



深蓝学院
shenlanxueyuan.com

Project 4 思路提示



主讲人 云行月下



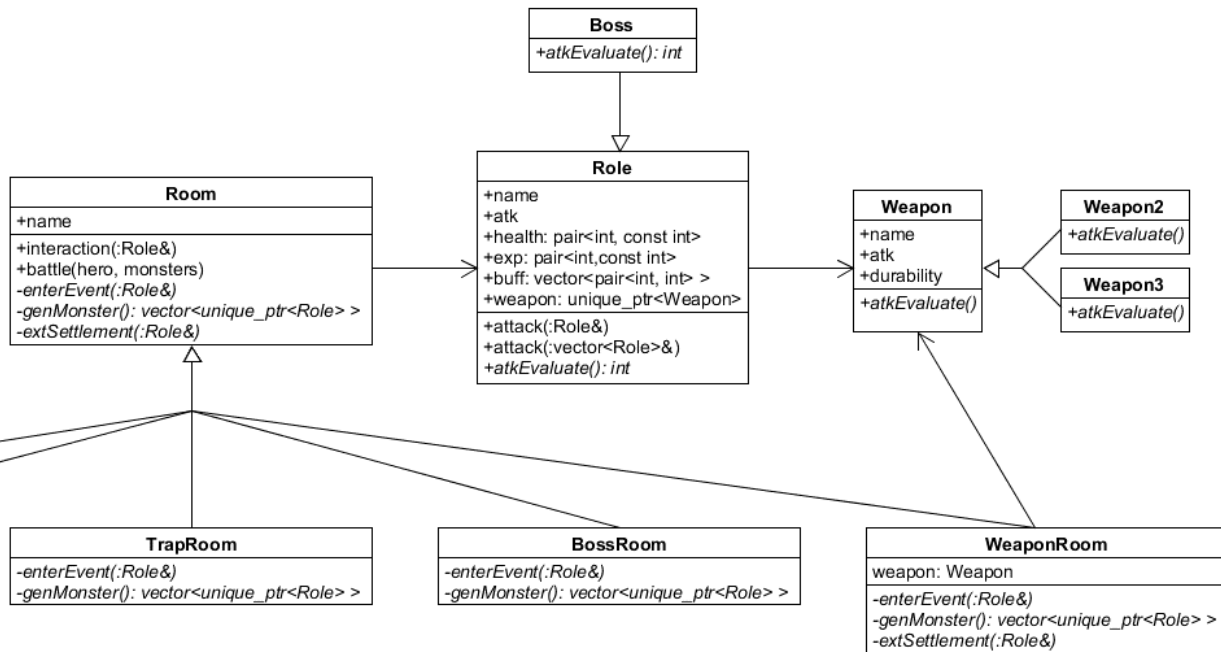
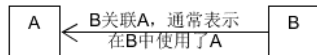
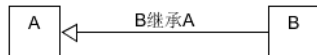
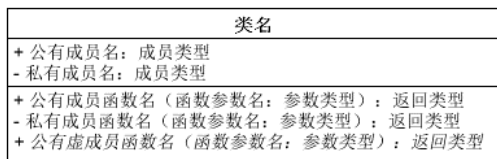
- 第一部分：整体思路
- 第二部分：面向对象设计
- 第三部分：一些细节

- 本次作业题目具有较强的面向对象属性，这些对象可以分成三类，即房间、角色、武器，我们可以对应定义三个基类，其他对象的类型都从这三个基类进行派生
- 我们应该尽可能的将处理逻辑进行泛化并在基类中实现，和具体类型相关的特殊逻辑定义为虚函数并在派生类进行重写
- 房间的处理流程和战斗的处理流程在每个对象中都是基本相同的，以房间为例不同之处仅在于进入事件、生成怪物、结算不同，那么我们可以将这三处不同定义为三个虚函数并在派生类重写

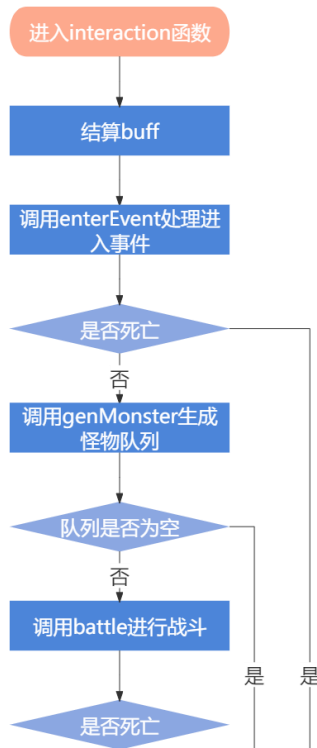
- 第一部分：整体思路
- 第二部分：面向对象设计
- 第三部分：一些细节

面向对象设计

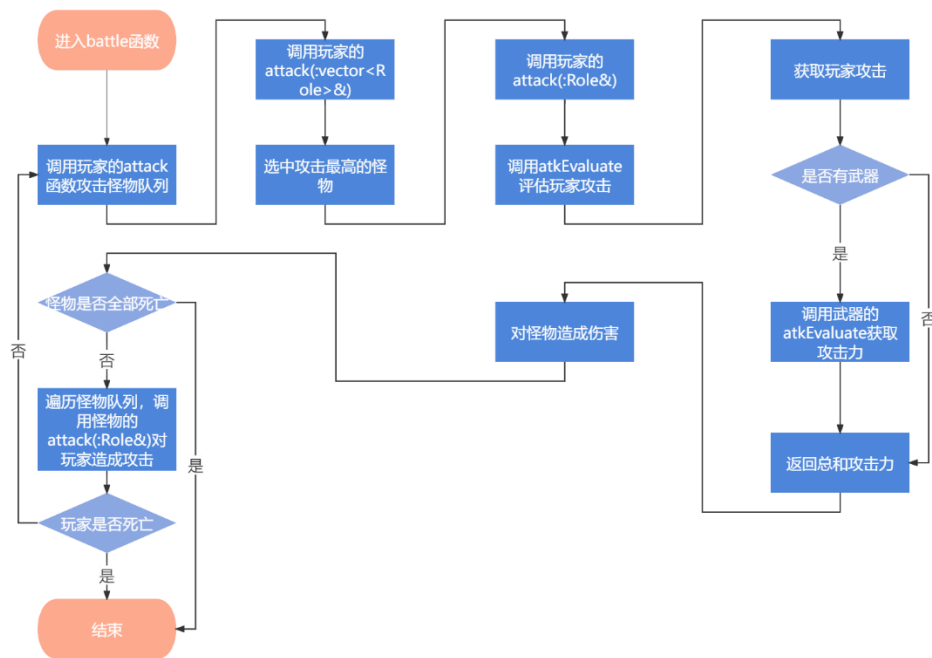
图例



- 房间的处理逻辑可以表示为以下流程图，在类图中对应函数为 interaction



- 战斗逻辑可以表示为以下流程图，在类图中战斗逻辑入口函数为battle
流程图中每一竖列代表一个函数



- 第一部分：整体思路
- 第二部分：面向对象设计
- 第三部分：一些细节

- 房间、怪物和武器的生成可以使用工厂函数产生，以武器为例实现如下

```
std::unique_ptr<Weapon> createWeapon()
{
    int r = rand() % 3;
    switch (r)
    {
        case 0:
            return std::unique_ptr<Weapon>(new Weapon());
        case 1:
            return std::unique_ptr<Weapon>(new Weapon2());
        case 2:
            return std::unique_ptr<Weapon>(new Weapon3());
    }
}
```

一些细节

- 在每行输出结束时延时可以通过如下方式实现，等待时间可用另外的变量表示

```
struct Endl{};

inline ostream& operator<<(ostream& out, const Endl&)
{
    Sleep(1000);
    out << std::endl;
    return out;
}

int main()
{
    std::cout << Endl{};
}
```

- 应当在角色基类中定义一个武器类型的指针来表示武器，如果指针为空表示当前未装备武器，建议使用unique_ptr代替裸指针



深蓝学院
shenlanxueyuan.com

感谢各位聆听 !
Thanks for Listening

