

2021-06



2021-06

项目选题：元气骑士

小组信息：2053182 王润霖

1. 功能点实现（基础）



(1) 基础功能

☒ 游戏开始界面、暂停及恢复功能、退出功能

☐ 背景音乐、音量控制

☒ 键盘控制

☒ 至少2类武器，包括不限于枪械类武器、近战武器；每类武器至少包括2种具体的武器，如枪械类武器包括不限于一次发射单发子弹的枪械、一次发射多发子弹枪械等，近战武器包括不限于可快速攻击的小范围武器、攻击速度较慢的大范围武器等

设计两种枪械类武器，射程较远，杀伤性较弱，其中一种为单发，另一种为三连发

设计两种近战武器（刀剑），杀伤性较强，但要求近身攻击

☒ 至少2种普通敌人类型，包括不限于近战型敌人和远程型敌人；每类普通敌人至少包括2类具体的敌人

设计思路与武器类似，远程型敌人攻击距离远，杀伤力小；近战型敌人攻击距离近，杀伤力大

☒ 至少两种道具，包括不限于恢复生命、增加护甲的道具，可以通过击败敌人获取

☒ 宝箱机制，开启可获得新武器或者道具等

☒ 安全地图与探险地图，安全地图指进入关卡前的地图，可以查看武器信息、敌人信息等，探险地图指关卡地图

☒ 关卡中主角死亡，探险终止并返回安全地图

☒ 至少包括2个关卡

2. 功能点实现（创新）



(2) 创新功能

☐ 每次进入即使相同关卡也会有不同的游戏地图、敌人、道具等 (Roguelike!)

☒ 游戏小地图，可以显示已经探索过的地图，可以参考元气骑士的小地图

将地图放缩后固定在右上角

通过检测人物位置，实时更新地图探索状态，以探索过的地区用“explored”标记

☒ 多个可选主角，每个主角拥有不同的属性加成（近战攻击速度提升、枪械子弹速度提升、生命值增加等）或者主动技能（移速增加、翻滚躲子弹）

在安全地图中选择主角，同时主角的各种属性显示在安全地图中

☒ buff机制，主角可以获得多种buff，包括但不限于移速增加、无敌等

☒ debuff机制，主角会获得多种debuff，包括但不限于移速减慢等

buff、debuff的产生方式包括：击败敌人随机产生、击败BOSS无敌

通过特殊道具，可以增强buff的性能，或削弱debuff的性能

☒ 金币机制，击败敌人可以获取金币

☒ 主角等级系统，高等级主角拥有更强的属性或能力，主角可以通过消耗金币升级（在安全地图中）

在安全地图中升级主角，同时主角的各种属性和能力显示在安全地图中

☒ 安全地图中商店系统，可以在进入关卡前购买或抽取特殊道具、武器等

☒ 探险地图中的老爷爷系统，可以在关卡中购买/获赠礼物，如武器/道具/buff等

☒ 特殊地形，包括但不限于可被破坏区域（如木墙）、移速受限区域（如沼泽）、通过受伤区域（如岩浆）、机关地形（如时不时出现的地刺）等

设计沼泽、岩浆与地刺，影响人物速度、血量等属性

☒ BOSS型敌人秘籍，如十倍攻击力、无敌等

BOSS的血量更多，击败后获得“无敌”能力



3. 设计思路

(1) 设计**三个场景**，包括游戏封面、安全地图、探险地图

在安全地图中，实现 金币机制、角色选择、角色升级；在探险地图中，实现 地图探索、相互攻击、获赠道具

(2) 设计**人物类**

为骑士设置 血量、魔法值、**buff**、**debuff**等各类属性

(3) 设计**敌人类**

为敌人设置 血量属性，并设置 随机移动、射程检测、自动攻击

(4) 设计**武器、子弹类**

为武器设置 攻击性能、射程属性，实现武器与子弹对应，实现子弹 按射击角度旋转与定时消失



4. 技术难点

(1) 技术难点

在本项目中，出现了各种影响人物属性的机制。例如：buff机制、debuff机制、人物升级机制、角色选择机制、特殊道具机制等。实现**各类影响性能的机制准确叠加**，从而让角色升级与选择、buff/debuff等真正发挥作用，是本项目的技术难点之一。

(2) 解决方案

人物升级机制、角色选择机制所带来的属性，在人物属性初始化时进行设定。

buff、debuff机制所带来的影响，在相应的功能被执行时加入。

部分特殊道具可以改变buff/debuff的影响力，在buff/debuff产生影响时加入。

(3) 解决过程

在人物选择与升级之后，根据角色及其等级设定Grade、Type，并以此设定人物属性初始化时的参数。根据buff/debuff，为相应的参数设定倍率，例如，没有速度buff时，speed为人物属性初始化值 $\times 1.0$ ；而拥有速度buff时，speed为人物属性初始化值 $\times 1.2$ 。而部分特殊道具，会对buff/debuff的倍率产生影响。由此，**准确地得到真实的人物属性**。

项目在设计人物攻击时，还原了游戏中根据所持武器的射程，自动检测距离最近敌人的玩法。

```
Enemy* Enemy::NearestEnemy(int BulletType, Vec2 knightPos, Vector<Enemy*> enemyArray){  
    double distance = 300;  
    if (BulletType == 3 || BulletType == 4)//近身武器  
        distance = 80;  
    for (Enemy* enemy : enemyArray){  
        Vec2 enemyPos = enemy->getPosition();  
        double temp = pow((pow(fabs(knightPos.x - enemyPos.x), 2)) + (pow(fabs(knightPos.y - enemyPos.y), 2)),  
0.5);  
        if (temp < distance && enemy->getAlive() == true){  
            distance = temp;  
        }  
    }  
    //部分省略 }
```