



深蓝学院
shenlanxueyuan.com

Project 4 作业提示

主讲人 周奕端



- 第一部分：流程设计
- 第二部分：类设计
- 第三部分：实现细节

第一部分：流程设计

- 流程主循环：

```
While(true){  
    checkEntry(); // 进入事件  
    checkDeath(); // 死亡判定  
    checkBattle(); // 战斗事件  
    checkDeath(); // 死亡判定  
    checkEnd(); // 结算  
}
```

- 当且仅当checkDeath()内部会跳出主循环。

第一部分：流程设计

● 战斗流程：

```
While(true){
```

```
    // 探险者根据怪物攻击力，随机攻击一个攻击力最高的怪物
```

```
    // 更新怪物数量，删除死亡怪物
```

```
    foreach monster in monsters:
```

```
        // 每个怪物依次攻击探险者，每次均判断探险者情况
```

```
}
```

- 第一部分：流程设计
- **第二部分：类设计**
- 第三部分：实现细节

第二部分：类设计

- 主要的类：

- 角色类

- 探险者、怪物、首领有着不同的行为模式，无需继承同一基类；
 - 探险者除了生命值和攻击力，还需要记录武器和buff；

- 房间类

- 每个房间都会触发同样的事件，行为模式相同，可以使用虚函数实现多态；

- buff类

- 包括持续时间和效果；

- 随机数

- 系统需要大量的随机数，可以使用一个类封装一个随机数生成器。

第二部分：类设计

●角色类

- 记录攻击力、生命值、生命上限、经验值和经验上限；
- 探险者用一个数据结构记录当前获取的全部buff，为了方便添加和删除操作，哈希表（unordered_set）是比较合理的选择；
- 探险者和怪物需要记录持有的武器，可以考虑optional模板来处理没有武器的情况。

●房间类

- 房间有着相同的行为模式，可以设计一个虚基类将各类事件接口写好，其他具体的房间类实现自己的行为；
- 由于不是所有房间都有某些特定的事件，虚基类应给出自己的默认实现而非直接设为纯虚类；

第二部分：类设计

●buff类

- 使用一个成员变量记录buff剩余时常；
- 不同的buff有着自己的效果，如果用虚函数的方式会使得每种不同效果都要有一个新的类；
- 考虑用buff类记录一个函数指针，在需要新的函数时给该函数指针赋一个lambda表达式，实现匿名函数伪装的“多态”；

```
explorer.buffs.insert(Buff{ 2, [](Explorer& explorer){}});
```

●随机数

- 可以考虑STL库里的<random>文件，如uniform_real_distribution用于生成给定范围的均匀分布。

- 第一部分：流程设计
- 第二部分：类设计
- 第三部分：实现细节

第三部分：实现细节

- 主循环判断死亡后需要跳出循环，需要注意break的时机；
- 武器也是对象，注意在合适的时机更改耐久并删除破损武器；
- C++虚函数实现多态只能在基类指针指向子类对象或者基类引用子类对象才能起作用；
- 每一个行为之后都会有结果打印，需要处理打印位置以防重复打印；
- 本次作业逻辑上很简单，难点在于大量使用面向对象思想且代码量较大，在调试过程中需要理清思路。



感谢各位聆听 !
Thanks for Listening

