**关于俄罗斯方块的简单说明**

--------作者:姜楠,张卓

**(1).** **设计意图**

俄罗斯方块，作为一款经典小游戏，它上手简单，通过方块的移动、旋转和摆放，使之排列成完整的一行或多行并且消除得分，老少咸宜，早已家喻户晓。它看似简单却变化无穷，每一个简单的变换都蕴含着几何与动态之美，令人上瘾。

怀着对经典的崇敬之心，本团队尝试着用手头上的工具，实现了这款经典游戏。除了游戏中方块的移动、旋转、摆放与消除并得分、显示分数的功能外，还添加了增减游戏难度、显示时间等功能，并提供了多种界面风格。全程使用Java语言实现，界面十分清爽。

**(2).** **流程图**

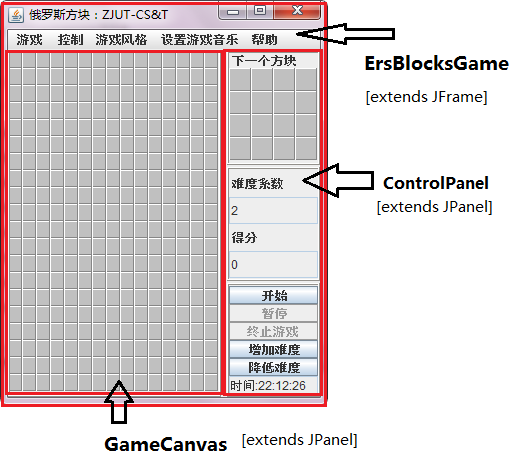


图1.游戏的总体结构。

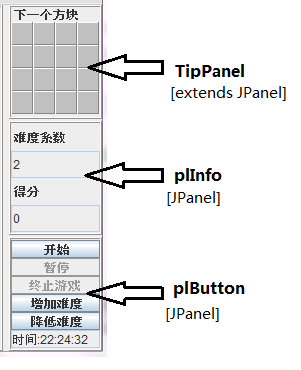
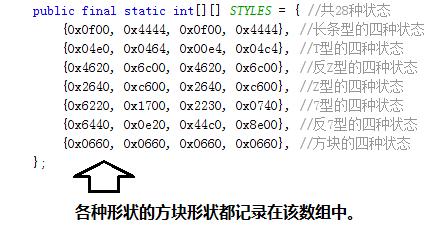


图2 ControlPanel的主要组成

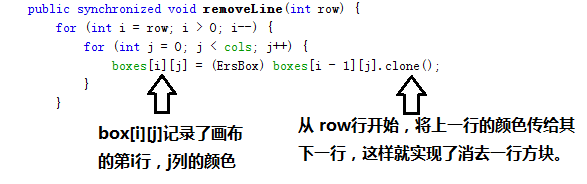
**问题一：如何“产生”，“旋转”，“移动”各种“俄罗斯方块”？**



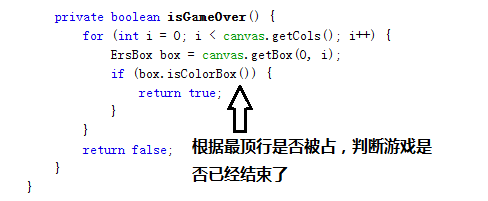
**问题二：当一行满后，如何才能消去这一行的所有方块？**



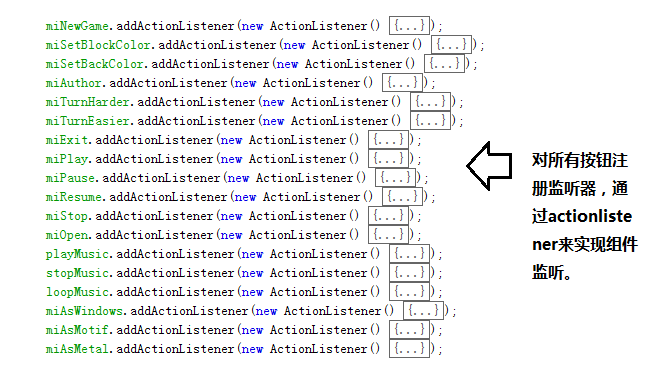
**图3.1**



**问题三：如何判断游戏结束？**



**问题四：如何实现控制按钮对游戏的控制？**



**问题五：如何实现播放音乐的功能？**

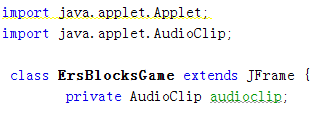


图 5.1 首先将AudioClip设置为ErsBlockGame的私有成员，主要是通过这个接口来实现播放功能。

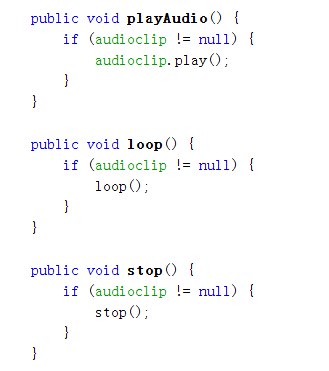


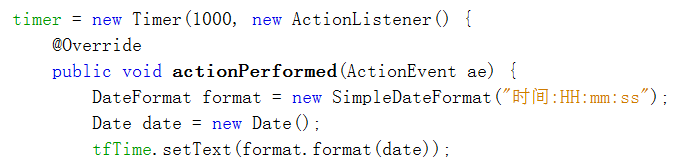
图5.2 AudioClip有三个public的函数play(),loop(),stop();这里通过再次申明，转化为主程序的内部成员,然后通过dialog选择音乐文件，将音乐的地址传给AudioCilp.代码见下图.



**问题六：如何显示系统时间？**



图7.1 右下角显示的系统时间是一个名叫tfTime的JTexeField。再设计一个计时器，每隔1000ms(即1s)更新上面显示的时间，见下图。



**(3).如何使用软件**



图1.1 打开软件，首先看到这个界面



图1.2 点击菜单栏的“游戏风格”项，可以设置游戏界面风格，本游戏一共提供了三种风格：

Windows，Motif，Metal。下面给出三种风格的截图：

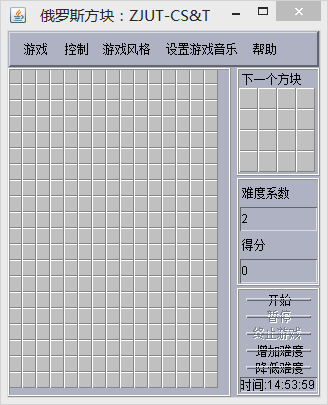


图1.3 三张截图分别为Windows，，Motif，Metal风格的界面。



图1.4 如图，点击“设置游戏音乐”一栏，可以在玩游戏的同时聆听动感的音乐（本功能只支持wav，MIDI格式，不支持mp3，mp4，flac，ape等格式）。

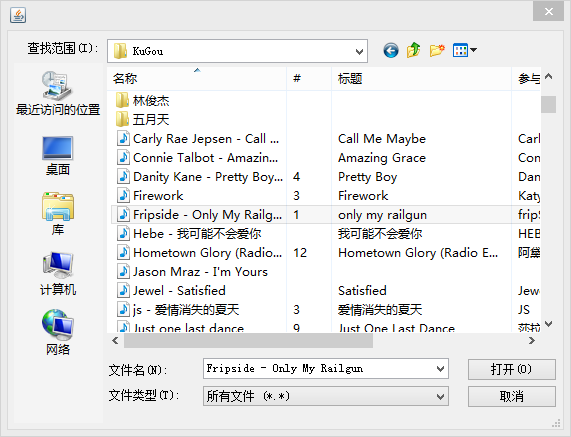


图1.5 点击“设置游戏音乐”一栏中的“打开”项，弹出上面的“音乐选择对话框”，选择自己音乐喜欢的，就可以播放了。“设置游戏音乐”一栏中还有“停止”，“循环”项，可以停止播放音乐，循环播放音乐，因截图原因无法显示。



图1.6 点击“帮助”一栏的“关于”项，弹出有关作者的信息。



图1.7 这是有关作者的信息的对话框



图1.8 点击“游戏”一栏的“新游戏”项，可以重新开始游戏。点击“设置方块颜色”项，弹出颜色选择界面，设置俄罗斯方块的颜色。点击“设置背景颜色”， 弹出颜色选择界面，设置背景的颜色。点击“升高游戏难度”，加快方块下落速度。点击“降低游戏难度”，减慢方块下落速度。点击“退出”，游戏结束，自动退出。

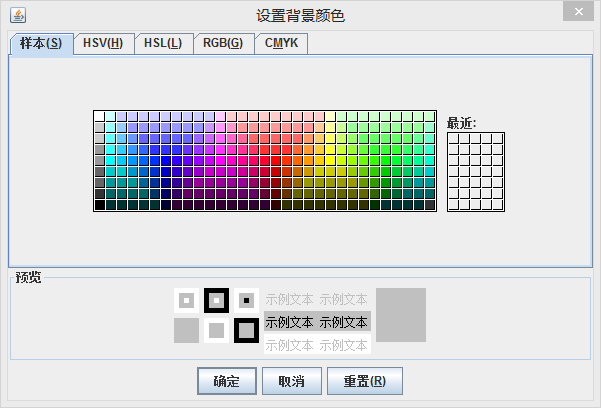


图1.9 上面两幅截图分别为“设置方块颜色”项和“设置背景颜色”的颜色选择界面



图1.10 点击“控制” 菜单栏下的“开始”，开始俄罗斯方块的游戏。点击“暂停”，暂停游戏。

点击“恢复”，继续中止的游戏。点击“终止游戏”，结束游戏。

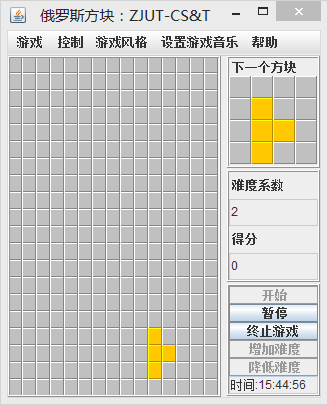


图1.11 点击“开始”后显示的俄罗斯方块，右上角的方块为下一个将要出现的方块。下面是“难度系数”一栏和“得分”一栏，右下角显示“当前系统的时间”，提醒玩家注意时间，不要玩得过度。



图1.12 上面截图的红色方框区域是本游戏的“控制面板”，分别设置了“开始”，“暂停”，“终止游戏”，“增加难度”，“降低难度”。功能与上面介绍的“控制”菜单栏中的“开始”，“暂停”，“终止游戏”，“增加难度”，“降低难度”菜单项相同。



图 1.13 如上图显示，当点击“暂停”，游戏暂停后，该按钮会显示“继续”，若再次点击，则游戏会继续进行。

**(4)** **数据结构和算法的实现**

该程序由5个类组成，分别是：

1. ErsBlocksGame;
2. GameCanvas;
3. ControlPanel;
4. ErsBlock;
5. ErsBox;

|  |  |  |
| --- | --- | --- |
| ErsBlocksGame | GameCanvas | ErsBlock |
| + PER\_LINE\_SCORE : int  + PER\_LEVEL\_SCORE : int  + MAX\_LEVEL : int  + DEFAULT\_LEVEL : int | - rows : int  - cols : int  - score : int  - scoreForLevelUpdate : int  - boxWidth : int  - boxHeight : int  - boxes : ErsBox[][]  - backColor : Color  - frontColor : Color | + BOXES\_ROWS : int  + BOXES\_COLS ： int  + LEVEL\_FLATNESS\_GENE ：int  + BETWEEN\_LEVELS\_DEGRESS\_TIME：int  + BLOCK\_KIND\_NUMBER : int  + BLOCK\_STATUS\_NUMBER: int  + STYLES : int[][]  - canvas : GameCanvas  - boxes : ErsBox[][]  - style : int  - y : int  - x : int  - level : int  - pausing : int  - moving : int |
| +Main()  +ErsBlocksGame()  +getCanvas()  +getCurBlock()  +setLevel()  +getLevel() : int  +get Score() : int  + reset()  + isPlaying() : boolean  + getCurBlock() : ErsBlock  + getCanvas() : GameCanvas  + playGame()  + pauseGame()  + resumeGame()  + stopGame()  + getScoreForLevelUpdate(): int  + levelUpdate() : boolean  - play()  - reportGameOver()  - creatMenu()  - setWindowStyle() | + GameCanvas()  + setBackgroundColor()  + getBackgroundColor()  + getBlockColor() : Color  + getRows() : int  + getCols() : int  + getScore() : int  + getScoreForLevelUpdate(): int  + resetScoreForLevelUpdate()  + getBox() : ErsBox  + paintComponent()  + fanning()  + removeLine()  + reset() | + ErsBlock()  + run()  + moveLeft()  + moveRight()  + moveDown()  + turnNext()  + pauseMove()  + resumeMove()  + stopMove()  - erase()  - display()  - isMoveAble() : boolean  - moveTo() : boolean  - isTurnAble() : boolean  - turnTo() : boolean |

|  |  |
| --- | --- |
| ControlPanel | ErsBox |
|  | - isColor  - size : Dimension |
| + ControlPanel()  + setTipStyle()  + getLevel()  + setLevel()  + setPlayButtonEnable()  + setPauseButtonLabel()  + reset()  + fanning() | + ErsBox()  + isColorBox() : boolean  + setColor()  + getSize() : Dimension  + setSize()  + clone() : Object |

**ErsBlockGame:游戏主类，继承自JFrame类，负责游戏的全局控制**

public ErsBlockGame(String title)主游戏类的构造方法

public void reset() 游戏复位

public boolean isPlaying()判断游戏是否还在进行

public ErsBlock getCurBlock()得到当前活动的块

public GameCanvas getCanvas()得到当前画布

public void playGame()开始游戏

public void pauseGame()游戏暂停

public void resumeGame()让暂停中的游戏继续

public void stopGame()用户停止游戏

public int getLevel()得到游戏者设置的难度

public void setLevel(int level)用户设置游戏难度

public int getScore() 得到游戏积分

public int getScoreForLevelUpdate()得到自上次升级以来的游戏积分，升级以后，此积分清零

public boolean levelUpdate()当积分累积到一定数值时，升一次级

private void play()游戏开始

private void reportGameOver()报告游戏结束了

private void creatMenu()建立并设置窗口菜单

private void setWindowStyle(String plaf)根据字串设置窗口外观

private class Game implements Runnable一轮游戏过程，实现了Runnable接口 一轮游戏是一个大循环，在这个循环中，每隔100毫秒， 检查游戏中的当前块是否已经到底了，如果没有，就继续等待。如果到底了，就看有没有全填满的行， 如果有就删除它，并为游戏者加分，同时随机产生一个 新的当前块人 让它自动落下。 当新产生一个块时，先检查画布最顶上的一行是否已经 被占了，如果是，可以判断Game Over 了。

private class gameOverDialog extends JDialog implements ActionListener定义

GameOver对话框

public static void main(String[] args)程序入口函数

class GameCanvas extends JPanel画布类，内有<行数>\*<列数> 个方格类实例。 继承自JPanel类。 ErsBlock线程类动态改变画布类的方格颜色，画布类通过

public GameCanvas(int rows, int cols, Color backColor, Color frontColor)画布类的构造函数

public void setBackgroundColor(Color backColor)设置游戏背景色彩

public Color getBackgroundColor()取得游戏背景色彩

public void setBlockColor(Color frontColor)设置游戏方块颜色

public Color getBlockColor()取得游戏方块色彩

public int getRows()取得画布中方格的列数

public int getCols()取得画布中方格的行数

public int getScore()取得游戏成绩

public int getScoreForLevelUpdate()取得自上一次升级后的积分

public void resetScoreForLevelUpdate()升级后，将上一次升级以来的积分清零

public ErsBox getBox(int row, int col)得到某一行某一列的方格引用

public void paintComponent(Graphics g)覆盖JComponent类的函数，画组件。

public void fanning()根据窗口大小，自动调节方格的尺寸

public synchronized void removeLine(int row)

当一行被游戏者叠满后，将此行清除，并为游戏者加分

public void reset()重置画布，置积分为零

**class ErsBox implements Cloneable方格类，是组成块的基本元素，用自己的颜色来表示块的外观:**

public ErsBox(boolean isColor)方格类的构造函数

public boolean isColorBox()此方格是不是用前景色表现

public void setColor(boolean isColor)设置方格的颜色

public Dimension getSize()得到此方格的尺寸

public void setSize(Dimension size)设置方格的尺寸

public Object clone()覆盖Object的Object clone(),实现克隆

**class ErsBlock extends Thread块类，继承自线程类（Thread） 由4 × 4个方块（ErsBox）构成一个方块， 控制块的移动·下落·变形等:**

public final static int BOXES\_ROWS = 4;一个块占的行数是4行

public final static int BOXES\_COLS = 4;一个块占的列数是4列

public final static int BOXES\_COLS = 4;让升级变化平滑的因子，避免最后几级之间的速度相差近一倍

public final static int LEVEL\_FLATNESS\_GENE = 3;让升级变化平滑的因子，避免最后几级之间的速度相差近一倍

public final static int BETWEEN\_LEVELS\_DEGRESS\_TIME = 50;相近的两级之间，块每下落一行的时间差别为多少（毫秒）

public final static int BLOCK\_KIND\_NUMBER = 7;方块的样式数目为7

public final static int BLOCK\_STATUS\_NUMBER = 4;每一个样式的方块的反转状态种类为4

public final static int[][] STYLES;分别对应7种模型的28种状态

public ErsBlock(int style, int y, int x, int level, GameCanvas canvas)构造函数，产生一个特定的块

public void run()线程类的run()函数覆盖，下落块，直到块不能再下落

public void moveLeft()块向左移动一格

public void moveRight()块向右移动一格

public void moveDown()块向下移动一格

public void turnNext()块变型

public void pauseMove()暂停块的下落，对应游戏暂停

public void resumeMove()继续块的下落，对应游戏继续

public void stopMove()停止块的下落，对应游戏停止

private void erase()将当前块从画布的对应位置移除，要等到下次重画画布时才能反映出来

private void display()让当前块放置在画布的对因位置上，要等到下次重画画布时才能看见

public boolean isMoveAble(int newRow, int newCol)当前块能否移动到newRow/newCol 所指定的位置

private synchronized boolean moveTo(int newRow, int newCol)将当前块移动到newRow/newCol 所指定的位置

private boolean isTurnAble(int newStyle)当前块能否变成newStyle所指定的块样式，主要是考虑 边界以及被其他块挡住，不能移动的情况

private boolean turnTo(int newStyle)将当前块变成newStyle所指定的块样式

**class ControlPanel extends JPanel控制面板类，继承自JPanel。 上边安放预显窗口，等级，得分，控制按钮 主要用来控制游戏进程。:**

public ControlPanel(final ErsBlocksGame game)控制面板类的构造函数

btPlay.addActionListener(new ActionListener()开始游戏

btPause.addActionListener(new ActionListener()暂停游戏

btStop.addActionListener(new ActionListener()停止游戏

btTurnLevelUp.addActionListener(new ActionListener()升高难度

btTurnLevelDown.addActionListener(new ActionListener()降低游戏难度

public void setTipStyle(int style)设置预显窗口的样式

public int getLevel()取得用户设置的游戏等级。

public void setLevel(int level)让用户修改游戏难度等级。

public void setPlayButtonEnable(boolean enable)设置“开始”按钮的状态。

public void reset()重置控制面板

public void fanning()重新计算TipPanel里的boxes[][]里的小框的大小

private class TipPanel extends JPanel预显窗口的实现细节类

public TipPanel()预显示窗口类构造函数

public TipPanel(Color backColor, Color frontColor)预显示窗口类构造函数

public void setStyle(int style)设置预显示窗口的方块样式

public void paintComponent(Graphics g)覆盖JComponent类的函数，画组件。

public void fanning()g根据窗口的大小，自动调整方格的尺寸.