

崩坏.io项目文档

项目信息介绍

1. 项目名

cocos2dx开发的io类游戏——崩坏大作战

2.项目成员

董 震宇 1852143

杨 础 1852344

3.小组成员贡献比例

董 震宇 50%

杨 础 50%

4.项目github地址

<https://github.com/yc111348/->

5.项目进度时间线

2019.05.19 搭建环境

2019.05.20 新建项目并上传 `github`

2019.05.30 加入 `主界面`、`登录界面`

2019.05.31 增加 `输入昵称` 功能,并存入 `UserDefault`

2019.06.01 加入 `选择武器` 和 `选择人物` 界面,实现播放 `背景音乐`

2019.06.02 加入 `人物` 和 `武器` 界面,实现 `人物动画`

2019.06.03 加入 `选择单机/联机模式` 界面,加入 `联机模式` 界面,实现 `弹出对话框`

2019.06.04 `创建地图` 并添加

2019.06.05 实现 `人物移动`

2019.06.06 实现人物移动与移动动画结合

2019.06.07 实现人物与地图的 `碰撞检测`

2019.06.08 实现 `人物攻击`

2019.06.09 实现 `游戏视角跟随人物移动`

2019.06.10 加入 `敌人AI`

2019.06.11 产生 `随机道具`,实现 `拾取道具`

2019.06.12 实现 `实时积分`,人物与AI `血条` 功能

2019.06.13 实现子弹与人物/地图的碰撞检测

2019.06.14 加入人物&AI属性,多个人物&AI的加入

2019.06.15 加入 游戏结束 界面

-

项目开发文档

1. 项目选用引擎

cocos2dx

2. 游戏整体架构设计及小组分工

2.2 小组分工

董震宇:

UI界面,地图

杨 础:

人物&AI

3. 用到的设计模式

4. 实现的功能点+实现思路

- ☒ 支持鼠标和键盘操作
- ☒ 支持道具拾取功能
- ☒ 支持动画
- ☒ 支持多种不同类型的武器
- ☒ 实现电脑AI加入
- ☒ 更多的动画
- ☒ 实现本地排行榜

鼠标和键盘操作

简单的侦听器,用map容器存储方向键对应的布尔值。
创建侦听器,在键盘按下和放回的时候改变对应按键的布尔值。在update里扫描map容易各个按键的布尔值,调用对应方向的移动函数。
update里记录当前方向与上一次扫描的方向,如果当前方向不同于上一次的方向,改变运行的动画数组,从而达到移动动画的卡顿问题,当所有键对应的布尔值都为0时,调用之前方向对应的站立动画,并将之前方向设为0,使站立动画避免重复调用而抽搐。

道具的创建和拾取

创建10个道具精灵类的组合，包括两个属性---种类和是否存在的布尔类型。

创建时候扫描10个精灵的是否存在值布尔值。若有0的出现变使该精灵变为新精灵，保证了地图上最多有10个道具的出现。

创建的时候random一个坐标作为道具的位置（需要进行碰撞检测避免道具出生在墙壁内），之后random一个0到10的值。根据random的区间确定刷新的种类并记录为道具的属性，从而实现道具的刷新概率。

Update里面随时调用hero与道具坐标的碰撞检测，结果为1时根据道具的属性值对主角的属性做出改变

动画

见 键盘操作

电脑AI

地图最多有3个敌人出现，创建3个敌人的精灵组，包含属性 血量 射程 攻击时间间隔 刷新闻隔 是否存在 移动间隔 速度 射速 攻击力。

包含函数 创建 移动 碰撞 方向计算 攻击 死亡 update整合。

update整合里面包含了需要在update里面调用的所有函数，使update看起来更简洁。

在 是否存在为0 与 刷新闻隔为1 的时候调用创建函数 每次创建都会提升敌人的速度 攻击力 射速 以增加游戏性，创建时random坐标并使用碰撞检测。

在移动间隔为1的时候调用移动函数，移动函数里面调用方向计算函数，计算方向时先取第一个方向为敌人与英雄最能接近的方向，之后进行与墙壁碰撞检测，碰撞检测为1时 从剩下的3个方向 random一个值并继续碰撞检测。

在攻击间隔为1的时候调用攻击函数，首先直接用自身坐标和英雄坐标计算出攻击的最佳方向的单位向量，之后用射程乘以单位向量之后调用子弹函数。

在血量为0时调用死亡函数，将攻击间隔移动间隔等所有值全调为0，并完成闪烁两下的动作，闪烁动作通过不断调用 延迟 和 可见度的函数实现，闪烁之后移除父类，并给出刷新闻隔。

本地排行榜

使用 *UserDefault* 创建包含当局得分 *Score* 在内的六组键值对,在每次进入排行榜界面时调用,用冒泡排序进行排列展示.

5. 项目实施过程遇到的困难+解决方案、解决过程

6. 亮点部分