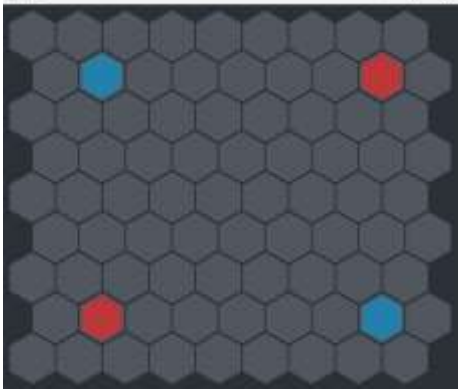
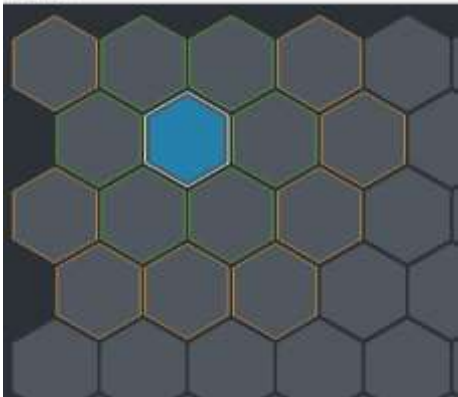

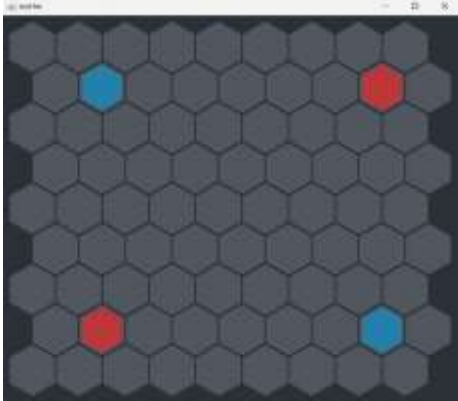
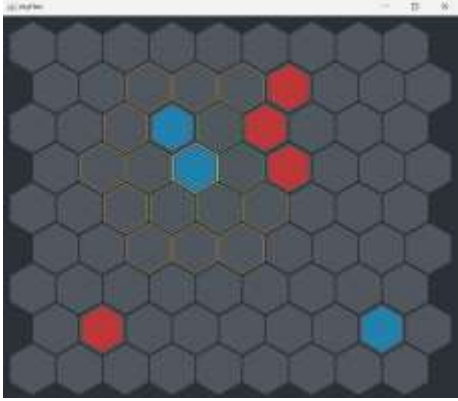
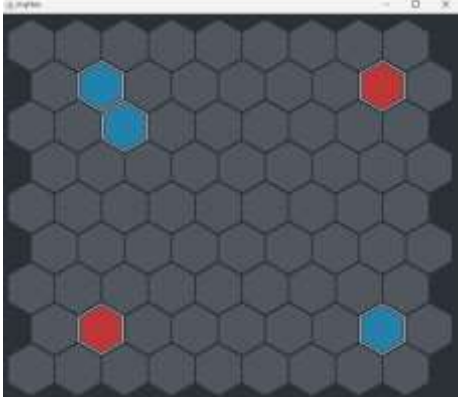


大作业实验报告

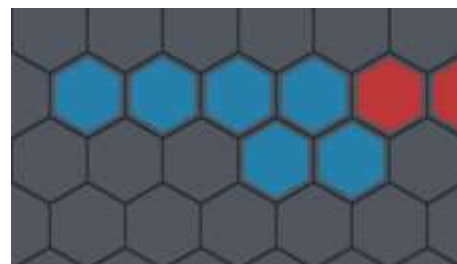
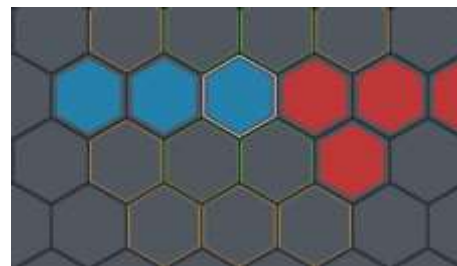
苟芳菲 2021011837 软件 13

一、游戏操作方式

游戏界面	
绘制棋盘	
绘制棋子	
棋子移动提示	
<p>绘制游戏结果：</p> <ol style="list-style-type: none">1. 将未充满的棋盘充满胜方棋子2. 弹窗显示结果3. 关闭弹窗后重新开始	

游戏模型	
初始化	
<p>棋子移动</p> <ol style="list-style-type: none">1. 点击当前棋子后点击合法位置完成移动2. 落子位置不合法则无反应3. 待移动棋子并非当前执棋方则棋盘所有棋子白色标亮	
<p>棋子移动提示：</p> <ol style="list-style-type: none">1. 点击当前棋子周围出现高亮提示2. 移动后高亮消失3. 若之后点击另一个合法棋子，则高亮提示换为该棋子周围4. 点击非法格子无反应（非法格子包括：该格子上无棋子、该格子上的棋子并非当前执棋方）	

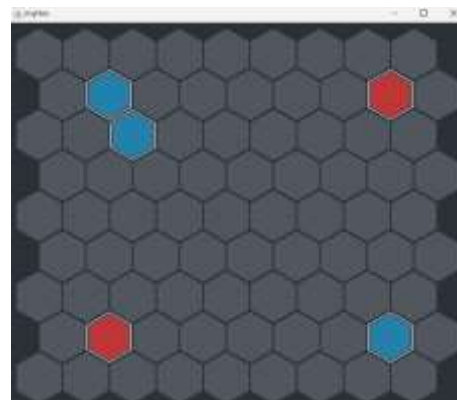
棋子俘获

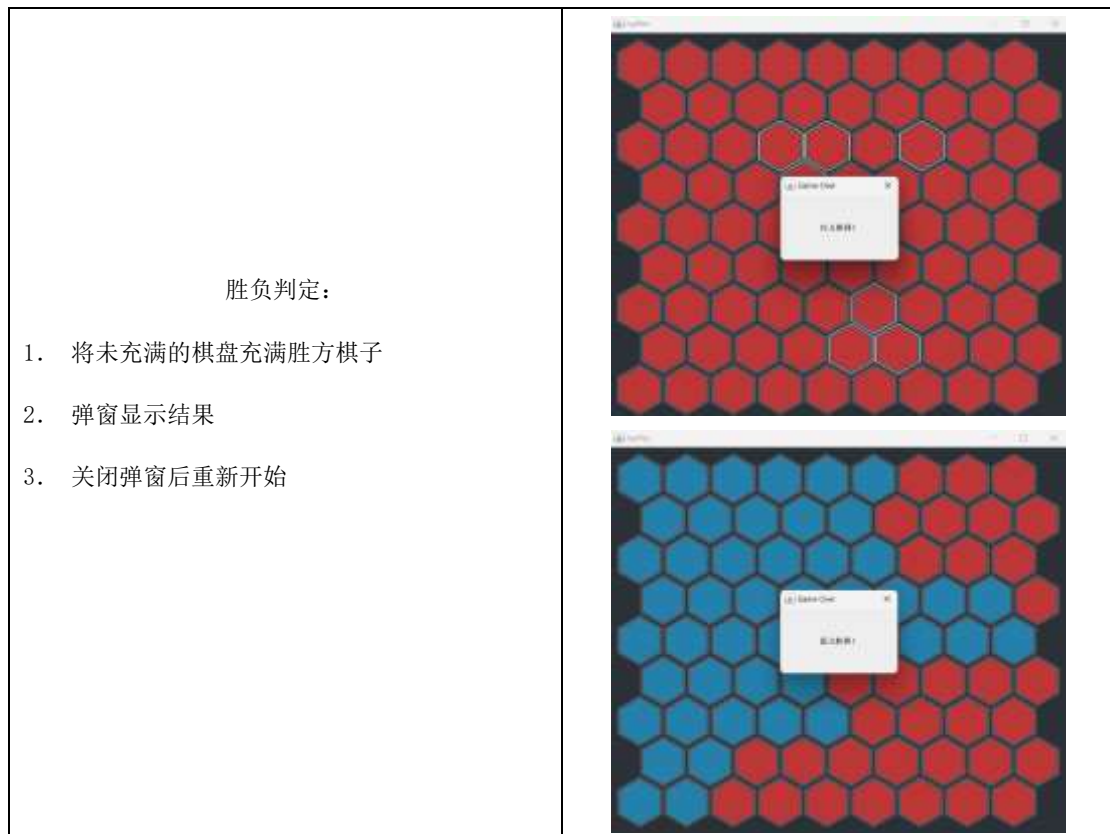


交替落子

1. 待移动棋子并非当前执棋方则棋盘所有棋子

白色标亮





二、项目架构与设计

一开始写的时候是将棋子作为一个类单独描述的，但是在后来继续完善的过程中发现这种方法流畅性一般，所以采用了 hexSet 数组存储每个格子对应的棋子状态，用 hint 数组存储高亮提示，在每次鼠标操作之后对这两个数组进行操作，然后在 paint 函数中调用 update 函数，在 update 中根据棋子和高亮状态调用 drawHexagon 和 drawHint 函数，重新绘制棋子和高亮提示。下图是各个函数的调用关系图：

