# Rogue-like动作游戏需求分析文档

## 导论

### 1.1系统目标

一款可以在Android平台供玩家游玩的游戏,类型为rogue-like + act。

### 1.2系统范围

1.开发意图：基于兴趣以及丰富手机游戏类型。

2.作用范围：爱好rouge-like或者action类型的青年玩家(大学生)。

### 1.3项目的目标和成功的标准

项目目标：研发出一款可玩性高的rogue-like + act游戏。

成功的标准:

1. 游戏可以在android平台流畅运行,基本无bug。

1. 游戏的可玩性高(内容丰富,操作流畅,数值平衡)。
2. 游戏的画面基本过关。

### 1.4定义、首字母缩写词和缩写词

|  |  |  |
| --- | --- | --- |
| 缩写词 | 全称 | 含义 |
| UI | User Interface | 用户界面 |
| Rand | Random | 随机 |
| Btn | Button | 按钮 |

### 1.5总结

本次项目主要是要研发一款特定类型的手机游戏,对流畅度和可玩性要求比较高,画面有基本要求。面向群体主要为爱好此类游戏的当代青年团体。

## 2.当前系统

在当前系统中，玩家控制一个角色，在通过若干关卡后获得游戏的胜利。每个关卡有随机生成的BOSS以及普通怪物，玩家通过击杀怪物和BOSS完成本关卡，每通过一个关卡都会获得随机的道具以及随机掉落的精华，拾取道具可以存储在道具栏中适时使用，拾取精华可以解锁新的技能以及完成玩家控制角色的形态转换，以此体验到全新的玩法。

其中玩家在当前系统中可以进行交互，获得道具后，玩家可以点击查看道具的使用效果；拾取精华解锁新的技能时，系统会提示玩家查看新的技能效果，玩家也可以随时点击查看所有已拥有的技能；玩家点击个人信息可以查看当前角色的各种属性；在图鉴模块中玩家可以查看游戏中的所有信息，包括武器信息、怪物信息和剧情发展等。

## 3.建议的系统

### 3.1 概述

本系统以玩家的视角罗列了25个的功能需求和8个非功能需求，并标识出了5个场景与4个用例，之后建立了基本的对象模型,并以顺序图的形式给出了系统的动态模型。

### 3.2 功能需求

支持一类用户：玩家

|  |
| --- |
| 玩家可以销毁已有存档，重新开始游戏。 |
| PC端玩家可以更改操作杆的按键 |
| 游戏中玩家可以看到自己的属性和已拾取道具的种类和数量，已有技能的效果 |
| 玩家操作角色攻击敌人时可以知道敌人的血量。 |
| 游戏的主界面有图鉴栏，玩家可以看到自己已经使用过的道具的功能，已碰到敌人的技能属性等 |
| 有操作介绍界面，玩家可以通过这个了解游戏的基本操作的过程 |
| 玩家在进行一场游戏时通过某个按钮来暂定或更改某些游戏设置（比如调节音量等） |
| 玩家在进行一场游戏时通过某个按钮退出游戏到游戏主界面或重新开始游戏 |
| 玩家需要有一个精致的欢迎界面。 |
| 玩家在主场景中可以选择开始游戏，图鉴,成就。 |
| 玩家可以通过控制杆来操作角色行动。 |
| 玩家需要一个核心游戏模块来体验这个游戏的内容与乐趣。 |
| 玩家可以切换角色的形态 |
| 玩家需要一个成就模块来满足其成就感。 |
| 玩家需要一个图鉴模块来查看游戏中每个元素的信息来提高自己。 |
| 玩家需要击败有挑战性的怪物来获得荣誉感 |
| 玩家需要每一轮多出一些新事物来抹去重复游戏的乏味感 |
| 玩家需要一些道具来辅佐其击败挑战性的怪物 |
| 玩家战斗时需要一个状态界面来查看自己的状态 |
| 玩家需要一个小地图来明白自己所处的位置 |
| 玩家需要强大的BOSS以激发其挑战的热情 |
| 玩家需要角色有流畅的动作衔接以获得爽快感 |
| 玩家需要平衡的数值以循序渐进的游戏 |
| 玩家需要游戏的策略性 |
| 游戏结束后玩家会会进入游戏结束界面。 |
| 在通过关卡后，玩家会获得相应的剧情进展。 |
| 玩家在游戏结束后可以选择重新开始，退出游戏. |
| 玩家在任意一个场景中都需要合适的背景音乐。 |
| 玩家在游戏中能够随时查看游戏的进程和控制角色的相关信息。 |
| 玩家在游戏中获得新的道具或者技能时能够查看相应的信息。 |
| 玩家在游戏中可以随时暂停、存储和退出。 |

### 3.3 非功能需求

|  |
| --- |
| 低的操作代价要求。 操作者必须能够独自安装一个系统，无需购买额外的软件组件。在游戏过程中操作者可以经过不超过两次的尝试就熟悉游戏的整个操作流程。 |
| 剧情可扩展性要求。 游戏的剧情可以不断的更新，而不是一成不变或者一旦改变导致原版本的剧情,操作方式等发生大的改变。 |
| 低资源占有消耗要求。游戏在运行过程中进程占用的设备资源少，以及游戏本身占用存储的空间尽量低。 |
| 游戏流畅性要求。操作者进行游戏时游戏流畅而不会出现卡顿死机等情况。 |
| 高存储性能要求。操作者在结束游戏后第二次进入游戏时拥有以前的进度，而不是每次都从头开始或突然出现存档消失的情况。 |
| 随机性要求。操作者每次进行游戏时都有不同的体验，不会感到重复无聊。 |
| 设备适应要求。游戏的界面等布局要适应不同大小的设备，在不同设备下载游戏要做到UI的界面布局不会重叠等。 |
| 数量可扩展性需求。可以灵活的增加玩家的职业数量和BOSS数量。 |

### 3.4 系统模型

#### 3.4.1 场景

欢迎场景,主场景,战斗场景(胜利,失败,状态),图鉴场景,成就场景。

|  |  |
| --- | --- |
| 场景名称 | 欢迎界面 WelcomeScene |
| 参与者角色 | player：玩家 |
| 事件流 | 1.玩家点击屏幕中的任意地方，界面转换主界面（Main） |

|  |  |
| --- | --- |
| 场景名称 | 主界面 MainScene |
| 参与者角色 | Player：玩家 |
| 事件流 | 1.玩家点击“开始游戏”按钮，界面转换到游戏场景，开始一场游戏 |
| 2.玩家点击“图鉴”按钮，界面转换到图鉴场景，可以翻阅图鉴 |
| 3.玩家点击“成就”按钮，界面转换到成就场景，可以查看已达成成就 |
| 4.玩家点击“设置”按钮，出现设置菜单。 |
| 5.玩家点击“设置”菜单上的“操作介绍”按钮，出现操作介绍界面 |
| 6.玩家点击“操作介绍”界面之内，操作介绍界面到下一页 |
| 7.玩家点击“操作介绍”界面之外，操作介绍界面退出，回到“设置”菜单 |
| 8.玩家点击“设置”菜单外，操作菜单消失，回到主界面 |

|  |  |
| --- | --- |
| 场景名称 | BattleScene (战斗场景) |
| 参与者角色 | 玩家 |
| 事件流 | 1.玩家用手拖动屏幕上的方向键,角色在场景中往相应方向移动,若角色已在房间的边缘位置,则不会该方向的位移。 |
| 2.玩家用手点击攻击键,角色执行相应的攻击动作,若击中怪物则怪物掉血。在动画执行过程中屏蔽方向键 |
| 3.玩家点击角色头像,则展开玩家属性层。 |
| 4.在玩家属性层中,玩家按住某个图标会出现详细介绍的小对话框 |
| 5.玩家击败本个房间内所有怪物,会掉落道具, |
| 6.玩家击败本个房间内所有怪物,房间的出口全部打开 |
| 7.玩家走到出口位置,重新加载这个场景（随机） |
| 8.玩家击杀怪物,充能道具的能量增加 |
| 9.玩家每次被怪物击中,一般情况下血量减一 |
| 10.玩家击败BOSS或变异怪,掉落道具 |
| 11.玩家通关,播放相应结局动画,然后返回主界面 |
| 12.玩家被怪物杀死,出现游戏失败层,点击返回主界面 |
| 13.玩家拾取立即道具,获得相应的效果 |
| 14.玩家使用道具,出现相应的效果 |
| 15.玩家拾取非立即道具,放入道具存储栏 |
| 16.玩家走在陷阱上,血量-1 |
| 17.玩家在宝箱周围按攻击键,打开宝箱 |

|  |  |
| --- | --- |
| 场景名称 | 图鉴场景（BookScene） |
| 参与者角色 | 玩家 |
| 事件流 | 1.玩家从主场景中点击图鉴按钮，进入图鉴场景。 |
| 2.玩家点击返回按钮，返回到上一状态。 |
| 3.玩家点击武器按钮，右侧出现全部的十六种武器图片，点击武器图片会出现对应的武器名称和持有角色。 |
| 4.玩家点击技能按钮，右侧出现角色的四种形态，点击一种形态，会出现该形态对应的全部技能介绍。 |
| 5.玩家点击怪物按钮，右侧出现怪物的四种种族，点击一种种族，会出现该种族的所有玩家遇到过的怪物的图片，点击一种怪物图片，会出该怪物的描述。 |
| 6.玩家点击BOSS按钮，右侧出现所有已击败过的BOSS的图片以及对应的名称，点击BOSS图片，会出现该BOSS的介绍。 |
| 7.玩家点击道具按钮，右侧出现所有的已解锁道具的图片以及对应的名称，点击道具图片，会出现该道具的介绍。 |
| 8.玩家点击结局按钮，右侧出现故事所有的结局，玩家可以点击查看已解锁的结局。 |

|  |  |
| --- | --- |
| 场景名称 | 成就场景（AchievementScene） |
| 参与者角色 | 玩家 |
| 事件流 | 1.玩家从主场景中点击成就按钮，进入成就场景，场景中会出现成就图标和成就对应的名称、解锁条件、奖励。 |
| 2.玩家点击返回按钮，返回到上一状态。 |
| 3.玩家点击左右两侧的箭头，进行翻页查看。 |
| 4.玩家点击解锁条件，会出现对应成就的解锁条件介绍。 |
| 5.玩家点击奖励按钮，会出现完成该成就后可以获得的奖励。 |

#### 用例模型

|  |  |
| --- | --- |
| 用例名称 | 查看图鉴（ViewingBook） |
| 参与者 | 玩家 |
| 事件流 | 1.用户点击主界面（Main）的图鉴按钮（） |
|  | 2.界面转换到图鉴界面（Book） |
|  | 3.用户在图鉴界面查看相应图鉴 |
|  | 4.用户点击图鉴内的相关按钮，图鉴界面转换成对应的界面 |
|  | 5.用户结束查看点击返回按钮到主界面 |
| 入口条件 | 用户在主界面点击图鉴按钮 |
| 出口条件 | 界面回到主界面 |

|  |  |
| --- | --- |
| 用例名称 | 查看成就(ViewingAchievement) |
| 参与者角色 | 玩家 |
| 事件流 | 1.玩家点击主界面成就按钮,界面跳转到成就界面 |
| 2.玩家点击左划右划按钮,内容翻页 |
| 3.玩家按住某项成就,在手指处出现小对话框,显示成就获得时间 |
| 4.玩家点击激活的宝箱,可以开启宝箱,解锁皮肤 |
| 5.玩家点击返回按钮,返回主界面 |
| 入口条件 | 点击主界面成就按钮 |
| 出口条件 | 玩家关闭游戏或者玩家点击返回按钮 |

|  |  |
| --- | --- |
| 用例名称 | 进行战斗(Battling) |
| 参与者参与 | 玩家 |
| 事件流 | 1.用户点击主界面的开始游戏按钮。 |
| 2.界面转换到战斗界面。 |
| 3.玩家操作控制杆。 |
| 4.玩家控制的角色接收到移动指令，进行相应的移动 |
| 5.玩家点击攻击键。 |
| 6.玩家控制的角色接收到攻击指令，进行相应的攻击。 |
| 7.玩家控制的角色受到怪物的攻击，产生被攻击的效果（被击退，被击倒），扣除相应的生命值。 |
| 8.玩家进入地图中的陷阱、路障等特殊区域。 |
| 9.玩家控制的角色收到特殊区域的碰撞检测，产生在特殊区域的相应效果。 |
| 10.玩家点击拾取道具。 |
| 11.玩家控制的角色收到道具的碰撞检测，若道具栏未满，将道具存储到道具栏中，否则无响应。 |
| 12.玩家被怪物击杀，进入游戏结束界面。 |
| 13.玩家通关，进入游戏结束界面。 |
| 入口条件 | 点击主界面开始游戏按钮。 |
| 出口条件 | 1.玩家生命值降为零。 |
| 2.玩家通过所有关卡。 |
| 3.玩家点击退出按钮。 |

|  |  |
| --- | --- |
| 用例名称 | 查看战斗信息(ViewingBattleState) |
| 参与者角色 | 玩家 |
| 事件流 | 1.玩家在战斗界面点击人物头像。 |
| 2.界面显示玩家控制的角色信息。 |
| 3.玩家在战斗界面点击道具。 |
| 4.界面显示相应道具的使用效果。 |
| 5.玩家点击战斗界面的技能图标。 |
| 6.界面显示相应的技能效果。 |
| 入口条件 | 玩家点击相应的信息图标。 |
| 出口条件 | 玩家点击信息界面的退出按钮。 |

#### 对象模型

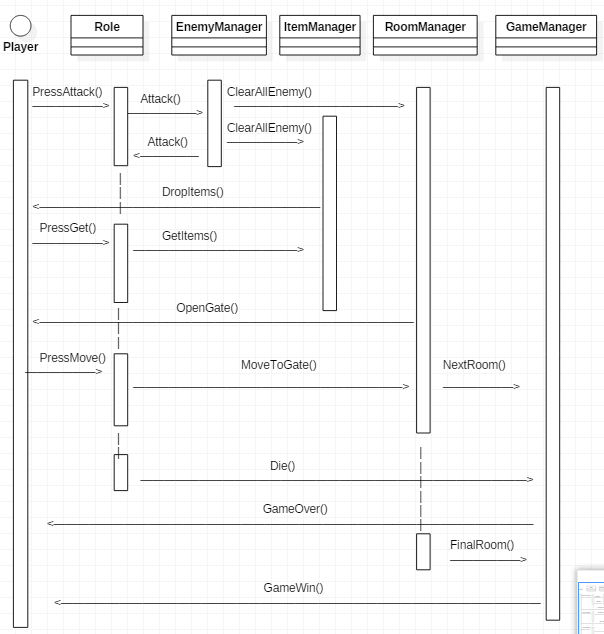
|  |  |
| --- | --- |
| 实体对象 | 定义 |
| 角色（Player） | 角色被玩家控制，对玩家输入的信息进行响应，直接与玩家进行交互。 |
| 怪物(Enemy) | 怪物由敌人管理者(EnemyManager)在每个关卡随机生成，并随着游戏的进行数量相应增加。怪物可以对玩家控制的角色进行攻击，也会承受来自角色的攻击。 |
| 道具（Items） | 道具由道具管理者（ItemsManager）在每个关卡结束后随机生成，可以被角色拾取和使用。 |
| 精华（Essence） | 精华由关卡结束后随机掉落，可由玩家拾取，不同种类的精华可以解锁不同的技能。 |
| 技能（Skill） | 技能由道具或精华获得，玩家控制的角色可以通过技能获得正面或负面效果。 |
| 游戏工具（GameUtil） | 游戏工具控制游戏中的实用工具，包括当前时间，音量大小等。 |
| 动作状态机（ActionStateMachine） | 动作状态机对应玩家的攻击按键，不同的按键组合顺序对应不同的状态机，根据状态机产生相应的连续攻击动作。 |
| 效果（BUFF） | 效果由道具、技能或地图中的特殊区域（陷阱、路障等）产生，直接对玩家控制的角色产生作用。 |
| BOSS | BOSS由BOSS管理者（BossManager）在每个关卡的最后随机生成，被玩家控制的角色击败后标志本关卡的结束。 |
| 怪物管理者（EnemyManager） | 怪物管理者控制关卡中怪物的随机生成、怪物的攻击、怪物的移动、怪物数量的变化、怪物出现的位置等。 |
| 道具管理者（ItemsManager） | 道具管理者控制关卡中道具的随机生成。 |
| BOSS管理者（BossManager） | BOSS管理者控制关卡最后的BOSS生成、BOSS出现的位置、BOSS的攻击、BOSS的移动等。 |
| 用户界面管理者（UIManaager） | 用户界面管理者统一管理游戏的UI界面，包括按键的布局，游戏信息的布局，界面字体的格式等。 |
| 游戏管理者（GameManager） | 游戏管理者管理整个游戏的进行，进入游戏战斗场景后对相应实体进行初始化，记录游戏的进行过程，判断游戏的进程是否结束。 |
| 音效管理者（SoundManager） | 音效管理者在玩家进行操作时产生相应的响应，播放音乐片段；音效管理者根据场景的变化，播放相应的背景音乐。 |
| 技能管理者（SkillManager） | 技能管理者存储所有的技能，初始全部为未解锁，根据玩家控制的角色获得的技能，判断是否已解锁，若该技能未解锁则角色获得此技能，若已解锁则没有响应。 |
| 特效管理者（EffectManager） | 特效管理者存储所有的特效，在玩家进行相应的操作时，特效管理者会判断此操作有无对应的特效，若有则播放相应的特效。 |

|  |  |
| --- | --- |
| 边界对象 | 定义 |
| “开始游戏”按钮 StartButton | 用户通过此对象开始一场游戏，跳转到游戏界面 |
| “图鉴”按钮 BookButton | 用户通过此对象进入图鉴界面 |
| “成就”按钮 AchievementButton | 用户通过此对象进入成就界面 |
| “设置”按钮 SystemButton | 用户通过此对象查看设置菜单 |
| “武器”按钮 ArmsButton | 用户通过此对象在图鉴中查看武器对象 |
| “技能”按钮 SkillButton | 用户通过此对象在图鉴中查看技能对象 |
| “怪物”按钮 EnemyButton | 用户通过此对象在图鉴中查看怪物对象 |
| “Boss”按钮 BossButton | 用户通过此对象在图鉴中查看Boss对象 |
| “道具”按钮 ItemsButton | 用户通过此对象在图鉴中查看道具对象 |
| “结局”按钮 EndingButton | 用户通过此对象在图鉴中查看结局对象 |
| “返回”按钮 BackButton | 用户通过此对象返回主界面 |
| “往左”按钮 LeftButton | 用户通过此对象在成就界面中往左查看下页成就 |
| “往右”按钮 RightButton | 用户通过此对象在成就界面中往右查看下页成就 |
| “奖励”按钮 RewardButton | 用户通过此对象在成就界面中领取奖励 |

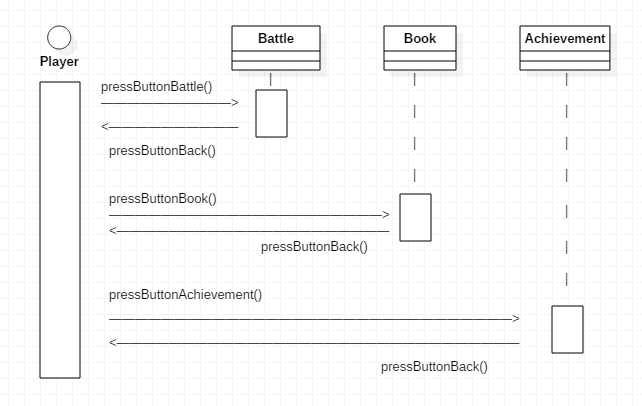
|  |  |
| --- | --- |
| 控制对象 | 定义 |
| UI控制 | 负责实体对象与UI控件间的数据传输 |
| 存档控制 | 负责从存档中读入数据,并转发到相应实体 |

#### 动态模型

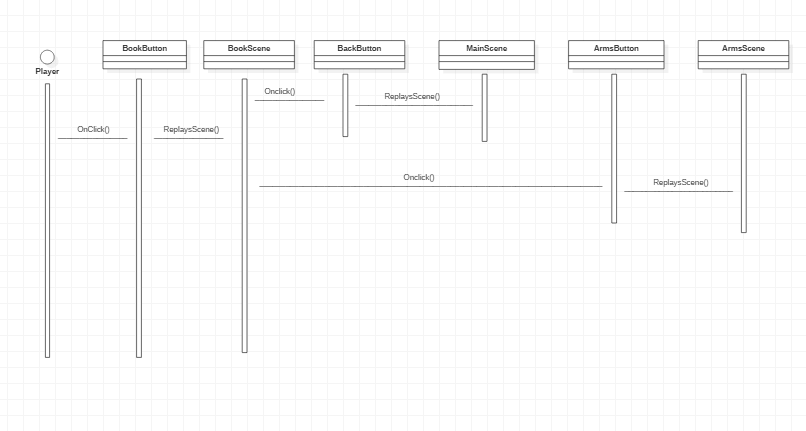
战斗顺序图：



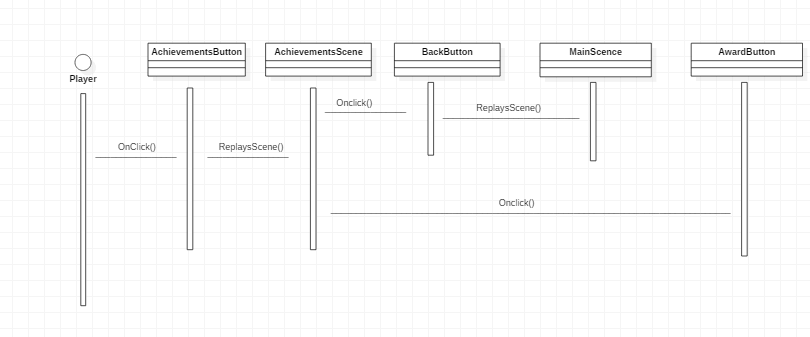
主界面顺序图：



查看图鉴顺序图



查看成就顺序图



## 4.术语表