魔兽世界1文档

xxxxxxxxx xxx

# 说明

在对应的文件夹中有类图classdiagram.drawio，使用<https://app.diagrams.net/> 可查看。

# 需求分析

首先提取相应的名词，决定要设计哪一些类，这里初步提取出“司令部”、“城市”、“武士”、“事件”等名词。

# 需求实现

## 司令部、城市

司令部和城市有共性，所以让它们都继承一个公共的基类Building，以分类对象，方便以后的需求修改，同时实现使用上的抽象依赖。

## 武士

题目中提到有5种武士，所以每种武士对应一个类，并且他们都需要继承一个公共的Warrior类，方便将通用的方法往基类放，比如每种武士都有编号，生命值和攻击力，所以这些成员都应该放在基类中，并提供对应的getter和setter。

## 武士建造者

由于武士需要被建造，考虑到将来不同的武士的建造方式可能有所不同，故在此采用建造者（或许应该叫工厂？比较模糊）模式来建造武士。每种武士写一个建造者类，比如Dragon就写一个DragonBuilder专门用于生产Dragon这种武士。所有的武士建造者类都继承一个公共的WarriorBuilder类，原因同上。由于建造者需要生产武士，所以建造者类需要life和attack两个成员属性分别指示这种武士的初始生命值和攻击力，并在构造武士时使用武士的setter方法设置进去。

## Logger

由于需要打印对应的信息，所以写一个Logger类，专门用于打印输出。

## 事件系统

由于游戏以事件和时间的方式进行，所以考虑观察者模式（发布订阅模式）。首先编写一个公共的基类GameObject，使得游戏中的所有对象都继承这个基类，GameObject中有name和side两个成员变量，分别表示这个游戏对象的名字和属于那一边（红或者蓝），除此以外，还在其中提供一个acceptEvent方法，用于给子类重写以处理EventSystem发送的事件。在EventSystem类中使用一个Map记录所有订阅到事件系统的游戏对象，并提供注册和反注册方法，任何一个游戏对象都可以通过注册方法注册到事件系统，也可以通过反注册方法从事件系统中移除出去。EventSystem中还有一个notify方法，根据游戏对象的名字发送具体的消息，也就是在Map中查找对应的对象，调用它的acceptEvent方法。最后因为是模拟到一定的时间就停止模拟，不一定所有的游戏对象都会执行反注册，所以需要一个reset方法，清空Map。

## 事件

既然有事件系统，肯定有事件，这里的事件指的是生成武士的事件，暂时称之为TimeEvent，因为考虑到以后可能会有更多的对象存在，所以这里让TimeEvent继承一个公共的事件基类Event。在TimeEvent中有一个time的成员变量，指示这个事件发生的时间，同时重写toString方法，用于输出时间。

## 游戏时钟

既然游戏是以时间为单位触发事件的，那就写一个Clock类，在这个类中，有一个time的成员变量初始化为0，指示现在的时间，然后写一个start方法，做无限循环，每次将time设置到TimeEvent中，并调用事件系统的notify方法将这个时钟事件依次通知给红方和蓝方的司令部，最后将time + 1，完成时钟的模拟，直到所有的游戏对象都从事件系统中反注册后时钟循环停止，游戏结束。

## 武士生成

首先司令部需要将生成好的武士存储起来，所以做一个Map<Class, List>，使用某种武士的Class作为键，一个保存武士的列表作为值。除此以外，司令部还需要有武士建造者，所以使用一个队列存储所有的武士建造者类。司令部有生命元，使用一个成员变量lifeElements存储，司令部还需要给武士定编号，所以需要有一个counter成员变量指示现在正在生产第几号武士。

具体的生成逻辑在acceptEvent中实现，首先如果发现事件的类型是TimeEvent，首先使用for循环遍历队列的大小，然后每次从队列中取出一个武士建造者，如果发现可以建造，就调用建造者的建造方法构造一个武士，然后将自己的边（红蓝）赋给建造好的武士，因为武士和司令部都是GameObject，所以直接调用setSide和getSide方法即可。紧接着将counter作为编号设置到生成好的武士中，并将其放入到Map中完成建造的工作，最后调用Logger类打印对应的信息（包括使用这个武士的Class类从Map中获取List的大小并输出），使用return语句返回，表示本次事件处理完毕。

如果遍历某一个武士建造者时发现不能建造这种武士，那就将其放回队尾，继续下一轮循环，如果循环完所有的武士建造者都发现无法建造武士时就打印停止建造的信息并将自己从事件系统中反注册。

## Main方法

最后是Main方法，对于每一组输入，构造对应的武士建造者，构造司令部，设置司令部的side和name，将武士建造者添加到司令部中，将司令部注册到事件系统中，构造Clock，调用Clock的start方法，最后调用事件系统的reset方法，任务完成。