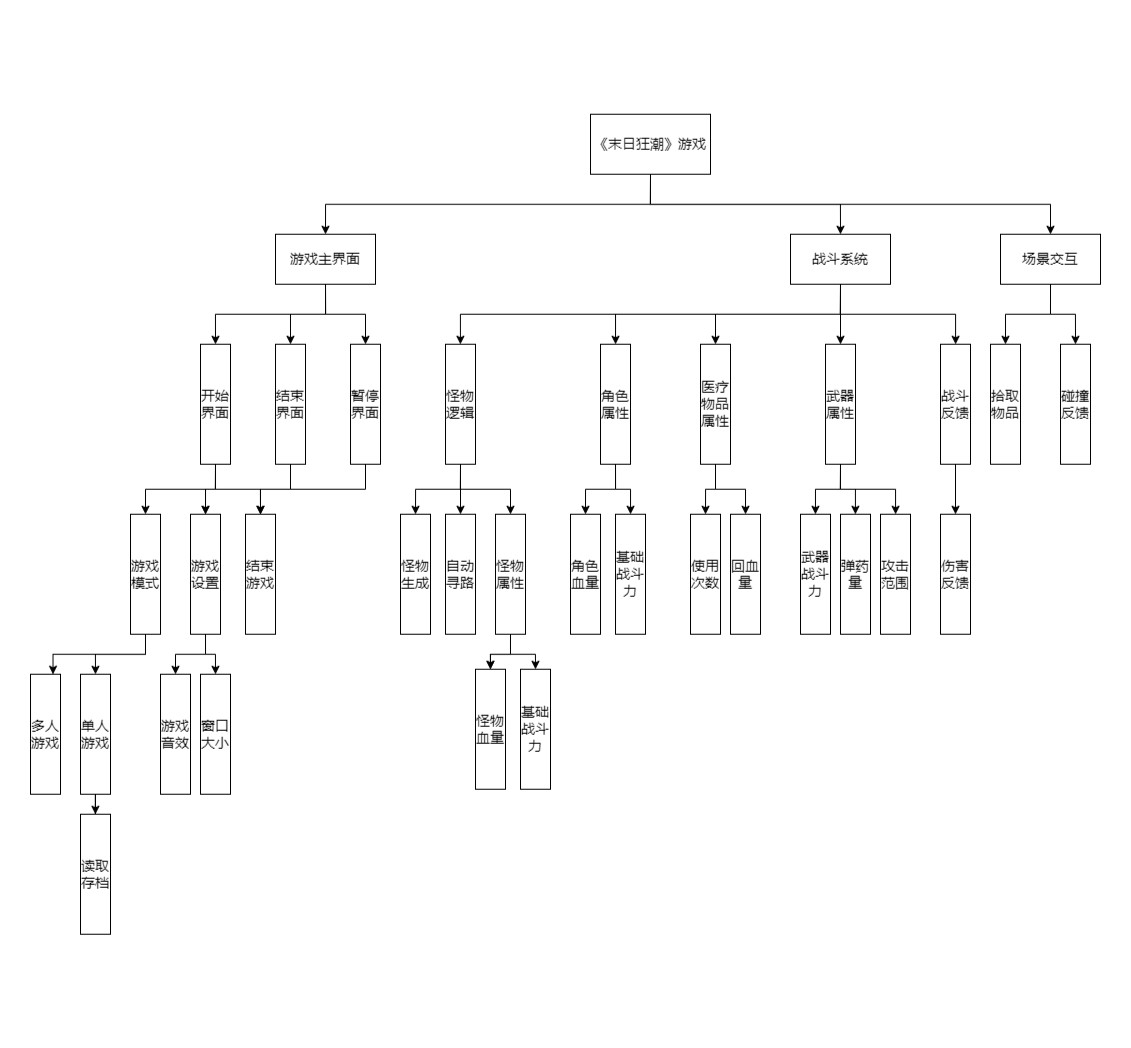


图1功能结构图

## 2.3 系统部署

* 游戏系统应在WINDOS XP以及以上版本下运行。
* 游戏系统运行的最低配置应为CPU Pentium4 1GHz以上、显卡64M BIT或者128M BIT以上硬件条件下运行。

# 3. 功能模块设计

## 3.1 功能模块划分

将系统功能划分为若干个模块，并对每个模块的功能进行简要描述。

|  |  |
| --- | --- |
| 功能类别 | 子功能 |
| 游戏主界面 | 开始界面 |
| 结束界面 |
| 暂停界面 |
| 战斗系统 | 怪物逻辑 |
| 角色属性 |
| 医疗物品属性 |
| 武器属性 |
| 战斗反馈 |
| 场景交互 | 拾取物品 |
| 碰撞反馈 |

## 3.2 功能描述

### 3.2.1开始界面

玩家进入游戏后看见的界面，可以在这个界面选择游戏模式，进行游戏设置，调整游戏音量和窗口大小。

### 3.2.2结束界面

玩家进入游戏界面，如果需要退出游戏，可以选择结束，到达结束界面。

### 3.2.3暂停界面

玩家进入游戏界面，如果需要暂停游戏，可以选择暂停，到达暂停界面。

### 3.2.4怪物逻辑

怪物具有基础血量和基础攻击力，根据算法自动生成，自动寻找玩家所在位置。

### 3.2.5角色属性

玩家可以查看当前游玩角色的血量，角色具有空手基础攻击力。

### 3.2.6医疗物品属性

玩家可以使用背包内的医疗物品来恢复血量，清除负面状态，不同的医疗物品可使用的次数不同。

### 3.2.7武器属性

玩家可以装备武器来增加伤害，不同的武器提供的攻击力不同，伤害范围也不同，同时武器使用后会消耗一定耐久值，耐久值归零武器就无法使用了。

### 3.2.8战斗反馈

玩家在攻击怪物时，怪物会反馈出受击动画与伤害反馈；怪物在攻击玩家时，玩家会反馈动画与伤害反馈。

### 3.2.9拾取物品

玩家能自动拾取在场景内的掉落物，将掉落物放进玩家背包。

### 3.2.10碰撞反馈

玩家在碰撞场景内不可拾取动物品时会受到阻挡或反弹等动作，玩家在碰撞怪物时也会受到阻挡或反弹等动作。

# 4. 数据结构设计

## 4.1 数据结构概述

介绍系统中涉及的数据结构类型、特点和应用场景。

## 4.2 数据实体定义

### 4.2.1角色实体

代表游戏中的角色，包括角色的名称、等级、技能、生命值、攻击力等数据。

### 4.2.2装备实体

代表游戏中的装备，包括装备的名称、等级、属性、特效、使用次数等数据。

### 4.2.4医疗物品实体

代表游戏中的道具，包括道具的名称、等级、效果、使用次数等数据。

### 4.2.5敌人实体

代表游戏中的敌人，包括敌人的名称、等级、生命值、攻击力等数据。

### 4.2.6场景实体

代表游戏中的场景，包括场景的名称、地图、障碍物等数据。

# 5. 技术选型与实现

## 5.1 技术框架

Godot游戏引擎是Juan Linietsky 和Ariel Manzur 发起的，是一款制作游戏的软件，可以制作2D和3D游戏。通过基于节点的架构来设计游戏，3D渲染器设计可以增强3D游戏的画面。具有内置工具的2D游戏功能以像素坐标工作，可以掌控2D游戏效果。面向团队的设计从架构和工具到VCS集成，Godot专为团队中的每个人设计。编辑器可在Windows、Mac OS和Linux系统中运行，支持导出游戏到Windows、Mac OS、Linux、Android、iOS、UWP和HTML5等平台。

Godot通过GDScript来编写游戏代码 ，GDScript是Godot的内置脚本语言。这是Godot完全原创的语言，能够以较少的代码体现Godot的潜力，使新手和专家开发人员都能尽可能快地利用Godot的优势。如果用户曾经用过像Python这样的语言写过任何东西，那么就可以快速上手。使用GDScript有几个原因，特别是在进行原型设计时，在项目的alpha/beta阶段，或者没有创建下一个3A大作，但最突出的原因是整体复杂度降低。GDScript减少了启动和运行Godot所需的时间，使开发人员能够快速接触引擎。同时减少了维护的总体负担，减少了问题的维度，并允许引擎的开发人员专注于排除错误并改进与引擎核心相关的功能，而不是花费大量时间来尝试在一大堆语言中获得一小组增量功能。

## 5.2 关键技术

对战游戏中的角色动作动画是提升游戏体验的重要因素。通过使用骨骼动画技术，可以实现流畅且自然的角色动作。

游戏渲染技术影响着游戏的画面质量。现代游戏引擎通常采用实时渲染技术，如光线追踪、阴影技术、后处理特效等，以呈现出高品质的画质。

物理引擎负责模拟游戏世界中的物体碰撞、摩擦力等因素。通过物理引擎，可以实现更加真实且多样的游戏交互。

# 6. 测试与维护

## 6.1 测试策略

测试目标是确保游戏功能的完整性、稳定性、性能和用户体验。根据本游戏的需求，确定需要测试的游戏主界面、战斗系统以及场景交互等功能。测试范围包括游戏3d对战玩法、界面、音效、角色动画、物理引擎、网络连接等。

## 6.2 维护策略

第一步收集游戏在运行过程中遇到的各类问题，包括玩家反馈、用户调查等。

第二步对收集到的问题进行分类、筛选和分析，找出具有代表性、严重性和优先级较高的问题，以便在后续的维护工作中重点解决。

第三步根据问题分析结果，制定相应的维护计划，包括需要修复的问题、修复的时间节点、预期的完成时间等。

第四步按照维护计划，对游戏进行修改和优化，修复各类问题。在修复过程中，需要充分考虑游戏的稳定性、性能和用户体验，确保修复方案的可行性和有效性。

第五步在问题修复完成后，进行相应的测试和验证，确保修复措施能够解决实际问题，且没有引入新的问题。测试过程可能需要包括功能测试、性能测试、兼容性测试等。

第六步在测试通过后，将修复后的游戏发布给用户，并进行相应的宣传和推广。

第七步在游戏更新后，持续收集用户的反馈和评价，了解游戏在实际运行过程中的表现和问题，为下一轮维护提供参考。

第八步根据用户反馈和问题收集，对游戏进行持续优化，以提高游戏的质量和用户体验。