# 系统设计文档

## 介绍

### 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 设计目标

1.易上手性：  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。  
2.可扩展性性： 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。  
3.低消耗性：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。  
4.流畅性：操作者进行游戏时游戏流畅而不会出现卡顿死机等情况。  
5.随机性：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。  
6.设备适应性：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的界面布局等不会重叠等。

### 1.3 定义、只取首字母的缩写词和缩写

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

### 1.4 参考文献

Unity3D 2D rogue-like实例教程

### 1.5 概述

本次要设计的是一款类型为rogue-like + act的安卓平台游戏,采用unity3D做为游戏引擎与C#作为脚本语言来研发。要求画面过关，运行流畅，可玩性高。

## 当前软件体系结构

点击游戏图标，进入到游戏的主界面，即游戏的第一层子系统。在当前系统中，玩家可以点击“开始游戏”、“图鉴”和“成就”三个按钮进入到按钮对应的下一层子系统，也可以点击退出键退出当前系统。

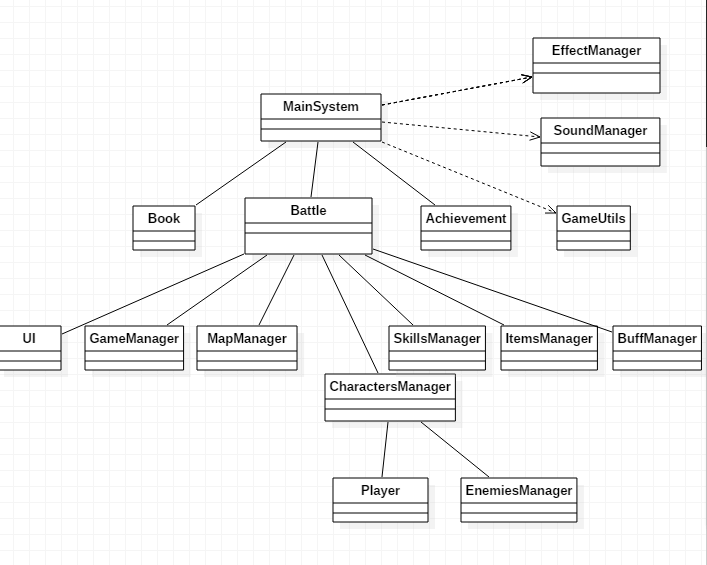
在第二层的战斗子系统中，玩家首先要选择重新开始或是读取存档，选择读取存档系统将读取玩家上次存储的所有数据，选择重新开始系统将会将存储的数据清空。游戏结束后玩家可以选择重新开始或者退出，选择重新开始玩家依旧会停留在当前子系统，选择退出则会返回上层子系统。在第二层的图鉴子系统中，游戏里的各种信息会随着游戏的进行而解锁，玩家可以访问解锁的数据，未解锁的数据玩家将无法访问，成就子系统和图鉴子系统类似。

## 建议软件体系结构

### 概述

游戏主系统下有三个子系统，分别为战斗、图鉴和成就子系统，其中每个系统都依赖于特效管理者（EffectManager），音效管理者（SoundManager）和游戏工具（GameUtils）。游戏子系统中有更多子系统的划分，是游戏的核心，玩家主要在此系统中进行游戏；图鉴子系统可供玩家查看游戏中怪物、技能等相关的信息；成就子系统可供玩家查看已获得的成就，并且完成成就后可以领取奖励。

### 3.2子系统分解



### 3.3硬件/软件映射

无。

### 3.4持久性数据管理

游戏首选项 使用XML存储

玩家存档数据 使用XML存储

注:怪物属性数据不算在持久数据中!

### 3.5访问控制与保密性

玩家在战斗系统中遇到的怪物，获得的道具，解锁的技能，击败的BOSS等会根据其类的标签进行标识，并在图鉴系统中将其解锁；玩家每完成一个关卡都会被标识，并在成就系统中解锁，玩家仅对解锁的数据具有访问的权限

### 3.6全局软件控制

游戏界面转换通过每个button来实现，当某个button被触发onClick()时，发生对应的事件流，从而控制整个软件的进程。

当一个button触发onClick()时，在产生事件流的同时又有一个button2的onClick()被触发，则此次触发无效。

### 3.7边界条件

点击游戏图标为启动游戏系统的边界条件；在主界面点击手机的退出按钮为退出游戏系统的边界条件；当游戏遇到错误行为时（如玩家手机电量不足等），游戏将会按照退出的边界条件处理。

## 子系统服务

### 术语表

|  |  |
| --- | --- |
| buff | 增益效果和减益效果的统称 |