

# 卡坦岛 online 更新说明

## 重构

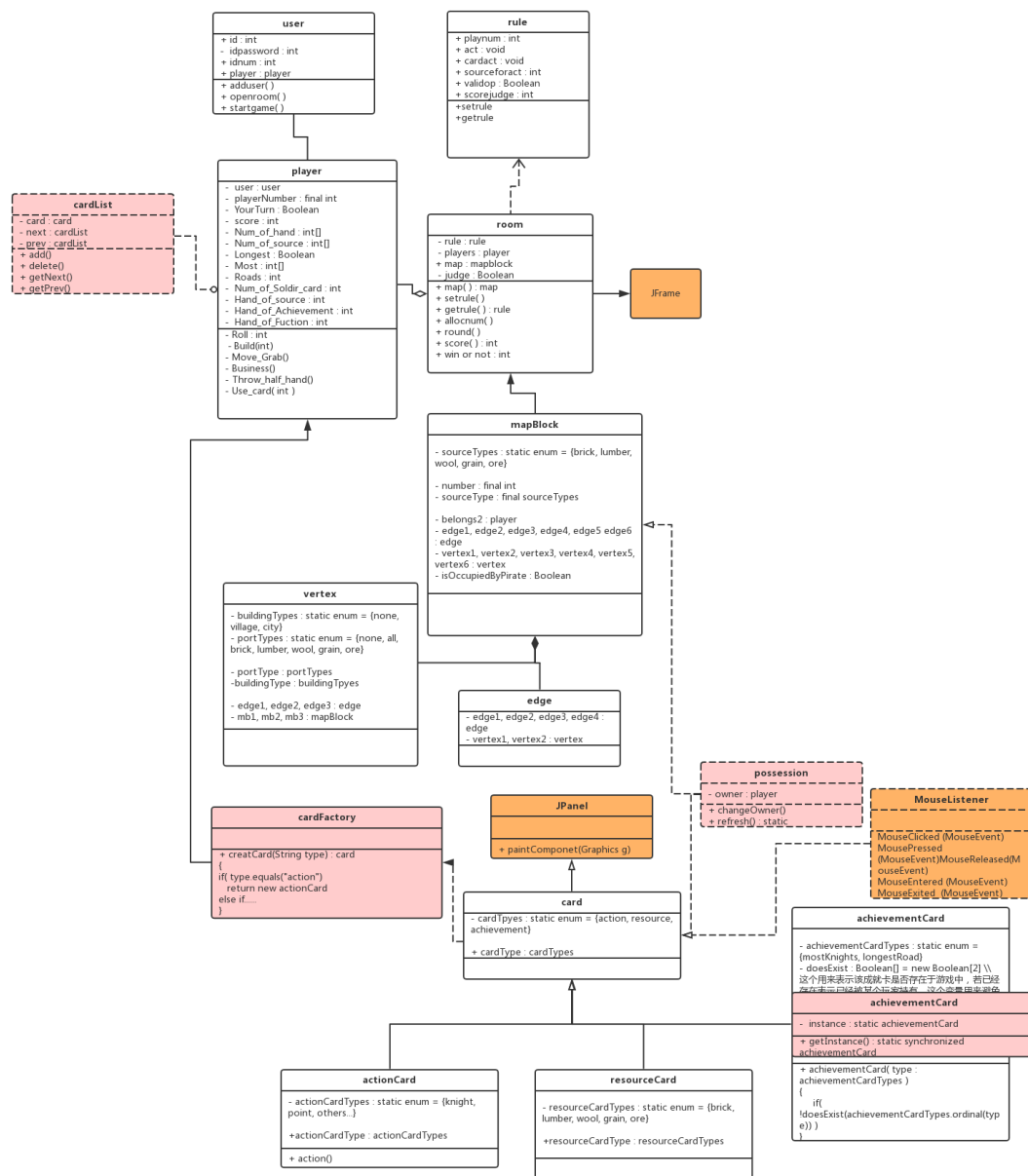
谢本壹、潘高培、宁可

2018.1

### 目录

总体 UML 图 .....	2
Patch V0.2（粉色） .....	3
卡牌类 .....	3
成就卡 .....	4
Patch V0.3（橙色） .....	4
GUI 和鼠标操作 .....	4

## 总体 UML 图



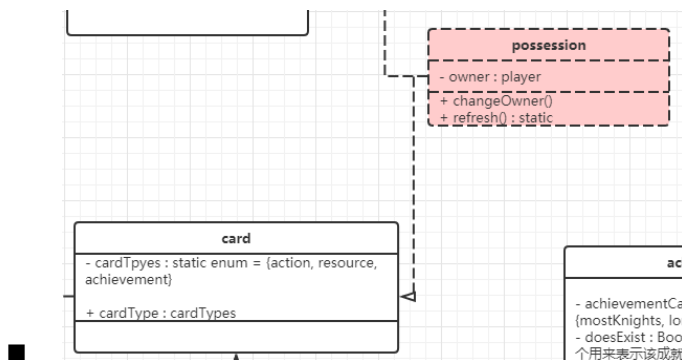
## Patch V0.2（粉色）

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。

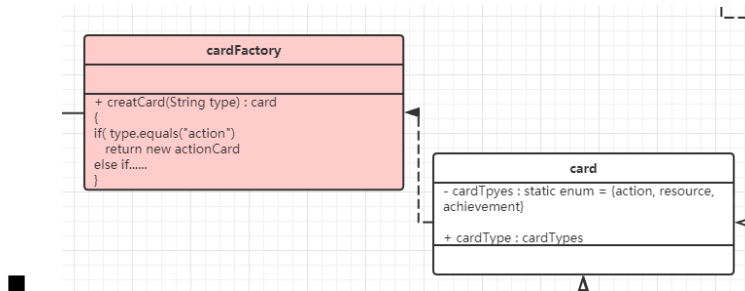
具体的重构：

### 卡牌类

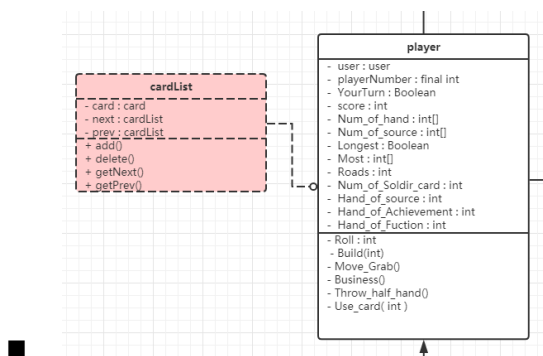
- 新增 possession 接口。方法中的 refresh 用于刷新持有者（player 对象）中的 cardList。所以手牌是通过计算出来的，可以一定程度上避免外挂直接修改手牌。地图块的 possession 同理。这里用到了类似于**观察者模式**的开发方法，区别是，这里并不用维护一个观察者表，因为玩家数固定了，根据 refresh 的参数里的玩家不同。比如 player1 夺取 player2 的卡牌，这是 refresh (player1, player2) 即可；又比如，强盗事件发生后 refresh (all) 即可。
  - 在 refresh () 中不仅要完成玩家手牌的重新计算，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。

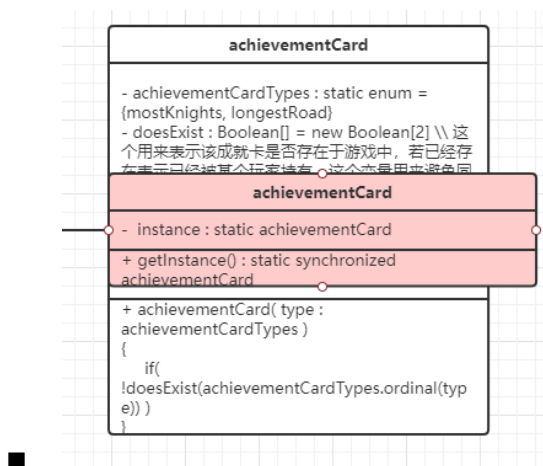


- 新增 cardList。相当于一个链表。



## 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。

