**public** **void** mouseClicked(MouseEvent e) {

// **TODO** Auto-generated method stub

}

**public** **void** mouseEntered(MouseEvent e) {

// **TODO** Auto-generated method stub

}

**public** **void** mouseExited(MouseEvent e) {

// **TODO** Auto-generated method stub

}

//鼠标按下mousePressed

**public** **void** mousePressed(MouseEvent e) {

// **TODO** Auto-generated method stub

/\*

\* System.out.println("X:"+e.getX()); System.out.println("Y:"+e.getY());

\*/

**if** (canPlay == **true**) {

x = e.getX();

y = e.getY();

**if** (x >= 10 && x <= 370 && y >= 70 && y <= 430) {

x = (x - 10) / 20;

y = (y - 70) / 20;

**if** (allChess[x][y] == 0) {

// 判断当前要下的是什么颜色的棋子

**if** (isBlack == **true**) {

allChess[x][y] = 1;

isBlack = **false**;

message = "轮到白方";

} **else** {

allChess[x][y] = 2;

isBlack = **true**;

message = "轮到黑方";

}

// 判断这个棋子是否和其他的棋子连成5连，即判断游戏是否结束

**boolean** winFlag = **this**.checkWin();

**if** (winFlag == **true**) {

JOptionPane.*showMessageDialog*(**this**, "游戏结束,"

+ (allChess[x][y] == 1 ? "黑方" : "白方") + "获胜！");

canPlay = **false**;

}

} **else** {

JOptionPane.*showMessageDialog*(**this**, "当前位置已经有棋子，请重新落子！");

}

**this**.repaint();

}

}

**public** **void** mouseReleased(MouseEvent e) {

// **TODO** Auto-generated method stub

}

**private** **boolean** checkWin() {

**boolean** flag = **false**;

// 保存共有相同颜色多少棋子相连

**int** count = 1;

// 判断横向是否有5个棋子相连，特点 纵坐标 是相同， 即allChess[x][y]中y值是相同

**int** color = allChess[x][y];

// 判断横向

count = **this**.checkCount(1, 0, color);

**if** (count >= 5) {

flag = **true**;

} **else** {

// 判断纵向

count = **this**.checkCount(0, 1, color);

**if** (count >= 5) {

flag = **true**;

} **else** {

// 判断右上、左下

count = **this**.checkCount(1, -1, color);

**if** (count >= 5) {

flag = **true**;

} **else** {

// 判断右下、左上

count = **this**.checkCount(1, 1, color);

**if** (count >= 5) {

flag = **true**;

}

}

}

}

**return** flag;

}

// 判断棋子连接的数量

**private** **int** checkCount(**int** xChange, **int** yChange, **int** color) {

**int** count = 1;

**int** tempX = xChange;

**int** tempY = yChange;

**while** (x + xChange >= 0 && x + xChange <= 18 && y + yChange >= 0

&& y + yChange <= 18

&& color == allChess[x + xChange][y + yChange]) {

count++;

**if** (xChange != 0)

xChange++;

**if** (yChange != 0) {

**if** (yChange > 0)

yChange++;

**else** {

yChange--;

}

}

}

xChange = tempX;

yChange = tempY;

**while** (x - xChange >= 0 && x - xChange <= 18 && y - yChange >= 0

&& y - yChange <= 18

&& color == allChess[x - xChange][y - yChange]) {

count++;

**if** (xChange != 0)

xChange++;

**if** (yChange != 0) {

**if** (yChange > 0)

yChange++;

**else** {

yChange--;

}

}

}

**return** count;

}