// 保存最多拥有多少时间(秒)

**int** maxTime = 0;

// 做倒计时的线程类

Thread t = **new** Thread(**this**);

// 保存黑方与白方的剩余时间

**int** blackTime = 0;

**int** whiteTime = 0;

// 点击 开始游戏 按钮 ，根据鼠标的位置来判断是否开始游戏，JOptionPane.showConfirmDialog弹出的对话框的选项是“是”和“否

**if** (e.getX() >= 400 && e.getX() <= 470 && e.getY() >= 320

&& e.getY() <= 350) {

**int** result = JOptionPane.*showConfirmDialog*(**this**, "是否重新开始游戏?");

**if** (result == 0) {

// 现在重新开始游戏

// 重新开始所要做的操作: 1)把棋盘清空,allChess这个数组中全部数据归0.

// 2) 将 游戏信息: 的显示改回到开始位置

// 3) 将下一步下棋的改为黑方

**for** (**int** i = 0; i < 19; i++) {

**for** (**int** j = 0; j < 19; j++) {

allChess[i][j] = 0;

}

}

// 另一种方式 allChess = new int[19][19];

message = "黑方先行";

isBlack = **true**;

blackTime = maxTime;

whiteTime = maxTime;

**if** (maxTime > 0) {

blackMessage = maxTime / 3600 + ":"

+ (maxTime / 60 - maxTime / 3600 \* 60) + ":"

+ (maxTime - maxTime / 60 \* 60);

whiteMessage = maxTime / 3600 + ":"

+ (maxTime / 60 - maxTime / 3600 \* 60) + ":"

+ (maxTime - maxTime / 60 \* 60);

t.~~resume~~();

} **else** {

blackMessage = "无限制";

whiteMessage = "无限制";

}

**this**.canPlay = **true**;

**this**.repaint();

}

}

// 点击 游戏设置 按钮

**if** (e.getX() >= 120 && e.getX() <= 200 && e.getY() >= 450

&& e.getY() <= 485) {

String input = JOptionPane

.*showInputDialog*("请输入游戏的最大时间(单位:分钟),如果输入0,表示没有时间限制:");

**try** {

maxTime = Integer.*parseInt*(input) \* 60;

**if** (maxTime < 0) {

JOptionPane.*showMessageDialog*(**this**, "请输入正确信息,不允许输入负数!");

}

**if** (maxTime == 0) {

**int** result = JOptionPane.*showConfirmDialog*(**this**,"设置完成,是否重新开始游戏?");

**if** (result == 0) {

**for** (**int** i = 0; i < 19; i++) {

**for** (**int** j = 0; j < 19; j++) {

allChess[i][j] = 0;

}

}

// 另一种方式 allChess = new int[19][19];

message = "黑方先行";

isBlack = **true**;

blackTime = maxTime;

whiteTime = maxTime;

blackMessage = "无限制";

whiteMessage = "无限制";

**this**.canPlay = **true**;

**this**.repaint();

}

}

**if** (maxTime > 0) {

**int** result = JOptionPane.*showConfirmDialog*(**this**,

"设置完成,是否重新开始游戏?");

**if** (result == 0) {

**for** (**int** i = 0; i < 19; i++) {

**for** (**int** j = 0; j < 19; j++) {

allChess[i][j] = 0;

}

}

// 另一种方式 allChess = new int[19][19];

message = "黑方先行";

isBlack = **true**;

blackTime = maxTime;

whiteTime = maxTime;

//计算黑白棋的时分秒

blackMessage = maxTime / 3600 + ":"

//小时

+ (maxTime / 60 - maxTime / 3600 \* 60) + ":"

//总计分钟数减去小时整数部分

+ (maxTime - maxTime / 60 \* 60);

//减去分钟数

whiteMessage = maxTime / 3600 + ":"

+ (maxTime / 60 - maxTime / 3600 \* 60) + ":"

+ (maxTime - maxTime / 60 \* 60);

t.~~resume~~();

**this**.canPlay = **true**;

**this**.repaint();

}

}

} **catch** (NumberFormatException e1) {

// **TODO** Auto-generated catch block

JOptionPane.*showMessageDialog*(**this**, "请正确输入信息!");

}

}

// 点击 游戏说明 按钮

**if** (e.getX() >= 225 && e.getX() <= 300 && e.getY() >= 455 && e.getY() <= 480) {

JOptionPane.*showMessageDialog*(**this**,"这个一个五子棋游戏程序，黑白双方轮流下棋，当某一方连到五子时，游戏结束。");

}

// 点击 关于 按钮

**if** (e.getX() >= 330 && e.getX() <= 400 && e.getY() >= 455

&& e.getY() <= 480) {

JOptionPane.*showMessageDialog*(**this**,"本程序由Java课程设计第5组，组长金美辰，组员刘亚平、郭海桃完成制作");

}

// 点击 退出 按钮

**if** (e.getX() >= 400 && e.getX() <= 470 && e.getY() >= 370

&& e.getY() <= 400) {

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

System.*exit*(0);

}

}

//按钮部分

**public** **void** run() {

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

// 判断是否有时间限制

**if** (maxTime > 0) {

**while** (**true**) {

**if** (isBlack) {

blackTime--;

**if** (blackTime == 0) {

JOptionPane.*showMessageDialog*(**this**, "黑方超时,游戏结束!");

}

} **else** {

whiteTime--;

**if** (whiteTime == 0) {

JOptionPane.*showMessageDialog*(**this**, "白方超时,游戏结束!");

}

}

blackMessage = blackTime / 3600 + ":"

+ (blackTime / 60 - blackTime / 3600 \* 60) + ":"

+ (blackTime - blackTime / 60 \* 60);

whiteMessage = whiteTime / 3600 + ":"

+ (whiteTime / 60 - whiteTime / 3600 \* 60) + ":"

+ (whiteTime - whiteTime / 60 \* 60);

**this**.repaint();

**try** {

Thread.*sleep*(1000);//睡眠1000毫秒

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.println(blackTime + " -- " + whiteTime);

}

}

}