



深蓝学院
shenlanxueyuon.com

项目四 思路讲解



助教

梁健



纲要

- 第一部分：作业分析
- 第二部分：思路讲解

➤ 基本作业：

- 游戏基本框架设计

➤ 扩展一：

- 增加首领房间

➤ 扩展二：

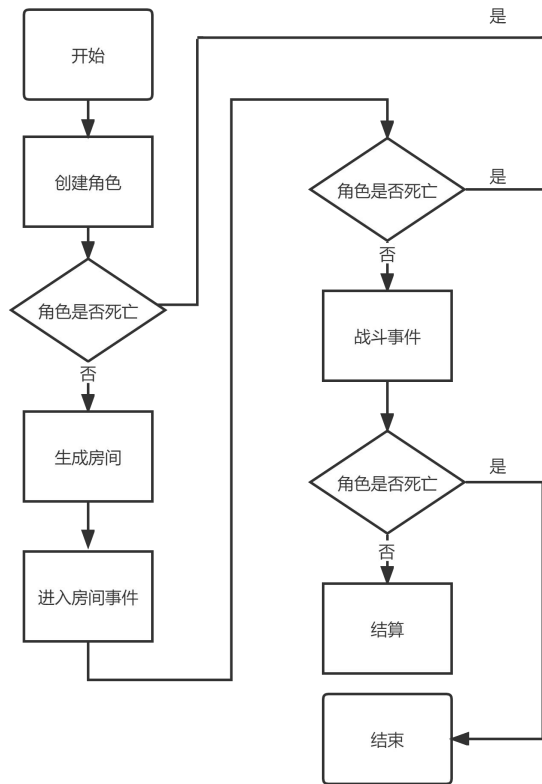
- 增加武器房间

纲要

- 第一部分：作业分析
- 第二部分：思路讲解

游戏基本内容

	营地	普通房间	陷阱房间	首领房间 (扩展一)	武器房间 (扩展二)
出现概率	10%	55%	15%	10%	10%
进入事件	恢复探险者全部的生命	10%概率恢复探险者10点生命, 并且每进入一个房间恢复5点生命, 持续2个房间	探险者损失当前10%的生命, 并且每次进入房间损失2点生命, 持续5个房间	恢复20点生命, 清除所有负面效果	记录冒险者当前的生命值
战斗事件	-	随机生成1-3个怪物, 每个怪物的属性值为怪物基本属性值的±40%, 进行 战斗	生成1个属性值为怪物基本属性值的两倍的怪物, 进行 战斗	生成一个怪物首领, 进行 战斗	生成1个属性值为探险者基本属性值0.4倍的怪物, 并装备一把随机武器, 进行 战斗
结算	-	每战胜一个怪物获得1点经验	获得1点经验	获得5点经验	获得2点经验, 恢复探险者进入房间时记录的生命值, 并装备怪物武器的全新复制



游戏基本内容

进入陷阱房间

触发事件, 损失10点生命, 造成持续损伤, 当前生命 (90/100)

遇到 怪物1 (20/20)

战斗开始

探险者攻击怪物1, 造成10点伤害, 怪物1 (10/20)

怪物1攻击探险者, 造成5点伤害, 探险者 (80/100)

探险者攻击怪物1, 造成10点伤害, 怪物1 (0/20)

怪物1死亡

获得1点经验, 当前经验 (1/10)

进入武器房间

持续损伤效果触发, 失去2点生命, 当前生命 (41/100)

持续恢复效果触发, 获得5点生命, 当前生命 (46/100)

遇到 怪物1 (40/40) //怪物装备了武器三, 但始终未触发效果

战斗开始

探险者攻击怪物1, 造成10点伤害, 怪物1 (30/40)

怪物1攻击探险者, 造成5点伤害, 探险者 (41/100)

探险者攻击怪物1, 造成10点伤害, 怪物1 (20/40)

怪物1攻击探险者, 造成5点伤害, 探险者 (36/100)

探险者攻击怪物1, 造成10点伤害, 怪物1 (10/40)

怪物1攻击探险者, 造成5点伤害, 探险者 (31/100)

探险者攻击怪物1, 造成10点伤害, 怪物1 (0/40)

怪物1死亡

获得2点经验, 当前经验 (6/10)

生命恢复, 当前生命 (46/100)

获得武器三

进入首领房间

持续损伤效果触发, 失去2点生命, 当前生命 (44/100)

持续恢复效果触发, 获得5点生命, 当前生命 (49/100)

生命恢复, 当前生命 (69/100)

持续损伤效果被移除

遇到 怪物首领 (40/40)

探险者攻击怪物首领, 造成11点伤害, 怪物首领 (29/40)

怪物首领攻击探险者, 造成15点伤害, 探险者 (54/100)

探险者攻击怪物首领, 造成12点伤害, 怪物首领 (17/40)

怪物首领攻击探险者, 造成16点伤害, 探险者 (38/100)

探险者攻击怪物首领, 造成14点伤害, 怪物首领 (3/40)

怪物首领恢复生命20点, 怪物首领 (23/40)

探险者攻击怪物首领, 造成14点伤害, 怪物首领 (9/40)

怪物首领攻击探险者, 造成17点伤害, 探险者 (21/100)

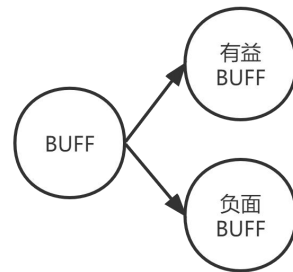
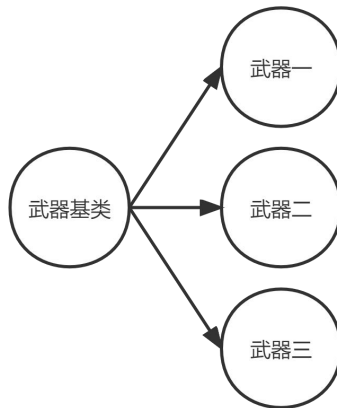
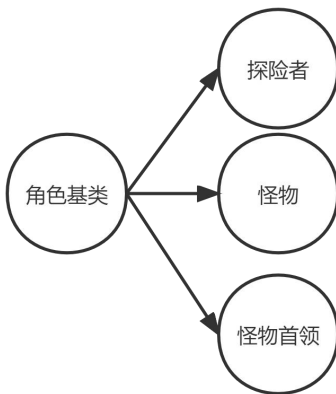
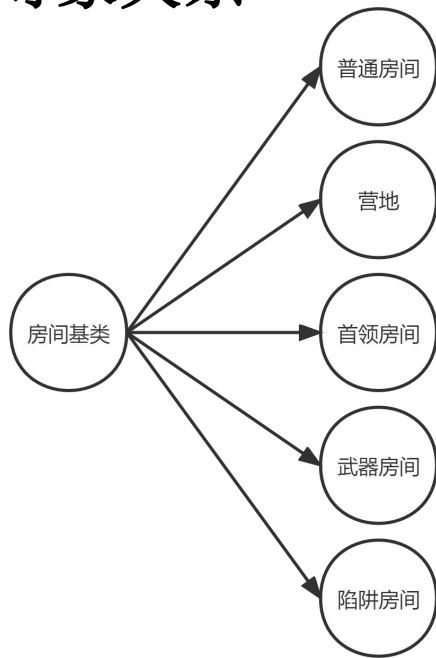
探险者攻击怪物首领, 造成14点伤害, 怪物首领 (-5/40)

怪物首领死亡

注意事项

1. 普通房间随机生成1-3只属性为基本属性 $\pm 40\%$ 的怪物（6-14）
2. 探险者首先攻击属性最高的怪物，怪物再依次攻击探险者
3. 武器3的双倍攻击力buff是在其存在时才有效
4. buff可以叠加，再次获得已有的buff时替换已有buff
5. 经验是在杀死怪物后获得的，升级是经验满之后即可升级（不是在结算时才能升级）

对象关系



房间对象

```
class Room
{
public:
    Room(ROOM_TYPE room_type, int before_ex_pt);
    virtual void enter_event(Hero& hero) = 0;
    virtual void battle_event(Hero& hero) = 0;
    virtual void settlement_event(Hero& hero) = 0;
    void room_event(Hero& hero);
    const int before_ex_pt_;
private:
    ROOM_TYPE room_type_;
};
```

```
class Normal_Room : public Room
{
public:
    Normal_Room(Hero& hero);
    void enter_event(Hero& hero);
    void battle_event(Hero& hero);
    void settlement_event(Hero& hero);
    void add_monster(Monster& new_monster);
private:
    std::vector<Monster> monsters_;
};
```

角色对象

```
class Npc
{
public:
    Npc(int hit_points, int damage, NPC_TYPE npc_type);
    int get_hit_pt();
    int get_damage_pt();
    virtual void change_hit_points(int change);
    bool is_living();
    int attack(Npc& enemy);
    friend std::string get_npc_name(Npc& npc);
    const int hit_limit_;

protected:
    int hit_points_;
    int damage_points_;
    bool living_flag_;
    NPC_TYPE npc_type_;

private:
};
```

```
class Monster : public Npc
{
public:
    Monster();
    Monster(const Monster& monster);
    Monster(int hit_point, int damage);
    int attack(Npc& enemy);
    bool is_equip_weapon();
    Weapon my_weapon_;
    void equip_weapon(Weapon& weapon);

private:
    bool equip_weapon_flag_;
};

class Boss : public Npc
{
public:
    Boss();
    Boss(const Boss& Boss);
    bool is_skill_available();
    void change_skill_flag();

private:
    bool skill_flag_;
};
```

陷阱房间：进入事件

```
void Trap_Room::enter_event(Hero& hero)
{
    std::cout << "Enter the Trap Room" << std::endl;
    int hit_loss = hero.get_hit_pt() * 0.1;
    hero.change_hit_points(-1 * hit_loss);
    std::cout << "The Hero lose " << hit_loss << " hit points. Hero("
        << hero.get_hit_pt() << "/100)" << std::endl;
    Buff new_buff(-2, 5);
    hero.add_buff(new_buff);
    hero.all_buff_work();
    hero.clear_invalid_buff();
}
```

陷阱房间：战斗事件

```
void Trap_Room::battle_event(Hero& hero)
{
    std::cout << "Meet The Monster1 (20/20)" << std::endl;
    while(hero.is_living())
    {
        if(monster_.is_living())
        {
            int attack_damage = hero.attack(monster_);
            std::cout << attack_info(hero, monster_, 1, attack_damage) << std::endl;
            Sleep(500);
        }
        if(!monster_.get_hit_pt())
        {
            std::cout << "The Monster1 is dead." << std::endl;
            hero.change_experience_pt(1);
            break;
        }
        int attack_damage = monster_.attack(hero);
        std::cout << attack_info(monster_, hero, 1, attack_damage) << std::endl;
        Sleep(500);
        if(!hero.is_living())
        {
            std::cout << "The Hero is dead." << std::endl;
            break;
        }
    }
}
```

陷阱房间：战斗事件

```
void Trap_Room::battle_event(Hero& hero)
{
    std::cout << "Meet The Monster1 (20/20)" << std::endl;
    while(hero.is_living())
    {
        if(monster_.is_living())
        {
            int attack_damage = hero.attack(monster_);
            std::cout << attack_info(hero, monster_, 1, attack_damage) << std::endl;
            Sleep(500);
        }
        if(!monster_.get_hit_pt())
        {
            std::cout << "The Monster1 is dead." << std::endl;
            hero.change_experience_pt(1);
            break;
        }
        int attack_damage = monster_.attack(hero);
        std::cout << attack_info(monster_, hero, 1, attack_damage) << std::endl;
        Sleep(500);
        if(!hero.is_living())
        {
            std::cout << "The Hero is dead." << std::endl;
            break;
        }
    }
}
```

陷阱房间：结算事件

```
void Trap_Room::settlement_event(Hero& hero)
{
    int ex_pt = hero.get_experience_pt() - before_ex_pt_;
    if(ex_pt < 0)
        ex_pt +=10;
    std::cout << "Leaving the Trap Room get " << ex_pt
               << " experience_pt." << std::endl;
    sleep(1000);
}
```



感谢各位聆听 !
Thanks for Listening

