



深蓝学院
shenlanxueyuan.com

C++ 基础与深度解析 Project4思路分享

助教

陈志伟



- 整体思路
- 面向对象设计
- 作业要求的细节

- 本次作业主要考察的是面向对象的设计，从作业描述中可以看出主要有三个不同属性的对象，分别是房间、角色、武器。具体的类型均是从这三个基类进行派生。
- 尽可能地将同一属性共性的内容放在基类实现，同一属性特例化的内容在基类定义为虚函数，并在子类重写。
- 尽可能通过动态绑定来进行实例化。

面向对象设计：运行流Flow

- Run () :
- InitRoom () :
- RoomEnter () :
- RoomAttack () :
- RoomAccount () :

面向对象设计：Room类

- EnterEvent ()
- AttackEvent ()
- AccountEvent ()

Ps. Flow类中RoomAttack和Room类中AttackEvent的区别：

在RoomAttack中会调用AttackEvent，同时会进行探险者的死亡检查

面向对象设计：Character类

- 通用基本属性

health:

health_base:

attack:

name:

- 探险者特有属性

buff:

experience:

experience_limit:

- Boss特有属性

health_up_chance:

- 通用基本功能：

isDead()

HealthDown()

HealthUp()

GetAttack()

GetHealth()

GetName()

- BOSS特有功能：

Check Health()

AttackUp()

- 探险者特有功能：

GetBuff();

RemoveDBuff();

CheckBuff();

SetWeapon();

CheckWeapon();

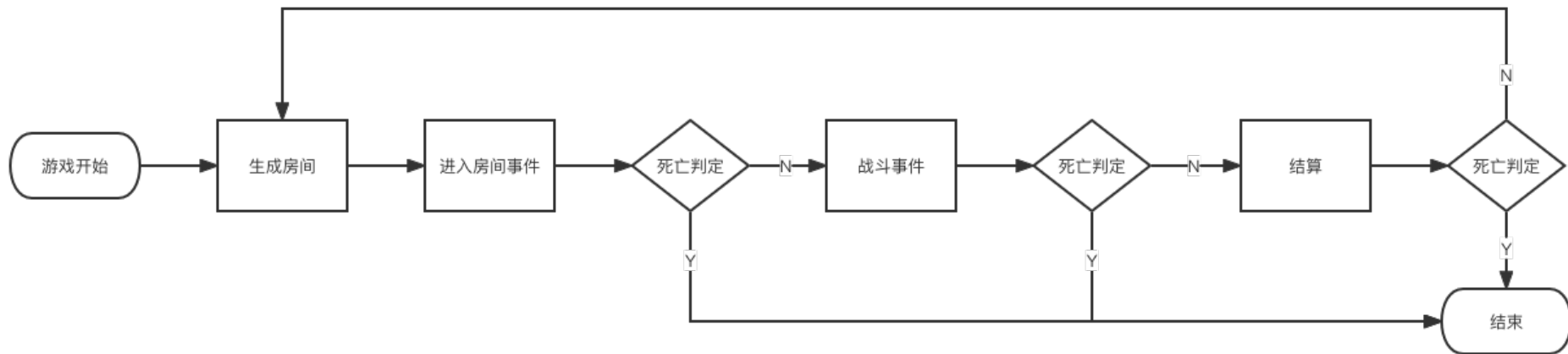
SetHealth();

MakeHealthMax();

Ps. 探险者的攻击计算需要加上武器的攻击力，需要对基类的函数进行重写。

武器类比较简单，不展开讲了

面向对象设计：处理流程



面向对象设计：一些细节

- 显示延时的实现：

Linux系统可以使用#include <unistd.h>头文件下的sleep函数。Windows可以使用#include <windows.h>头文件下的Sleep函数。

- 基于工厂模式的房间对象生成(下面只是一个简单例子)

```
class RoomFactory {
public:
    virtual std::shared_ptr<Character> CreatCharacter(int& health, int& attack, std::string& name);
};

class NormalRoomFactory : RoomFactory {
public:
    std::shared_ptr<Character> CreatCharacter(int& health, int& attack,
                                             std::string& name) override {
        return std::make_shared<Monster>(health, attack, name);
    }
};

class BossRoomFactory : RoomFactory {
public:
    std::shared_ptr<Character> CreatCharacter(int& health, int& attack, std::string& name) override {
        return std::make_shared<Boss>(health, attack, name);
    }
};
```

Ps. 也可以把生命、攻击力、名称的初始化放在CreatCharacter里面

作业要求的细节

- 对于武器三的描述：

武器的效果是“持有该武器攻击时有50%的几率使自身攻击力永久翻倍”，其自身指代的是武器本身而不是角色本身。如果使角色本身攻击力翻倍的话，很容易超过数据类型本身上限。

- 对于BUFF：

相同BUFF应该是叠加且独立计算时间。

- 对于怪物生成：

陷阱房间是生成“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”而武器房间是“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”。有同学把怪物基本属性和探险者基本属性搞混了。



深蓝学院
shenlanxueyuan.com

感谢各位聆听

Thanks for Listening

