

東南大學成賢學院

畢業設計報告（論文）

題目：掃雷遊戲設計

所屬系： 計算機工程系

專 業： 計算機科學與技術

學 號： 01111620

姓 名： 徐彩旭

指導教師： 王 征

起訖日期： 2015.3.2—2015.6.15

設計地點： 東南大學成賢學院

东南大学成贤学院毕业设计报告（论文）

诚信承诺

本人承诺所呈交的毕业设计报告（论文）及取得的成果是在导师指导下完成，引用他人成果的部分均已列出参考文献。如论文涉及任何知识产权纠纷，本人将承担一切责任。

学生签名：

日 期：

扫雷游戏设计

摘 要

21 世纪以来，随着人类经济迅速发展，计算机领域的应用也普及到经济和社会生活的各个领域，计算机在人们的日常生活中扮演着越来越重要的角色。随着计算机广泛的应用和深入的发展，当前的计算机应用早已不单局限于工作，计算机的娱乐性同样也越来越得到人们的关注。在日常生活中，单击智力小游戏占据了相当重要的地位，它可以满足人们在完成日常工作之余娱乐的需求，同时也不会浪费太长的时间去完成比赛。本课题主要是以 Java 语言作为开发工具，运用面向对象的思想，结合相应的数据结构算法去实现主要功能类似于 Windows 自带的扫雷游戏。本文重点介绍扫雷的实现，包括详细的类、各类之间的关系、重要算法的流程图以及游戏的测试。

关键词：swing 包；扫雷；面向对象程序设计

Design Of Mine Game

Abstract

Since the 21st century, with the rapid economic development of the human, applications in the computer field has also spread to all areas of economic and social life, and computer plays more and more significant role in people's daily life. With the computer widely used and in-depth development, the computer application has not only limited to the work, computer entertainment also attracts more and more attention. In daily life, the click intellectual games occupy a very important position, and it can satisfy people in spare time when people has completed their work. The game won not waste too much time to finish the game. This paper is mainly to use Java language as a development tool, simultaneously, including the idea of object oriented and corresponding data structure algorithm, to achieve the main function similar to windows built-in Mine game. This paper focus on the implementation of mine, including detailed class, the relations of these class, flow-process diagram of main algorithm and the test of software.

Keywords: swing package; mine game; OOP

目 录

摘 要.....	I
Abstract.....	II
第一章 引言.....	1
1.1 课题背景.....	1
1.2 国内外研究现状.....	1
1.3 论文的主要工作内容.....	1
1.4 论文的组织结构.....	2
第二章 相关理论与技术介绍.....	3
2.1 面向对象程序设计简介.....	3
2.2UML 简介.....	3
2.3Swing 简介.....	4
2.4Java 语言的发展.....	4
第三章 面向对象分析（OOA）.....	5
3.1 需求获取.....	5
3.2 需求分析.....	5
3.3 用例图.....	6
3.4 活动图.....	8
3.5 系统中涉及的类.....	10
3.6 本地所需文件.....	12
第四章 面向对象设计（OOD）.....	13
4.1 详细类图.....	13
4.2 系统实体类关联图.....	19
4.3 交互图（时序图、协作图）.....	23
4.4 程序流程图.....	28
第五章 软件测试.....	31
5.1 类测试.....	31
5.2 游戏软件黑盒测试.....	32
第六章 结束语.....	37
6.1 游戏特点.....	37
6.2 游戏局限.....	37
6.3 个人总结.....	37
致 谢.....	39
参考文献（References）.....	40

第一章 引言

1.1 课题背景

自 21 世纪以来，人类的经济飞速地发展，与此同时，计算机方面的应用也普及到经济和社会生活的各个应用领域。我们不仅可以运用计算机来完成许多重要的、精确度要求较高的工作，而且计算机在人们的日常生活中扮演着越来越重要的角色。随着计算机越来越广泛的应用和它与时俱进地不断深入发展，目前的计算机应用早已不单局限于工作，计算机单击的智力小游戏同样也越来越得到人们的青睐，近几年来游戏产业规模快速发展。权威市场调研机构国际数据集团先前发表的报告显示，在 2014 年全球游戏市场的产值达到数百亿美元，并预测 2015 年将达到千亿美元。这远远超过了电影产业以及音乐等众多娱乐事业的收益，从而成为目前娱乐事业的最大主流之一。

游戏行业发展可谓是一日千里，自 2000 年以来，若干游戏迅速涌入中国的市场，单击游戏同时在中国也得到了快速的发展。从某种程度上来说，该行业极大地影响和改变了人们的生活和娱乐的方式，游戏为广大消费者提供了丰富多彩的虚拟空间，使消费者可以自在地享受在虚拟世界的游戏所带来的乐趣。也正因为如此，很多人都瞄准了中国游戏这个巨大的市场，数以千计的游戏开发公司和团队，在中国这个拥有巨大市场的国家迅速发展壮大起来。

随着电脑游戏产业的规模越来越大，集成环境为广大的开发者提供了方便与便捷，极大地缩短了一个小游戏开发的周期，所以开发一款成功的电脑桌面小游戏靠个人之力已非难事。扫雷游戏早已经是 Windows 操作系统自带的一款小游戏，近年来虽然 Windows 操作系统历经数次换代更新，它的界面变得越来越美观，操作变得越来越方便，各方面完善得越来越好，功能同时也越来越强大，但是扫雷这款游戏依然是该系统装机必备的软件之一，仍依旧保留了下来，由此我们可以显而易见地看到这款单击小游戏自身具有着强大的生命力，正因为它长期的存在，我们可以得出这样的结论：这款游戏爱好者的数目是大常庞大的。

1.2 国内外研究现状

扫雷最原始的版本可以追溯到 1973 年一款名为“方块”的游戏。不久以后，“方块”被改写成了游戏“Rlogic”。在“Rlogic”里，玩家自己的身份是作为美国海军陆战队队员，他的最终目的探出一条没有地雷的安全路线，如果能找到一条安全的路线就算赢，如果路全被地雷堵死就算输。两年后，汤姆·安德森在“Rlogic”的基础上又编写出了游戏“地雷”，由此奠定了现代扫雷游戏的雏形。1981 年，微软公司的罗伯特·杜尔和卡特·约翰逊两位工程师在 Windows3.1 系统上加载了该系戏，扫雷游戏才正式在全世界推广开。

之后，MSN 中也出现了网络版的扫雷游戏，允许两个人同时游戏。再到后来腾讯游戏中的挖金子，可以说是扫雷游戏的变体。

1.3 论文的主要工作内容

扫雷游戏是比较经典而又非常有趣的一款小游戏，实现它的方法有很多，可以用很多不同的算法和语言去实现这款游戏，如 C、C++、VB、Java 及 Python 等语言。本论文是采用 Java 语言进行游戏软件的设计，其主要工作内容是设计和实现一款扫雷游戏，Windows 提供的扫雷游戏功能和界面都非常完善，本论文主要编写一个实现主要功能类似于 Windows 自带的扫雷游戏，将给出详细的类、各类之间的关系、重要算法实现的流程图以及及游戏程序的测试。

1.4 论文的组织结构

本文分 6 个部分，其具体组织如下：

第一章引言部分：介绍了本文的课题背景、国内外研究现状、论文的主要工作内容以及论文的组织结构。

第二章相关理论与技术部分：介绍了面向对象程序设计、统一建模语言（UML）、Swing 包、Java 语言的发展。

第三章面向对象分析（OOA）部分：介绍需求规定、需求获取、用例图、活动图、系统中涉及的类、本地所需文件。

第四章面向对象设计（OOD）部分：介绍详细类图、系统实体类关联图、交互图（时序图和协作图）、程序流程图。

第五章软件测试部分：介绍类测试、游戏软件黑盒测试。

第六章结束语部分：介绍游戏特点、游戏局限、个人总结。

第二章 相关理论与技术介绍

2.1 面向对象程序设计简介

面向对象程序设计简称为 OOP (Object Oriented Programming), OOP 是软件工程、结构化程序设计、数据抽象等为人所熟知的众多领域理论的发展。OOP 就是运用以对象作为基本元素的方法, 运用计算机语言描述并采取合理的方法处理一个特定问题。它符合客观世界本身的特点和人们分析问题的思维方式。按照面向对象的观点, 即客观世界是由各种各样的事物即“对象 (object)”组成, 而在计算机科学中, 对象是系统中用来描述客观存在事物的一个实体, 它是用于构成系统的一个重要的基本单位, 而系统又可以看作是由众多一系列相互通信的对象组成。

在 OOP 中, 一个类定义了同类对象所具有的公共属性和方法, 属性用数据结构表示, 方法用函数表示, 所以易得出这样的结论: 类=数据结构+对数据进行操作的函数。OOP 把数据和处理数据的过程 (即函数) 当成一个整体来对待, 便是我们通常所说的对象, 从现实客观的世界出发, 采用对象来描述实际要解决问题的实体, 同样在计算机世界里用程序代码模拟现实世界的对象, 使得程序设计过程更加自然、更加直观。结构化程序设计是以功能为中心划分系统的, 而面向对象程序设计是以数据 (即属性) 为中心来划分系统, 数据相对于功能而言具有更好的稳定性、可维护性。

OOP 同时也提供了各个对象之间的通信机制。程序由一系列相互通信的对象所构成, 对象之间的交互通过发送消息来实现。程序通过执行对象的各种行为方法, 来改变对象的状态 (即属性数据), 从而使该对象发生某些事件。当对象发生某事件时, 通常需要向其他对象发送消息以便获得它们的一些处理操作。OOP 易于把一个复杂的问题分解成多个功能独立的对象来处理, 然后把把这些对象进行有机组合起来去解决一个更加复杂的问题。20 世纪末期由于 Windows 系统的广泛使用, 可视化面向对象编程工具随后的推出, 系统的进一步发展更加使得 OOP 进入了它的黄金时期。在硬件技术飞速发展的今天, 程序设计方法的改进也越来越快, 从 20 世纪 60 年代的结构化程序设计, 到 80 年代的面向对象程序设计, 众所周知, 今天我们已进入基于组件的程序设计, 未来可能将迈向面向 Agent 的程序设计。

2.2 UML 简介

UML (Unified Modeling Language) 作为一种统一的软件建模语言, 它具有更加广泛、更加强大的建模能力。UML 是在进一步消化和进一步吸收当今已经存在的所有软件建模语言的基础上提出的, 集合了众多建模语言的优点, 是软件建模语言的一次重大发展。UML 还进一步突破了软件的限制, 广泛吸收了其它众多领域的建模思想, 并根据建模的一般性原理, 结合了软件本身的特点最终被人们所提出, 因此从某种意义上来说具有坚实的理论基础和广泛的应用性。

UML 取决于对事物实体、事物性质、事物关系、事物结构、事物状态、事物动态变化过程的详细描述和反映。UML 可以从不同视角描述人们所观察到的软件视图, 也可以进一步描述在不同开发阶段中的软件形态。UML 不仅可以建立需求模型、设计模型, 还可以建立实现模型, 因此它在软件设计中起着重要的作用。

作为一种建模语言, UML 有相对较严谨的语法和语义规范。UML 建立在元模型理论上, 包括四层元模型结构, 分别是基元模型、元模型、模型和用户对象。四层结构层层抽象, 下一层是上一层的实例。UML 中所有概念和要素均具有相对较严格的语义规范。

UML 采用一组图形符号来描述软件的模型, 这些图形符号具有简单、直观、规范的等特点, 开发人员学习和掌握起来也相对比较简单, 所描述的众多软件模型可以让我们直观的理解和阅读。由于 UML 具有易学性、规范性, 所以能够保证模型的准确、一致, 同时从某种程度上缩短了软件的开发周期。

2.3 Swing 简介

设计 Java 图形用户界面的基本元素主要由图形用户界面工具包 Swing 提供, 包括用户在开发软件过程中所用到的界面组件、处理事件的模型、图形和图像工具、布局管理器等等, 它们主要都可以在 `javax.swing.*` 中获得。Swing 是 Java 包的重要组成部分, 它为大规模的 GUI 开发提供了非常丰富的基础结构。javax.swing 包主要由组件类、事件类、接口、布局类、菜单类等所组成。

Swing 的优点有以下几点:

(1) Swing 工具包作为 AWT 的扩展, 同时又以 AWT 为基础, Swing 为 Java 程序增添了一种新的功能, 它可以根据一个给定的环境而调整它们的外观, 进而使得一个程序可以同时适用于多种操作平台, 它具有良好的跨平台性, 因此深受广大用户的喜爱。

(2) Swing 除了具有丰富的组件类型, 它还提供了十分广泛的标准组件, 受益于它良好的可扩展性, 除了标准组件, Swing 还提供了众多的第三方组件。许多商业或开源的 Swing 组件库在近几年都已经可以方便地获取并且便捷地使用。

(3) Swing 和 AWT 一样是 JRE 中的标准库, 它们都是与平台无关的。

(4) Swing 完全由 Java 代码实现, 所以几乎不会存在有兼容性的问题, Swing 在每个平台上都具有相同的性能, 不会有明显的性能差异。

2.4 Java 语言的发展

Java 语言是诞生于 20 世纪末期, 从它的刚开始至正式出现以来, 它飞速的发展已经让整个 Web 网络世界发生了翻天覆地的变化。它的问世时间虽然不是很长, 但是由于它很强的影响力, 已经得到了业界广泛认可, IBM、Apple、HP 和 Microsoft 等大型软件公司都已经购买了 Java 语言的许可证。微软公司还从其网页浏览器 Explorer 3.0 版本开始广泛增加对 Java 语言的支持。同时, 越来越多的软件开发商也开发了许多支持 Java 的产品。在目前网络膨胀的时代, 不支持 Html 和 Java 语言, 从一定程度上来讲, 就意味着应用程序的应用范围也只能限于局域网的环境。

Java 语言是一种跨平台性好、适合于分布式计算环境的面向对象编程语言。它具有众多优点, 如可移植性、面向对象、平台无关性、分布式、解释性、安全性、高性能、简单性、可靠性、多线程、动态性等等。目前的 Java 技术主要包括以下三个方面: 1. Java SE (Java Platform Standard Edition)。以前的为人熟知的版本也被称为 J2SE, 是 Java 平台的标准版, 是用于工作站、PC 的 Java 标准平台。它也从一定程度上体现了 Sun 公司的开放精神, 被称为“Internet 网络上的世界语言”; 2. Java ME (Java Platform Micro Edition), 以前的为人熟知版本也被称之为 J2ME, 是 Java 语言平台的精简版, 同时也是致力于消费产品和嵌入式设备的最佳选择方案。Java ME 是移动商务应用的最佳典范, 无线通信、手机等小型电子装置均可采用 Java ME 作为开发工具及应用平台。它可以提供了 HTTP 等高级 Internet 协议, 可以使众多移动设备以 B/S 模式直接访问 Internet 上的全部信息, 不同的客户端访问不同的文件, 此外还能访问本地存储区, 提供最高效率的无线交流。3. Java EE (Java Platform Enterprise Edition), 以前的版本称为 J2EE, 是 Java 平台的企业版, 它是为企业为环境而开发应用程序最佳的选择方案。它提供了非常强大的企业电子商务架构以及 Web Services 服务, 其优越的跨平台能力与开放的标准, 深受广大企业用户的喜爱。目前它已经快速成为开发商创建电子商务应用的事实标准。

第三章 面向对象分析（OOA）

3.1 需求获取

3.1.1 任务概述

(1) 目标:

在当今的社会，随着人们工作压力变得越来越大，如今更多的人开始通过游戏来排解内心的压力，扫雷是一款便捷、有趣而又无需连网的单机小游戏，Windows 所有版本的系统都提供扫雷游戏，它是一个十分能缓解压力的小游戏，在此欲通过 Java 语言来设计一款与其类似的扫雷游戏（实现重要功能），使一些新手用户能有较好的游戏体验。

(2) 用户特点

本游戏使用者为广大使用电脑的人，他们可以在工作忙碌的同时，来放松自己，通过小游戏来进行消遣、放松，扫雷游戏不需要连网的环境，因此单机游戏从某种意义上讲更加方便用户的使用。

(3) 假定和约束

由于本产品未投入使用，开发周期较短，个人能力有限，所以游戏难免会存在不足，还有部分缺陷难以在有限次的测试中测试出来。

3.1.2 需求规定

(1) 程序功能要求:

窗口游戏菜单中包含初级、中级、高级三个级别可供用户选择。

窗口上面板中包含显示“探雷”标记个数的文本框、重新开始按钮和显示用户游戏过程中所花费时间的文本框。

窗口下方面板中包含用户进行游戏的扫雷区域。

(2) 程序性能要求:

操作方式应该能够满足鼠标的左击和右击。

程序键盘式的输入与输出，所有输入与输出均通过菜单与按钮的方式给出。

3.1.3 运行环境设定

(1) 设备:

运行本软件要求处理器在奔腾 III 以上，内存在 256MB 以上的计算机

(2) 支持软件:

本扫雷游戏支持的系统软件有：Windows 系列系统、Java 虚拟机

3.2 需求分析

3.2.1 需求描述

(1) 扫雷游戏分为初级、中级和高级三个级别，这些级别可供不同的用户人群选择，用户可以很方便的通过菜单栏中的游戏菜单选项选择“初级”、“中级”和“高级”中的任意一个级别。

(2) 在用选择级别后，在面板中将会出现相对应级别的扫雷区域，在用户开始游戏时右上方将自动启动计时器，计时器计算用户最终完成游戏或者中途游戏失败所花费的时间。

(3) 用户若想要揭开其中任意一个方块时，左键单击此方块即可。如果所揭开的方块下是“地雷”，

用户便输掉了这场比赛,与此同时将显示所有地雷所在的位置并停止计时;如果所揭开的方块下不是“地雷”,则显示一个数字(或者不显示一个,而是以此方块为中心的一圈数字),该数字代表以显示数字的方块为中心,周围 8 个方块中共有多少个“地雷”。

(4) 根据系统所提供的数字,如果用户认为某个方块下可能是正是“地雷”所在的位置,便可通过单击右键在此方块上标识,此方块上面便显示了红色的小旗以方便标识。用户每标记一个探雷标记(无论用户所给的标记是否正确),程序左上方的剩余探雷数将减少一个(若小于 0,用户便不可再标识,但用户可通过再次右键取消已标记的方块)。

(5) 扫雷胜利后(即用户标记所有地雷正确的位置),程序弹出对话框告知用户已经胜利,同时用户可以查看到文本框中所用的时间。否则,用户将输掉比赛,雷区所有方块均无效,用户无法再操作任何一个方块。

3.2.2 活动列表

- (1) 用户选择相应的难度系数(或点击重新开始按钮重新开始此难度的游戏)
- (2) 用户在扫雷区进行扫雷
- (3) 系统记录相应用户所花费的时间

3.3.3 程序人机界面描述

(1) 用户和扫雷游戏的界面:

图形用户界面(Graphics User Interface, GUI)是由 Java 图形用户界面工具包 Swing 提供的。通过此 Swing 包所提供的人机接口设计,如使用窗口、菜单、对话框等,使得扫雷游戏的图形界面比较简洁,上方面板有“探雷”数目文本框、重新开始按钮以及记录时间的文本框,正下方面板是扫雷的区域(即用户进行游戏的区域)。

3.3 用例图

3.3.1 游戏中的参与者



图 1 游戏中的参与者

说明

这里指任意用户,不区分系统管理员和其他普通用户。

3.3.2 游戏中的用例

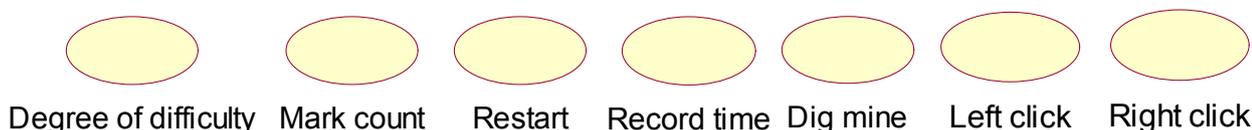


图 2 游戏中的用例

说明	
Degree of difficulty	游戏菜单栏中的难度系数（初级、中级、高级）
Mark count	左上方的文本框记录所剩“探雷”标记数
Restart	上方中间的重新开始按钮
Record time	右上方的文本框记录所用时间
Dig mine	用户进行游戏的主要动作，其余动作可以不执行，但必须执行扫雷
Left click	扫雷游戏的过程中用户必须用左键进行挖雷
Right click	扫雷游戏的过程中用户可以用右键标记可能是“地雷”的方块

3.3.3 总体用例图

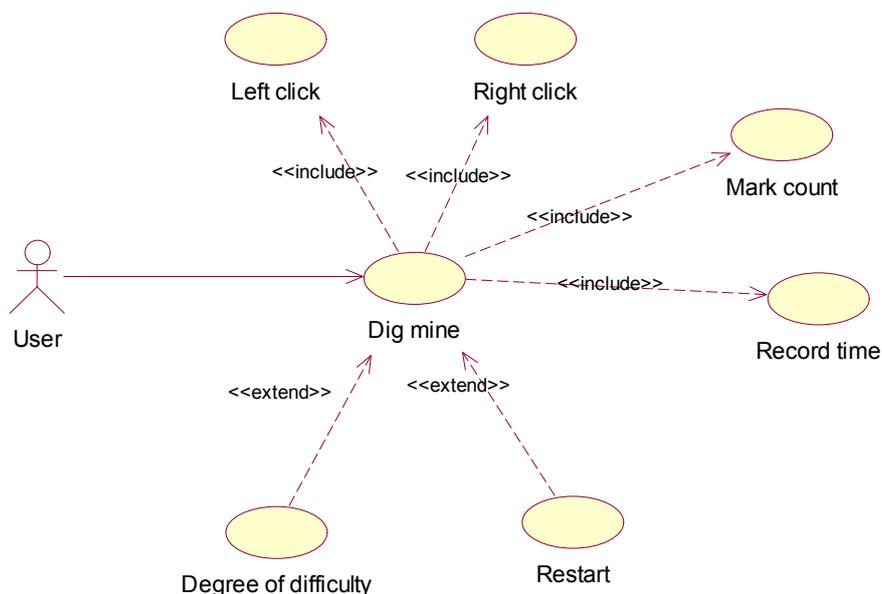


图 3 总体用例图

说明
<p>用户运行扫雷程序时，可以在菜单项进行游戏级别（及难度系数）的选择，也可以在必要时按重新开始按钮进行游戏的重置，以重新获取同等难度的随机布雷的序列。用户每左击一次便进行挖雷动作，扫雷游戏中包含记录“探雷”标记的数量使用情况，用户每右击一次就会减 1（直至减少至 0），还包含计时文本框，用户在最终胜利完成或失败时，文本框会记录用户游戏过程中所用的时间。</p>

3.4 活动图

3.4.1 Dig mine 用例活动图

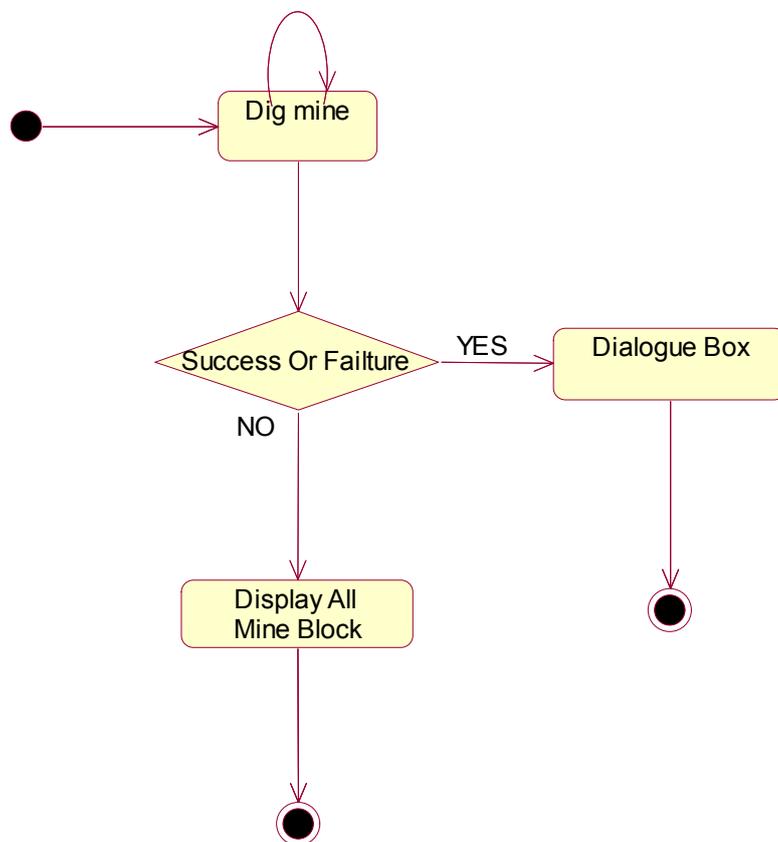


图 4 Dig mine 用例活动图

3.4.2 含 Degree of difficulty 用例的用例活动图

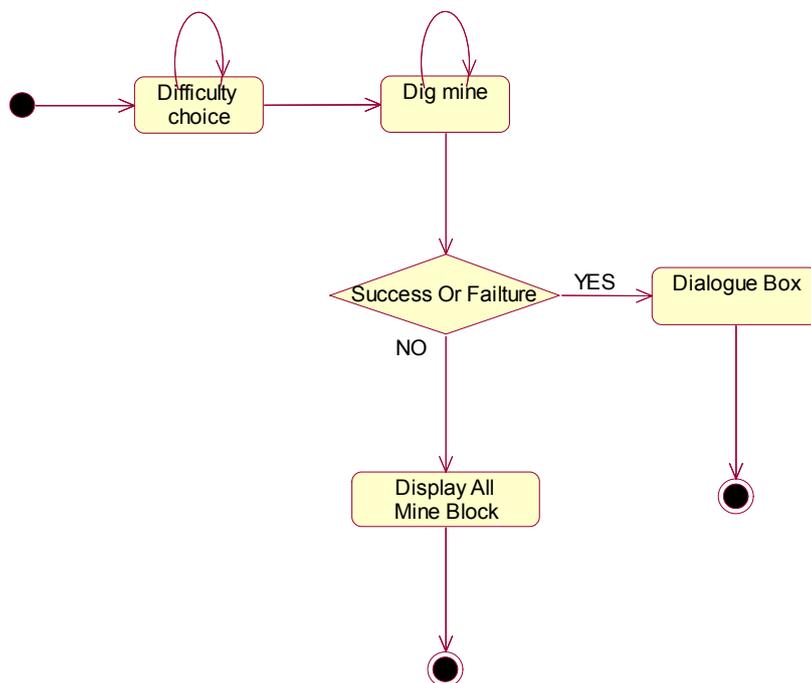


图 5 含 Degree of difficulty 用例的用例图

3.4.3 含 Restart 用例的用例活动图

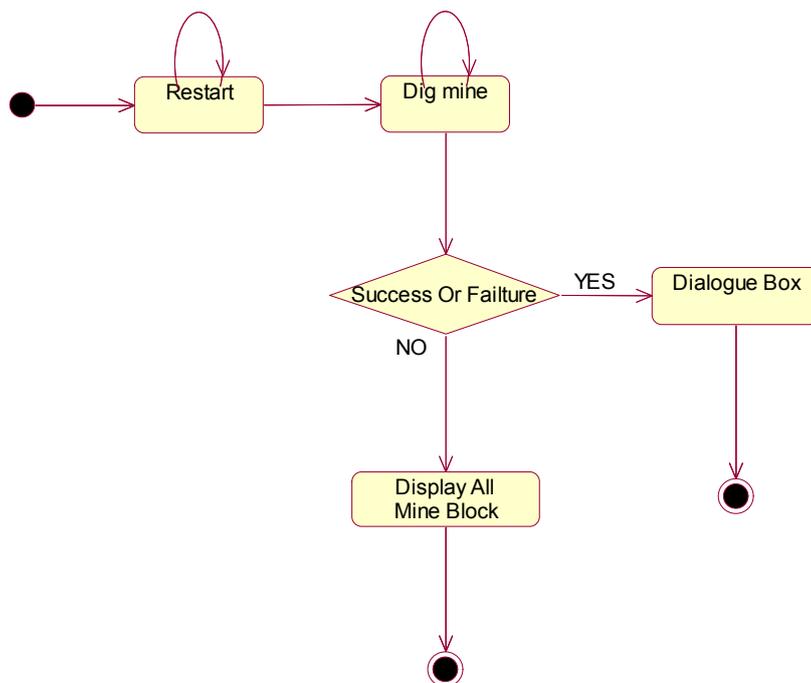


图 6 Restart 用例的用例活动图

说明

用户运行程序时（可以进行难度系数的选择，可以不选择直接进入默认的初级状态，也可以在游戏进行中重新开始游戏），在扫雷区域进行扫雷，重复执行扫雷的动作（即左击方块），若能挖出所有的“地雷”，则弹出对话框（告知用户已经取得胜利），并且用户可以在主窗口的右上方面板中查看自己完成扫雷游戏所花费的时间；若在游戏过程挖出的一个雷是错误的，则显示所有的雷区，并且终止用户的挖雷操作，从而结束游戏，用户同样可以查看自己所花费的时间。

3.5 系统中涉及的类

从以上需求中可以找出需要编写 5 个 Java 源文件，分别为 Game.java、GameRegion.java、Block.java、DisplayView.java 以及 PostMines.java，总体描述如下：

3.5.1 Game 类


图 7 Game 类

说明

Game 类主要是负责创建扫雷游戏的主窗口，该文件含有程序的入口函数 main()方法，扫雷游戏就从该类的此方法开始执行。Game 类主要有一种类型的成员：GameRegion。将在后面面向对象设计（OOD）中详细阐述 Game 类的主要成员的作用。

3.5.2 GameRegion 类


图 8 GameRegion 类

说明

GameRegion 类创建的对象是 Game 类最重要的成员，它代表“扫雷区域”。该类的成员变量中有 3 种重要类型的对象：Block 类、DisplayView 类和 PostMines 类。

3.5.3 DisplayView 类


图 9 DisplayView 类

说明

DisplayView 是 JPanel 容器的子类，DisplayView 创建的对象主要功能是负责为 Block 对象提供与用户交互的显示视图，用户以便通过该视图能方便地间接对 Block 对象操作。DisplayView 对象提供标签对象和按钮对象为 Block 对象提供显示视图，标签和按钮按照卡片布局(CardLayout)的方式叠放在一起，默认状态是显示按钮（起遮挡住标签作用）。当用户左键单击按钮后，如果 Block 对象是“地雷”，DisplayView 对象中的标签就显示的是雷的图标；如果 Block 对象不是雷，标签显示空白或者是周围 8 个方块内“地雷”的总个数。

3.5.4Block 类



图 10 Block 类

说明

Block 类是雷区中“方块”的属性以及有关方法的封装，它的主要功能是负责为雷区创建代表“方块”的 Block 对象，它起着虚拟方块的作用，其中的视图显示由 DisplayView 提供。

3.5.5PostMines 类

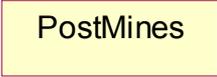


图 11 PostMines 类

说明

PostMines 类对象不需要视图，在游戏中不需要看见该对象。在初始化过程中，PostMines 对象是采用随机函数 random()指定 GameRegion 对象中的哪些 Block 对象是“地雷”，哪些 Block 对象不是“地雷”。

3.5.6 扫雷游戏所用到的类大体组织结构

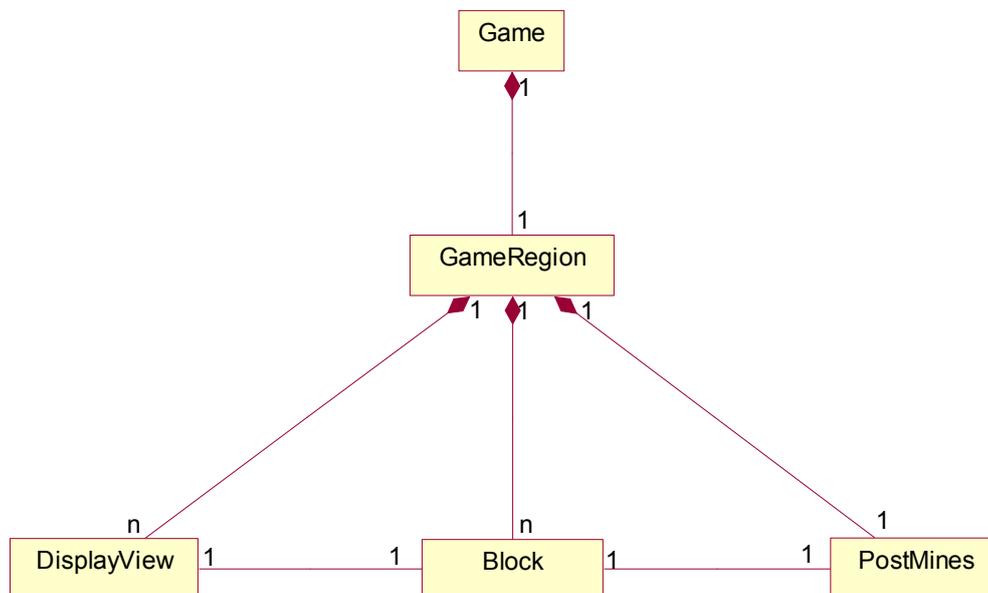


图 12 实体类大体组织结构

说明

在设计扫雷游戏时，需要编写 5 个 Java 源文件，这些源文件分别为：Game.java、GameRegion.java、Block.java、DisplayView.java、PostMines.java。扫雷游戏除了需要编写上述 5 个 java 源文件所给出的类外，还需要 java 系统本身提供一些起关键作用的类，如 JButton、JLabel、JTextField 和 JPanel 等类（这些类为系统提供，用户无需定义便可直接使用，只需在文件头中加入类似于 `import javax.swing.*;` 的方法标识告诉编译系统即可）。

3.6 本地所需文件

文件名称为 mark.gif 和 mine.gif 的图像文件，两幅图像文件分别是扫雷时所使用的“探雷标记”图像和“地雷”的图像： ，需将这两个文件放在与*.class 字节码文件的同一目录中。

第四章 面向对象设计（OOD）

4.1 详细类图

4.1.1 Game 类

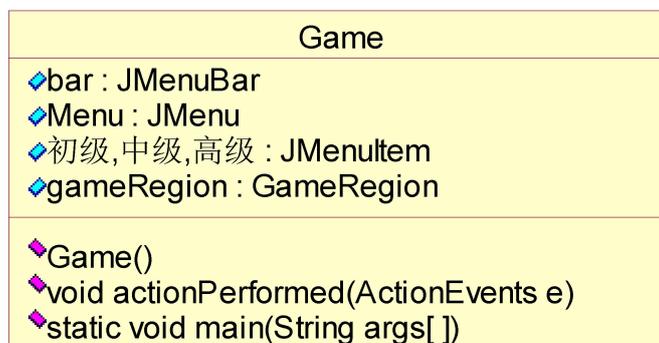


图 13 Game 类的详细类图

4 个类型的属性，3 个方法：

属性	
bar	JMenuBar 类型对象；java 菜单工具条，提供菜单操作
Menu	JMenu 类型对象；是 bar 菜单工具条的第一个菜单，即游戏菜单
初级、中级、高级	JMenuItem 类型对象；“游戏”菜单中包含 3 个菜单项：初级、中级和高级
gameRegion	GameRegion 类型对象（由 GameRegion.java 提供）；创建扫雷区域对象，提供雷区初始化工作以及有关扫雷相关操作的主要功能

操作	
Game()	函数无返回值；Game 类的构造函数，进行相关参数的初始化，从而完成最终窗口的初始化工作：1.菜单界面初始化，使得游戏菜单栏中有初级、中级、高级三个级别的选项，并分别为这三个菜单项添加监听事件；2.调用 GameRegion()方法初始化雷区，5 行 5 列，随机分布 3 个“地雷”，默认的级别为初级；3.设置 Game 的窗体，并初始化一些必要的参数。
actionPerformed(ActionEvent e)	Game 类实现 ActionListener 接口中的方法，该方法负责执行菜单发出的有关命令。用户选择菜单中的菜单项可触发 ActionEvent 事件，导致该方法执行相应的操作：1.如果事件源是初级菜单项，调用 GameRegion 类中的 initGameRegion()方法设置相应规模的雷区与雷数（即难度系数）2.如果事件源是中级菜单项，调用并设置相应规模的雷区与雷数 3.如果事件源是高级菜单项，调用并设置相应规模的雷区与雷数
main()	类的入口函数（主函数），是程序运行的入口方法，所执行的操作：1.构造类 Game 的对象

4.1.2 GameRegion 类

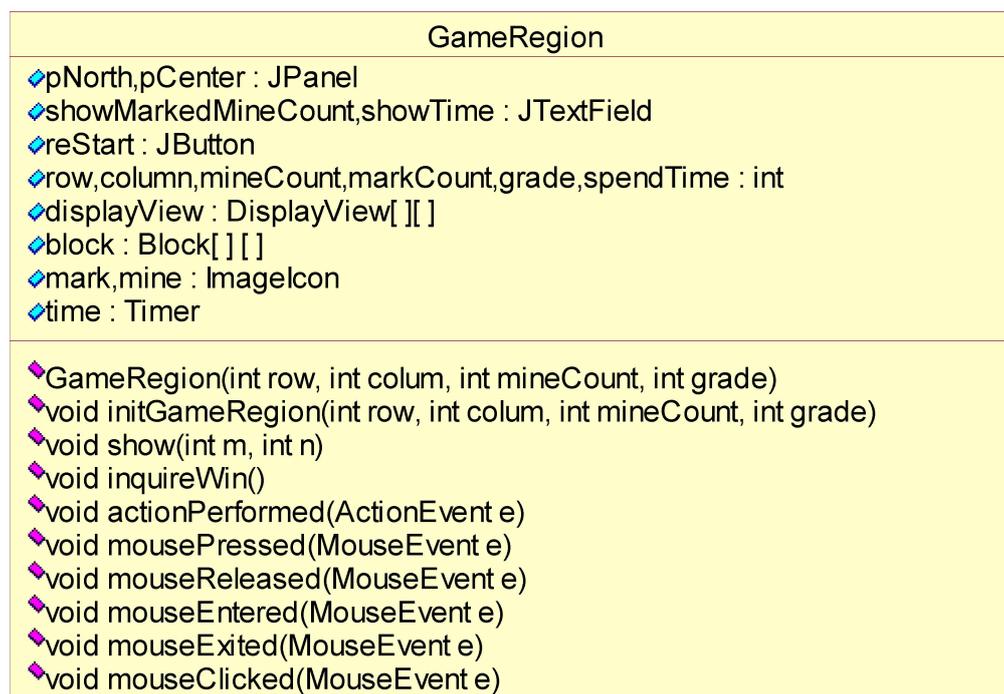


图 14 GameRegion 类的详细类图

8 个类型的属性，6 个有效方法：

属性	
pNorth, pCenter	JPanel 类型对象；可以用它来容纳其它组件；pNorth 用来容纳 2 个 JTextField 类型对象的文本框和 1 个 JButton 类型对象的重新开始按钮，pCenter 用来容纳雷区所需的组件。
showMarkedMineCount showTime	JTextField 类型对象；showMarkedMineCount 用来显示“探雷”标记的数目；showTime 用来显示用户进行游戏所消耗的时间。
reStart	JButton 类型对象；是一个按钮对象，主窗口上方面板中的重新开始按钮，用户可以单击它快速重新开始游戏。
row column mineCount markCount grade spendTime	int 型对象；row 表示雷区的行数；column 表示雷数列数（通常情况下 row 等于 column）；mineCount 表示“地雷”的个数；markCount 表示“探雷”标记的数目；grade 表示用户所选择的难度等级（共 3 个级别：1 代表初级，2 代表中级，3 代表高级）；spendTime 表示计时器（起着计时器的功能），showTime 文本框中显示的即为 spendTime 的值。
displayView	DisplayView 类型对象（由 DisplayView.java 提供）；是一个二维数组，主要负责为 block 数组中的 Block 对象提供显示视图。当用户点击雷区某方块时，要么显示地雷、要么显示周围雷的数量或者是空白，此数组每一个元素就是提供这三种外观显示。
block	Block 类型对象（由 Block.java 提供）；是一个二维数组，该数组每一个元素对应雷区的每一方块（即虚拟方块），界面显示由 blockView 提供，block 数组元素的个数决定雷区共有多少需扫雷的方块。

mark, mine	ImageIcon 类型对象； mark 表示探雷标记的图标， mien 表示“地雷”标记的图标
time	Timer 类型对象； 计时器功能， 设置 1000ms 触发事件， 从而让 spendTime 自增 1。

操作	
GameRegion(int, int, int, int)	函数无返回值； 类 GameRegion 的构造函数， 初始化的作用： 1.将雷数文本框 showMarkedMineCount 对象添加到 pNorth 面板中并进行相应的设置 2.将重新开始按钮 reStart 对象添加到 pNorth 面板中并进行相应的设置 3.将计时器 showTime 对象添加到 pNorth 面板中并进行相应的设置 4.调用初始化雷区此类中的 initGameRegion()方法完成 pCenter 面板的初始化工作。
initGameRegion(int ,int ,int ,int)	函数无返回值； 初始化 pNorth 面板容器和 pCenter 面板容器中的内容： 1.将 showMarkedMineCount 文本框中的内容设置为“探雷”标记的个数（即为 mineCount 的值）； 2.调用 PostMines 类的 postMinesForBlock()函数为虚拟方块 block 对象随机分布地雷； 3.将 block 的信息传递给 DisplayView 类型的对象， 并用 blockView 类的 JButton 对象遮盖需要显示的内容。
show(int , int)	函数无返回值； 如果此块不是雷， 则供 actionPerformed(ActionEvent e)方法调用 show(int , int)显示此方块的 blockNameOrIcon 属性： 若此方块周围含“地雷的方块”大于 0， 则显示地雷方块的个数； 若此方块周围无含“地雷的方块”， 则递归调用 show(int , int)， 再显示外围， 直至显示数值为止。 此函数的具体实现详见后面的程序流程图部分。
InquireWin()	函数无返回值； 该方法用来判断用户是否扫雷成功： 遍历所有属于雷区的方块， 计算所有未被挖开的方块的个数， 若个数恰好为所设置的雷的个数， 则终止游戏， 并且弹出对话框告知用户已经取得胜利。 此方法供此类的 actionPerformed(ActionEvent e)方法调用。
actionPerformed(ActionEvent e)	函数无返回值； 该方法是 GameRegion 类实现的 ActionListener 接口中的方法： 1.当用户单击 blockView 中的某个 JButton 方块时， actionPerformed(ActionEvent e)方法便能监听到， 便会执行该方法： 首先判断是否为 blockView 中的某个 JButton 方块触发此方法的， 若是为方块所触发， 则利用双重循环遍历所有属于该区域的方块以获取方块的具体位置（即坐标）； 再判断是否为“地雷”， 若为“地雷”， 则将所有方块设置为不可用并让用户输掉比赛； 否则调用 show()方法显示周围的方块信息； 若为重新开始按钮， 则调用本类中 initGameRegion()方法进行初始化工作； 2.当 1000ms 到时， 同样会触发此方法， 设置相应显示时间（spendTime 自增 1）； 3.最后调用 inquireWin()判断最终输赢。 关于此函数的具体实现详见后面的程序流程图部分。
mousePressed(MouseEvent e)	函数无返回值； 是 GameRegion 类实现 MouseListener 接口中的方法。

	当用户按下右键时，如果此时方块已经被标记为红旗，则取消标记，同时修改 showMarkedMineCount 文本框对象的值；若此方块未被标记为红旗，则标记并修改 showTime 文本框对象的值。
mouseReleased() mouseEntered() mouseExited() mouseClicked()	这些函数均无返回值；并且函数内什么都不执行，是空语句，因为此类是继承接口 MouseListener 的，所以抽象方法必须在子类中实现，所以必须在 GameRegion 类中实现这些函数的方法，但这些方法内部是空语句。

4.1.3 DisplayView 类



图 15 DisplayView 类的详细类图

3 个类型的属性，5 个有效方法：

属性	
blockNameOrIcon	JLabel 类型对象；用来显示 Block 对象（即虚拟方块）的 aroundMineNumber 和 mineIcon 的属性值，即踩中地雷便显示 mineIcon 图标，若未踩中，则显示以此方块为中心周围 8 个方块中“地雷”的数目，或者显示为空。
blockCover	JButton 类型对象；用来遮挡 blockNameOrIcon 标签，使得用户看不到 blockNameOrIcon 标签上的内容。
card	CardLayout 类型的对象；卡片式布局，它将容器中的每个组件看作是一张卡片，每次用户只能看到顶层的内容，底层内容被顶层内容所遮盖。

操作	
DisplayView()	函数无返回值；DisplayView 类的构造方法：1. 设置布局方案为卡片式布局 2. 创建标签 blockNameOrIcon 对象并将其添加加入此对象中 3. 创建按钮 blockCover 对象并将其添加加入此对象中
giveView(Block)	函数无返回值；给指定参数的 Block 对象提供视图：1. 如果 Block 对象的 isMine 属性为真（即此块是“地雷”），那么 blockNameOrIcon 标签上设置的图标就是 Block 对象中的属性 mineIcon 指定的图标(mine.gif 图片)，即设置由 blockCover 遮挡的标签控件的雷的图标；2. 如果 Block 对象的 isMine 属性的值为假，blockNameOrIcon 标签上设置的文本是 Block 对象的 aroundMineNumber 属性的值，即以此块为中心周围“地雷”的数目（数目需大于等于 1）；否则，标

	签上无显示的文本（即空白）。
seeBlockNameOrIcon()	函数无返回值；就是让用户看到卡片式视图中的标签 blockNameOrIcon 的属性值，按钮 blockCover 对象被置于底层。
seeBlockCover()	函数无返回值；显示卡片式视图中 JButton 类型的 blockCover 按钮对象，将标签对象 blockNameOrIcon 置于底层，不让用户看到标签上的信息（即“地雷”的分布情况）。
getBlockCover()	函数的返回值为 JButton 类型；获取到 blockCover 对象。

4.1.4Block 类



图 16 Block 类的详细类图

5 个类型的属性，10 个有效方法：

属性	
aroundMineNumber	int 型对象；表示以此方块为中心的周围 8 个方块内“地雷”的数目。
mineIcon	ImageIcon 类型对象；表示方块上“地雷”的图标。
isMine	布尔类型变量；判断方块是否为“地雷”，初始值为 false。
isMark	布尔类型变量；判断方块是否被右键点击的“探雷”标记图标所标记，初始值为 false。
isOpen	布尔类型变量；判断方块是否被鼠标左键点击过（即是否被挖开），初始值为 false。

操作	
setAroundMineNumber(int)	函数无返回值；设置以此方块对象为中心的周围 8 个方块中“地雷”的数目。

getAroundMineNumber()	函数返回值类型为 int 类型；获取此方块对象周围 8 个方块中为“地雷”的数目。
setIsMine(boolean)	函数无返回值；设置此方块对象是否为“地雷”
isMine()	函数返回值类型为布尔类型；判断此方块对象是否为雷，并返回 true 或者 false。
setMineIcon(ImageIcon)	函数无返回值；设置此方块对象上的“地雷”标记的图标。
getMineIcon()	函数返回值类型为 ImageIcon 类型；获取此方块对象上的“地雷”标记图标。
setIsOpen(boolean)	函数无返回值类型；设置此方块对象是否被挖开，挖开则设置为 true。
getIsOpen()	函数返回值类型为布尔型；获取此方块对象是否已被用户挖开。
setIsMark(boolean)	函数无返回值类型；设置此方块对象是否被鼠标右键点击的“探雷”标记所标记，被用户标记则设置为 true。
getIsMark()	函数返回值类型为布尔型；获取此方块是否被鼠标右键点击的“探雷”标记所标记。

4.1.5 PostMines 类



图 17 PostMines 类的详细类图

1 个类型的属性，2 个有效方法

属性	
mineIcon	ImageIcon 类型的对象；用来指向“地雷”的 mine.gif 图像。

操作	
PostMines()	函数无返回值；PostMines 类的构造方法，为 mineIcon 开辟相应的内存空间。
postMinesForBlock(Block[][],int)	函数无返回值；该函数将 mineCount 个地雷随机分散在二维数组 Block 中（将 block 对象串成链表并间接借助随机函数 random()来实现的），并将这些为“地雷”的方块图标属性设置为 mineIcon 图像；再次遍历所有方块，当此方块不是地雷时，计算出以此方块为中心的周围几个方块中是“地雷”的数目，即设置方块 block 对象的 aroundMineNumber 的值。

4.2 系统实体类关联图

4.2.1 Game 类

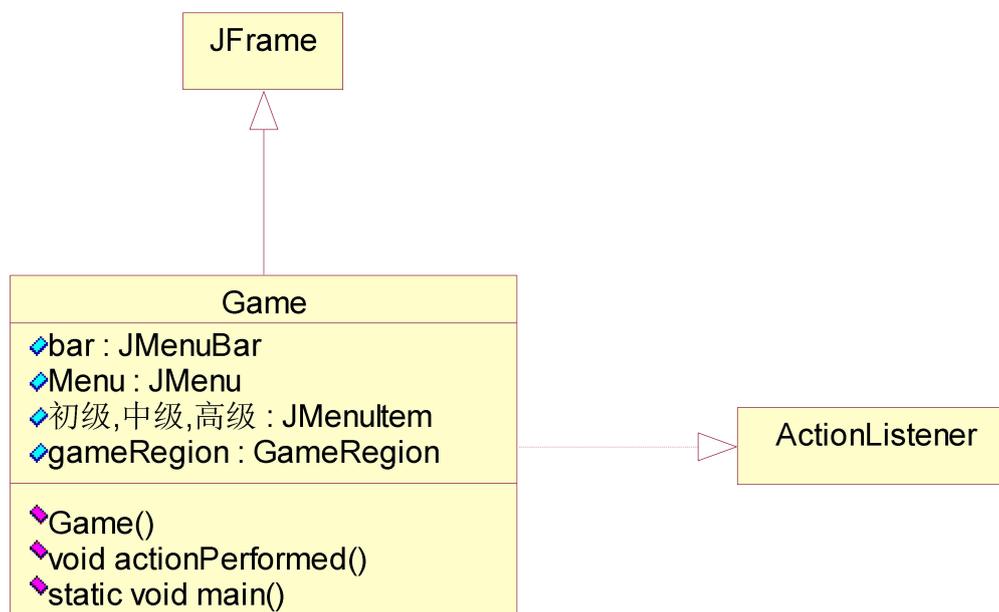


图 18 Game 类的关联图

说明

Game 类是 javax.swing 包中 JFrame 类的一个子类，并实现了 ActionListener 接口。

4.2.2 GameRegion 类

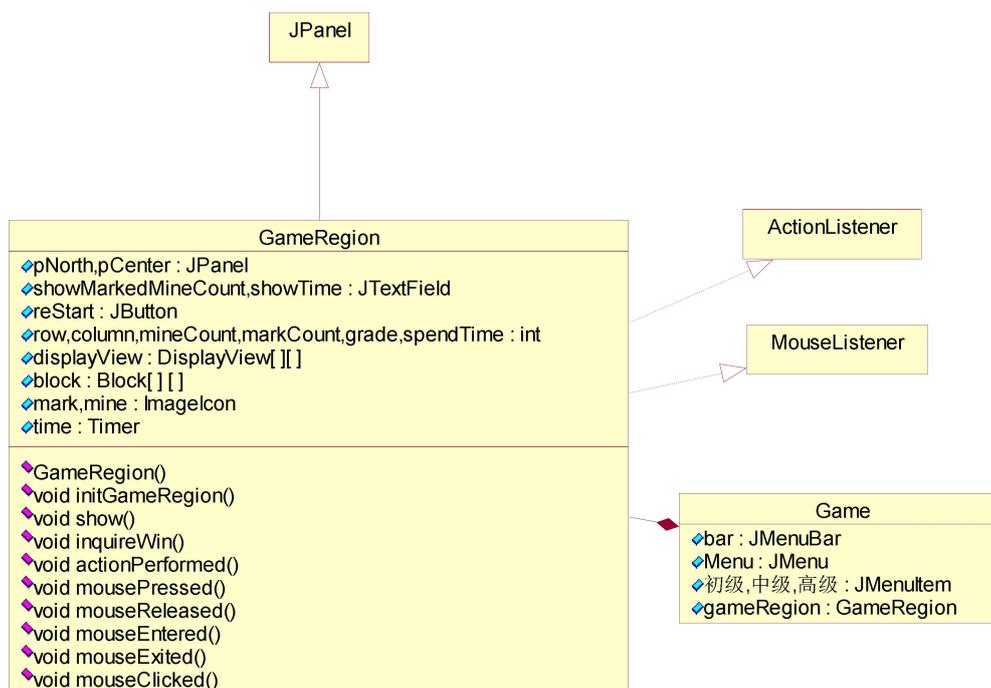


图 19 GameRegion 类的关联图

说明

GameRegion 类是 javax.swing 包中 JPanel 类容器的子类, 并实现了 ActionListener 和 MouseListener 接口, 所创建的对象: gameRegion 是 Game 类中最重要的成员之一, 它作为一个容器被添加到 Game 窗口的中心。

4.2.3 Block 类

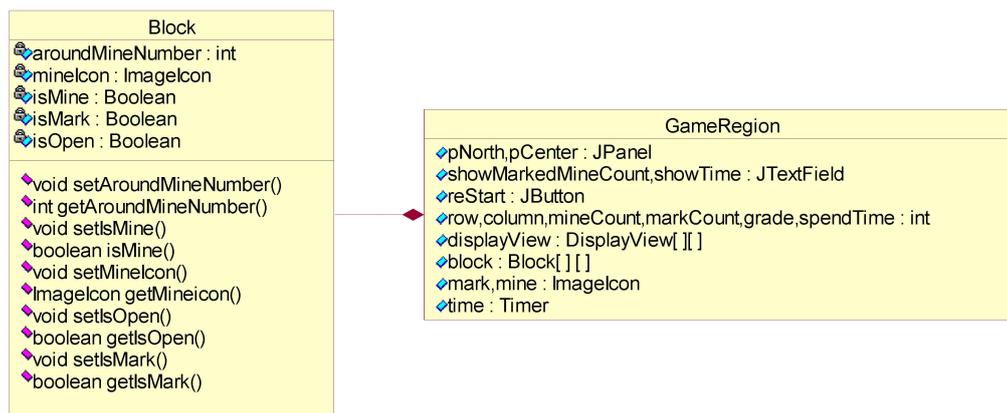


图 20 Block 类的关联图

说明

Block 类创建的对象是二维数组 block 对象单元元素中的一个对象。Block 类型的二维数组 block 是 GameRegion 类的重要成员之一。

4.2.4 DisplayView 类

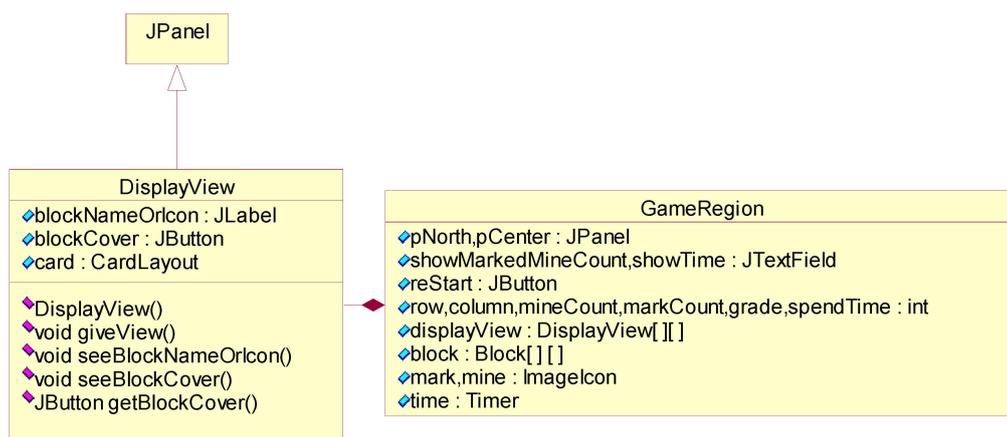


图 21 DisplayView 类的关联图

说明

DisplayView 类是 javax.swing 包中 JPanel 类的一个子类, 创建的对象是二维数组 blockView 的单元中的对象。DisplayView 类型的二维数组 blockView 是 GameRegion 类的重要成员之一。

4.2.5 PostMines 类



图 22 PostMines 类的关联图

说明

PostMines 创建的对象 post 是 GameRegion 中的重要成员。

4.2.6 程序初始化（或者点击游戏菜单中选项）的各类关联图

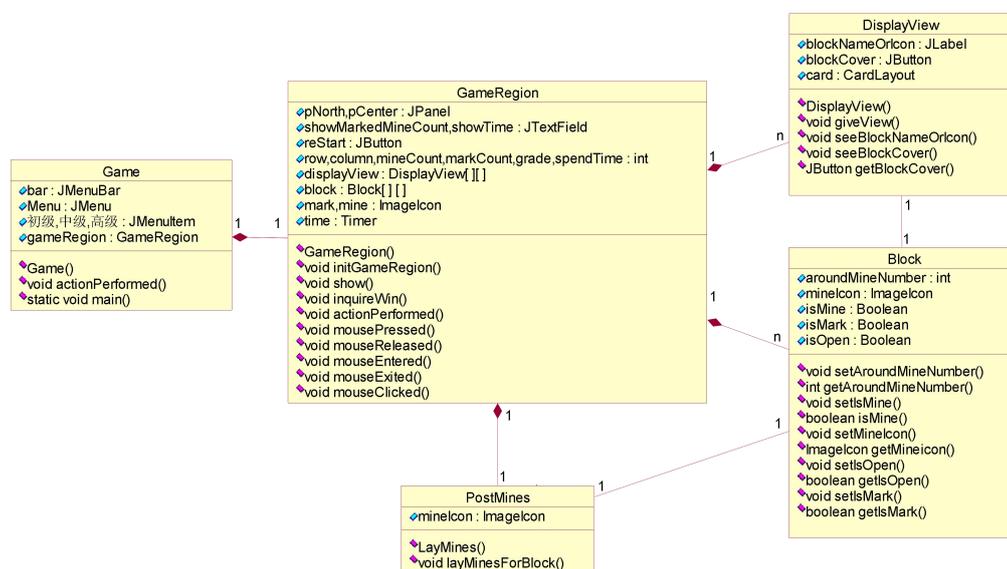


图 23 程序初始的各类关联图

说明

扫雷游戏刚开始打开运行时，需进行初始化工作，初始化涉及到以上 5 个实体类，Game 包含 GameRegion 类的对象，GameRegion 类中含有 Block 类、DisplayView 类和 PostMines 类的对象，而 PostMines 类和 DisplayView 类与 Block 类的关系是 1 对 1 的关联关系。

4.2.7 单击重新开始行为的各类关联图

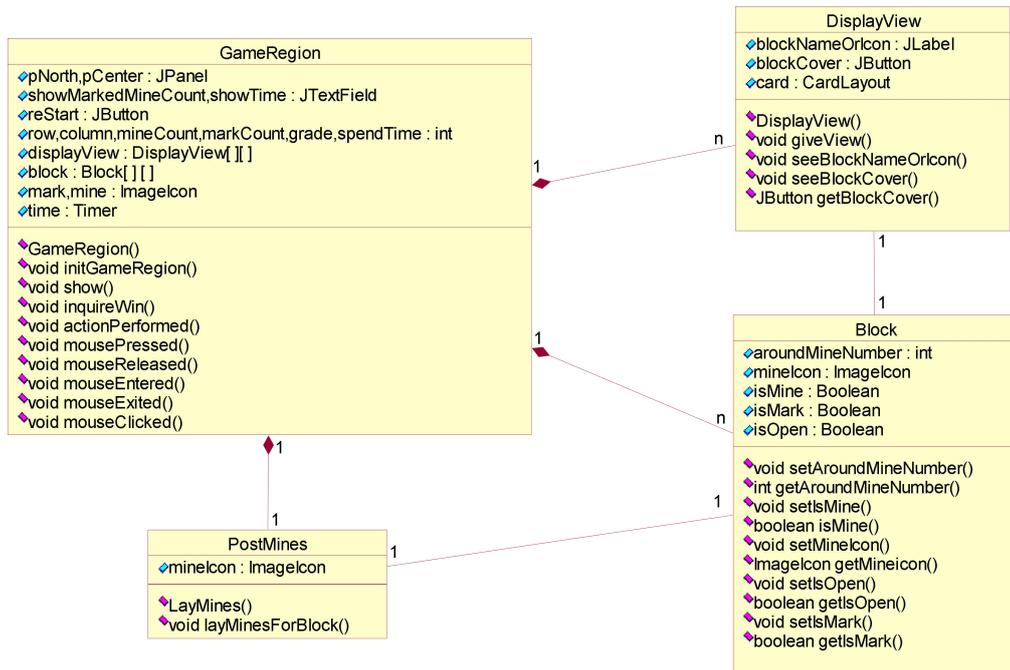


图 24 单击重新开始行为的各类关联图

说明

扫雷游戏初始化完成后，若用户点击重新开始按钮，将涉及到以上 4 个实体类，GameRegion 类中含有 Block 类、DisplayView 类和 PostMines 类的对象，而 PostMines 类和 DisplayView 类与 Block 类的关系是 1 对 1 的关联关系。

4.2.8 单击（左击或右击）雷区方块行为的各类关联图

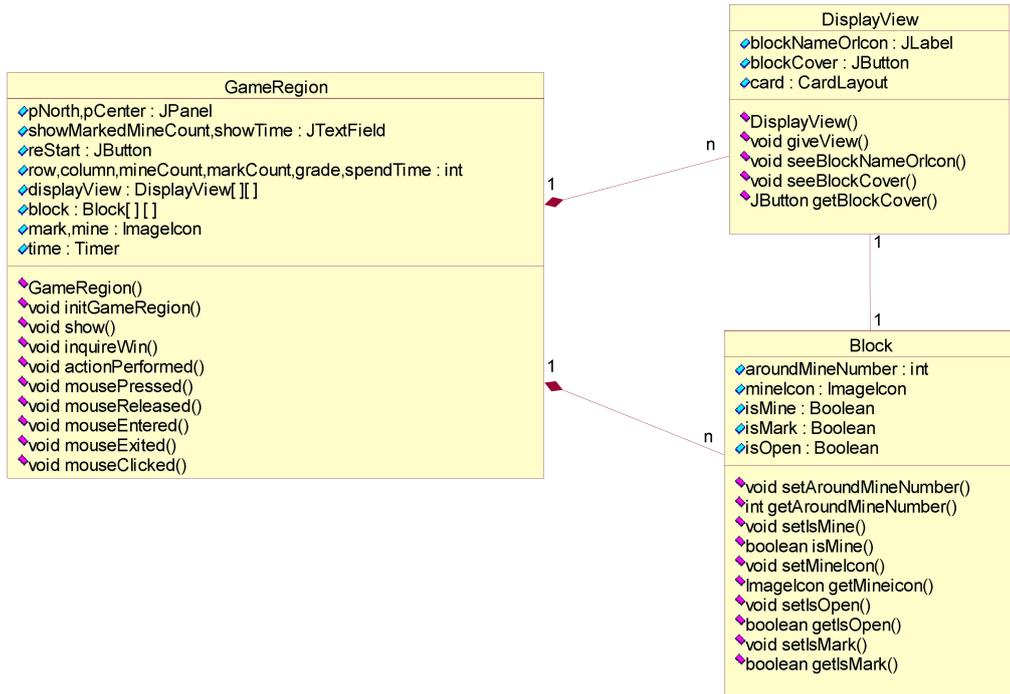


图 25 单击雷区方块行为的各类关联图

说明

当用户单击（左击或者右击）雷区域的方块时，将涉及到以上 3 个实体类，GameRegion 类中含有 Block 类、和 DisplayView 类对象，而 DisplayView 类和 Block 类是 1 对 1 的关联关系。

4.3 交互图（时序图、协作图）

4.3.1 程序初始化（初次运行）序列图和协作图

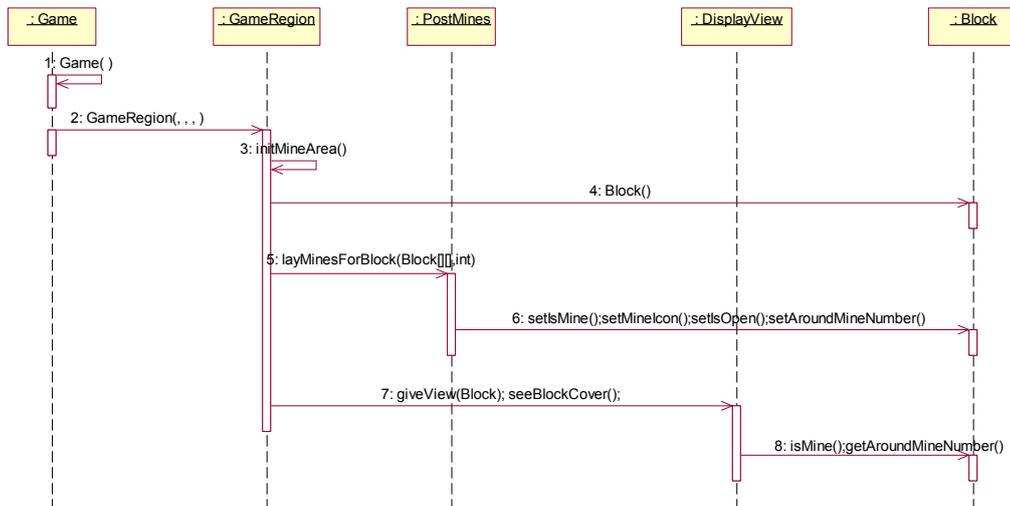


图 26 程序初始化序列图

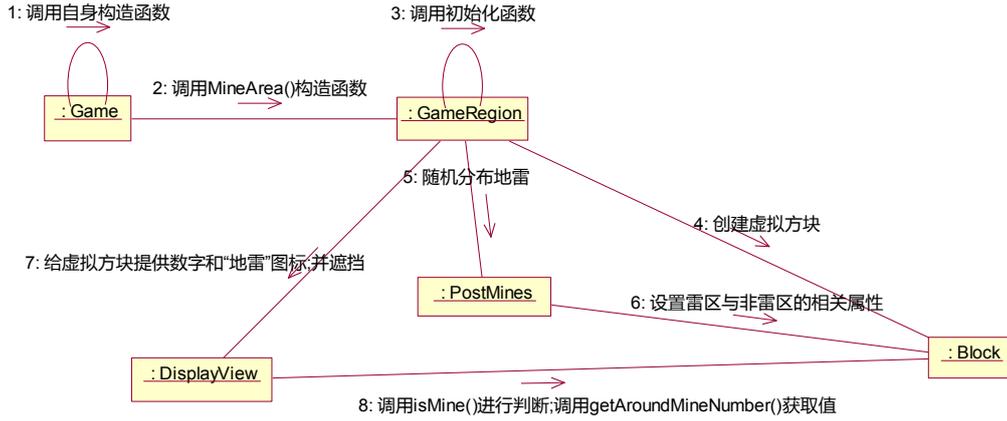


图 27 程序初始化协作图

说明

以上序列图和协作图显示了程序初始化所进行各类之间的方法调用，从而完成程序一开始的初始化工作，即用户在启动游戏时，未对游戏进行任何点击操作时的状态，便是初始状态，初始状态确认后，程序会等待用户的操作，从而触发相对应的事件。初始化工作相关类之间的时序及协作之间的关系，如上图所示。

4.3.2 用户选择难度序列图和协作图

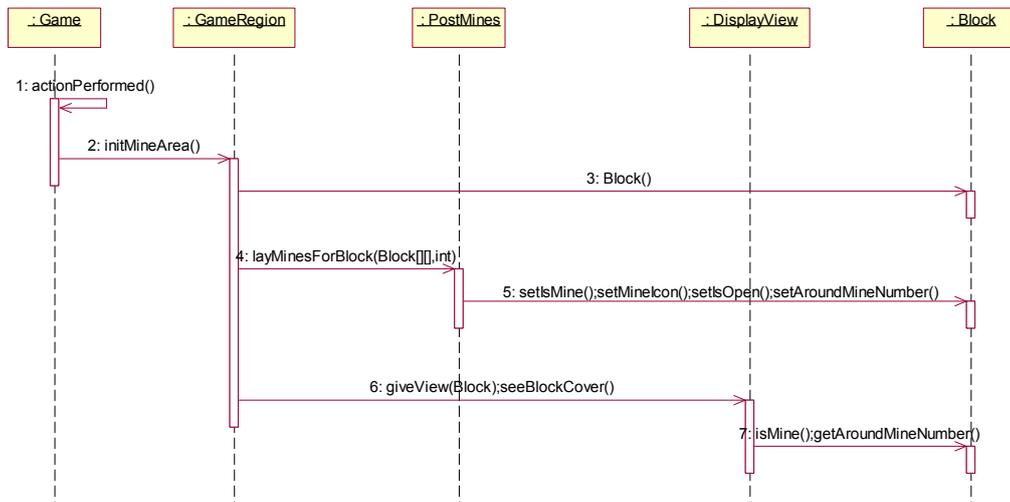


图 28 用户选择难度序列图

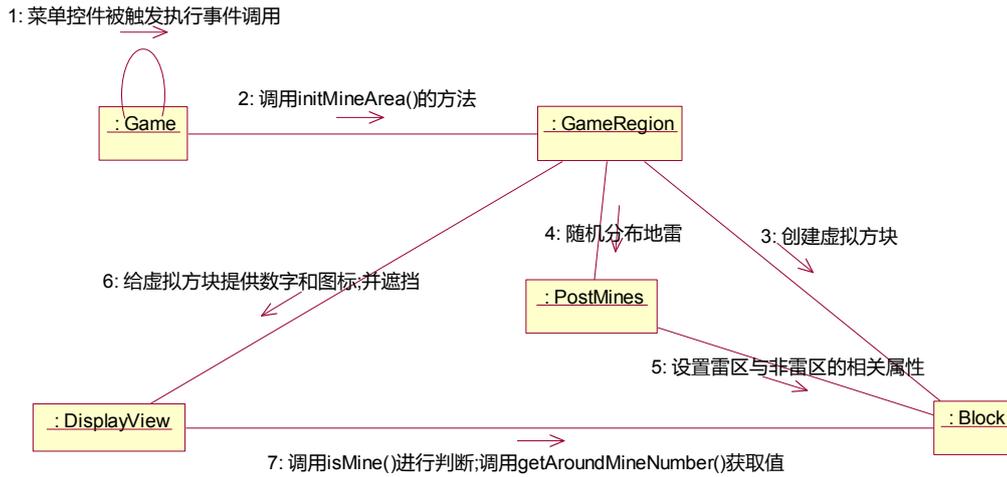


图 29 用户选择难度协作图

说明

初始化的工作完成之后，用户如果不想在默认初级的难度系数开始游戏，便可以在游戏菜单项中进行难度系数的选择，选择初级、中级或者高级这三者中任意一个难度系数，在点击任一难度系数时相应的事件被触发。被触发时各个类之间的时序及协作的关系，如上图所示。

4.3.3 用户点击重新开始按钮的时序图和协作图

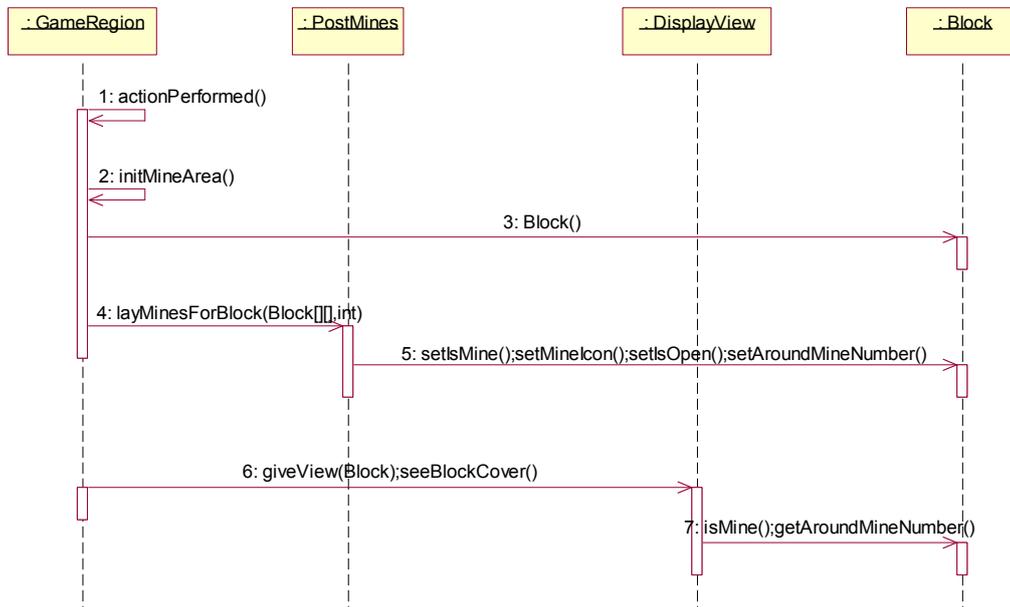


图 30 用户点击重新开始按钮的时序图

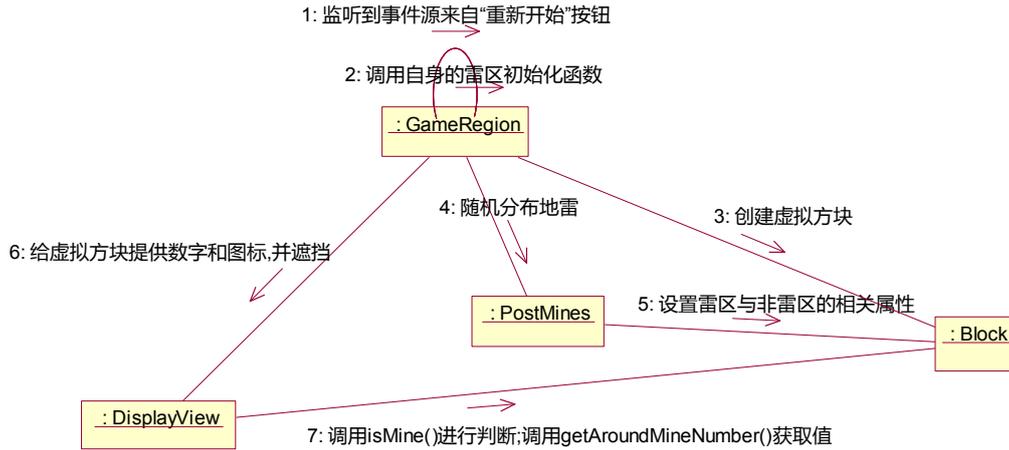


图 31 用户点击重新开始按钮的协作图

说明

在程序初始化完成后，用户可能分为几种情况用户再次点击上方面板中间的“重新开始”按钮：1、用户在开始游戏中，进行点击时便踩中了地雷，游戏结束，用户想重新开始游戏便点击重新开始按钮；2、用户在扫雷的过程中，用户所点击的方块所给出的数显使用户很难判断周围地雷的分布情况，用户想重新开始游戏，便点击重新开始按钮，重新获取随机分布的地雷序列；3、用户在扫雷结束后，最终获得了胜利，也不想菜单栏中选择其他的难度级别，在弹出胜出模式对话框点击确认后，再点击“重新开始”按钮便可以方便地重新开始当前难度级别的游戏。重新开始被监听到并执行时，各个类之间的时序和协作的关系，如上图所示。

4.3.4 用户左键点击雷区方块（按钮）的时序图和协作图

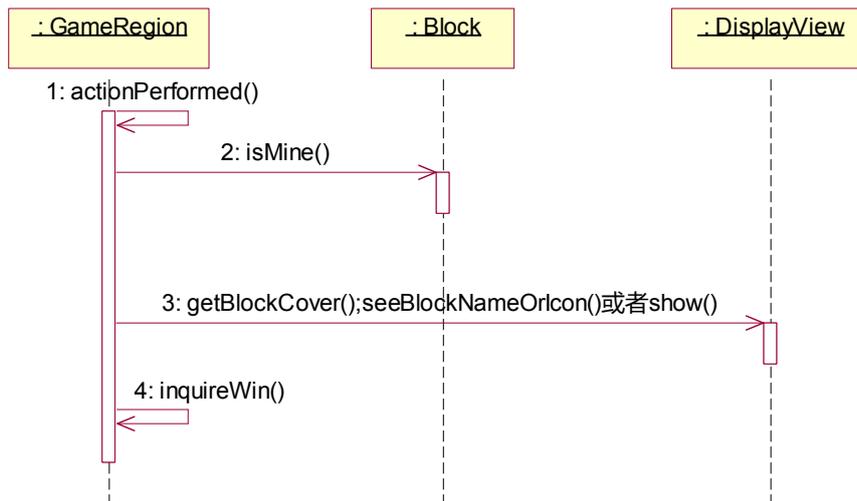


图 32 用户左键点击雷区方块的时序图

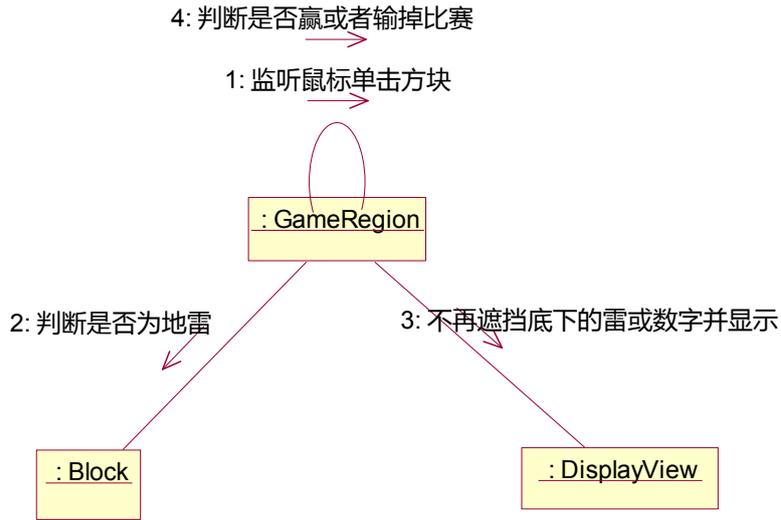


图 33 用户左键点击雷区方块的协作图

说明

程序初始化完成后，或者在用户选择完适当的难度系数后，单击雷区的方块，进入游戏时，每左击一次，便产生一次以上的类信息传递的关系，直至踩中地雷或成功完成扫雷游戏，类 GameRegion 中的 actionPerformed() 方法逻辑详见下面的程序流程图。用户左键单击雷区方块时，各个类之间的时序和协作的关系，如上图所示。

4.3.5 用户右键点击雷区方块（按钮）的时序图和协作图

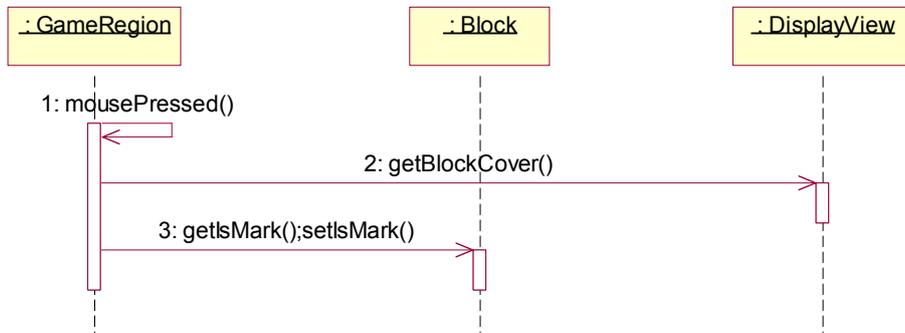


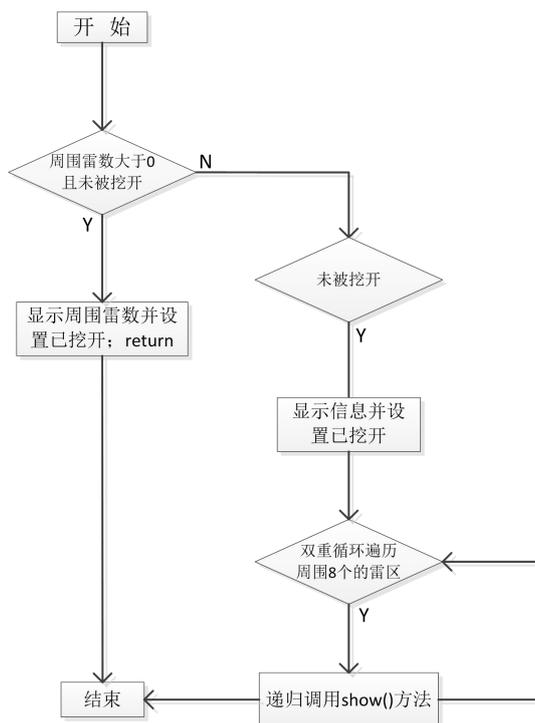
图 34 用户右键点击雷区方块的时序图



图 35 用户右键点击雷区方块的协作图

说明

程序初始化完毕后，当用户开始进行游戏时，用户可以对可能是“地雷”的方块进行标记，当对已标记的方块进行取消标记时，可以再次右键点击此方块方可取消。`GameRegion` 类中的 `MousePressed()` 方法逻辑详见下面的程序流程图。用户右击雷区方块时，各个类的时序和协作的关系，如上图所示。

4.4 程序流程图**4.4.1 类 `GameRegion` 中的 `show()` 方法的程序流程图**图 36 类 `GameRegion` 中的 `show()` 方法的程序流程图**说明**

函数 `show()` 供本类中 `GameRegion` 中 `actionPerformed(ActionEvent e)` 调用，若判断此方块周围雷数大于 0 且未被挖开，则显示周围雷数并退出本函数；否则周围方块若未挖开，则分别遍历此方块的上下左右斜共 8 个方块，并循环递归调用 `show()` 方法，直至周围雷数大于 0 需要被显示或者都被挖开为止。

4.4.2 类 GameRegion 中的 actionPerformed(AcationEvent e)方法的程序流程图

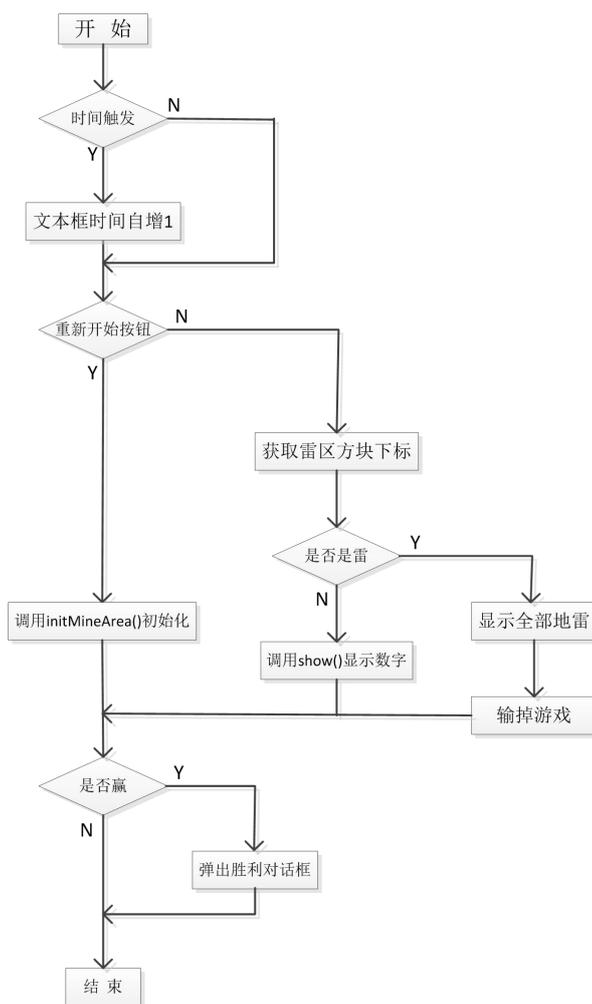


图 37 类 GameRegion 中的 actionPerformed(AcationEvent e)方法的程序流程图

说明

首先此监听方法捕捉到，判断是否为时间触发，若为真，则将时间文本框中的数值自增 1；其次判断是否为用户用鼠标左键点击雷区方块，若左键单击了此方块，则判断是否在雷区按下的左键：若是雷区按下的，则利用双重循环定位方块的位置，再判断是否为雷，如果按下的是雷，则显示全部地雷并结束比赛；否则调用此类的 show()方法显示数字；若不是鼠标左键点击的方块，则就为重新开始按钮所触发的，则调本类的 initGameRegion()方法进行初始化操作，否则不执行任何语句。最后调用本类的 inquireWin()方法判断是否赢得最终的比赛，若赢得比赛，则弹出胜利的提示对话框，否则不执行任何语句，最终结束此监听方法。

4.4.3 类 GameRegion 中的 mousePressed(MouseEvent e)方法的程序流程图

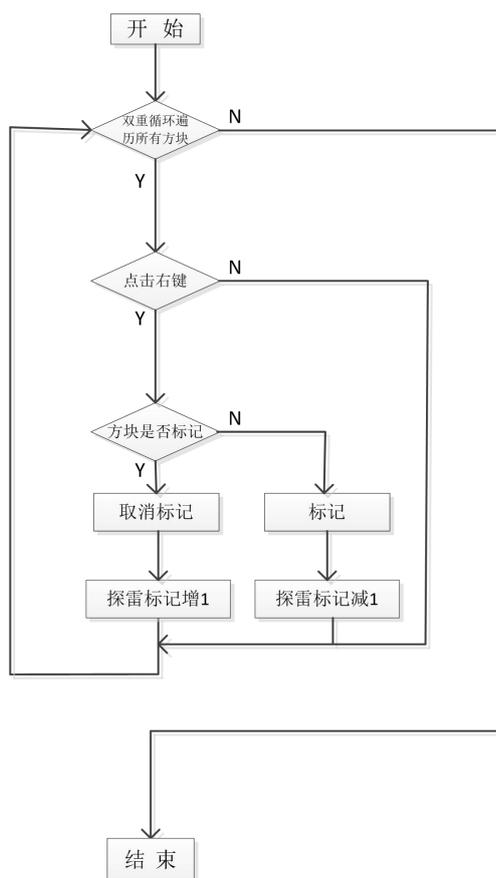


图 38 类 GameRegion 中的 mousePressed(MouseEvent e)方法的程序流程图

说明

此监听按钮的方法捕捉用户是否用鼠标点击雷区的方块，当捕捉到用户右键按下鼠标，利用双重循环，双重循环中进行判断哪个方块被右击了，若判断出此方块被右击，则再判断方块是否被标记，若被标记，则取消标记并修改上方面板中文本框中的数字；否则标记并修改上方面板中文本框中的数字；若判断出此方块未被右击，则不执行任何语句，再次转向循环。双重循环遍历所有的方块后结束。

第五章 软件测试

5.1 类测试

5.1.1 Game 窗口

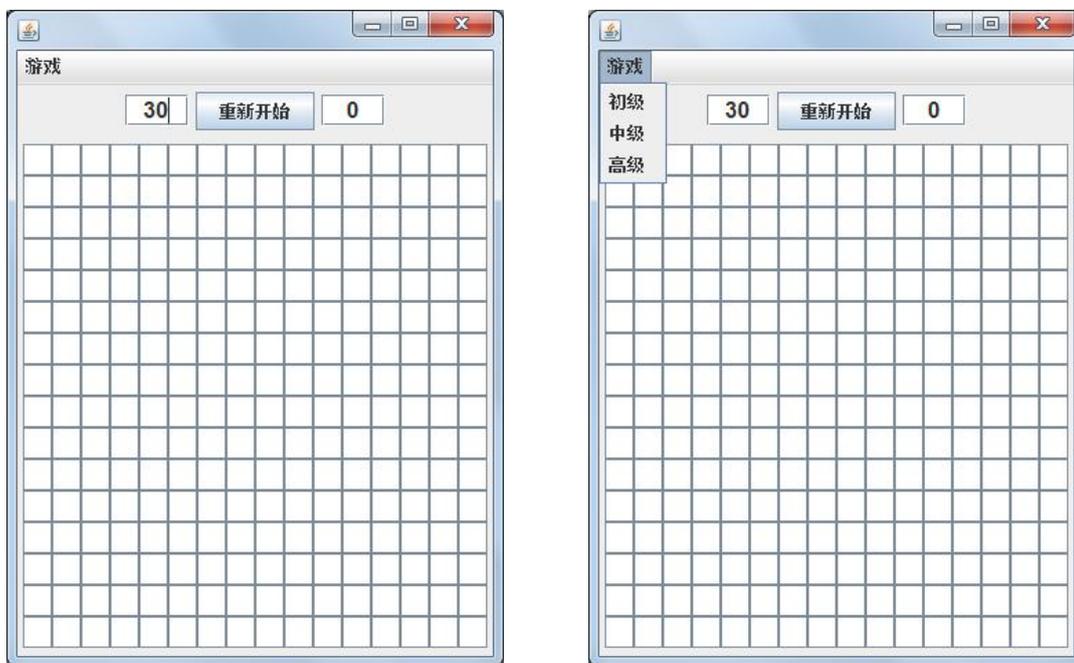


图 39 Game 类创建的窗口效果图

5.1.2 GameRegion 创建的扫雷区域

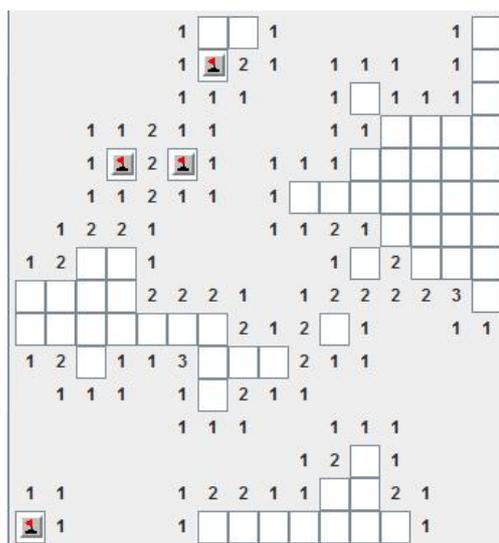


图 40 扫雷进行中的效果图

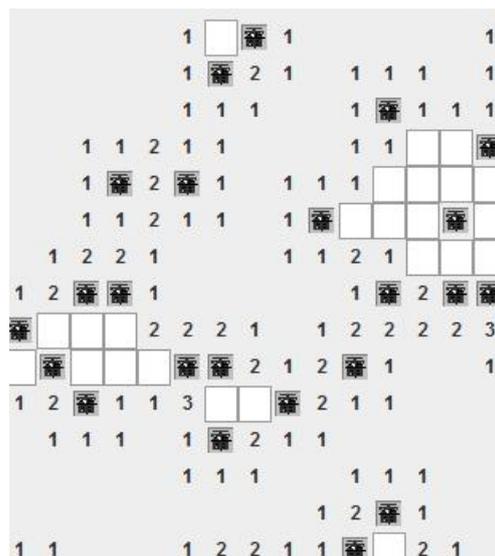


图 41 扫雷失败的效果图

5.1.3 DisplayView 类的效果图

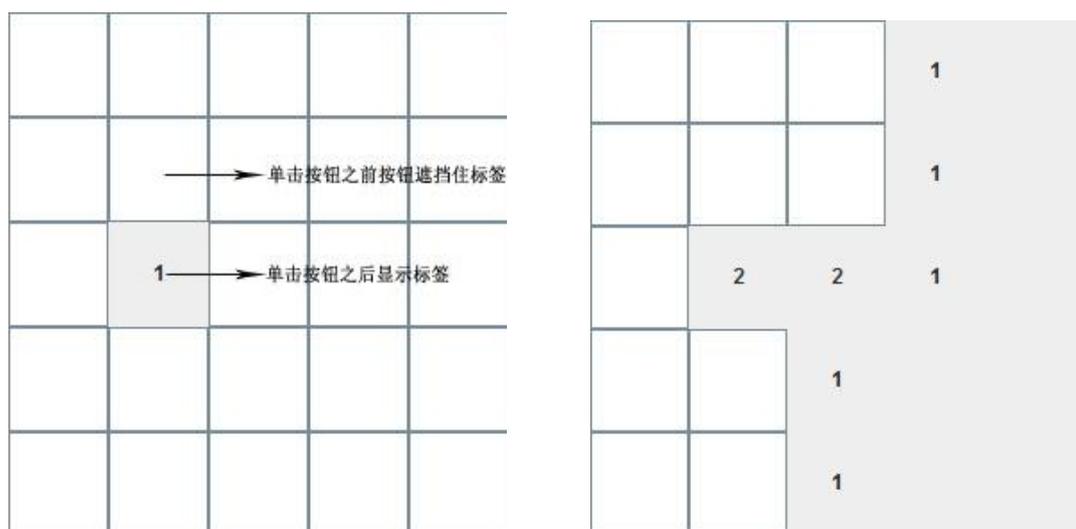


图 42 单击按钮前后的标签显示状况效果图

说明

通过以上的每个类单独的测试，可以得出各个类显示界面和基本功能已经实现。

5.2 游戏软件黑盒测试

5.2.1 不同游戏级别的界面

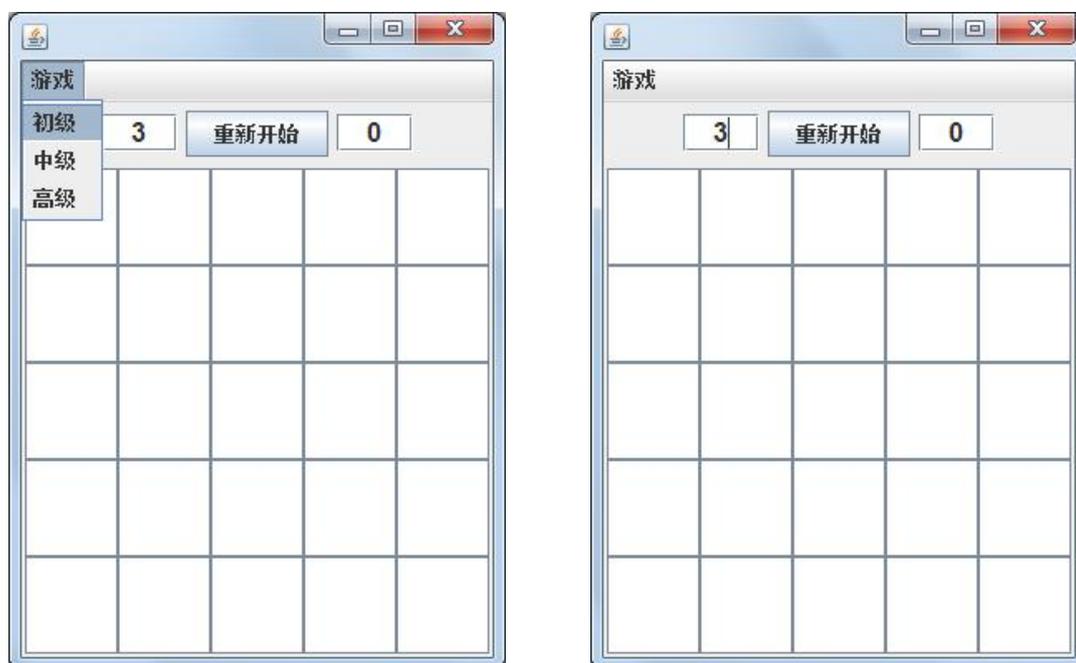


图 43 游戏的初级界面

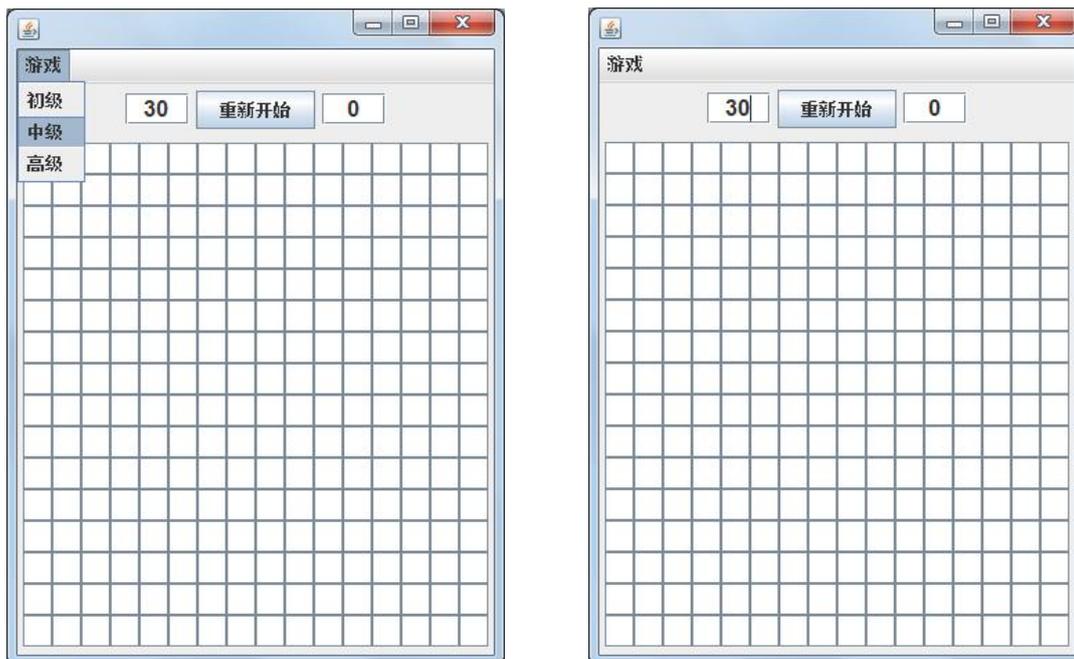


图 44 游戏的中级界面

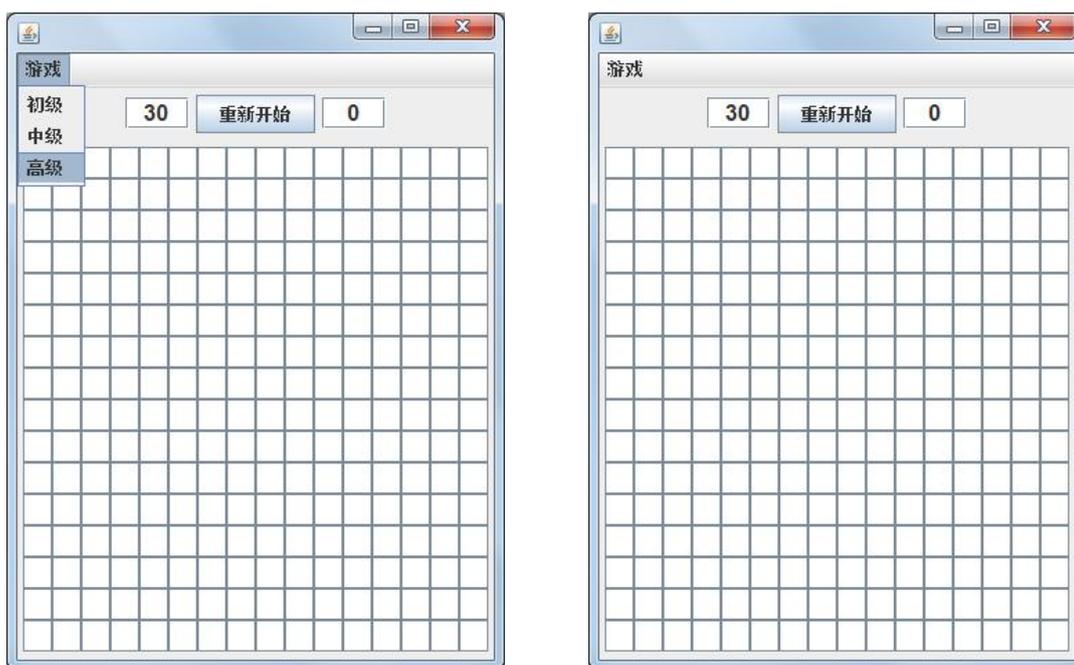


图 45 游戏的高级界面

5.2.2 游戏进行时的界面



图 46 游戏进行中的界面

5.2.3 游戏失败时的界面

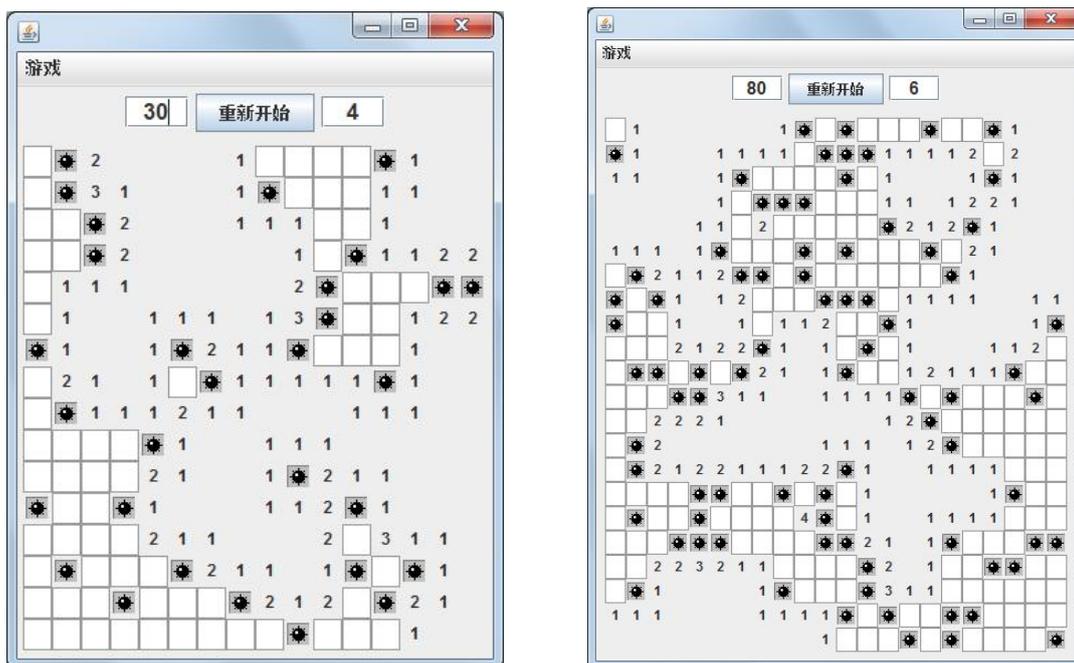


图 47 游戏失败时的界面

5.2.4 游戏胜利时的界面



图 48 扫雷成功的效果图

说明

通过以上的黑盒测试，并没有发现较大的逻辑性错误，可以得知游戏软件已经基本可以运行，由于开发时间和能力的限制，功能和界面还有待完善，还有部分缺陷还有待继续修正。后面结束语中已经总结出了在测试过程中所存在的缺陷。

第六章 结束语

6.1 游戏特点

1、由于本游戏实现的功能较为简单，所以游戏的大小非常之小，最终制作 JAR 文件发布编写软件的大小仅有 9.71KB，远远小于 Windows 自带的扫雷游戏的大小。

2、本游戏只适应于初学者，对功能及界面要求较高的用户难以满足需求，所以此游戏还需不断地完善，使其功能界面更加友好。

3、本游戏可以在任何一个安装了 Java 运行环境（版本号需高于 1.5 的计算机上）的系统上运行，运行游戏前需将扫雷时所使用到的“探雷”标记图像和“地雷”标记图像与*.jar 文件放在一个目录中，只有这样，用户才能在游戏过程的界面中通过左击右击使用相应的图标。

6.2 游戏局限

1、由于开发与能力有限。游戏实现的功能较少，较 Windows 自带的扫雷游戏的功能相差较远，我仅仅实现了游戏菜单选项中的初级、中级和高级三个按钮的功能。而 Windows 自带的扫雷游戏不仅含有以上的功能，还含有自定义、扫雷英雄榜等友好的功能界面，所以此扫雷游戏在功能和界面方面还需不断完善。

2、性能缺陷：1、当鼠标右键在雷区方块则标记“探雷”标记时，正常情况下（Windows 自带的扫雷游戏中不可再左击），而目前自己编写的扫雷，用户仍可左键单击方块；2、输掉游戏时，正常情况下，不可再右击标记方块，而目前自己编写的扫雷用户仍可在非地雷的方块上使用“探雷”标记。

3、界面不足：可视化界面与 Windows 中的扫雷游戏相差甚远，Windows 中的扫雷游戏具有立体效果，给用户呈现一种立体感，而自己编写的扫雷是平面效果，给用户的体验相对较差。Windows 自带的扫雷游戏清晰友好的用户体验，而自己编写的扫雷游戏界面并非十分友好。

6.3 个人总结

从开始接触到课题到现在，在这期间也经历了很多事情。但是值得高兴的是，通过自己的努力，灵活运用所学的专业知识，毕业设计最终可以运行起来，论文也在不断地完善当中。毕业设计课题—《扫雷游戏设计》应该说比较顺利地实现了最基本的功能。它是一个功能简单、界面简洁的一款单击智力小游戏。

扫雷游戏是一个比较有意义的娱乐游戏，在人们忙碌中工作之余，为广大使用者提供消遣，缓解人们的工作压力，此游戏还是一款智力小游戏，更加有利于开发小孩子的智力，使小孩思考问题的能力更强。

此扫雷游戏开发过程中采用的是 Java 语言，Java 语言是一种跨平台的面向对象编程语言，Java 语言提供的处理图形用户界面的包 javax.swing 包，此包更加方便用户对图形图像的处理，使用户便于实现了更加强大的图形处理功能。

在毕业设计刚开始设计的时候，对整个游戏的设计并不是很了解，只能抱着一些与扫雷游戏相关的参考书籍看。不过在指导老师的仔细帮助下，我渐渐地对该扫雷游戏设计的一些基本的功能需求有了浅浅的认识，与此同时，我也在网上查阅了很多资料。到后来慢慢地开始了解程序的设计过程，并在已有参考代码的书目中设计并修改了一些简单的模块，在更多同学的帮助下，最终完成了毕业设计。可以说这么说，在整个课题的设计过程中我都在不断地进步，同时也学到了很多，进一步加深了对所学的专业知识的认识，巩固了所学的专业知识，提高了软件开发的技术水平，为今后的工作与进一步发展打

下了坚实的基础。但是由于设计的时间与个人能力有限，许多方面还存在缺陷，许多功能还需要进一步的改进和完善。游戏的局限部分已经在上一小标题中列出，还需通过学习更多的知识得以完善该游戏。

致 谢

经历了近5个多月的毕业设计的论文撰写工作之后，便是临近结尾的时候，这一刻既是毕业论文的结束，还是大学时代四年生活的即将结束！

回顾自己近几个月毕业设计的时光，从寒假中就开始逐渐开始构思自己的毕业设计，看着别人都已经准备的比较好，都已经有了基本的思路，但自己心中却还没有任何基本的想法。感谢在王征老师的催促下自己通过看书完成了一些简单的任务。

首先感谢王征老师在整个毕业设计过程中对我的督促与帮助！王老师是一个工作负责、为人慷慨的一位老师。在做毕业设计期间，他定期检查我毕业设计完成的情况，有时甚至我深夜问他问题，他都能及时给我答复并给我一些建议，这对我完成毕业设计有很大的帮助。我还要对计算机科学与技术学院的赵勃悖老师表示由衷的感谢，由于我的毕业设计导师是东南大学的，面对面交流可能有点困难，在此期间，赵老师也帮助了我不少忙，向我推荐书籍并给我详细的讲解，他在专业知识上也给予了我很大的指导，为我整个课题的开发给予了很大的帮助。还要感谢帮助过完成毕业设计的同学们，在遇到困难，是他们陪我找出问题，然后大家再共同解决问题，才保证了毕业设计的如期完成。

同时，特别感谢大学老师们四年来对我的谆谆教诲，使我理解了计算机科学与技术这门学科，并在我今后的进一步深造学习中起着不可磨灭的作用。最后，我要感谢我的母校—东南大学成贤学院，是它为我提供了良好的学习环境，在这里结识了一起学习计算机的同学们，很高兴能在这里度过自己的本科生活。

参考文献 (References)

- [01]陈国君. Java 程序设计基础 (第 3 版) [M]. 北京: 清华大学出版社, 2011.
- [02]谭浩强. C 语言程序设计 (第 2 版) [M]. 北京: 清华大学出版社, 2008.
- [03]王育坚. Visual C++面向对象编程教程 (第 2 版) [M]. 北京: 清华大学出版社, 2007.
- [04]耿祥义、张跃平. Java 课程设计 (第二版) [M]. 北京: 清华大学出版社, 2008.
- [05]陶华亭. 软件工程实用教程 (第 2 版) [M]. 北京: 清华大学出版社, 2012.
- [06]卫红春、朱欣娟. 信息系统分析与设计 (第二版) [M]. 西安: 西安电子科技大学出版社, 2008.
- [07]Graig Larman. UML 和模式应用[M]. 北京: 机械工业出版社, 2002.
- [08]李文书. 数据结构与算法应用实践教程[M]. 北京: 北京大学出版社, 2012.
- [09]狄国强、杨小平、杜宾. 软件工程实验[M]. 北京: 清华大学出版社, 2008.
- [10]刘腾红. 项目管理[M]. 东软电子出版社, 2012.