算法

本程序的架构借鉴了Spring Boot的设计思想，将块的移动、块相关数据的记录与主界面的相关实现方法分开，便于分段开发，也便于程序的迭代更新。

移动算法：

从配置文件读入一个数组，该数组共53个数，第一个数代表已经走过的步数，然后接下来50个数每5个数一组，分别代表每个块的X轴位置，Y轴位置，宽度，高度，名称（1代表张飞，2代表黄忠，3代表关羽，4代表赵云，5代表马超，6代表曹操，7代表卒），最后两个数代表两个空着的块。

考虑到华容道只有四种不同形状的块（长方形块横竖算两种），所以只需要给每种块写一个移动的算法即可。

我们用“层层设防”布局来解释该算法，如下图。

|  |  |  |
| --- | --- | --- |
| R1 | R2 | R3 |
| R4 | R5 | R6 |
| R7 | R8 | R9 |
| 空 | R10 | 空 |

设置一个400\*500的pane，这样一共就只有100\*100，100\*200，200\*100，200\*200共计四种大小的块。但是在具体算法中，我们还是要以100\*100的块为最小单位，并对这个pane划分并编号（如下图）。

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 |

每个块的左上角的坐标代表该块的坐标，如1号块的坐标是（0，0）。这样我们就可以根据坐标来算出某一块的编号。用one，two记录两个空着的块的编号。

然后对鼠标点击的位置进行判断，将每一个不同大小的块分成四个区域（如下图）。

|  |  |
| --- | --- |
| 1 | 2 |
| 3 | 4 |

如果点击的是1，返回1，以此类推。

然后对每一种大小的块，写一个函数来判断是否能满足用户的要求，函数传入该块左上角那个100\*100块的编号，两个空块的编号和鼠标位置函数的返回值。然后根据块的相互位置来判断是否可以完成移动，如果可以返回一个数字（可以向上返回1，可以向右返回2，可以向下返回3，可以向左返回4），如果不可以则返回0。

最后根据返回值和该块与边界的相对位置来移动块，每次移动100单位长度。