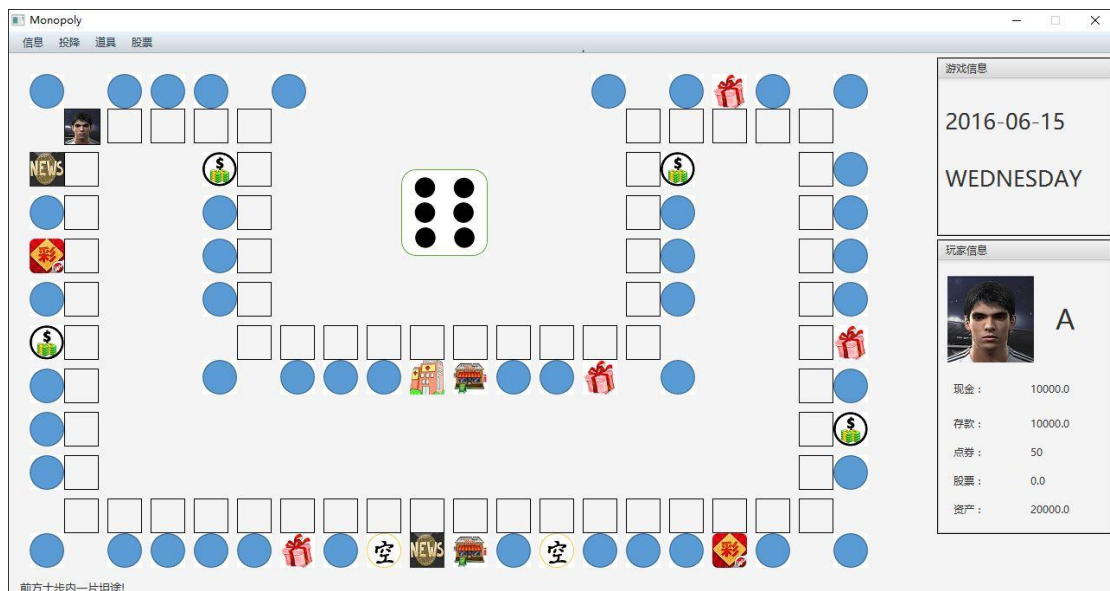


1. 图形界面设计与相关截图

- 所有的图形界面均采用 JavaFX 实现
- 程序入口在 application 包下 Monopoly.java
- 首先是**选择玩家数**



- 点击确认进入主界面



- 主界面的 UI 设计包含在 Map.fxml 中，其中定义了顶部菜单栏、中间地图、右侧信息栏与底部状态栏。主界面的部分样式在 Map.css 中定义。
- 地图实际包含了两圈，正方形框体为自定义的 Point（定义在 Point.fxml 与 Point.java 中），玩家点击骰子后其头像将在 Point 内移动，放置的路障也会显示在 Point 中。
- 另一圈是 Point 的标志(Mark)，初始的土地设置为蓝色的圆圈，新闻、银行、医院、商店、空地、彩票、送点券（道具）的点分别有图标标志。

菜单栏



- 点击信息可以选择查看玩家信息与查询具体位置信息，实现效果如下

玩家信息							
姓名	现金	存款	点券	房产	股票	资产	
A	10000.0	10000.0	50	0	0.0	20000.0	
B	10000.0	10000.0	50	0	0.0	20000.0	

查看位置信息

?

您想查看前后哪一步的具体信息(后方用负数表示)

1

确定取消

信息

i

名称：邯郸路22号
拥有者：无主
等级：0 价格：500

确定

查看位置信息

?

您想查看前后哪一步的具体信息(后方用负数表示)

-1

确定取消

信息

i

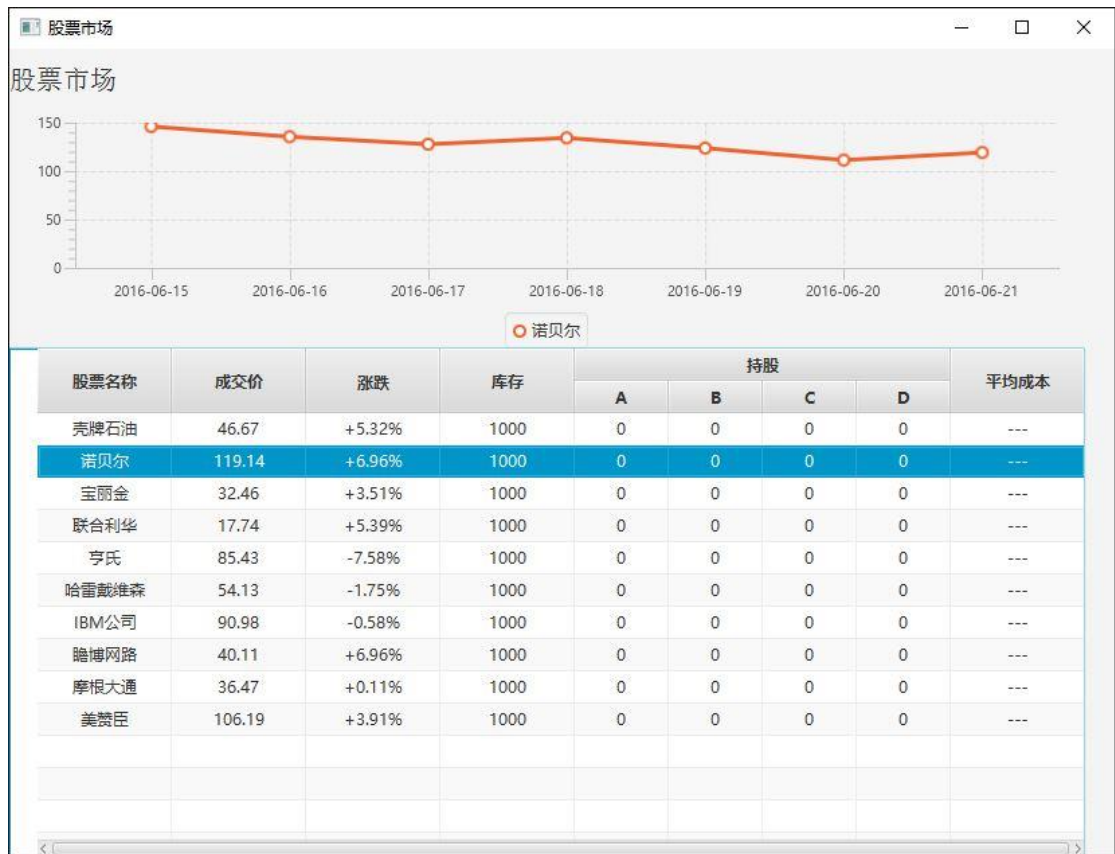
名称：大新闻(杭州路分社)

确定

- 点击投降系统会将当前玩家设置为投降，若只剩下一个玩家则会结束游戏
 - 点击道具后可以使用道具、丢弃道具、查看当前拥有道具与道具信息
- 界面以路障的实现为例：

The chart displays the daily closing price of Shell Petroleum (壳牌石油) over a seven-day period. The y-axis represents the price, ranging from 0 to 50. The x-axis shows dates from 2016-06-15 to 2016-06-21. The price starts at approximately 43.5 on June 15, rises to 45.5 on June 16, dips to 44 on June 17, peaks at 47 on June 18 and 19, falls to 44 on June 20, and ends at 46 on June 21.

Date	Price
2016-06-15	43.5
2016-06-16	45.5
2016-06-17	44.0
2016-06-18	47.0
2016-06-19	47.0
2016-06-20	44.0
2016-06-21	46.0



宝丽金

宝丽金

请选择操作,退出请选择“取消”

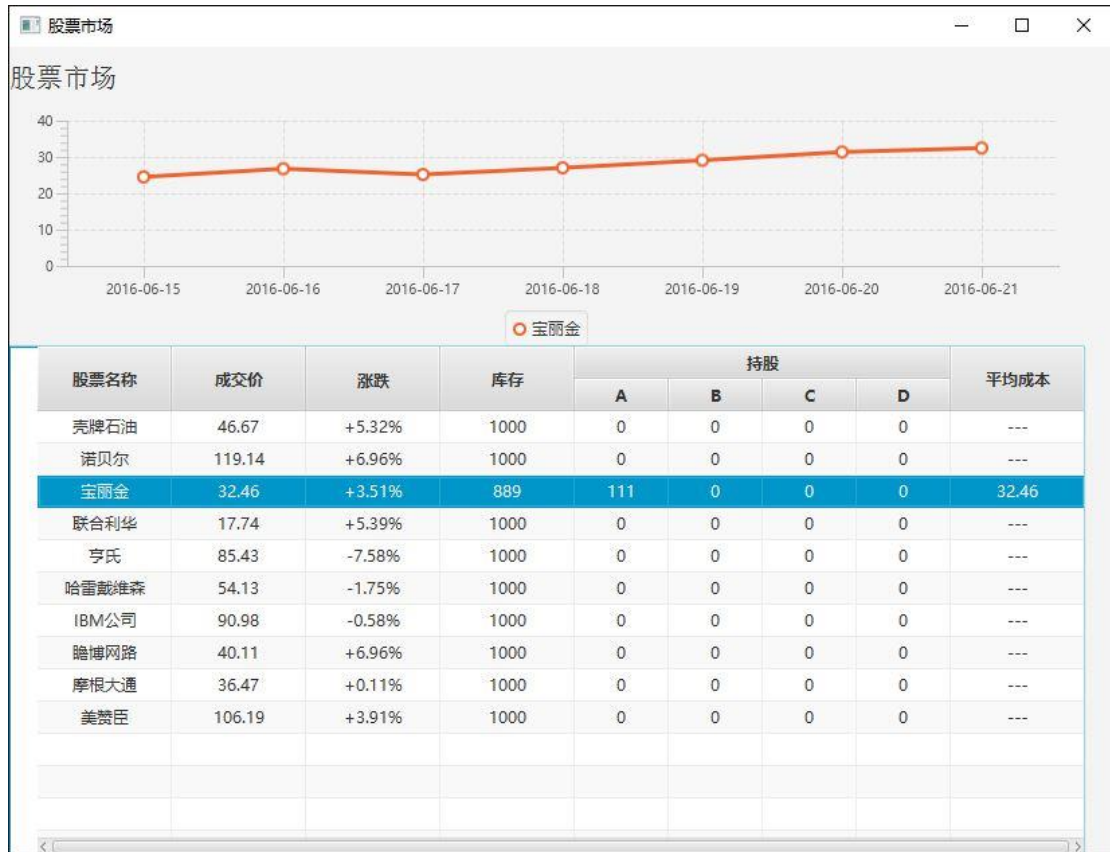
买 卖 取消

宝丽金

您的现金为8190.0元
您的存款为10000.0元
今日剩余库存为1000股
系统计算出当前您可以购买的股票数目范围为(0-560),请输入需要购买的股数

111

确定 取消



其他地点触发时间界面如下

银行

欢迎光临人民银行(国权路支行)

请选择操作,退出请选择“取消”

存款 取款 取消

欢迎光临人民银行(国权路支行)

您的现金为10000.0
请输入您想要存入的金额数(0-10000.0),金额将保留两位小数

122

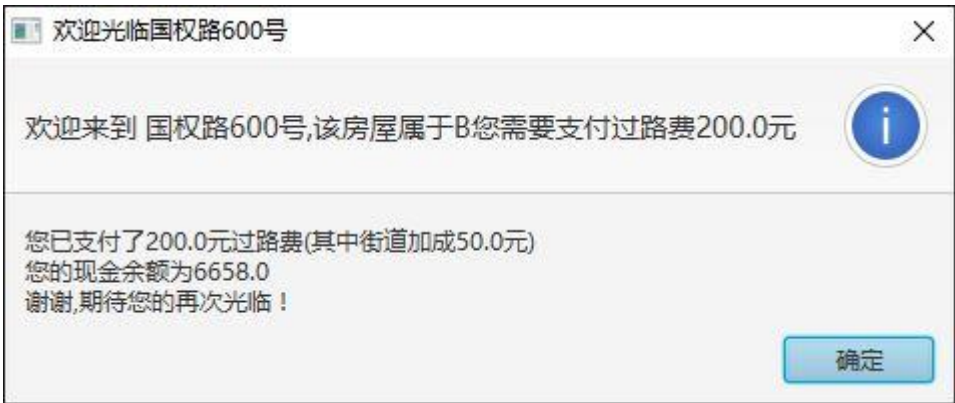
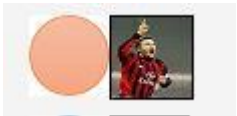
确定 取消

存款成功

您的存款从10000.0元增加至10122.0元
您的现金从10000.0元减少至9878.0元

确定

土地：购买、购买后、过路费、升级

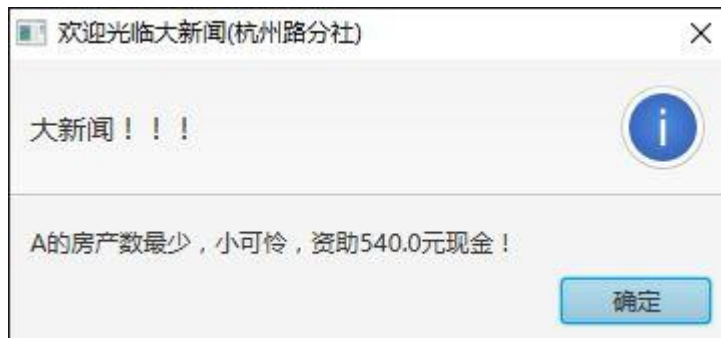


送点券，送道具：





新闻:

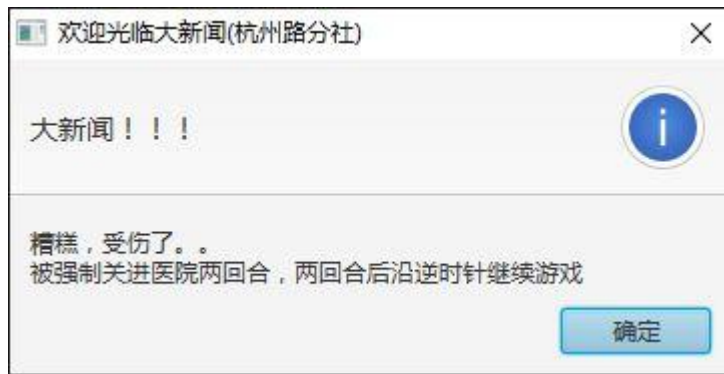


商店:



医院与受伤:

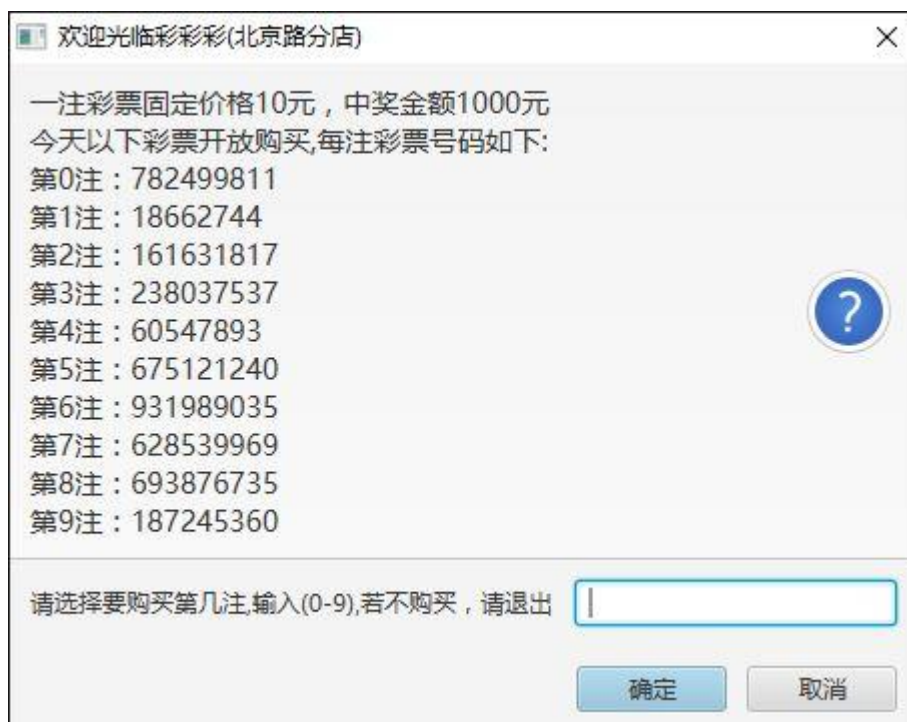




空地：



彩票店：

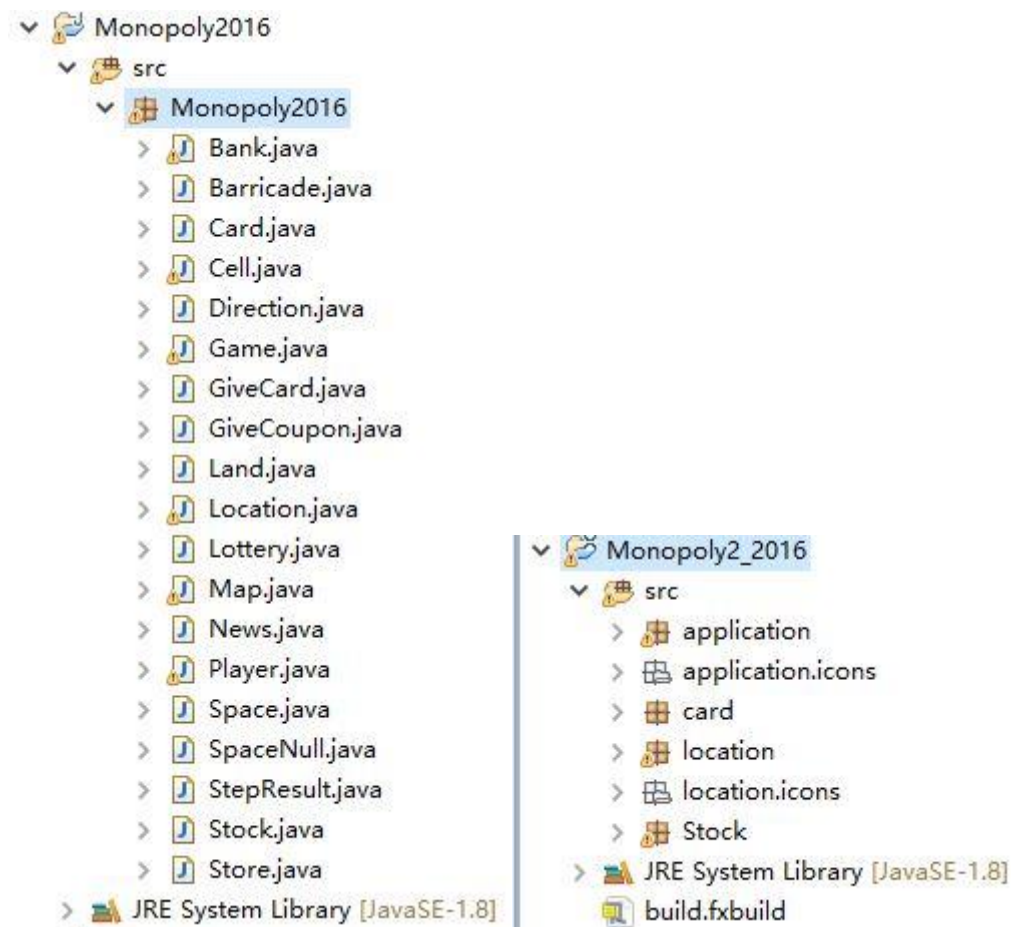




月底发利息



2. 代码改动与新增，之前设计的不足



A. 相比 PJ1，这次将各个类分到了不同的包下，application 下包括了主界面与地图的相关

文件，card 包中为卡片道具的实现，location 为地图中各个点与玩家的实现，Stock 为彩票的实现。这样分类方便知道各个类的实现的内容大致是什么，也方便查找

- B. 因为现在是图形界面，所以更换了所有“System.out.println”。Card 包中所有有关道具的实现、location 中除了 Player 类外的实现，业务逻辑都没有改变，仅仅改变了信息的提示方式
- C. 删除了 Map 类，现在 Map 的实现通过 application 包下 Map.fxml、Map.css 与 Controller 实现。这种实现方式将 UI 设计与事件响应分离。通过修改 Map.fxml、Map.css 可以很容易地对地图中点的排列，样式等进行更改，设计多个 fxml 文件就相当于设计多个 Map。在 Monopoly 主程序入口中，只需要载入不同的 fxml 文件，最后通过一个 Controller 类来管理 Map。
- D. 在股票 Stock 的实现中，新增了很多。其一，图形界面设计同样采用 fxml 与 Controller 结合的方式。其二，拓展了 Stock 类的属性以实现库存、走势图、持股和平均成本等。
- E. 之前的设计是每一个位置是一个 Cell，现在将 Cell 拓展为 Point，Point 是自行拓展的 VBox 类。拓展的好处是在 fxml 中，可以直接使用自定义的这种控件来布局。这就简化了原来需要将 Label 与 Cell 一一对应的实现。
- F. 之前设计的是将每一个 Cell 都与 Map 一一对应，这样设计其实是不需要的。程序初始化后，只有一个 Map 也就不需要这样的对应关系。而之前用到这个对应关系的地方实际上是用到了 Cell 的集合，因此在这次的更改中，我删除 Map 后，将 Point 的集合作为参数来代替之前 Map 作为参数的实现，更改发生在 Card 的使用与 Player 的 step、moveTo 方法。
- G. 之前很多循环写的很复杂，这次一些实现刻意使用了 Java8 的 lamda 表达式简化表达。

3. 此次的问题

- A. 我的设计是每一个 Point 都如同一个实际位置，这个位置可能有多个 Location，如可能同时有商店、医院和新闻社（比照现实的大楼）。但实际上 PJ 的内容不需要这样，每一个点只能是商店或者是其他，所以设计的 Point 类显得有些累赘。
- B. PJ 的图形要求将每一个点都还有一个单独的标签来显示当前的位置信息。本来可以将 Point 的类中加入两个标签，直接实现这一对应关系，但实际写的时候，发现 VBox 的旋转很麻烦，因此一个 Point 有两个标签界面会很难看。因此最后只能重新加上这一对应关系。
- C. application 包下的 Monopoly 文件有一个 static 变量，在 Controller 中用到。这里用 static 是因为在 JavaFX 中，从启动类往界面控制类传参很麻烦，而这里只需要用到一个变量，所以选择了 static 这种最简单的一种实现，不太符合 OOP 思想
- D. 此次面临的最大问题的使用 JavaFX 构建 GUI 时，fxml 与 Controller 之间的常会出现奇怪的问题，最终都还是解决了。JavaFX 这种新的 GUI 构建方式，我认为更符合 OOP 的思想，也更有助于 RIA 的实现。