**卡坦岛online更新说明**

**重构**

**谢本壹、潘高培、宁可**

**2018.1**

目录

[总体UML图 2](#_Toc503521449)

[Patch V0.2（粉色） 3](#_Toc503521450)

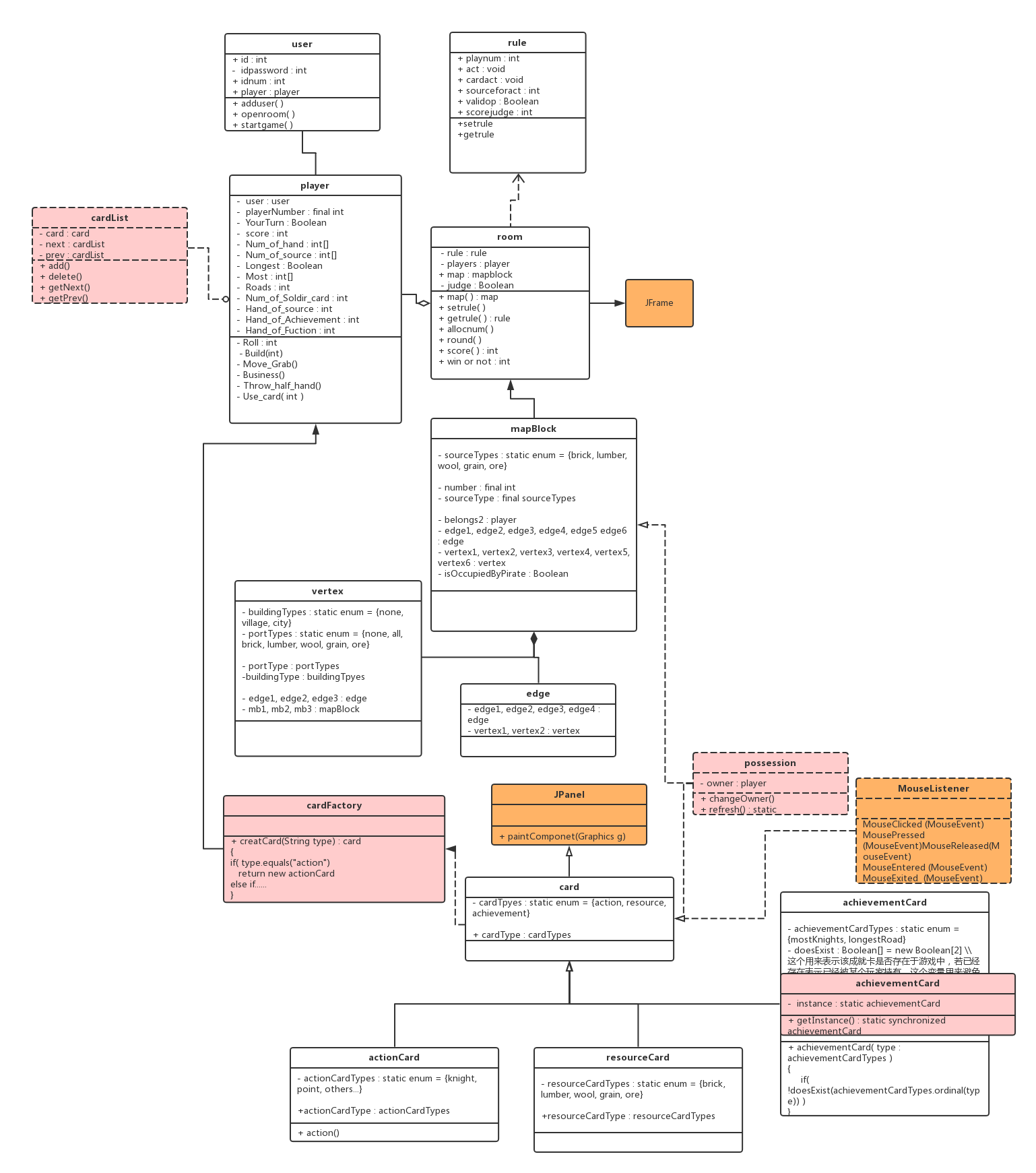
[卡牌类 3](#_Toc503521451)

[成就卡 3](#_Toc503521452)

[Patch V0.3（橙色） 3](#_Toc503521453)

[GUI和鼠标操作 3](#_Toc503521454)

# 总体UML图

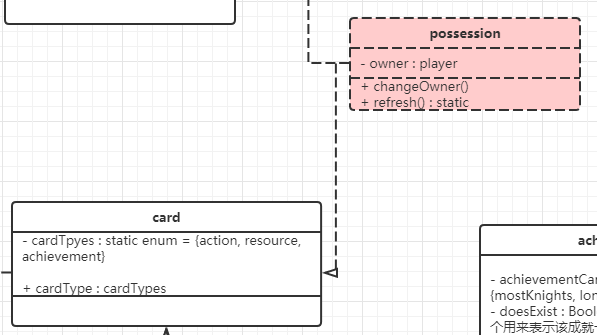
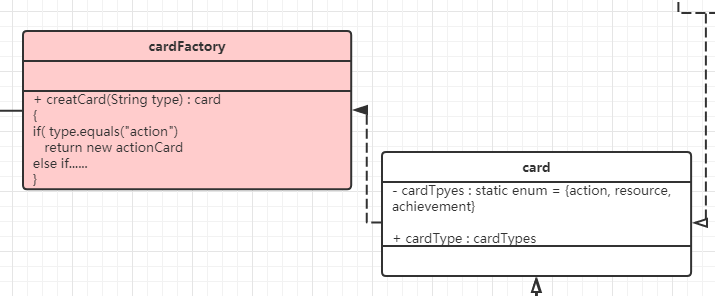
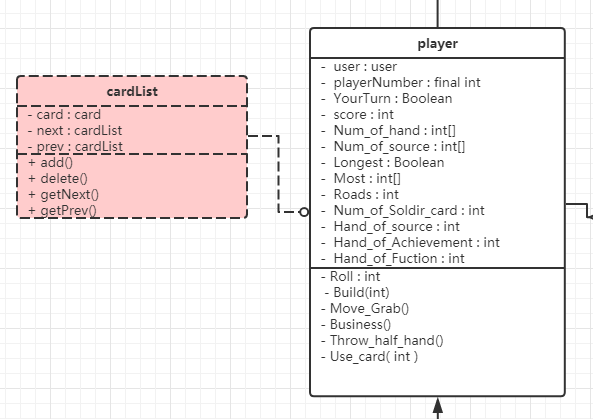


# Patch V0.2（粉色）

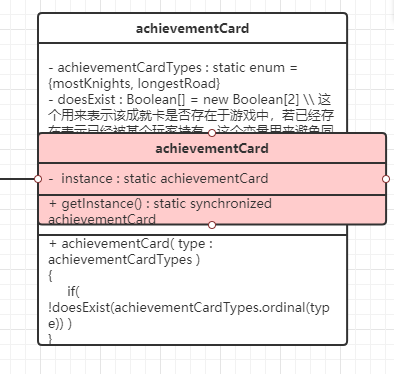
改动主要是围绕着“把card从int值变成一个个对象”。这样的好处是在后续版本中加入图形界面和鼠标点击时只需要给card一个paint接口和mouseListener的接口即可。

具体的重构：

## 卡牌类

* 新增possession接口。方法中的refresh用于刷新持有者（player对象）中的cardList。所以手牌是通过计算出来的，可以一定程度上避免外挂直接修改手牌。地图块的possession同理。这里用到了类似于观察者模式的开发方法，区别是，这里并不用维护一个观察者表，因为玩家数固定了，根据refresh的参数里的玩家不同。比如player1夺取player2的卡牌，这是refresh（player1，player2）即可；又比如，强盗事件发生后refresh（all）即可。
  + 在refresh（）中不仅要完成玩家手牌的重新计算，还有完成成就判断（比如最长路径和最多骑士），有refresh（）来调用判断成就的方法。
  + 
* 新增cardFactory，工厂方法。玩家每次摸牌相当于用cardFactory实例化一个card，但是具体是摸actionCard, resourceCard or achievementCard只有在游戏过程中才能确定。
  + 
* 新增cardList。相当于一个链表。
  + 

## 成就卡

* 每种成就卡场上只存在一张。
  + 在“设计”中用的是一个Boolean数组doesExist来表征场上是否有对应的成就卡。比如doesExist[0] = false 表示最多骑士成就还没有被实例化。这样的做法是每次实例化和删除对象时都要对doesExist检查修改。
  + 现在用singleton懒汉模式进行了修改，保证了每张成就卡只有一张，且代码简洁了许多。
  + 

# Patch V0.3（橙色）

## GUI和鼠标操作

应用了观察者模式

* JPanel类中含有paintComponet的方法，让card继承后，覆盖这个方法。因为actionCard、resourceCard和achievementCard都是card的子类，所以每个子类各自在覆盖paintComponet（）时先调用super.paintComponet()然后在添加各自独特的绘制方法即可
* Listener
  + 先实现Listener接口中的每一个方法（可以空着），然后用addXXXListener注册即可。

