



# 安徽理工大学

ANHUI UNIVERSITY OF SCIENCE & TECHNOLOGY

## 工程训练 B 实训报告

题 目 : 基于 Java swing 的俄罗斯方块游戏

学 院 : 计算机科学与工程学院

专 业 班 级 : 计算机 19-5 班

学 号 : 2019304295

学 生 姓 名 : 程诗沁

指 导 老 师 : 孙克雷

2021 年 07 月 25 日

## 工程训练 B 实训执行计划书

实训名称	工程实训 B		
实训题目	基于 Java swing 的俄罗斯方块游戏		
学期	2021-2022 (1)	专业班级	计算机 19-5 班
实训目的和要求	<p><b>实训目的</b> Java 项目开发</p> <p><b>实训要求</b></p> <p>(1) 问题分析和任务定义： 根据给定的项目素材完成相应的功能</p> <p>(2) 逻辑设计和数据结构的选择： 选择使用数组和集合中的 list 集合来存储数据.</p> <p>(3) 详细设计和编码： 编码中采用窗体类和画布类搭建项目框架, 使用 IO 来读取素材图片, 使用线程来操作项目中的移动业务.</p> <p>(4) 上机调试： 使用 Debug 操作调试项目</p>		
	<p>(1) Java 鼠标监听时间, 键盘监听事件, IO 流, 文件存储, 文件读取, 多线程等</p> <p>(2) JavaWeb 开发 JSP, HTML, Tomact, Servlet, JDBC 等</p> <p>(3) 数据库</p> <p>测试数据： Debug 调试</p>		
实训的组织	<p>(包括分组、负责教师名称、专业班级、学生人数、方式、起止时间):</p> <p>分组情况： 程诗沁、孙逊、方文建、张凯健、盛洋</p> <p>指导教师： 孙克雷</p> <p>专业班级： 计算机 19-5 班</p> <p>学生人数： 5</p> <p>方 式： 实践</p> <p>起止时间： 19-20 周</p>		

<b>实训进度安排</b>	<p>2021.07.12-07.13 根据设计要求，查找相关资料，完成需求分析；</p> <p>2021.07.14-07.16 进行程序主体的概要设计；</p> <p>2021.07.17-07.20 进行具体程序的详细设计和程序的编写；</p> <p>2021.07.21-07.22 对程序进行调试分析，写出项目的实训报告。</p>
<b>系主任审核意见</b>	<p style="text-align: right;">系主任（签章）： 徐辉</p>
<b>学院审核意见</b>	<p style="text-align: right;">教学院长（签章）： 周华平</p>



# 目录

1. 研究状况.....	1
1.1 背景.....	1
1.2 游戏发展现状.....	1
2. 需求分析.....	2
2.1 功能分析.....	2
2.2 环境需求.....	2
3. 概要设计.....	3
3.1 功能描述.....	3
3.2 模块划分及介绍.....	3
3.3 系统功能示意图.....	4
4. 详细设计.....	5
4.1 模块描述.....	5
4.2 模块初始化.....	6
4.3 设置窗口菜单.....	7
4.4 构造方法.....	8
4.5 功能函数.....	10
4.6 游戏运行.....	12
4.7 游戏结束.....	15
5. 测试分析.....	17
5.1 设计思路.....	17
6. 实训总结.....	19
参考文献.....	20

# 1. 研究状况

## 1.1 背景

在 PC 日益普及的今天，一些有趣的桌面游戏已经成为人们在使用计算机进行工作学习之余休闲娱乐的首选，而俄罗斯方块游戏是人们最熟悉的小游戏之一，它以其趣味性强，易上手等诸多特点得到了大众认可。为此我们设计了一款简单的俄罗斯方块 Java 游戏程序，此程序可以为用户提供一个可在普通个人电脑上运行的，界面美观的，易于控制的俄罗斯方块游戏。以便更好的满足广大电脑工作者闲暇之余的娱乐消遣，并且也让我们提升编程能力和团队合作意识。

## 1.2 游戏发展现状

俄罗斯方块牵扯到多个数学问题，最常见的一个问题就是：俄罗斯方块是否一定能够永无止境？用更准确地语言来描述，就是假设双方都是理想状态不出额外错误，对于任意组合的图形顺序，是否都能保证无限消除而不会陷入无解状态。这个问题在俄罗斯方块诞生后的多年时间里都有人持续研究，1988 年一篇论文指出，俄罗斯方块游戏存在不能无限玩下去的情况，同时给出了一种算法可以保证游戏机能够将死玩家。

目前俄罗斯方块的游戏版权正式归属于 Tetris Company LLC，由它授权给其他公司发行新的游戏，如之前的 EA，以及现在的育碧都曾分别获得授权开发俄罗斯方块的新作，而这款游戏也成为史上移植平台最多的游戏，从最早的 PC 开始，它在无数平台上都留下了自己的足迹，从 GB 到 PSP，从 FC 到 PS3，更不用说以手机为代表的移动平台，或是网络在线游戏服务，每一个新兴的系统模式都少不了它的存在，而就在三十周年之际，它也同样会登陆 PS4 和 Xbox One 这最新一代的主机，在高清化的时代继续书写传奇。

俄罗斯方块是独一无二的，从诞生那天开始，它就已经进化到完美阶段，不需要再做任何多余的修改，三十年前的 DOS 版和现在的次世代版没有什么本质的不同，从操作方式到呈现形态全无二致，三十年的时间仿佛没有在它身上留下痕迹，而经过整整三十年的时间，它从一款游戏的生造单词变成了一个人尽皆知的专有名词，它创造了一类游戏类型，影响了游戏发展的进程，当然也在各个意义上改变了无数人的生活，在某个时候成为他们日常的一分子，或许在未来也会和围棋，纸牌一样成为人类社会根深蒂固的游戏文化。

## 2. 需求分析

### 2.1 功能分析

运行界面，包括游戏图形区域界面、游戏速度的选择更新界面、游戏分数的显示更新界面、下一个图形方块的显示更新区域、开始游戏按钮、重新开始游戏按钮以及退出游戏按钮。

功能扩展：菜单设置，窗口多样式，图形显示界面背景颜色，方块颜色等，以使该俄罗斯方块界面更为美观，功能更加齐全。

### 2.2 环境需求

本次开发本组选择编程语言是 Java，它是一种可以撰写跨平台应用程序的面向对象的程序设计语言。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网，同时拥有全球最大的开发者专业社群。此次设计使用的 IDE 环境是 Eclipse 和 IDEA，两个平台优势互补，最后将代码合并，完成了本次的实训项目。

运行环境：Windows10, Linux, MacOS

开发环境：Eclipse, IDEA

编程语言：Java

## 3. 概要设计

### 3.1 功能描述

俄罗斯方块游戏应有如下功能：

1. 在二维的平面里用各种随即产生的方块堆积木，每填满一行消去一行，当到达顶部时，游戏结束。
2. 玩家能通过方向键来控制方块的转动，左移，右移和直落。
3. 游戏能够在玩的过程中，给出玩家的分数，分数是由随即方块的类型决定的，每堆一个方块，就将分数累加到总分中。
4. 游戏有暂停、开始和结束、游戏规则、游戏说明等控制。

### 3.2 模块划分及介绍

本系统基于游戏的各项功能来设计游戏的各个功能模块系统功能模块示意  
本游戏主要有两大模块：游戏界面区，游戏控制区。游戏界面区分显示玩家可选操作、显示玩家操作结果两个部分。游戏控制区分更改颜色、开始、更改游戏等级、自定义下落速度、更改背景、退出以及其他等一些功能模块。

根据游戏的功能实现分为 6 个功能类：游戏主类、控制面板类、画布类、方块类和方格类。它们的作用如下：

RussiaBlocksGame 类：游戏主类，创建菜单栏，添加游戏，控制，帮助等菜单，游戏菜单里添加新游戏、退出等按钮，控制菜单里添加开始、暂停等按钮，帮助菜单里添加游戏规格和关于本游戏等按钮。

ControlPanel 类：控制面板类，继承 JFrame 类创建一个框架，再创建一个控制面板，加上开始、暂停、增加难度等按钮。设置预显窗口的样式，调整合适的大小。

ErsBlock 类：方块类，继承自线程类（Thread）由  $4 \times 4$  个方块（ErsBox）构成一个方块，控制块的移动、下落、变形等

GameCanvas 类：画布类，内有〈行数〉\*〈列数〉个方格类实例，继承自 JPanel 类，通过检查方格颜色（白色和灰色）来体现 ErsBlock 块的移动情况。

Ersbox 类：方格类，组成块的基本元素，用自己的颜色（默认白色）来表示块的外观。

musicStuff 类：背景音乐类，主要使用了 javax.sound 包，通过调用本地路径下的音频文件实现了游戏背景音乐的添加。



3.3 系统功能示意图

本游戏的系统功能模块示意图，如图 3-1 所示：

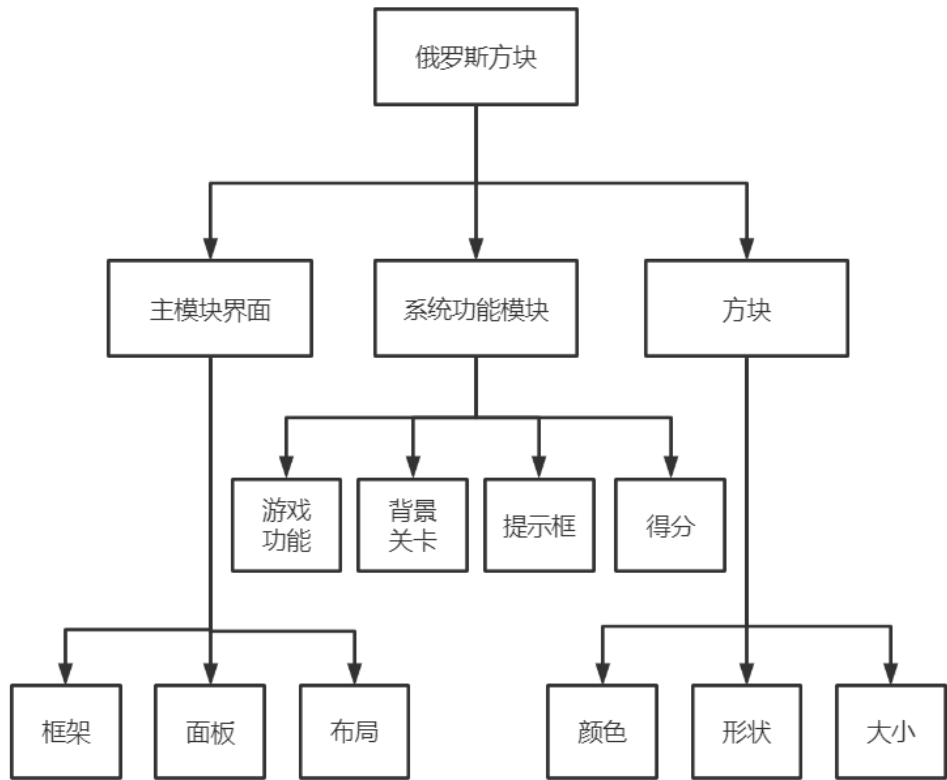


图 3-1 功能模块设计图

本游戏的菜单栏功能示意图，如图 3-2 所示：

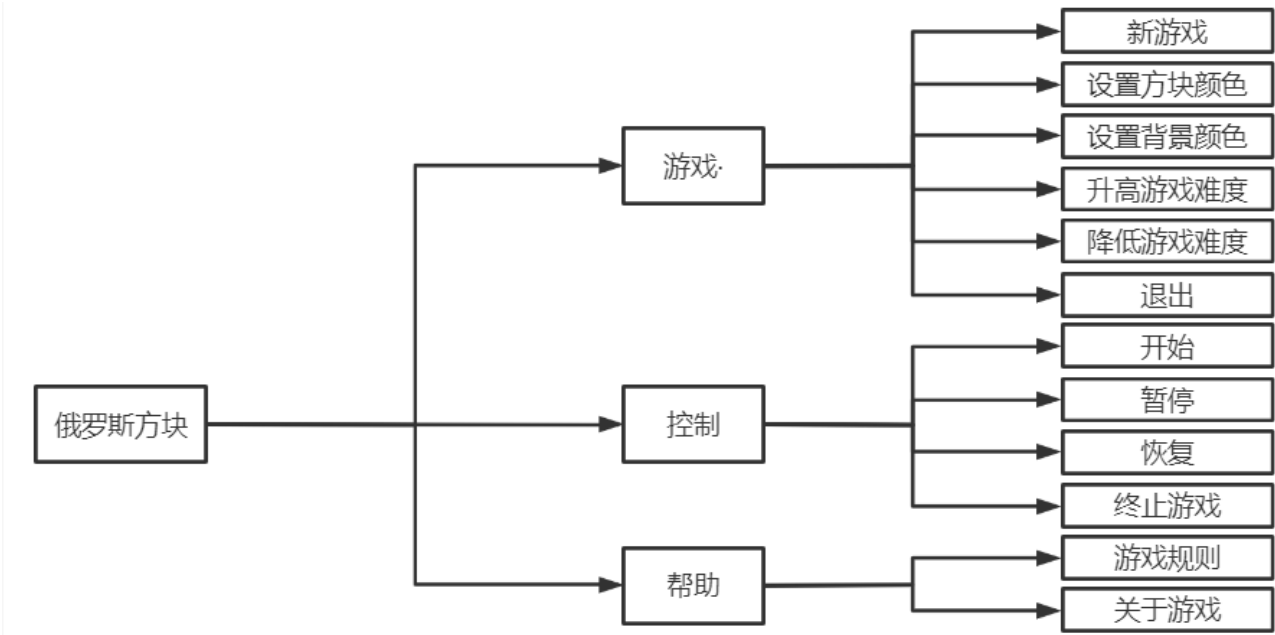


图 3-2 菜单栏功能

## 4. 详细设计

### 4.1 模块描述

RussiaBlocksGame 类是我负责编写的部分，该类是游戏的主类，继承自 JFrame 类，负责游戏的全局控制。内含一个画布类(GameCanvas)的实例对象，一个保存当前活动块(RussiaBlock)实例的对象，一个保存当前控制面板(ControlPanel)实例的对象。

首先是初始化一些常量和对象以及初始化菜单栏，其次是建立并设置窗口菜单并为按钮添加监听事件，然后是主游戏类的构造方法，在此构造方法中建立了窗口，注册窗口事件，并且能够根据窗口大小自动调节画布中方格的尺寸。后续代码定义了一些函数，其中包括：

play(): 游戏开始

reset(): 游戏复位

isPlaying(): 判断游戏是否进行

getCurBlock(): 得到当前活动的块

getCanvas(): 得到当前画布

playGame(): 用户开始游戏

pauseGame(): 用户暂停游戏

resumeGame(): 让暂停中的游戏继续

stopGame(): 用户退出游戏

getLevel(): 得到游戏者设置的难度

setLevel(): 用户设置游戏难度

getScore(): 得到游戏积分

levelUpdate(): 当积分累积到一定数值时，升一级

reportGameOver(): 宣告游戏结束

getScoreForLevelUpdate(): 获取自上次升级以来的游戏积分

随后是一轮游戏的过程，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并让它自动落下。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。最后就是 GameOver 对话框，它会在游戏结束后弹出，显示玩家获得的分数，并提供两个选项“再玩一局”和“退出游戏”(已添加监听事件)供玩家选择。

## 4.2 模块初始化

最开始的是初始化一些常量和对象。其中包括每一行的计分数，升级所需的积分数，最高难度等级，初始难度等级，以及画布类 (GameCanvas)、方块类 (RussiaBlock)、控制面板类 (ControlPanel) 所实例化出来的对象，随后就是初始化菜单栏。

```
public final static int PER_LINE_SCORE = 100;
public final static int PER_LEVEL_SCORE = PER_LINE_SCORE * 20;
public final static int MAX_LEVEL = 10;
public final static int DEFAULT_LEVEL = 1;
private GameCanvas canvas;
private ErsBlock block;
private boolean playing = false;
private ControlPanel ctrlPanel;
private JMenuBar bar = new JMenuBar();

private JMenu mGame = new JMenu(" 游戏"),
    mControl = new JMenu(" 控制"),
    mInfo = new JMenu("帮助");

private JMenuItem miNewGame = new JMenuItem("新游戏"),
    miSetBlockColor = new JMenuItem("设置方块颜色..."),
    miSetBackColor = new JMenuItem("设置背景颜色..."),
    miTurnHarder = new JMenuItem("升高游戏难度"),
    miTurnEasier = new JMenuItem("降低游戏难度"),
    miExit = new JMenuItem("退出"),
    miPlay = new JMenuItem("开始"),
    miPause = new JMenuItem("暂停"),
    miResume = new JMenuItem("恢复"),
    miStop = new JMenuItem("终止游戏"),
    miRule = new JMenuItem("游戏规则"),
    miAuthor = new JMenuItem("关于本游戏");
```

## 4.3 设置窗口菜单

窗口菜单分为游戏栏、控制栏以及帮助栏。其中游戏栏包括新游戏、设置方块颜色、设置背景颜色、升高游戏难度、降低游戏难度以及退出；控制栏包括开始、暂停、恢复以及终止游戏；帮助栏包括游戏规则和关于本游戏。在建立完菜单后，给菜单中的按钮添加监听事件，调用了 swing 中的 `addActionListener()` 方法，实现了在点击该按钮时，程序作出相对应的响应。以下程序是给游戏栏添加按钮以及设置“新游戏”和“设置方块颜色”的监听事件。

```
mGame.add(miNewGame);
mGame.addSeparator();
mGame.add(miSetBlockColor);
mGame.add(miSetBackColor);
mGame.addSeparator();
mGame.add(miTurnHarder);
mGame.add(miTurnEasier);
mGame.addSeparator();
mGame.add(miExit);

miNewGame.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        stopGame();
        reset();
        setLevel(DEFAULT_LEVEL);
    }
});

miSetBlockColor.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        Color newFrontColor =
            JColorChooser.showDialog(RussiaBlocksGame.this,
                "设置方块颜色", canvas.getBlockColor());
        if (newFrontColor != null) {
```

```

        canvas.setBlockColor(newFrontColor);
    }
}
});

```

窗口设置完毕后如图 4-1、图 4-2 和图 4-3 所示：

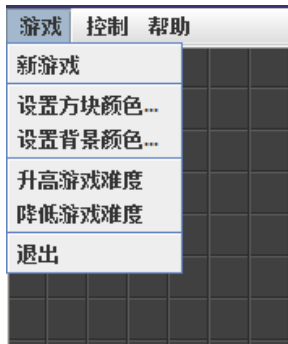


图 4-1



图 4-2



图 4-3

## 4.4 构造方法

在主游戏类 `RussiaBlocksGame` 的构造方法中首先是设置窗口的标题以及利用 `setSize()` 和 `setLocationRelativeTo()` 方法来确定大小和位置。随后用 `creatMenu()` 和 `getContentPane()` 方法创建菜单栏并获取内容面板，设置完窗口的布局管理器后新建游戏画布和控制面板，分别位列与界面的左右。

在此之后，重写 `awt` 中的 `addWindowListener()` 方法来注册窗口事件，当点击关闭按钮时，结束游戏，系统退出。再重写 `addComponentListener()` 方法，调用画布类 (`GameCanvas`) 中的 `adjust()` 函数来实现根据窗口大小自动调节方格的尺寸的功能。以下程序是构造方法的全部内容：

```

public RussiaBlocksGame(String title) {
    super(title);
    setSize(500, 600);
    setLocationRelativeTo(null);
    creatMenu();
    Container container = getContentPane();
    container.setLayout(new BorderLayout(6, 0));
}

```

```

canvas = new GameCanvas(20, 15);
ctrlPanel = new ControlPanel(this);
container.add(canvas, BorderLayout.CENTER);
container.add(ctrlPanel, BorderLayout.EAST);
addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent we) {
        stopGame();
        System.exit(0);
    }
});
addComponentListener(new ComponentAdapter() {
    @Override
    public void componentResized(ComponentEvent ce) {
        canvas.adjust();
    }
});
setVisible(true);
}

```

构造完成后的界面如图 4-4 所示：

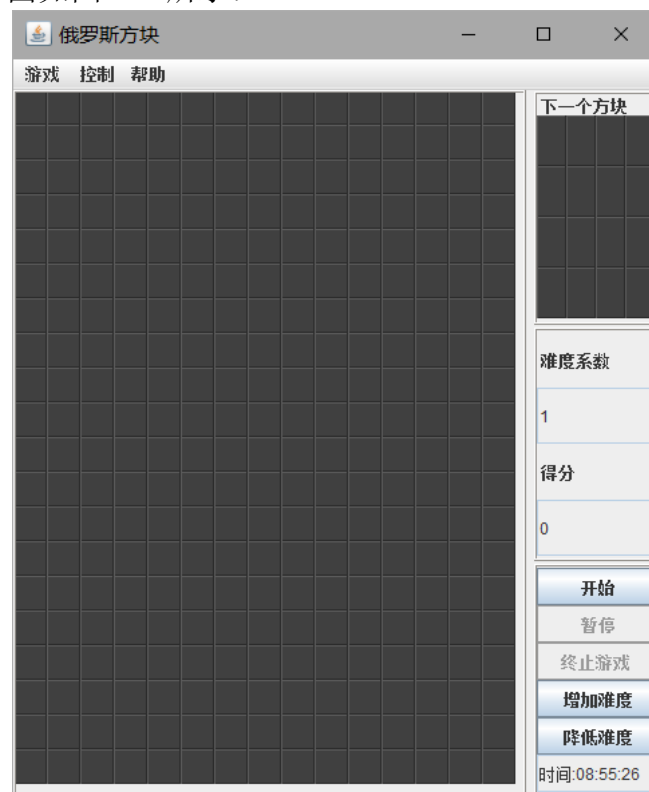


图 4-4 界面构造完成

## 4.5 功能函数

在此主游戏类我还定义了供全类调用的函数。其中包括 play() 游戏开始；reset() 游戏复位；isplaying() 判断游戏是否进行；getCurBlock() 得到当前活动的块；getCanvas() 得到当前画布；playGame() 用户开始游戏；pauseGame() 用户暂停游戏；resumeGame() 让暂停中的游戏继续；stopGame() 用户退出游戏；getLevel() 得到游戏者设置的难度；setLevel() 用户设置游戏难度；getScore() 得到游戏积分；levelUpdate() 当积分累积到一定数值时，升一级；reportGameOver() 宣告游戏结束；getScoreForLevelUpdate() 获取自上次升级以来的游戏积分。这些功能函数在 RussiaBlocksGame 类中的一些地方被多次调用，用来保证游戏运行的逻辑可行性。一下程序为部分功能函数：

```
public void stopGame() {
    playing = false;
    if (block != null) {
        block.stopMove();
    }
    ctrlPanel.setPlayButtonEnable(true);
    ctrlPanel.setPauseButtonEnable(false);
    ctrlPanel.setPauseButtonLabel(true);
    ctrlPanel.setStopButtonEnable(false);
    ctrlPanel.setTurnLevelDownButtonEnable(true);
    ctrlPanel.setTurnLevelUpButtonEnable(true);
    miPlay.setEnabled(true);
    miPause.setEnabled(false);
    miResume.setEnabled(false);
    miStop.setEnabled(false);
    miTurnHarder.setEnabled(true);
    miTurnEasier.setEnabled(true);
    reset();
}

public int getScoreForLevelUpdate() {
    if (canvas != null) {
        return canvas.getScoreForLevelUpdate();
    }
}
```

```

        return 0;
    }

    public boolean levelUpdate() {
        int curLevel = getLevel();
        if (curLevel < MAX_LEVEL) {
            setLevel(curLevel + 1);
            canvas.resetScoreForLevelUpdate();
            return true;
        }
        return false;
    }

    public int getScore() {
        if (canvas != null) {
            return canvas.getScore();
        }
        return 0;
    }
}

```

部分功能实现截图如图 4-5 和图 4-6 所示：

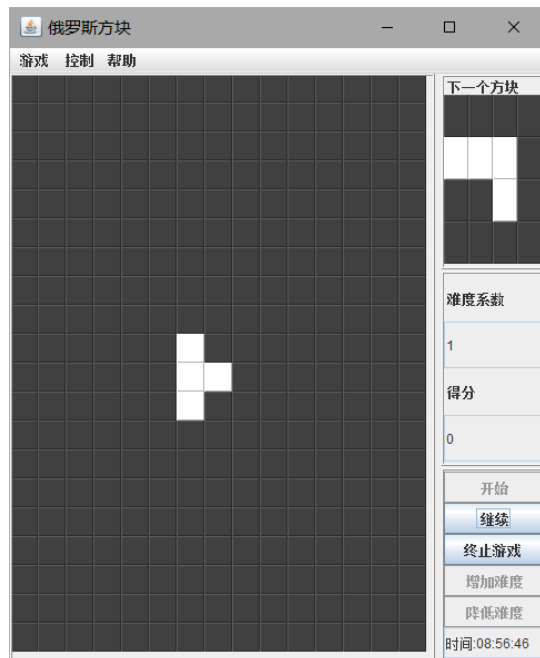


图 4-5 暂停游戏

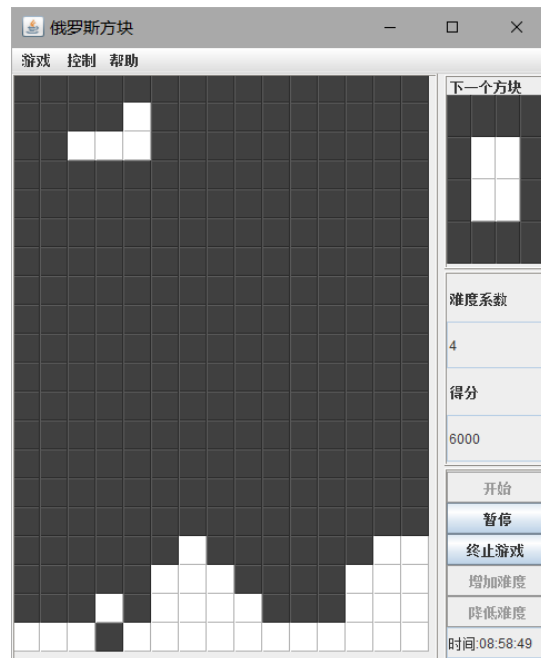


图 4-6 难度升级



## 4.6 游戏运行

实现了 Runnable 接口，一轮游戏是一个大的 while 循环，在这个循环中，每执行一次循环线程休眠一小段时间，检查游戏中的当前块是否已经到底并且为下一个方块的掉落做准备，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并让它自动落下。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，如果是，可以判断 Game Over。

首先利用 Math.random() 随机生成 col 和 style 来分别表示方块落下的 x 轴位置以及方块的形状。随后就是一个 while 循环，利用变量 playing 来控制循环的进行，在此循环内每隔 500 毫秒检查一次检查游戏中的当前块是否已经到底并且为下一个方块的掉落做准备，随后就是检查是否有全填满的行，如果有就删除。执行完判定满行的操作后判定游戏是否结束，若未结束，就生成新的方块并且启动相应线程，重新随机生成 col 和 style，并且在在控制面板中显示。以下程序是游戏运行的全部代码：

```
private class Game implements Runnable {
    @Override
    public void run() {
        int col = (int) (Math.random() * (canvas.getCols() - 3));
        int style = ErsBlock.STYLES[ (int) (Math.random() * 7)][(int)
(Math.random() * 4)];
        while (playing) {
            if (block != null) {
                if (block.isAlive()) {
                    try {
                        Thread.currentThread();
                        Thread.sleep(500);
                    } catch (InterruptedException ie) {
                        ie.printStackTrace();
                    }
                    continue;
                }
            }
            checkFullLine();
        }
    }
}
```

```

if (isGameOver()) {
    reportGameOver();
    ctrlPanel.setPlayButtonEnable(true);
    ctrlPanel.setPauseButtonLabel(false);
    ctrlPanel.setStopButtonEnable(false);
    miPlay.setEnabled(true);
    miPause.setEnabled(false);
    miResume.setEnabled(false);
    miStop.setEnabled(false);
    return;
}
block = new ErsBlock(style, -1, col, getLevel(), canvas);
block.start();
col = (int) (Math.random() * (canvas.getCols() - 3));
style = ErsBlock.STYLES[ (int) (Math.random() * 7)][(int)
(Math.random() * 4)];
ctrlPanel.setTipStyle(style);
}
}

```

游戏在运行时的截图如图 4-7 所示：

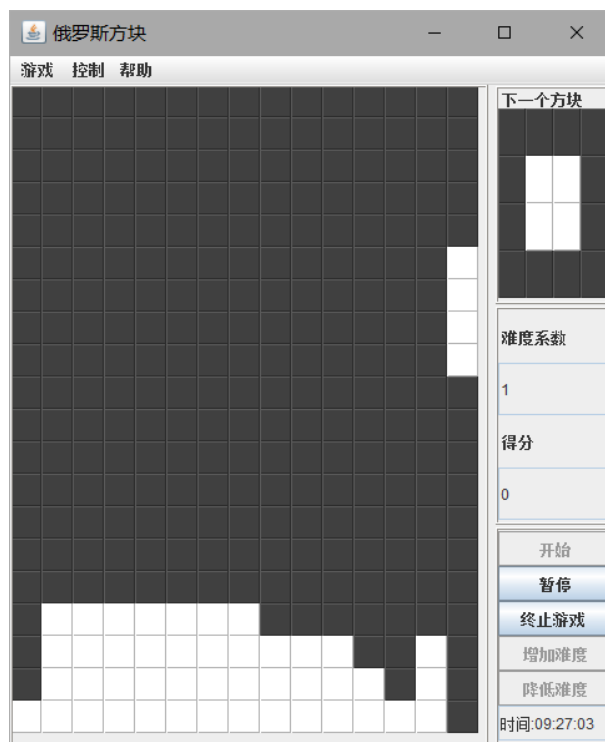


图 4-7 运行截图

游戏运行的整体流程图如图 4-8 所示：

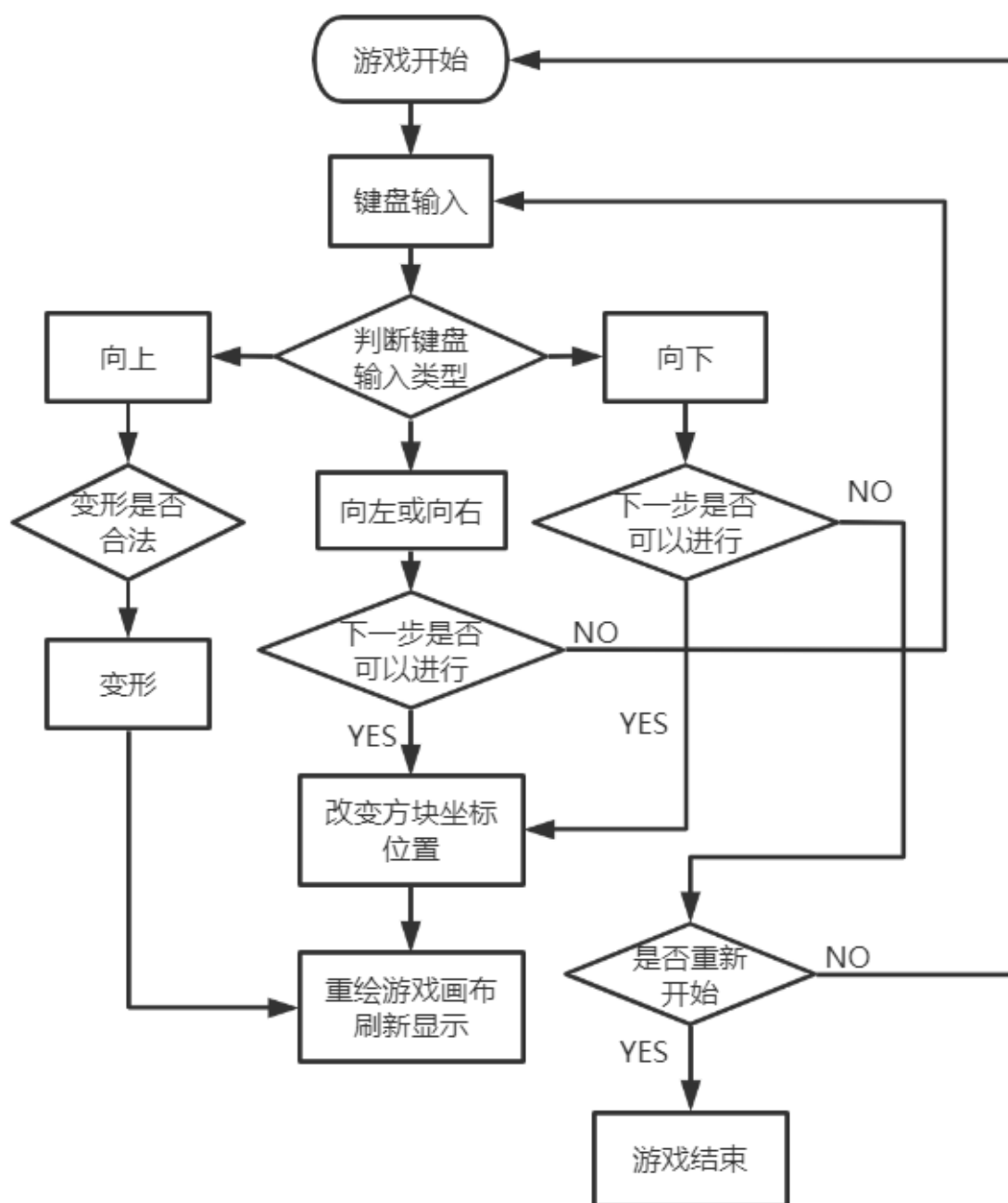


图 4-8 运行流程图

## 4.7 游戏结束

游戏结束时会弹出 GameOver 对话框，该对话框会使用 swing 中的 JPanel 以及 awt 中的 setLocationRelativeTo 重新定义并且设置一个全新的窗口，显示玩家获得的分数。并利用 swing 中的 JButton 提供两个选项“再玩一局”和“退出游戏”的按钮供玩家选择后续的操作，其中通过调用 swing 的 actionPerformed 方法来实现对按钮添加监听事件。以下程序是在游戏结束后弹出对话框的全部代码：

```
private class gameOverDialog extends JDialog implements ActionListener {
    private JButton againButton, exitButton;
    private Border border = new EtchedBorder
        (EtchedBorder.RAISED, Color.white, new Color(148, 145, 140));
    public gameOverDialog(JFrme parnt, Strng title, String message) {
        super(parent, title, true);
        if (parent != null) {
            setSize(240, 120);
            this.setLocationRelativeTo(parent);
            JPanel messagePanel = new JPanel();
            JPanel choosePanel = new JPanel();
            messagePanel.add(new JLabel(message));
            messagePanel.setBorder(border);
            Container container = this.getContentPane();
            container.setLayout(new GridLayout(2, 0, 0, 10));
            container.add(messagePanel);
            choosePanel.setLayout(new GridLayout(0, 2, 4, 0));
            container.add(choosePanel);
            againButton = new JButton("再玩一局");
            exitButton = new JButton("退出游戏");
            choosePanel.add(new JPanel().add(againButton));
            choosePanel.add(new JPanel().add(exitButton));
            choosePanel.setBorder(border);
        }
        againButton.addActionListener(this);
        exitButton.addActionListener(this);
    }
}
```

```

        this.setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == againButton) {
            this.setVisible(false);
            reset();
        } else if (e.getSource() == exitButton) {
            stopGame();
            System.exit(0);
        }
    }
}

```

游戏结束后弹出的对话框如图 4-9 所示：

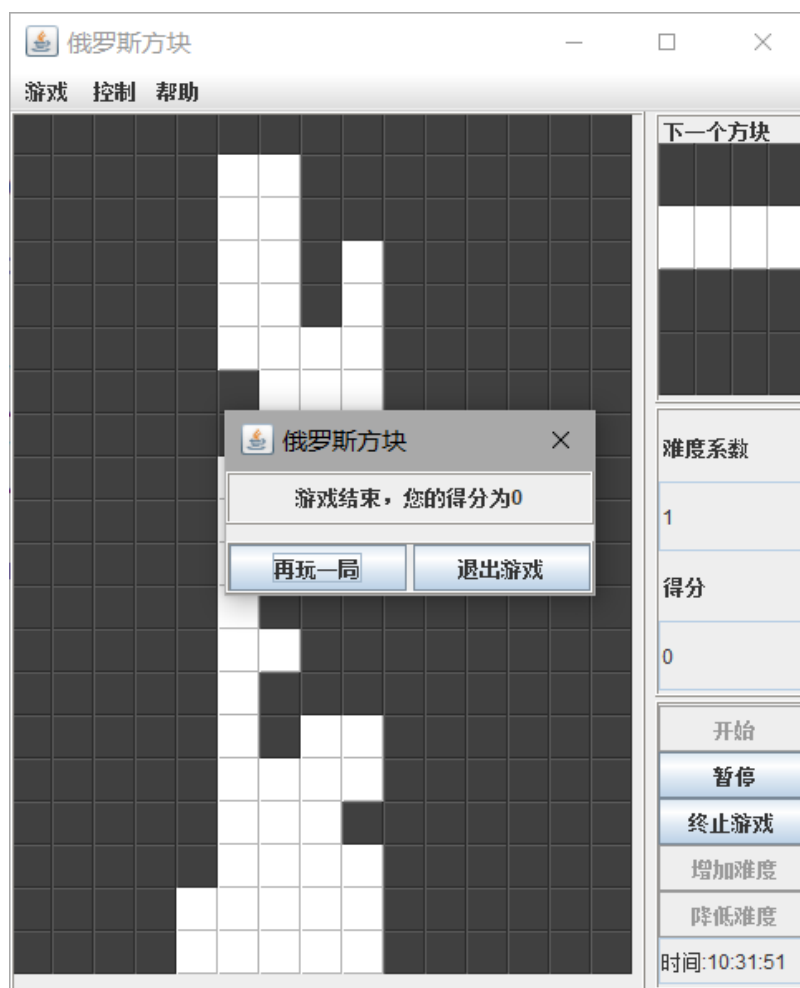


图 4-9 弹框图

## 5. 测试结果

### 5.1 各控制功能测试

#### (1) 暂停游戏

运行程序，点击开始，游戏成功运行，点击暂停按钮，游戏停止运行，即暂停游戏功能测试成功。

#### (2) 终止游戏

运行程序，点击开始，游戏成功运行，点击终止游戏按钮，游戏重新开始，即终止游戏功能测试成功。

暂停和终止功能示意图如图 5-1 和 5-2 所示

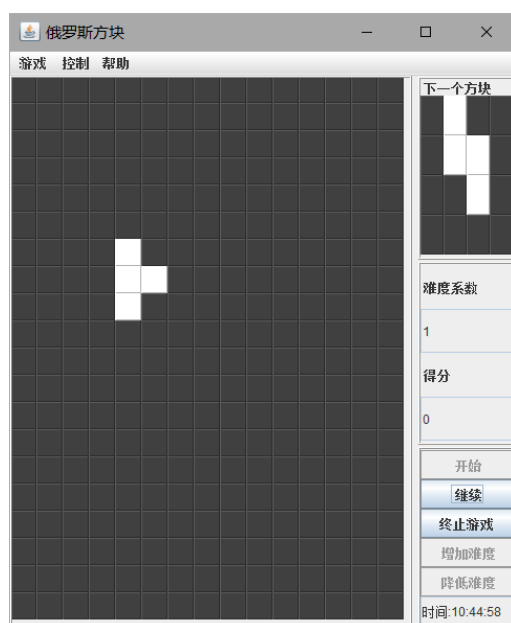


图 5-1 暂停游戏

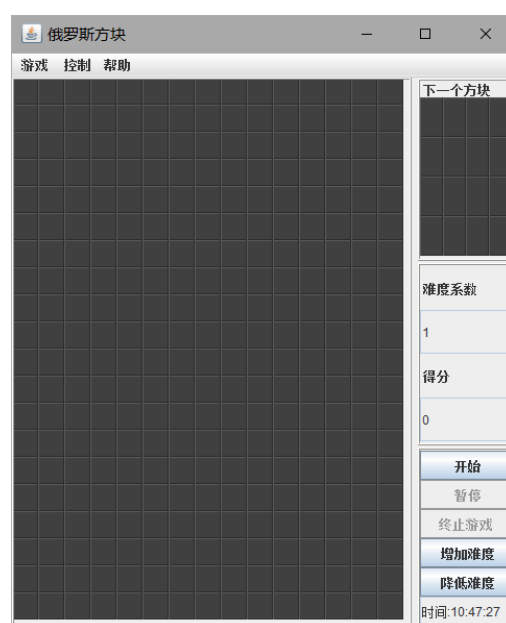


图 5-2 终止游戏

#### (3) 增加难度

运行程序，点击开始，游戏成功运行，点击增加难度按钮，对游戏难度进行增加设置，难度系数为 4，即增加难度功能测试成功。

#### (4) 降低难度

运行程序，点击开始，游戏成功运行，点击降低难度按钮，对游戏难度进行降低设置，难度系数为 3，即降低难度功能测试成功。

#### (5) 改变窗口风格

点击菜单栏游戏按钮，选择设置方块颜色，设为黄色；选择设置背景颜色，设为蓝色，设置后，游戏成功运行，即改变窗口风格功能测试成功。

(6) 运行程序，点击开始，游戏成功运行，可以菜单选择，可以控制难度，可以看到下一个方块图形，以及得分等。

以下功能的运行如图 5-3，5-4，5-5，5-6 所示：

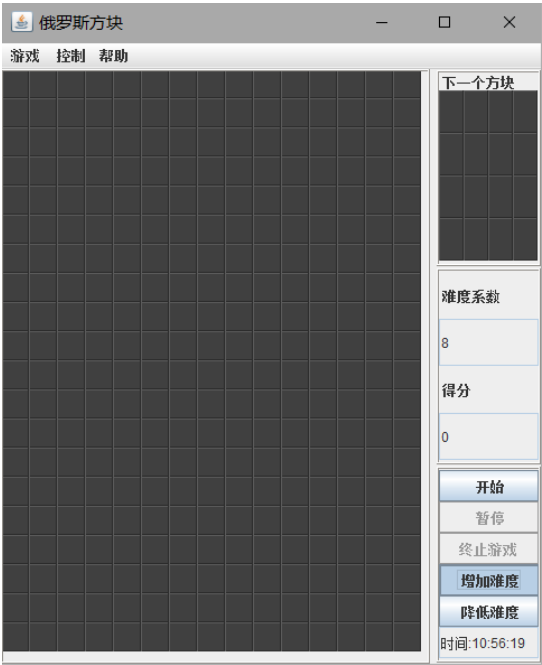


图 5-3 增加难度



图 5-4 降低难度

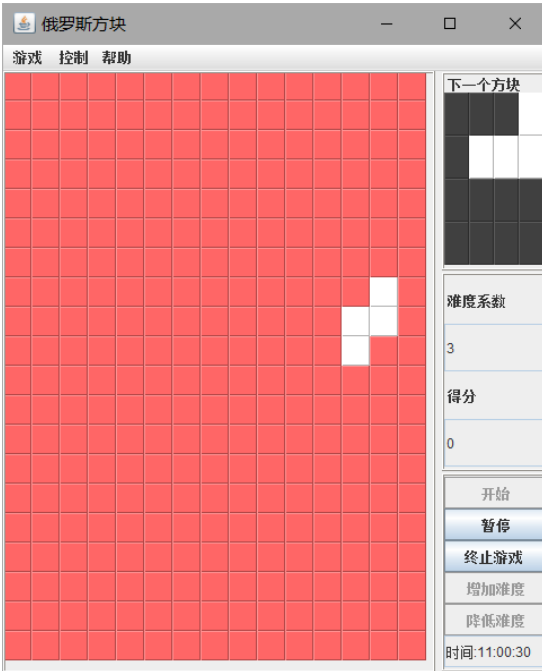


图 5-5 改变窗口风格

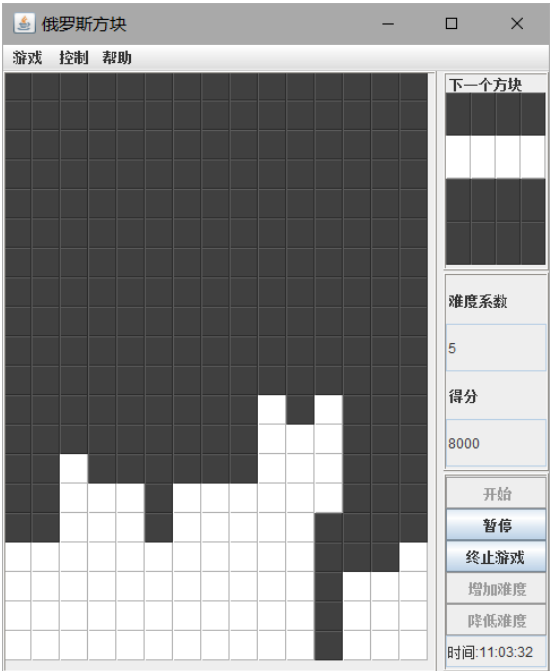


图 5-6 运行程序

## 6. 实训总结



## 参考文献

- [1] 荣钦科技. Java2 游戏设计. 清华大学出版社, 2004.
- [2] 朱福喜. Java 程序设计技巧与开发实例[M]. 北京: 人民邮电出版社, 2004. 2.
- [3] 高凌琴 陈青华. 俄罗斯方块游戏关键技术探讨[J]. 信息技术与信息化, 2008 年第二期
- [4] 赵东跃. 俄罗斯方块编程[J]. 电脑编程技巧与维护, 1998 年第六期.
- [5] 宋波. Java 应用开发教程[M]. 北京: 电子工业出版社, 2002.