**《西游记》游戏模拟**

**一、功能分析**

**1.1游戏运行界面**

要完成一个《西游记》游戏模拟­——当然它是基于文本方式的（它不是图形的，标准C++并没有提供图形操作的功能。如果要实现图形编程，那还要学习其他库的使用），通过文本的输出反应程序的运行状态。此外，它只是简陋的模拟了《西游记》的主干思路，在某些地方为了实例的需要还做了一些简化的调整（很显然，这个游戏离原著的差距很大）。

最后生成的xiyou.exe的运行界面非常简单。

由于xiyou.exe的输出信息很多，直接运行可能会发现刷了几次屏之后，无法再看到最初的输出的内容，这时候需要调整以下窗口的输出缓冲区的大小。

**1.2游戏角色的划分**

《西游记》的角色应该包括：

普通角色：如普通行人；

和尚：专指取经的和尚，如唐三藏等；

妖怪：会吃人，总是和取经的和尚作对，如白骨精、黑熊怪；

神仙：功力强大，会在危难关头帮助和尚，如观音菩萨等。

角色名单依赖于配置的角色表，程序根据角色表动态构造《西游记》中的角色。

所有的角色都有男女之分，如：和尚的性别总是男的，白骨精就是女的。

所有的角色都有功力大小之分，如：孙悟空的功力与二郎神差不多，但是比猪八戒厉害，等等。随着时间的推移和战力的情况，每个角色的功力会发生变化，如：妖怪每天功力都会有所增加。

角色中还包括一种妖怪，譬如：猪八戒，他在取经之前就是一个地道的妖怪，绰号“猪刚鬣”，整天干的也是吃人的勾当。但是入了佛们之后，上了西天路，便是和尚了，除了多吃点，与唐三藏没有什么区别。

妖怪除了通过角色表来构造，为了增加游戏的难度，还需要增加若干个小怪物，这个数目可能很大，如50个。

**1.3战斗的形成**

《西游记》游戏总共包含81关，每一关又包含多少个回合，由双方的战斗情况而定。唐三藏一行只有顺利通过了81关，或者消灭掉所有的妖怪，才可以胜利结束游戏。

唐三藏总是在每一关的开始被妖怪抓住，如果在限定时间内（5天）唐三藏没有被徒弟们救出，就会被妖怪吃掉，游戏直接以失败告终。

最初，唐三藏只有且必须有一个徒弟，即孙悟空（必须要有一个孙悟空，否则唐三藏在第一关就会挂掉），一路上他们会依次收服白龙马、猪八戒和沙和尚。

每一回合存在着战斗的两方，即唐三藏所率领的出国取经团（简称“我方”）和各种各样的妖怪（简称“敌方”），双方通过功力与兵器的比较结果决定胜负。战斗结束后，胜利的一方，功力会提升，负方的功力会减少。

如果我方战败，则可以再求助于神仙；再败，则再求，直到胜利为止。当然，前提是唐三藏还活着，一旦他被妖怪吃了，游戏直接结束。

如果敌方战败，则根据实际实力的悬殊决定敌方是否被消灭还是逃跑。

战斗结束后，战胜方的功力值会增加，战败方如果没有被消灭，他们的功力值会减少。

**1.4兵器的分配**

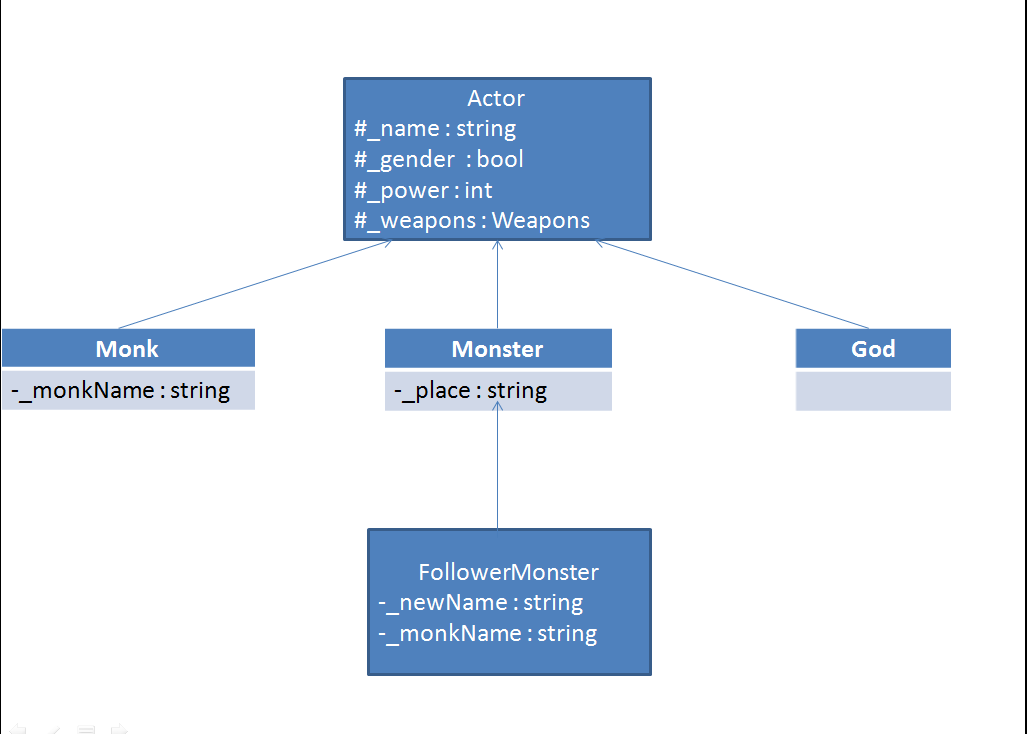
所有的角色都有可能佩带兵器，兵器有威力大小之分，如:孙悟空的如意金箍棒比猪八戒的九齿钉耙厉害。

战斗结束时，负方的兵器会被没收，胜负会缴获负方的兵器。苦行僧一路奔波，取经的和尚只能携带一种兵器，而妖怪则可以同时携带N种兵器。

**二、设计思路**

**2.1使用类和继承完成角色的设计**

根据面向对象编程的思想，可以将所有的角色划分成多个具有层次关系的类，如下图：



解释如下：

Actor: 角色类，作为共同的基类存在；

Monk: 和尚类，继承于Actor。如: 唐三藏属于Monk类；

God： 神仙类，继承于Actor。如： 观音菩萨属于God类；

Monster: 妖怪类，继承于Actor。如： 白骨精属于Monster类；

FollowerMonster： 随从妖怪类，继承于Monster。随从妖怪类用以描述等待跟随唐三藏的妖怪，如： 白龙马、猪八戒和沙和尚。

**2.2使用文件流加载角色列表**

角色列表通过文本文件的形式提供，如下示出程序采用的配置文件的格式：

####演员表

####和尚列表

monk=唐三藏，唐玄奘，1

weapon=九环锡杖，20

monk=孙悟空，孙悟空，72

weapon=如意金箍棒，99

####随从列表

follower=小白龙，20，蛇盘山鹰愁涧，白龙马，白龙马

follower=猪刚鬣，48，云涧洞，猪八戒，猪悟能

weapon=九齿钉耙，48

follower=流沙河怪，36，流沙河，沙和尚，沙悟净

weapon=降妖仗，36

####神仙列表

god=如来佛祖，true，1000

god=观音菩萨，false，500

角色列表文件允许以“#”开头引起行注释，如果某一行没有任何字符，则不被程序处理。每个角色都如下描述:

monk=唐三藏，唐玄奖，1

等号“=”前面的monk代表角色的类别(由于C++没有类的反射机制，所以此处不能真接采用类名，如: Monk),后面的序列则用以指定该角色的属性。如上面的语句说明:定义一个和尚，名字唐三藏，法名唐玄类，功力1 。

不同的角色类，其属性也不样， 如: follower (即FollowerMonster) 小白龙，其名字小白龙，功力20,其所在地为蛇盘山鹰愁涧，收服之后的名字为白龙马，法名白龙马。

每个角色可以佩带兵器，其兵器的配置置于角色定义之后:

monk=孙悟空，孙悟空，72

weapon=如意金箍棒，99

这样即表明，为和尚孙悟空佩带上如意金箍棒，其威力为99。可以同时为一个角色佩带多个兵器:

monster=金角大王，true, 50，平顶山莲花洞

weapon=红葫芦，80

weapon=玉净瓶，8

相信看过80版《西游记》电视剧的读者都会记住金角大王的红葫芦和玉净瓶，以上的配置体现出来了这个一对多的关系。

**2.3使用虚函数实现角色的多态**

角色中除了继承性，还存在着多态性，这些多态性具体体现在：

每种角色说话的时候可能语气会不一样；

随着时间的推移，妖怪每天的功力会自动增加2。而其他的角色不会增加；

战斗结束时，战胜方没收战败方的兵器，但神仙除外，神仙不参与哄抢兵器的行列；

战斗结束时，战胜方没收战败方的兵器，但前面提过，苦行僧一路奔波，取经的和尚只能携带一种兵器，而妖怪则可以携带N种兵器，这也是一种多态；

战斗结束时，战败方要求交出兵器，但像FollowerMonster猪八戒这样的 ，不必交出兵器；

妖怪被消灭时，会被杀死（销毁），但像FollowerMonster猪八戒这样的，它会同时构造一个新的和尚角色，改名，加入取经人的行列。

针对这些多态性，程序将做如何实现，很显然，需要用到虚构函数。基类Actor提供了虚函数。

尽管在析构的时候，每个角色所做的动作没有什么不同，但是作为一种惯例，仍然将析构函数声明为虚函数。

使用onXXX()作为函数名遵循了另外一种不成文的惯例，它提醒用户这类似与对某种事件的处理，如：onLose()，它的含义是：当战败了该执行的动作。on可以理解成“当”。onEveryDay()、onLose()、onWin()、onKilled()会被外围的主控程序，如：Game，由它来在遇到某种具体的事件的时候激活。

**2.4使用操作符的重载**

尽管不赞成过多的操作符重载，但有几个可以正当使用到操作符函数的场合：

●重载Actor的“+=”: 完成功力值的提高，如:

fctor += 2；

●重载Actor的“\_=”： 完成功力值的减少，如：

actor -= 2;

●重载 Actor的“<<":完成新兵器的添加，如:

actor << Weapon(" 如意金箍棒"，99);

●重载Actor的“>>”:完成兵器的解除，如:

actor>> Weapon;

●重载Weapon的“<”和“>”:完成兵器功力的比较，这个在容器和排序算法中可能会用到，如:

priority\_ queueWeapon, vector<Weapon> > weaponsi

●重载全局的“<<”:完成角色信息的输出，如下形式:

ostream & operator<< (ostream & cout, Actor 8 actor);

cout << actor << endl;

●重载全局的“<<”:完成兵器信息的输出，如下形式:

ostream & operatore<<(ostream & cout, Weapon & weapon);

cout << weapon << endl;

**2.5使用RTTI获取角色的类别**

有的时候需要根据某个Actor获取它的类型，使用RTTI可以达到这个目的。如下场合：

（1）组建敌方的时候，需要知道哪些角色是妖怪

（2）筛选神仙的时候，需要知道哪些角色是神仙

**2.6使用STL容器管理角色**

在本游戏中，所有的角色都是动态构造的（具体来说，它们是new出来的），因此需要容器将它们管理起来。STL容器是再好不过的选择。

**2.7使用STL容器管理兵器**

兵器（Weapon）也需要管理起来，譬如：每个角色所佩戴的兵器，被收缴的所有兵器，这些都需要管理起来。

管理角色的时候没有对角色之间进行排序，但是对于兵器，最好的选择是需要排序。譬如：收缴兵器的时候，按照人类贪婪的想法，肯定是先轮到的角色缴获最好的兵器，后轮到的角色只能从剩下的兵器中缴获最好的了。

**2.8使用循环和判断完成主控逻辑**

程序的主控逻辑应该是本游戏的核心，Game类提供了play()函数用以实现游戏的开始，play()函数制造了一个大的如来大循环（九九八十一难），每一个循环通过调用play(int level)函数来完成，如下图所示：

（1）play()函数的逻辑设计

《西游记》开始

组建取经者的队伍

i<=81?

是

i++

play(i)

否

《西游记》结束

（2）play(level)的逻辑设计

本关开始

组建敌方

逮住唐僧

在限期之内？

是

营救唐僧

成功营救？

否

求助神仙

是

收缴敌方武器

否 否

消灭掉妖怪

可以逃走？

是

妖怪逃走

本关结束

**三、实现详解：**

本游戏的实现包含如下文件：

weapon.h: 兵器类的头文件；

weapon.cpp: 兵器类的实现文件；

actor.h： 角色类的头文件；

actor.cpp： 角色类的实现文件；

monk.h： 和尚类的头文件；

monk.cpp： 和尚类的实现文件；

god.h: 神仙类的头文件；

god.cpp: 神仙类的实现文件；

monster.h: 妖怪类的头文件；

monstet.cpp: 妖怪类的实现文件；

follower.h： 随从类的头文件；

follower.cpp: 随从类的实现文件；

game.h: 游戏类、异常类的头文件；

game.cpp: 游戏类的实现文件；

gamexcept.h: 异常类的头文件；

functoers.h: 函数对象类的头文件；

xiyou.cpp: main()函数；