2048游戏项目报告

**一、需求分析**

在愈发紧凑的学习生活中，适当的放松是快速恢复精力并继续投入下一阶段学习的方法之一。2048数字游戏作为操作简单又锻炼智力的益智休闲类游戏，无疑是消磨闲暇时光的好选择。因此打算开发一款简洁轻便的2048游戏，以满足人们对休闲娱乐的需求。

**二、技术框架**

Ubuntu虚拟机linux操作系统

Java语言，

Eclipse编译器，

jdk编译环境。

游戏UI运用Java图形界面编程（AWT）。

**三、主要功能**

1.游戏的后台通过监听键盘方向键来移动数字方块；

2.利用随机数思想产生新的方块；

3.记录并更新分数；

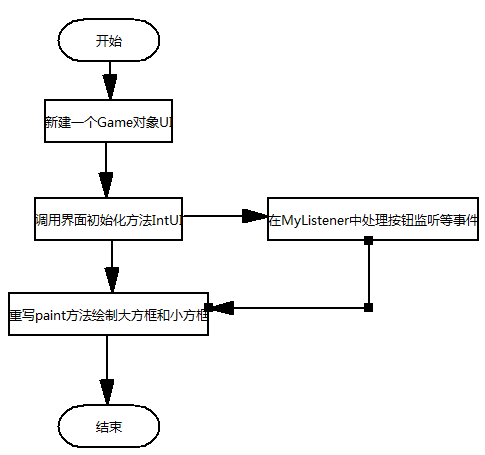
4.游戏操作说明的提示；

5.可选择的退步功能和输掉游戏后的起死回生功能；

**四、具体功能实现**

整个游戏有两个类，分别为游戏的主类Game.class和事件处理类MyListener.class，下面分别对Game.class和MyListener.class进行说明。

1.Game.class的简单程序框图如下：



游戏的主类Game.class是窗体程序JFrame的扩展类，主要负责界面的搭建，完成界面绘图的工作。

首先创建一个窗体，标题为“2048小游戏”，

接下来分别是 “新游戏” 、“帮助” 、“退一步” 的按钮，创建三个新游戏的图片按钮，为达到美观的显示效果，把聚焦，边线等特征设置为false.

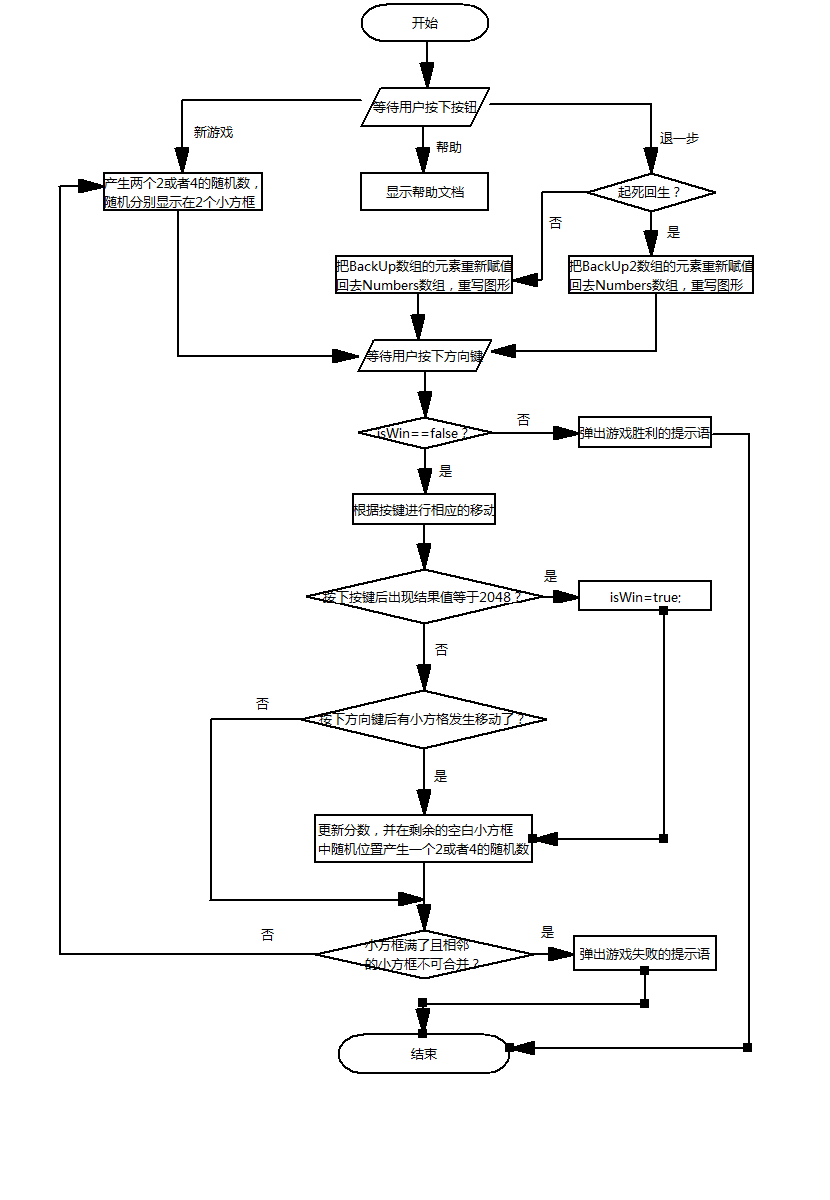
布置好控件后，为了彻底结束进程，使用 System exit 方法退出应用程序，同时由于按钮都是相对窗体固定坐标的，所以不允许用户随意改变窗体大小，最后把界面设置为可见。

然后便是游戏中大方框和数字小方框的绘制，这里用到了paint方法来绘制容器。

先绘制出大矩形框，为了美观，这里绘制圆角矩形框，接下来，通过双重循环，绘制4\*4的小方框，每一个小方框距离边框及相邻方框的大小相等，能达到相对美观的效果。

由于2048游戏里面可能显示的数值有2、4、8、16等等不同的数值，为了便于区分，我们给显示不同数值的方框绘制不同的颜色。由于一位数字2与两位数字16甚至多位数字128或1024等来说，有些数字的显示会超出方框，影响观感，所以还要对数字的相对位置和大小做一定的调整。

2.MyListener.class是一个键盘监听事件类，该类的简单程序框图如下：



在该类的开始，先声明一些变量，用于接收传递进来的参数。

接下来是按钮的监听，通过e.getSource()取得按钮对应的值，进行相应的处理。

“新游戏” 按钮作为游戏的开始，将会在按下该按钮时初始化数据，

“帮助” 按钮功能简单，按下后弹窗显示游戏规则。

“退一步” 按钮分了两种情况考虑，进入回退按钮的操作后，先判断是否是起死回生类型的回退，如果不是，则把非起死回生的分数备份还原成当前分数，然后还原上一步的状态；如果是，操作和前面是一样的，不同点在于在操作后，把起死回生回退的标志relive重新置为false。在完成这些操作后，重新绘图。

下面是按键监听的解析，按键监听通过相应的键值识别按键，然后运用switch开关语句控制不同按键的事件。

在完成移动及合并操作后，要判断相邻的方块是否还能合并，因为有可能出现移动后，相邻的方块又能进行新的合并，如果不加这个判断的话，当所有的方块全部填满时会直接判断游戏结束。

在完成上述操作后，再判断非0元素的个数。当非零元素个数为0时，即代表所有的小方块中已经有元素存在了，表明16个方块全部都满了。

在完成统计空格个数后，如果按下按键后发生了移动，就完成分数的更新，因为要移动一次后在随机的空的格子（这里指元素值为0的格子）里面产生新的基于2或者4的随机数，

接下来判断这次移动后，有没有出现 isWin = true ，如果出现了，就弹出游戏胜利的标志。

如果没有出现 isWin = true，判断是否是16个格子全满了，同时相邻的方格不能进一步合并了，如果是则把可以进行起死回生操作的标志relive置为true，同时弹出游戏结束的提示语。

最后，重新绘制图形。

**五、运行成功截图**

