分类号\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 密级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

UDC \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 学号

毕业设计（论文）

|  |  |
| --- | --- |
| 论文题目 | 基于Java的连连看游戏设计与实现 |
| **Thesis Topic** | Design and Implementation of Java-Based Link-Game |

2017年 6月5日

毕业设计（论文）任务书

第1页

|  |
| --- |
| 毕业设计（论文）题目：  基于Java的连连看游戏设计与实现  毕业设计（论文）要求及原始数据（资料）：  1．简述Java游戏开发现状和走向；  2．了解图形用户界面的概念；  3．熟练掌握Java的事件处理机制；  4．掌握Java小程序的工作原理和使用方法；  5．了解AWT的基本体系结构，掌握窗口的基本原理；  6. 掌握几种布局管理器的使用方法；  7. 了解Swing的基本体系结构，掌握Swing组件的使用方法；  8．训练检索文献资料和利用文献资料的能力；  9．训练撰写技术文档与学位论文的能力。 |

第2页

|  |
| --- |
| 毕业设计（论文）主要内容：  1．简述Java游戏开发现状和选题背景；  2．了解Java在游戏开发方面的相关技术与环境；  3．设计实现游戏计分功能：每消去成对图片时加10分；  4．设计实现游戏重置功能：玩家遇到困难难以消除下一对图片时，可按下”重置”按钮，图片重新排列，游戏重新开始；  5.设计实现游戏计时功能；  6．设计实现难度选择功能；  7．算法路径的实现。  学生应交出的设计文件（论文）：  1．内容完整、层次清晰、叙述流畅、排版规范的毕业设计论文；  2．包括毕业设计论文、源程序等内容在内的毕业设计电子文档及其它相关材料。 |

第3页

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 主要参考文献（资料）：   1. 张红旭. 浅谈Java的发展及特点[J]. 硅谷, 2010(6):50-50. 2. Burns B. Darkstar: the java game server[J]. 2007. 3. 吴越胜、李新磊.Eclipse 3.0程序开发技术详解[M]:清华大学出版社，2010.1~3 4. 岷江. Java的运行原理[J]. 开放系统世界, 2002(5):85-86. 5. 白亮. Java简介[J]. 厦门科技, 2005(1):27-29. 6. Piroumian V. Java GUI Development[C]:Sams, 1999. 7. 李钟蔚. Java开发实战宝典[M]. 清华大学出版社, 2010 8. 苏碧霞. 探析JAVA Swing工具包在图形用户界面设计中的应用[J]. 信息与电脑:理论版, 2013(5):143-144. 9. Hardy V J. Java 2D API Graphics[C]:Prentice Hall PTR, 2000. 10. 明日科技. Java从入门到精通[M]. 清华大学出版社, 2012. 11. 郑淑荣.Java的事件处理机制剖析.《办公自动化：综合月刊》，2011（10）：23-24. 12. 胡家芬. Swing可视化组件多线程操作机制研究[J]. 电脑知识与技术,2012(11):7481-7482. 13. 刘煜. 深入了解JAVA中的线程[J]. 中文信息, 2002(8):56-57. 14. 王继成, 高珍. 软件需求分析的研究[J]. 计算机工程与设计, 2002, 23(8):18-21. 15. 冯玲玲, 沈轶. 基于UML的需求分析与建模[J]. 科学技术与工程, 2005, 5(9):605-607. 16. Reenskaug T. The Model-View-Controller (MVC) Its Past and Present[J]. University of Oslo Draft, 2003. 17. 朱亚玲, 张睿敏. 基于java的连连看游戏设计与实现[J]. 数字技术与应用, 2016(3):161-162.  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 专业班级 |  | | | | | 学生 |  | | 要求设计（论文）工作起止日期 | | | | |  | | | | 指导教师签字 | |  | | | | 日期 |  | | 教研室主任审查签字 | | | |  | | 日期 |  | | 系主任批准签字 | | |  | | | 日期 |  | |

基于Java的连连看游戏设计与实现

摘 要

连连看是一种消除类益智游戏，核心要求是在规定的时间内，消除游戏界面中选中的两张相同的图案，直至完全消除所有图案。这款游戏操作简单(只需单击鼠标左键操作)、面向人群广泛，在限时操作的游戏要求下，吸引玩家自发地锻炼观察能力、判断能力和反应能力，故从推出至今颇受欢迎。消除类游戏如今也有很多类型，但连连看游戏因其操作简单依旧广受好评，该游戏不仅能调节精神状态，还能在游戏中锻炼我们的眼力视野和集中力，开发思维能力的同时培养面对困难时具备攻克困难的信心。

此次设计所使用的语言为Java语言，开发环境为JDK Version1.7.0，开发工具为Eclipse。主要应用Swing组件，Java小程序，事件及接口等。本系统主要是运用Swing组件和事件处理技术，进行界面的绘制，鼠标的操作，以及积分的显示和沙漏的计时的显示等功能。

关键词：游戏；连连看；二维游戏

**Design and Implementation of Java-Based Link-Game**

Abstract

Link-Gameis a kind of tile-matching puzzle games,the core requirement is that within the prescribed period eliminating two selected pictures which clicked on the before and after and had the same pattern until completely eliminating all of the same two.There are numerous reasons why this game is popular since launch as follow.The operation of the game is simple (just need clicking the mouse left key)and facing the crowd is widespread.But also in the limited time,game players have to victory attracts people exercise spontaneously observation ability,judgment ability and the ability to respond to.Now this kind of puzzle games also has a lot of types,but Link-Game because of its simple operation is still popular,not only can the game adjust the mental state,but also exercise our eyesight and concentrated force by enjoying the game.Furthermore,it develops the thinking ability.Meanwhile,it cultivates the confidence to overcome difficulties when we face the difficulties.

This design uses Java language,and it’s development environment is the Version 1.7.0 JDK on eclipse,the development tool.This system mainly use Swing components,Java applet,event and interface.By using event handling technology and Swing technology realizes the GUI,the mouse operation,showing the game credits as well as the function that timing as the hourglass.

**Keywords**: Games; Link-game; Two-dimensional Game

目录

[摘 要 1](#_Toc484463928)

[Abstract 2](#_Toc484463929)

[目录 2](#_Toc484463930)

[1. 绪论 1](#_Toc484463931)

[1.1. 课题背景 1](#_Toc484463932)

[1.2. Java游戏开发的现状 1](#_Toc484463933)

[1.2.1. Java日益趋向成熟 1](#_Toc484463934)

[1.2.2. Java极大的方便了游戏脚本的使用 2](#_Toc484463935)

[1.2.3. Java平台无关特性 2](#_Toc484463936)

[1.2.4. Java在手机游戏开发方面的优势 2](#_Toc484463937)

[1.2.5. Java在网络方面的良好表现 2](#_Toc484463938)

[1.2.6. 游戏速度 3](#_Toc484463939)

[1.3. 项目开发的目标 3](#_Toc484463940)

[1.4. 项目开发的意义 3](#_Toc484463941)

[1.5. 本文章节安排 4](#_Toc484463942)

[1.6. 本章小结 4](#_Toc484463943)

[2. 开发环境和相关技术简介 5](#_Toc484463944)

[2.1. 开发环境 5](#_Toc484463945)

[2.1.1. Eclipse 5](#_Toc484463946)

[Java 6](#_Toc484463947)

[2.1.2. 6](#_Toc484463949)

[2.2. 相关技术简介 7](#_Toc484463950)

[2.2.1. Eclipse的图形界面开发 8](#_Toc484463951)

[2.2.2. Java技术在游戏开发中的应用 8](#_Toc484463952)

[2.3. 本章小结 12](#_Toc484463953)

[3. 需求分析 13](#_Toc484463954)

[3.1. 可行性分析 13](#_Toc484463955)

[3.1.1. 技术可行性 13](#_Toc484463956)

[3.1.2. 经济可行性 13](#_Toc484463958)

[3.1.3. 社会可行性 14](#_Toc484463961)

[3.1.4. 操作可行性 14](#_Toc484463962)

[3.2. 连连看游戏需求描述及规则描述 14](#_Toc484463963)

[3.3. 系统需求分析 14](#_Toc484463964)

[3.3.1. 业务需求 15](#_Toc484463965)

[3.3.2. 用户需求 15](#_Toc484463966)

[3.3.3. 功能需求 16](#_Toc484463967)

[3.3.4. 非功能需求 17](#_Toc484463968)

[3.4. 本章小结 18](#_Toc484463969)

[4. 概要设计 19](#_Toc484463970)

[4.1. 模块设计 19](#_Toc484463971)

[4.1.1. 连连看系统总体结构 19](#_Toc484463972)

[4.1.2. 连连看系统模各模块介绍 19](#_Toc484463973)

[4.1.3. 游戏界面模块 20](#_Toc484463974)

[4.1.4. 处理模块 20](#_Toc484463975)

[4.2. 确定类 20](#_Toc484463976)

[4.3. 游戏素材 22](#_Toc484463977)

[4.3.1. 背景图片 22](#_Toc484463978)

[4.3.2. 游戏图片 22](#_Toc484463979)

[4.4. 本章小结 23](#_Toc484463980)

[5. 详细设计和实现 24](#_Toc484463981)

[5.1. 界面设计 24](#_Toc484463982)

[5.1.1. 界面实现 24](#_Toc484463983)

[5.1.2. 倒计时沙漏的设计 26](#_Toc484463984)

[5.2. 游戏执行的实现 28](#_Toc484463985)

[5.2.1. 点击图片的实现及其部分代码 29](#_Toc484463986)

[5.2.2. 记分、记关数的实现 30](#_Toc484463987)

[5.2.3. 刷新功能的实现 30](#_Toc484463988)

[5.2.4. 重新开始游戏功能的实现 32](#_Toc484463989)

[5.3. 游戏的连接算法 32](#_Toc484463990)

[5.3.1. 连连看连接方式的类型 32](#_Toc484463991)

[5.3.2. 连连看连接方式的分析 33](#_Toc484463992)

[5.3.3. 部分核心代码 35](#_Toc484463993)

[5.4. 本章小结 36](#_Toc484463994)

[6. 系统测试 37](#_Toc484463995)

[6.1. 软件测试目的及意义 37](#_Toc484463996)

[6.1.1. 测试目的 37](#_Toc484463997)

[6.1.2. 测试意义 38](#_Toc484463998)

[6.2. 测试方法分类 38](#_Toc484463999)

[6.3. 测试步骤 39](#_Toc484464000)

[6.4. 测试用例 40](#_Toc484464001)

[6.5. 测试结果 45](#_Toc484464002)

[6.6. 本章小结 45](#_Toc484464003)

[结 论 46](#_Toc484464004)

[致 谢 47](#_Toc484464005)

[参考文献 48](#_Toc484464006)

1. 绪论

以互联网产业为代表的中国信息产业蓬勃发展，并已成为国民经济和社会发展的重要组成部分。电子游戏产业作为电子技术的产物，涉及到电子游戏的开发、市场营销和销售等方方面面的经济领域，并且很快在全世界的娱乐领域占据主流位置。在社交网络飞速发展的影响下，电子游戏社交化助力推动电子游戏各个方面的发展，带来的经济效益也愈发庞大。网页游戏、单机游戏、网络游戏形形色色数量庞大，涌现出一批批颇具新意的游戏，为疲惫的生活带来轻松和愉悦，满足玩家的娱乐休闲要求。

* 1. 课题背景

连连看游戏顾名思义就是找出具有关联关系的事物并进行相应处理，经历了从桌面游戏、在线游戏再到社交游戏三个过程，形式多种多样。游戏的核心要求是将两个能用三根以内的直线连接的关联事物（相同事物）消除（或进行其他处理）。一般的连连看小游戏规则和操作都简单明了易上手，但也有很多新版本的连连看游戏加入了各类道具和游戏模式来丰富游戏内容，很好地增强了游戏的娱乐性，也带给玩家长期的挑战性和新鲜感。

对于90后大部分玩家来说，一定对网页游戏有很深刻的印象。最早，玩家们会在电脑课上接触到基于Flash的网页游戏。在线Flash游戏大都界面优良，操作简单，基本靠鼠标点击或者键盘方向键控制即可完成游戏。虽然当时网络游戏发展也很迅猛，但网页游戏仍旧依靠自身特点很快地吸引了一大批玩家。随着网络更多地服务于社交活动以及更多开放平台的兴起，很多游戏慢慢走向“社交化”，就是将游戏与个人空间结合，在社交网络和平台上主动或被动的通过玩家的“分享”快速地传播，应运而生出社交游戏。

无论是网页游戏还是网络小游戏，都拥有很多玩家受众，经典游戏也有很多，其中不乏有应用Java语言开发的经典游戏，比如俄罗斯方块、超级泡泡龙、贪吃蛇以及任天堂的超级马里奥等等，兼具游戏趣味和休闲益智的连连看便是这其中的佼佼者。

* 1. Java游戏开发的现状

提到游戏开发，更多地会提到C和C＋＋，这两个语言在游戏开发中有稳定和高效的表现。作为主流开发语言的Java是否也能在游戏开发方面展现良好的一面呢？从理论上讲，Java语言当然能被用来开发游戏，随着Java技术不断的更新发展，Jogl、Java3D等技术的出现，Java程序在显卡愈发强大的硬件加速基础上运行效果显著提升，开发出了许多基于Java的2D、3D游戏和大型的在线游戏。但Java作为开发语言开发游戏仍然被诟病，可以开发游戏不等同于适合开发游戏，因此，要讨论Java开发游戏的优势和意义。

* + 1. Java日益趋向成熟

Java语言从面世至今已过20载，它的发展也越来越符合人们对它的要求和需要。企业级应用的实践在某些方面可以看出该语言在大型软件开发方面颇为适用。当Java语言运行效率不再成为诟病时，相信Java语言跨平台、安全性、高抽象的优势在开发的大型游戏软件方面会有很好的表现。

* + 1. Java极大的方便了游戏脚本的使用

开发的过程往往是一个需要不断改进的过程，过程中会涉及许多方面的问题，游戏开发也不例外。所以改进的过程是否容易实现对于开发者来说也很重要，而游戏脚本就是为了改进的过程更容易实现。

如果将全部文件和数据写入程序，出现问题后要将整个程序重写编译发布，游戏是会不断地修改的，程序会越变越复杂。如果我们将可重复的数据写入外部文件，游戏引擎开发完毕通过读取外部文件运行的话，这样，当游戏出现问题时，修改外部文件就能解决问题，这样做极大地便利了游戏开发过程。Java本身就可以用来编写游戏脚本，目前也有例如groovy、beanshell等脚本语言可以无缝交互于Java语言，极大的方便了Java游戏编程。

* + 1. Java平台无关特性

提起Java语言特性，不得不提它的平台无关性。跨平台一直都是游戏开发面对的一个实际问题，Java在这方面做的相对于其他语言要好一些。当然，跨平台的特性可以使程序员减少工作量，公司减少投资，然而有些时候，“write once run everywhere”也需要付出一定的代价。在现实中，往往会出现这样的情况：对于Java applet来说，即使电脑在硬件上满足程序的要求，可能也无法让程序以同样良好的状态运行在每一台电脑上，除非使用者都是专业人士。利用Java跨平台特性开发软件在实现过程中有利也有弊，但Java在解决跨平台问题上比起其他技术要做的好一些。

* + 1. Java在手机游戏开发方面的优势

目前来看，游戏功能在手机功能中占据比重越来越大，手机游戏开发也将占据游戏开发市场更大的份额。手机搭载了各类操作系统也使Java语言因其跨平台的特性在游戏开发中颇受青睐。例如，安卓系统基于Java语言开发，安卓市场在手机市场所占比重很大，所以Java应用还是有一定的需求和不错的前景的。

* + 1. Java在网络方面的良好表现

Java在解决网络方面的问题上有不错的优势，体现在两个方面。一方面，基于applet和Java Web Start的游戏很容易更新，不需要用户手动下载新版本。另一方面，Java在网络交互的API相对友好。

在大型网络游戏开发中，分布式系统尤为重要，为了更好的推出硬件产品，Sun公司大力开发了用于网络服务器的Project DarkStar，DarkStar项目系统基础结构的设计方便了大型多玩家在线游戏的开发，并且有健壮的游戏运作能力，能动态的分配可用的服务器资源，还拥有性能改善功能，可以说，Project DarkStar的出现提高了Java在开发网络服务器方面的可能性。

* + 1. 游戏速度

基于Java的游戏执行效率方面一直颇受争议，在大型游戏应用方面的实例不多，更适用于中小型游戏开发。Java开发的游戏执行速度被很多程序员争议的地方在于Java程序运行时要启动Java虚拟机，具体表现如何，取决于开发项目的大小和软、硬件各方面的表现。与常用来开发游戏的C＋＋语言相比，Java开发出的游戏速度更慢，但Java开发的成本也相对较低。

* 1. 项目开发的目标

游戏软件的最直观的评判是它的操作界面，界面的好坏直接影响游戏玩家对这款游戏的接受程度和喜爱程度。对游戏评判的要求其次是功能齐全与否，功能越丰富，游戏体验的层次越丰富。在游戏操作简单的基础上丰富游戏内容，会使游戏变得更易普及。

针对连连看这款小游戏，本项目的开发目标是：游戏界面整洁美观，实现游戏的可操作性，使游戏操作简单易上手，并且在基础功能实现后增添一些理解简单、操作简单的功能用以丰富游戏内容。

实现连连看游戏具体要做到以下几点：游戏界面美观大方：通过有些界面要能对游戏操作过程有一定的理解；游戏具有一定的可玩性和吸引力：有一定内容，不局限于普通功能；游戏运行稳定可靠。

在设计与实现的过程中，力求对Java语言有新的认识和理解，对面向对象的概念和方法熟练应用和实践。在编写过程中，养成良好的编程习惯和良好的编程风格。在应用所学知识时，巩固理解Java程序设计的基本过程以及线程、网络、界面布局等编程技术。旨在培养分析问题的能力和解决问题的能力，着重培养设计分析、设计方法、设计操作与测试、观察设计过程和理解归纳问题的能力。

通过对游戏的开发，实现游戏的可操作性及美观性，进一步深化面向对象的程序设计思想，以及熟练运用Swing进行编程的能力，为将来的实际工作打下坚实的基础。

* 1. 项目开发的意义

项目开发的意义有两个层面。一方面，在项目开发过程中，开发人员通过将所学所感融入实际应用中，非常有利于学习和掌握开发过程，有利于进一步掌握Java面向对象的程序设计语言的基础知识和技能，有利于更直观地感受到面向过程的开发和面向对象的开发之间的不同点。

另一方面，项目开发往往能够很好地培养和锻炼个人专业技术能力和个人综合素质能力。本次项目开发，是开发人员和所有先辈们共同努力的结晶。在开发过程中，查阅资料，解读国内外优秀学子和老师的智慧结晶，扩展了开发人员解决问题的思路和看待问题的眼界，在实践过程中，感受开发的乐趣，可以更好的验证优秀的结论，也可以在前人的基础上提出新看法新角度，并在实践中证明。

项目开发的过程大大有利于锻炼开发人员对于目标问题的发现、分析以及解决过程中必不可少的沟通能力、分析能力、理解归纳能力、设计实现能力以及观察测试能力等各项能力。

通过此次毕业设计，能充分的展现项目开发的意义：

1. 进一步巩固、加深本科四年专业所学所感，掌握基本理论知识，结合理论完成实践，提高综合分析问题，解决问题的能力。
2. 开发小型应用系统，掌握运用Java语言编写调试应用程序，训练独立开发应用系统，进行数据处理的综合能力。
3. 掌握面向对象程序设计的方法。
4. 熟练掌握Java语言的基本语法，灵活运用各种数据类型，对面向对象的程序设计语言的学习有更好的理解。
5. 进一步掌握在集成环境下如何调试程序和修改程序。
   1. 本文章节安排

本文共分七章，具体章节安排如下：

第一章概述了选题背景、项目目标及意义，介绍了Java游戏开发现状，指明连连看游戏设计及实现的基本方向。

第二章概述了开发环境和相关技术，简述了Java和Eclipse，重点介绍了Java在游戏开发中的应用，包括GUI开发、绘图技术、事件处理机制等。

第三章介绍了可行性分析和系统需求分析。首先从技术、经济、社会三方面进行了可行性分析，然后简述了连连看游戏的一般游戏规则。最后，从业务需求、用户需求、功能需求和非功能需求四个方面进行了需求分析。

第四章进行了系统概要设计，主要将系统划分出界面模块和处理模块两大模块，确定了游戏风格和素材。

第五章介绍了详细设计和实现，包括界面、处理模块和游戏连接算法的设计及实现。

第六章进行了系统测试。简述了系统测试相关内容，给出了测试用例和测试结果。

第七章进行了全文总结，一方面总结了在完成过程中学习到的内容，另一方面说明了现存问题，展望了进一步工作。

* 1. 本章小结

　　本章通过对选题背景、项目目标和意义以及Java游戏开发现状的介绍，阐述了选题原因，指明了连连看设计和实现的基本方向，描述了项目开发意义。

1. 开发环境和相关技术简介

Eclipse 是一款优秀的跨平台的自由的集成开发环境（IDE），除了可作为Java的集成开发环境外，还可以作为编写其他语言（如C＋＋和Ruby）。Java是一种可以撰写跨越平台的[应用软件](http://baike.haosou.com/doc/3175127.html_blank)的面向对象的[程序设计语言](http://baike.haosou.com/doc/5797287.html_blank)，具有优越的通用性、高效性、安全性和平台移植性，广泛应用于PC、科学超级计算机、游戏控制台、数据中心、移动电话和互联网等各个领域。本章将围绕Eclipse和Java介绍项目开发环境和相关技术。

* 1. 开发环境

软件开发环境（Software Development Environment，SDE）指在基本硬件和数字软件的基础上，为支持系统软件和应用软件的工程化开发和维护而使用的一组软件。系统的软件环境主要由开发系统所用的开发工具和环境集成机制等几部分构成。良好的开发环境对于稳定项目组人员以及提高生产效率都有不可忽视的作用。本次开发环境的具体内容如表2-1开发环境表所示。

表 2-1　开发环境表

Table 2-1　 Development Environment

|  |  |  |  |
| --- | --- | --- | --- |
|  | **名称** | **版本** | **语种** |
| 操作系统 | Windows | Windows 8.1 | 简体中文 |
| 开发环境 | Java | JDKVersion1.7.0 | 简体中文 |
| 开发工具 | Eclipse | Eclipse-jee-mars | 简体中文 |

* + 1. Eclipse

Eclipse 是一款优秀的跨平台的自由的集成开发环境（IDE），从其功能上来讲是一个基于Java的、开放源代码的可扩展开发平台，就其本身而论，它是一个用于通过插件组件构建开发环境的框架、服务，并且附带一个标准的插件集，包括Java开发工具（Java Development Kit）。

Eclipse的目标不仅仅为提供一个IDE，它还包括了插件开发环境（Plug-in Development Environment，PDE），该环境使得开发人员可以根据所需添加新插件，使得Eclipse不断扩充越来越强大。整个Eclipse体系结构像一个拼图，可添加插件，还可以在已有插件上再添加插件，所以Eclipse的架构基本上是“内核＋核心插件＋定制插件”。

如图2-1所示，其中，Platform Runtime（平台运行库）是 Eclipse 平台核心，它是一个微内核，负责管理插件注册表和插件，其他所有的功能如 WorkSpace（工作空间）和 WorkBench（工作台）等都是以插件的形式提供。工作空间是负责管理用户资源的插件。工作台是提供用户界面的插件。SWT（Standard Widget Toolkit，标准小窗口工具箱）是Eclipse 平台自己开发的图形API包，作用和Java的AWT/Swing GUI API 一样，用来构建图形界面。

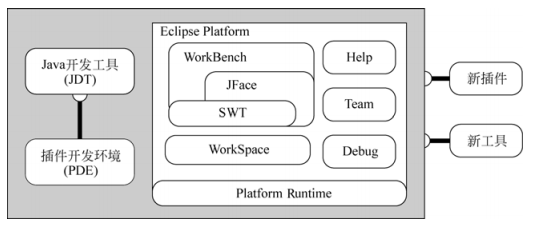


图 2-1　Eclipse体系结构图

Figure 2-1　The Architecture Of Eclipse

* + 1. Java

1. 体系

Java根据应用范围可以分为三个体系：Java SE、JavaEE、Java ME。

Java SE（J2SE，Java2 Platform Standard Edition，标准版）主要用于桌面应用软件的编程；JavaEE（J2EE，Java 2 Platform, Enterprise Edition，企业版）主要用于开发分布式的网络程序，例如电子商务网站和ERP系统；Java ME（J2ME，Java 2 Platform Micro Edition，微型版）主要用于嵌入式系统开发，例如手机和PDA的编程。

1. 工作原理

Java由四方面组成：Java编程语言，即语法；Java文件格式，即各种文件夹、文件的后缀；Java虚拟机(JVM)，即处理\*.class文件的解释器和Java应用程序接口(Java API)。当编辑运行一个Java程序时，需要同时涉及这四个组成。

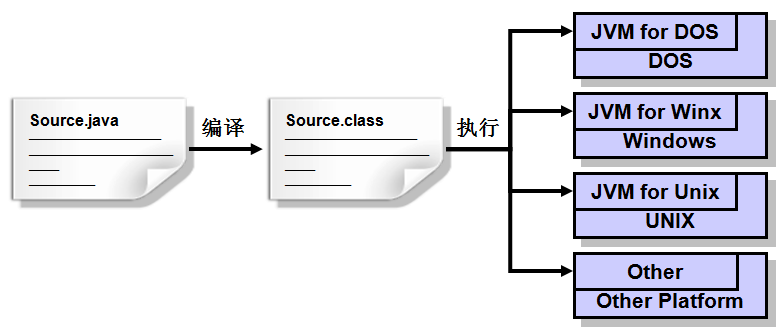


图 2-2　工作原理图

Figure 2-2　The Figure Of Working Principle

使用文字编辑软件（例如记事本、写字板、UltraEdit等）或集成开发环境（Eclipse、MyEclipse等）在Java源文件中定义不同的类，通过调用类（这些类实现了Java API）中的方法来访问资源系统，把源文件编译生成一种二进制中间码，存储在class文件中，然后再通过运行与操作系统平台环境相对应的Java虚拟机来运行class文件，执行编译产生的字节码，调用class文件中实现的方法来满足程序的Java API调用。

1. 特性

第一，Java语言是简单的。

一方面，语法与C语言和C＋＋语言接近，易学习易使用。另一方面，Java丢弃了C＋＋中很少使用的、不易理解的特性。 Java语言中不使用指针使用引用并且提供了自动的垃圾回收机制，这使得程序员不必为内存管理而担忧。

第二，Java是面向对象的语言。

Java提供类、接口和集成等原语，并只支持类之间的单继承，但支持接口之间的多继承，支持类与接口之间的实现机制（关键字 implements）。Java语言还全面支持动态绑定。Java程序使用类（class）来组织的，另外它还包括一个类的扩展集合，分别组成各种程序包（Package），用户可以在自己的程序中使用。例如，Java提供产生图形用户接口部件的类（java.awt包），这里awt是抽象窗口工具集（abstract windowing toolkit）的缩写，处理输入输出的类（java.io包）和支持网络功能的类（java.net包）。

第三，Java语言是分布式的。

Java设计成支持在网络上应用，它是分布式语言。Java既支持各种层次的网络连接，又以Socket类支持可靠的流（stream）网络连接，所以用户可以产生分布式的客户机和服务器。

第四，Java语言具有健壮性和安全性。

Java允许扩展编译时检查潜在类型不匹配问题，不支持隐式声明，不支持指针。它的强类型机制、异常处理以及垃圾回收机制保证了Java程序的健壮性。并且Java还提供了Null指针检测、数组边界检测、异常出口、Byte code校验等功能。Java通常被用在网络环境中，具有许多安全特性。

第五，Java语言具有多线程性。

Java的lang包提供一个Thread类，而线程必须由Thread类或其子（孙）类来创建。通常使用以下俩种方法创建线程：其一，使用Thread(Runnable)的构造将一个实现了Runnable接口的对象包装成一个线程。其二，从Thread类派生出子类并重写run方法。Java语言支持多个线程同时执行，并提供多线程之间的同步机制（关键字synchronized）。

第六，Java语言是可解释的。

C，C＋＋语言都是针对CPU芯片进行编译，生成机器代码。而Java程序在Java平台上被编译为字节码格式的“中间代码”，可以在有Java虚拟机（JVM）的人和系统上被解释执行。

第七，Java语言是动态的。

Java语言的设计目标之一是适应于动态变化的环境。Java程序需要的类能够动态地被载入到运行环境，也可以通过网络来载入所需要的类。这也有利于软件的升级。另外，Java中的类有一个运行时刻的表示，能进行运行时刻的类型检查。

* 1. 相关技术简介

本次项目开发基于成熟的Java语言，开发环境是Eclipse。Eclipse是一个开放源代码的软件开发项目，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台，它使用简单，且占用电脑资源也比较少，便于系统的学习Java。Java语言具有良好的特性，是一种应用广泛的面向对象的程序设计语言。

* + 1. Eclipse的图形界面开发

Eclipse拥有漂亮的开发界面，基于标准部件库（Standard Widget Toolkit，SWT）开发。SWT是基于Java环境下的新类库，它提供了Java环境下的图形编程接口，SWT中的图形库和工具包取代了AWT和Swing。SWT直接调用操作系统的图形库，这使得Java程序的运行速度得到了保证，但是SWT的缺点是支持的平台太少。Eclipse也可以开发基于Swing的程序。

* + 1. Java技术在游戏开发中的应用

1. 图形用户界面（Graphical User Interface，GUI）

图形用户界面（Graphical User Interface，GUI）与早期命令行界面相比，在视觉上对于用户来说更易于接受。但是，通过在显示屏的特定位置，以各种美观而不单调的视觉消息提示用户状态的改变，势必得比简单的消息呈现花上更多的计算能力。

在Java语言中，所有的组件都以类的形式给出，AWT组件放在java.awt包中，Swing组件放在java.swing包中。如图2-3所示，java.awt包中控件类的体系结构。

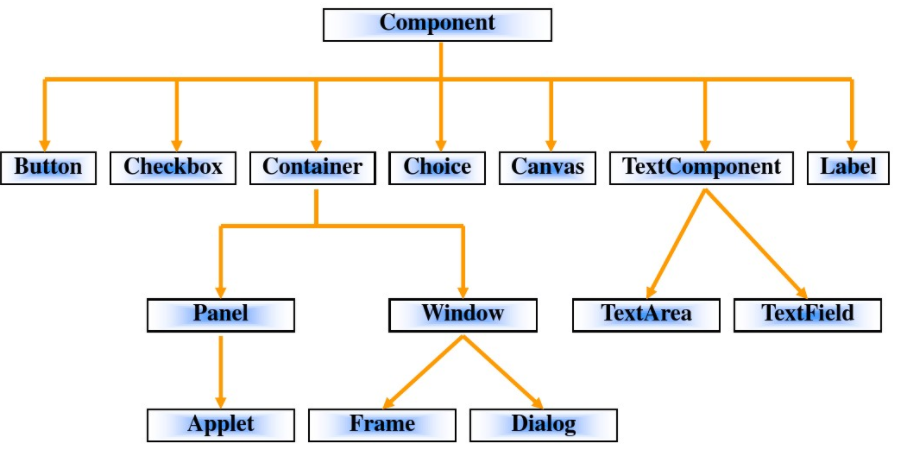


图 2-3　 java.awt包中控件类的体系结构图

Figure 2-3　 The Architecture Diagram For The Control Class In The java.awt Package

Java.awt是Java内置的包，属于Java基本库（JFC）一部分，要使用到该包中的类，必须显式声明import java.awt.\*。Component是所有组件的父类，Component类或之类创建的对象叫组件。Button、Scrollbar、Canvas、List、Checkbox、TextField、TextArea、Label是包java.awt中的类，并且是包中Component的子类。容器也是组件，可以嵌套。上图中，Applet不是java.awt中的类。

Javax.swing包包含了一系列Swing控件，如果要使用该包中的类，则必须显式声明import javax.swing.\*。Javax.swing包中类有：Japplet、JFrame、JDialog、Jcomponent。Swing组件类的设计原理建立在MVC（Model-view-Controller）结构基础上。MVC模式由图2-4所示。

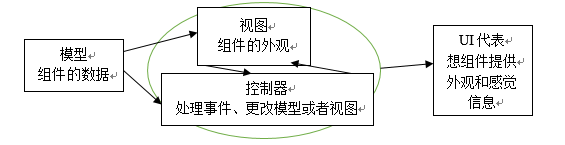


图 2-4　MVC模式图

Figure 2-4　MVC Model

抽象窗口工具包（Abstract Window Toolkit，AWT）是SUN专门针对Java GUI编程提供的最早的也是最初的开发工具包，用来建立和设置Java图形用户界面，这个工具包提供了一套与本地图形界面交互的接口。AWT使用的技术是对等设计模式即（AWT 中的图形函数与操作系统所提供的图形函数之间有着一一对应的关系）。所以它建立图形用户界面时，是利用操作系统所提供的图形库，这使得不同操作系统上显示不同的界面，因此AWT组件集遵循最大公约数原则，即AWT只拥有所有平台上都存在的组件的交集。AWT具有两面性，不被Sun推荐使用，只是为了确保向下兼容和支持Swing。

Swing是在AWT的基础上构建的图形界面系统，它最初是为解决AWT不能满足用户需求的发展。Swing提供了AWT能提供的所有功能，并且用纯粹的Java代码对AWT的功能进行了扩充。Swing遵从组件按照某种布局摆放在容器里的原理。当界面完成之后，必须能进行事件处理，AWT和Swing之间的事件处理基本是一致的。Swing组件是用Java实现的轻量级( light-weight)组件，没有本地代码，不依赖操作系统的支持，这是它与AWT组件的最大区别。Swing采用了一种MVC的设计范式，即“模型-视图-控制”。 Swing构建界面包括以下几个步骤：导入Swing包；选择界面风格；设置顶层容器；设置按钮和标签；将组件放到容器上；为组件增加边框；处理事件；辅助技术支持。Swing组件的层次结构图如图2-5所示。

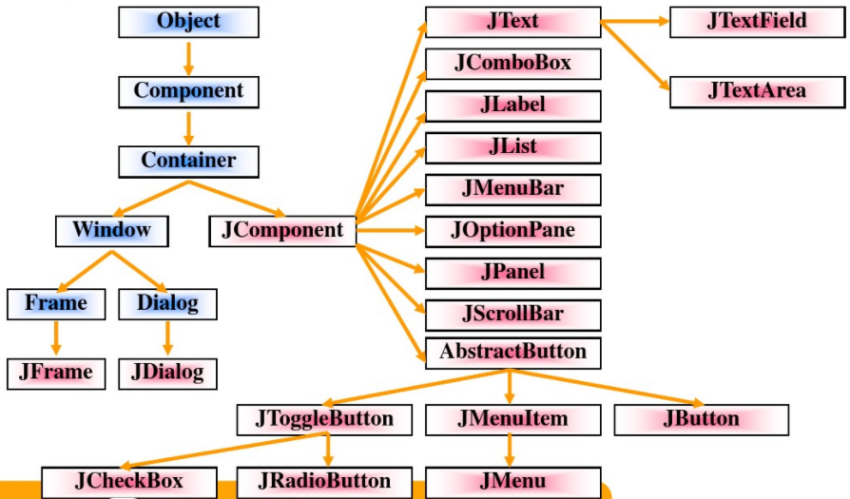


图 2-5　 Swing组件的层次结构图

Figure 2-5　 Hierarchy Diagram For Swing Components

SWT是Standard Widget Toolkit的缩写，是IBM为尝试解决AWT和Swing带来的诸多问题而构建的一个开源的GUI编程框架，它提供了比AWT更为丰富的组件集。与 AWT 的概念相比，SWT 是一个低级的 GUI 工具包，在构建SWT的过程中，构建者从 AWT 和 Swing 实现中学习了很多经验，他们试图构建一个集二者优点于一体而没有二者的缺点的系统。因此SWT可以说是AWT与swing的融合体。

1. 绘图技术

由Sun公司与Adobe系统公司合作推出的Java 2D API，提供了一个功能强大而且非常灵活的二维图形框架。Java 2D API扩展了java.awt包中定义的Graphics类和Image类，提供了高性能的二维图形、图像和文字，同时又维持了对现有AWT应用的兼容。

Java 2D API是JFC(Java Fundation Classes)的一员，加强了传统AWT的描绘功能。在 JDK1.2中已经支援 Java 2D 的使用。透过Java 2D API ，程序员可以轻松地描绘出任意的几何图形、运用不同的填色效果、对图形做旋转（ rotate）、缩放（ scale）、扭曲（ shear）等。

Applet程序中所采用的AWT的绘图机制主要涉及三个方法：paint()方法、update()方法和repaint()方法，update()方法和paint()方法都有一个Graphics类参数。Graphics是画图的关键，它可以支持两种绘图：一种是基本的绘图，如：画线、矩形、圆等；另一种是画图像，主要用于动画制作。处理图形则包括两个步骤：创建Graphics对象和使用Graphics对象绘制线条和形状、呈现文本或显示与操作图像。

Graphics中提供的图形绘制方法有：

paint( )：进行绘图的具体操作，必须有程序员重写，系统自动调用；

update( )：用于更新图形，先清除背景、前景，再调用paint()；

因为update()方法和paint()方法所传递的参数是Graphics类的对象，所以通常用于在动画程序中通过重载进行绘画。

repaint( )：用于重绘图形，在组件外形发生变化，即大小改变或位置移动repaint( )方法立即被系统自动调用，而实际上repaint()方法是自动调用update()方法。

1. 事件处理机制

除了友好的图形用户界面，流畅的人机交互也是必不可少的要素。在游戏过程中，正确且及时的对用户操作做出响应会给用户带来很好的游戏体验。所以说，游戏开发中的交互与其他技术有着相同的地位，是一个游戏必备的核心。Java的事件处理机制能够提供良好的交互。

Java的事件处理机制中，有以下几个概念：

1. 事件Event：一个状态的改变或者一个活动的发生。Java中的事件主要有两种：组件类事件、动作类事件。
2. 事件源Event Source：产生事件的对象。
3. 事件监听器Event Listener：监听事件源的对象，包含事件处理器，并负责检查事件是否发生，若发生就激活事件处理器进行处理的类叫做事件监听器类。其实例就是事件监听器对象。事件监听器类必须实现事件监听器接口或继承事件监听器适配器类。事件源通过调用相应的方法将某个对象作为自己的监听器。在事件发生时，事件源负责给予通知的一种对象。接收到事件通知的监听器进行处理。事件监听器接口和事件监听器适配器类也都包含在java.awt.event和javax.swing.event包中。
4. 事件处理器Event Handler：事件处理方法，接收事件并处理事件的方法。事件处理器包含在一个类中，这个类的对象负责检查事件是否发生，若发生就激活事件处理器进行处理。

事件处理机制图详见图2-6。当事情发生时，由事件源委托给事件监听器进行处理，继而执行事件处理代码，产生执行效果。

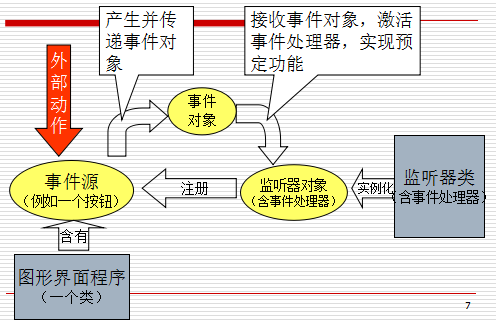


图 2-6　事件机制图

Figure 2-6　Event Mechanism Diagram

1. 多线程的应用

在Java中，线程有两个指代含义：其一是指java.lang.Thread类的一个实例；其二是指线程的执行。如图2-7所示，Java线程有五种基本状态：新建状态（New）、就绪状态（Runnable）、运行状态（Running）、阻塞状态（Blocked）和死亡状态（Dead）。

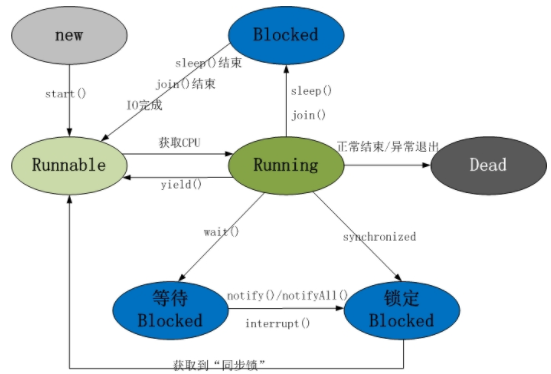


图 2-7　线程生命周期图

Figure 2-7　The Thread Lifecycle Diagram

在Java中，线程的创建常见的有以下三种形式。

第一种，继承Thread类，重写该类的run()方法。其中，run()方法的方法体也称之为线程执行体，是线程需要完成的任务。当创建此线程类对象时会创建一个新的线程，并且使得线程进入新建状态。通过调用线程对象引用的start()方法，使得该线程进入到就绪状态，此时此线程并不一定会马上得以执行，这取决于CPU调度时机。

第二种，实现Runnable接口，并重写该接口的run()方法，该run()方法同样是线程执行体，创建Runnable实现类的实例，并以此实例作为Thread类的target来创建Thread对象，该Thread对象才是真正的线程对象。

第三种，使用Callable和Future接口创建线程。具体是创建Callable接口的实现类，并实现clall()方法。并使用FutureTask类来包装Callable实现类的对象，且以此FutureTask对象作为Thread对象的target来创建线程。

对于线程的启动而言，三种方法都是调用线程对象的start()方法，需要特别注意的是：不能对同一线程对象两次调用start()方法。在Java中，一旦创建一个新的线程，就产生一个新的调用栈，每个线程都有一个调用栈，即使不在程序中创建任何新的线程，线程也在后台运行着。一个Java应用总是从main()方法开始运行，mian()方法运行在一个线程内，它被称为主线程。

JDK1.6之后增加了线程池的概念，使用线程池对线程的操作性能更优于传统的线程操作，线程操作更加灵活、安全。

* 1. 本章小结

本章主要介绍了Eclipse和Java的相关内容和本次项目涉及到的相关技术，通过对比GUI三种工作集：AWT、SWT、Swing，加深对图形用户界面开发的理解和对于AWT、Swing组件层次结构的掌握。通过学习总结Java的事件处理机制的概念和过程，为项目的设计和实现打下理论基础。

1. 需求分析

软件需求分析就是通过调查研究和有效的沟通以及建立系统的数据模型、功能模型和行为模型，使用户和开发人员就“系统必须做什么”这个问题达成一致意见，最后生成软件需求规格说明的过程。软件需求分析的过程就是对用户或企业意图不断揭示的过程，需要对系统进行可行性研究之后，对系统目标做出分为详细的描述。

* 1. 可行性分析

可行性研究的结果是清楚的回答：该系统是否值得开发。

下面将分别从技术、经济、社会、操作四个方面对基于Java的连连看游戏的设计与实现进行可行性分析和研究。

* + 1. 技术可行性

技术可行性分析一般可以理解为项目开发涉及到的技术、环境是否符合开发标

准，能够达到预期效果；它是项目成功运行的一个必不可少的条件；因此，对技术可行性分析需要进行全面化的评估。

本次连连看游戏采用的开发工具是功能强大的Eclipse，它具有十分强大的可扩展功能，只要下载安装所需要的相关插件，就能支持我们用纯Java实现此次的桌面游戏程序。使用Eclipse开发应用程序可按需下载插件，不过分依赖于硬件，不论是升级产品方面还是扩展功能方面都有很好的发挥空间。

本次项目使用Java的GUI开发技术。Java本身可提供类库，完善的类库和AWT包对桌面游戏开发提供了强大完整的技术支持，完全可以满足我们连连看小游戏的开发，在开发过程中，可以创建满足本次开发项目任务的直接的、简明的图形用户界面。

Eclipse在开发Java项目方面应用多而广，是一个成熟可行的开发平台。Java语言也是成熟的开发语言，能够满足此次要求。在开发方面，无论是开发平台，还是开发语言，对于开发人员来说都是熟悉且能易掌握的。在从硬件方面来看，时下的硬件容量不断扩大，可靠性日益渐涨，性价比越来越高，对于本项目开发而言，足够满足需求。

由以上分析可知，本软件在技术上是可行的。

* + 1. 经济可行性

经济可行性是指对项目的成本和收益进行分析，具体内容包括两个部分：新系统所能带来的经济效益和该经济效益是否超过系统开发和系统维护所需要的费用。

对于本项目而言，在系统开发和系统维护方面的开销很小，没有过多的硬件投入，调试方便，开发过程中，Eclipse是开源免费的，本游戏在此开发环境下，不需要太复杂的工具和服务器支持，在完成过程中，也仅需要一台个人电脑以及在完成过程中上网查询所用到的人力资源以及文献资源。

　　综上所述，本项目在经济上是可行的。

* + 1. 社会可行性

社会可行性广义上可分为三大部分：道德方面、法律方面、社会方面。接下来将就这几个方面，对此次项目进行分析。

从道德方面来看，本次开发无论从过程中还是最终完成的产品上看均不涉及传播低级黄色内容、社会负能量。本次开发目的主要是借毕业设计巩固所学专业知识，一切皆以个人为单位，不涉及广泛传播及使用，对社会基本不造成任何不良影响。

从法律方面来讲，本次开发过程和最终产品因开发目的不涉及版权、社会安全等法律问题。在开发过程中，参考的所有文献会列明，尊重知识产权，尊重专利版权，不会将产品投入社会使用。

从社会方面来谈，本次开发是对已有成果的一次学习和开发体验，虽不能带来很大的社会价值，但不同程度上实现了开发者个人价值。因不涉及产品推广，所以不涉及产品带来的社会经济效益和社会影响力。

由以上分析可知，本软件具有社会可行性。

* + 1. 操作可行性

操作可行性主要分析系统操作难度问题。本游戏界面以窗口形式给出，可视化强，整体风格简洁大气，各个功能键布局合理使得游戏功能一目了然。整个游戏过程操作简单易上手，完全依靠鼠标点击操作，在简单说明或培训下，操作上基本没有难度。

综上所述，本游戏具有技术可行性、经济可行性、操作可行性和社会可行性值得开发。

* 1. 连连看游戏需求描述及规则描述

游戏规则是模仿网络上普通的连连看游戏，即解决鼠标两次点击的图片能否消去的问题。当然，前提是点击两张相同的图片，若点击的是同一张图片或者两张不同的图片，则不予以处理。在两张相同图片所能连通的所有路径中，如果存在一条转弯点不多于两个的路径，就可以消去；如果没有，则不予以处理。

开始制作游戏时，主要需要解决的问题有以下几个方面：第一，如何设置整个游戏的界面。第二，如何控制连连看游戏中随机图片的生成且每种图片生成个数必须为偶数。第三，游戏开始后，需要判断鼠标两次点击的图片能否消去，即判断图片是否相同且片之间路径是否满足判断要求等。第四，如何实现游戏计时功能、刷新功能、重新开始新一局游戏功能。第五，相关游戏情况展示。

游戏往往会涉及游戏胜利或失败条件。对于连连看游戏来说，胜利条件是在规定时间内将游戏界面上所有图片全部依照一定规则消除掉。失败条件则相反，指未能或无法在规定时间内将棋盘上面的图片全部消除掉。

* 1. 系统需求分析

软件需求分为四个层次，分别是：业务需求、用户需求、功能需求和非功能需求。

业务需求主要反映了组织结构或客户对系统、产品高层次的目标要求，通常在项目定义与范围文档中予以说明；用户需求描述了用户使用产品必须要完成的任务，这在实例或方案脚本中予以说明；功能需求定义了开发人员必须实现的软件功能，使用户利用系统能够完成他们的任务，从而满足业务需求；非功能性的需求描述了系统展现给用户的行为和执行的操作等，它包括产品必须遵从的标准、规范和约束，操作界面的具体细节和构造上的限制。

* + 1. 业务需求

业务需求针对的是项目组或者公司，描述是公司想如何解决用户的问题，如何满足用户的欲望，并将利益最大化。

本项目课题是连连看游戏的设计与实现，在业务需求方面，第一，具备连连看基本游戏功能。选择两个图形，判断是否满足一定条件，条件即两个图形一致且二者之间存在转弯少于3的路径。当判定满足这一条件后，两张图片被消除。第二，具备附加功能：计时功能、重新开始功能、刷新当前界面功能等。

* + 1. 用户需求

本项目面向的用户是游戏玩家，对于用户而言，需求有两个方面，一方面，要求游戏界面整齐美观，长时间使用也应尽可能的减小对游戏的审美疲劳。另一方面，要求游戏体验良好。在运行游戏的过程中，能保证数据安全，游戏能稳定、高速地运行。对于游戏本身，用户往往要求游戏功能相对完善，游戏具备可娱乐性，并且游戏规则易理解，游戏操作既要简单同时也应有难度，在保证游戏操作顺滑、响应快速的情况下，游戏能具有一定的挑战性。

通过用例图能更清晰的分析用户需求，如图3-1连连看游戏的用例图。用户在连连看游戏中，能开始游戏、重新开始游戏、重排列游戏、结束游戏、退出游戏、消除图片、查看得分和闯关数等。

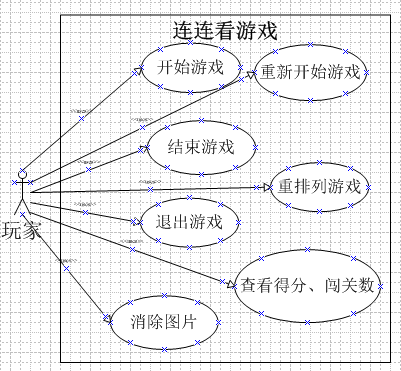


图 3-1　用例图

Figure 3-1　Use Case Diagram

　以“重排列游戏”用例为例，其活动图如下图3-2所示。

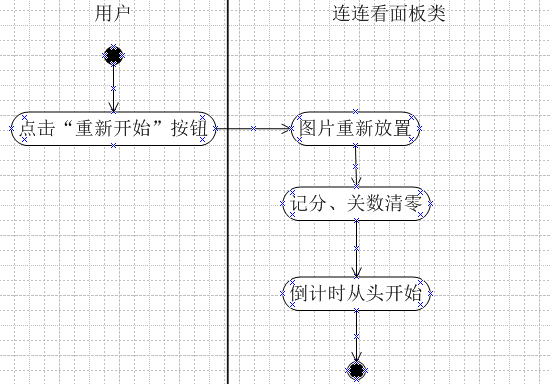


图 3-2　“重排列游戏”活动图

Figure 3-2　 Activity Diagram Of “Rearrange The Game”

* + 1. 功能需求

根据上文游戏规则描述以及需求描述，游戏规则是模仿网络上普通的连连看游戏，主要需求是解决鼠标两次点击的图片能否消去的问题。当然，前提是点击两张相同的图片，若点击的是同一张图片或者两张不同的图片，则不予以处理。在两张相同图片所能连通的所有路径中，如果存在一条转弯点不多于两个的路径，就可以消去；如果没有，则不予以处理。这也是游戏的基本功能，给定任意具有相同图形的两个格子，我们需要寻找这两个格子之间在转弯最少的情况下，经过格子数目最少的路径。如果这个最优路径的转弯数目少于3 ，则这两个格子可以消去。

根据难度的不同该游戏由4、8、12、15种不同的图片组成，游戏开始将会出现64张随机组合的图片，在规则下点击两张相同的图片后，图片将会消失。图片全部消完为游戏成功，将进入下一关。游戏还将设置刷新功能，重新开始功能的按钮，并实现该功能，方便用户进行操作。游戏的主要功能将通过程序中的线程、布局功能、事件监听进行有效实现。

本系统将划分为以下几个模块，如图3-3系统功能模块图。将连连看游戏按照功能划分为：

消除模块：完成连连看游戏消除动作的模块。

重新开始模块：完成连连看游戏中重新开始新一局游戏功能的模块。

刷新模块：完成连连看游戏中，将游戏中未消除的图片个数记录并重新随机放置等数量图片的功能模块。

选择难度模块：完成游戏中切换游戏难度并开始新一局游戏的模块。

计时模块：完成游戏中倒计时并显示提示信息的模块。

状态显示模块：完成游戏中对游戏得分、闯关数记录的模块。

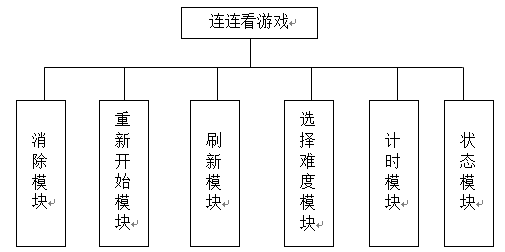


图 3-3　系统功能模块图

Figure 3-3　System Functional Module Diagram

为更好的实现该游戏，结合系统流程图可进一步将上述功能细化如下：

第一，图片的选择：选中的图片应该区别于其他图片。

第二，图片的排列：在开始新一局的游戏时，图片的排列要清晰，满足随机的同时，要保证成对出现。

第三，记分功能：在游戏界面上要有明确标识，直观的显示当前游戏状态，包括游戏得分情况和闯关关数情况。

第四，沙漏计时功能：沙漏状计时能将时间地流逝直观形象的表达，增加游戏难度的同时很好地刺激了游戏玩家的胜负欲。

第五，游戏的开始和重新开始：在游戏开始前，应对玩家有所提示。重新开始游戏要中断当前游戏，开始一局新的游戏，同样要有所提示。

第六，游戏关闭和结束：游戏关闭和结束是两个概念，游戏关闭相当于关闭程序，游戏结束是指游戏时间到，此处应该对玩家有所提示，游戏结束后游戏依旧在运行，可以开始新一局游戏，也可以结束游戏。

第七，游戏难度：游戏难度不要仅仅局限于一个难度，提升难度有利于激发玩家运行游戏，提高游戏对玩家的吸引力和游戏本身的可玩性。游戏难度应可选。游戏默认开始时游戏难度应该较低。现设置游戏难度为4个等级，每个等级显示总体个数相同但不同种类的图片，难度越高，种类越多，整体界面看起来越复杂。

第八，游戏闯关： 每一个难度下可以有很多关卡，每一个关游戏过关后可以直接进入下一关，最好能实现每一关较上一关略有难度。

第九，游戏提示功能（刷新功能）：在游戏过程中可以使用一次刷新功能，重新排列当前游戏未消除的图片个数的图片。

第十，游戏必要的提示：包括前面叙述的关卡数提示、得分提示、游戏开始和结束提示、倒计时提示。

第十一，游戏要最大限度地实现了易维护性和易操作性，游戏运行稳定、安全可靠。

* + 1. 非功能需求

非功能需求包括产品必须遵从的标准、规范和约束，操作界面的具体细节和构造上的限制。

首先，软件制作为了设计、实现和后期维护的方便，以及软件用户使用的便利，所以必须采取一定的设计原则。其主要设计原则有：

1. 简单性：这里指游戏操作简单易懂，结合上文，可以通过游戏界面实现部分要求，游戏整体要尽量操作简单易懂，这对于我们这款注重交互体验的游戏软件至关重要。
2. 针对性：本软件设计是基于java的连连看的需求定向开发设计，所以具有专业突出和很强的针对性。
3. 实用性：要求本软件能够满足基于java的连连看游戏的需要，因此具有良好的实用性。
4. 一致性：界面风格及命名规则一致性（整体界面的布局、地图样式和贴图保持风格一致；变量、类名和其他元素的命名规则保持一致）具体的来说：元素风格一致性是指界面元素的美观风格、摆放位置在同一个界面和不同界面之间都应该是一致的。变量命名规则的一致性具体是指变量用英文单词及英文词组进行命名，做到任意变量均能从上下文推断其义。

　　其次，游戏界面的具体细节也不可忽略。游戏一经推出，能否第一时间被喜欢、被欢迎，除了游戏本身操作方面的表现，更多时候取决于游戏界面的优美表现。时下，大型网络游戏、flash网页游戏、桌面小游戏，款式多、样式多，不乏有一些亮眼表现的好游戏、经典游戏。玩家在审美疲劳的同时练就了一双苛刻的眼力，对游戏界面的要求也越来越高。

本游戏主要针对广大游戏玩家的娱乐时间，在结束一段紧张的工作学习时间之后，通过玩游戏放松心情。结合连连游戏一般以图片形式给出，本次游戏界面也将采用图形用户界面设计准则。

在游戏界面中，将通过一个二维界面采用图片的形式给出一些可点击可选择的方块，在选中图片后应该能明显区分。整体布局合理，用户使用应以方便、灵活、快捷、准确为准。

游戏界面总体风格要简洁明了，对各个功能按钮、功能部件一目了然，通过简单的图片和组件，直观的“摆出”游戏规则和使用方法。方便游戏玩家接受和掌握游戏方法，达到舒缓压力的基本要求。

* 1. 本章小结

　　首先，本章对连连看游戏项目从技术、经济、社会三个方面进行可行性研究，研究证明项目可行。其次，从业务需求、用户需求、功能需求和非功能需求四个层次借助用例图和活动图描述了连连看游戏系统的需求。

1. 概要设计

　　概要设计的主要任务是把需求分析得到的系统扩展用例图转换为软件结构和数据结构。设计软件结构的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性、以及数据库的设计。显然，概要设计建立的是目标系统的逻辑模型，与计算机无关。

* 1. 模块设计

由于Swing组件类的设计原理建立在MVC（Model-View-Controller）结构基础上，MVC是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型(Model)、视图(View)和控制器(Controller)。所以根据此架构模式划分连连看游戏总体结构，设计各模块概要内容，进而划分出此次项目中的类，绘制类图。

* + 1. 连连看系统总体结构

此次设计连连看的主体构成为连连看图片以及一些功能键。从界面布局来看，面板的右面有重新开始、刷新按钮，难度选择选项卡（4种难度），得分、关卡状态显示栏，以及沙漏状的时间倒计时装置。面板左面是一块主要操作区域，用来显示待消除的图片排列情况。本连连看系统应该包括以下两大结构：游戏界面模块；处理模块。游戏界面模块下分两个模块：功能键模块和操作模块。如图4-1：

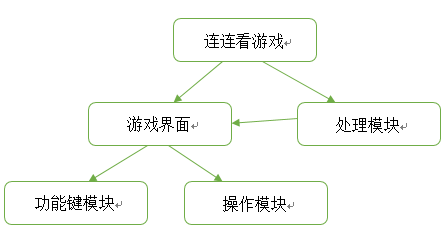


图 4-1　连连看游戏模块图

Figure 4-1　The Module Diagram Of Link Game

* + 1. 连连看系统模各模块介绍

　　图4-1所示的游戏模块图中，连连看游戏系统主要分成两个部分。第一部分是显示部分即游戏界面模块，该模块根据布局又分为功能键模块和操作间模块。第二部分是处理部分即处理模块，在该模块中，需要完成需求分析中的用户需求和功能需求。

* + 1. 游戏界面模块

游戏界面模块根据布局细分为操作模块和功能键模块。

操作模块：在游戏过程中，需要使用鼠标与游戏系统交互的主要模块。模块主要有排列的JButton构成，每个JButton上显示图片。所有JButton随机出现，且成对出现。借按钮组件及其事件处理机制，模拟游戏中选中图片并消除。当点击按钮组件时，这个外界刺激会触发事件监听器，事件监听器会接收事件并作出处理，就有了对这一操作的响应。

功能键模块：在游戏过程中，需要使用鼠标与游戏系统交互的次要模块。模块主要有JButton组件、下拉列表、JLabel等组件，构成完成游戏功能的控制部分，在此布局内，除了使用这些组件外，还要绘制一个沙漏状的动画，用以实现倒计时功能。同样，在此模块中所有组件除沙漏，均需要用鼠标操作，每个组件都应该有对应的响应事件和事件监听器。

为更好的区分开显示和实现，游戏界面模块将以布局为主要内容，具体的处理和响应将划分到处理模块。

* + 1. 处理模块

针对游戏的主要功能划分出一个模块：处理模块。这个模块中将实现一些功能，包括操作功能、计时功能、提示功能等。现将设计一下几个模块：

图像处理模块：程序运行后，这个模块将把资源文件中提供的图片进行排列。这个模块在初始化游戏布局和重新开始游戏以及刷新游戏时会用到。

响应模块：响应模块分两大类，一类是对操作模块中JButton的响应（即消除功能），一类是对功能键模块中各个组件的响应（每个响应都实现其对应功能）。

更改提示模块：这个模块主要负责记分、记录关数。

倒计时模块：倒计时模块是对沙漏动画的绘制，并在倒计时五秒处显示提示JLable，在倒计时结束后弹出提示框。

消除模块：判断选中的两个图片是否满足被消除的条件，此模块中，将实现判断。

* 1. 确定类

在图形用户界面GUI设计中，会使用到javax.swing中的JButton、JDialog、JFrame 、JLabel、JPanel等组件。将这些组件放在JFrame容器中，构成LianLianKan类继承于JFrame类。如图4-2所示。在LianLianKan类中声明内部类LianLianKanJPanel类，继承于JPanel类。如图4-3所示，在LianLianKanJPanel类中声明了若干JButton、Choice、ImageIcon类对象。

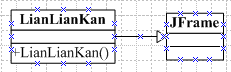


图 4-2　连连看类图

Figure 4-2　LianLianKan Class Diagram

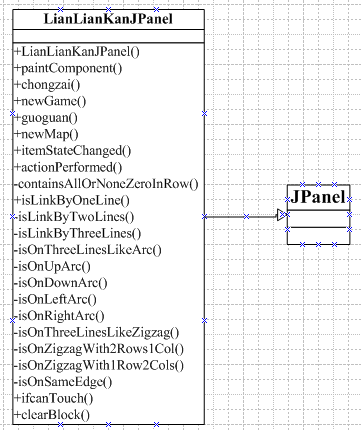


图 4-3　LianLianKanJPanel类图

Figure 4-3　LianLianKanJPanel Class Diagram

在LianLianKanJPanel类中再声明一个内部类loudou类。loudou类同样继承于JPanel类，在该类中创建Thread类、JLabel、JDialog类对象，漏斗形状的绘制通过调用paintComponent(Graphics g)方法完成。loudou类图如图4-4所示。Thread类图如图4-5所示

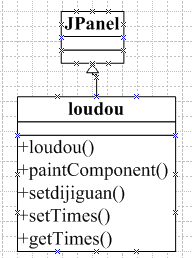


图 4-4　loudou类图

Figure 4-4　loudou Class Diagram

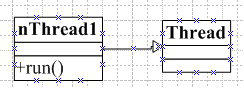


图 4-5　Thread类图

Figure 4-5　Thread Class Diagram

* 1. 游戏素材

本课题游戏显示将采用图片连连看的模式，界面要满足需求分析中提出的简洁明了、友好大方，保证数据存储安全可靠。游戏素材则包括两类图片，一类是背景图片，一类是游戏操作模块中显示用的游戏图片。

* + 1. 背景图片

在背景图片的选择中，考虑到游戏界面的简洁性，选定采用纯色背景图片。为了更加清晰的衬托出游戏图片以及功能键模块中的组件，选定纯黑色游戏背景图片，此处选择图片是为了方便以后便利地更换背景图片或者扩充其他游戏设置功能等。背景图片见图4-6。

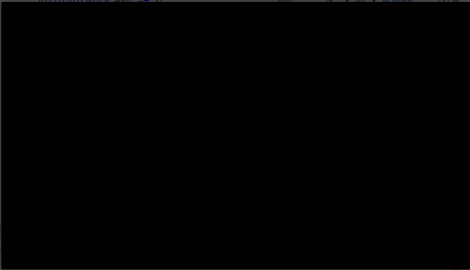


图 4-6　游戏背景图

Figure 4-6　Game Background

* + 1. 游戏图片

前文提到，根据游戏难度的不同将选择显示不同种类的图片，现将按照难度区分，依次展示简单模式、中等模式、困难模式和变态模式下游戏界面将出现的游戏图片素材。除简单模式，每一难度模式将新增几种图片并包含上一难度模式下的图片种类。

简单模式下将出现四种简单的游戏图片，是四种颜色不同的球体，详见图4-7。



图 4-7　简单模式下的游戏图片

Figure 4-7　The Game Picture In Simple Mode

中等模式下将包括八种图片，图4-8将展示新增图片。



图 4-8　中等模式下新增的游戏图片

Figure 4-8　New Image In Medium Mode

困难模式将包括十二种图片，图4-9将显示新增四种图片。



图 4-9　困难模式下新增的游戏图片

Figure 4-9　New Image In Hard Mode

变态模式将包括十五种图片，图4-10将展示新增的三种图片。



图 4-10　变态模式下新增的游戏图片

Figure 4-10　New Image In Hardest Mode

* 1. 本章小结

　　本章划分了连连看游戏总体结构。游戏界面方面确定了游戏界面中的素材和风格。处理模块下根据其功能，设计并描述了各个小模块的概要内容，进而划分出此次项目中的类，绘制了类图。

1. 详细设计和实现

详细设计是在概要设计的基础上进行细化的过程。详细设计的主要任务是设计每个模块的实现算法即实现游戏界面、组件的事件处理等。在详细设计阶段，除了要注意具体模块的功能分解，还要绘制活动图、状态图、顺序图等动态模型，丰富静态结构图。

* 1. 界面设计

游戏主界面以窗口形式给出，用户操作的游戏功能部分由各个组件构成，通过弹出的提示框开始游戏活动，用户通过点击窗口上的各个组件完成操作。游戏界面设计如下图5-1所示。



图 5-1　游戏界面设计图

Figure 5-1　Game Interface Design

* + 1. 界面实现

在游戏运行开始时，将弹出一个提示框以提示用户即将开始游戏，提示框如图5-2。此处弹出提示框一方面是提示作用，一方面是为便利扩展游戏功能，可将游戏扩展成多用户模式，在提示框实现处更改为用户登录模块，并联接数据库。鉴于本次任务时间有限此处未能实现。

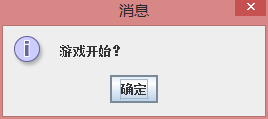


图 5-2　游戏开始对话框

Figure 5-2　The Beginning Of Game Dialog

部分代码如下：

**if** (remainTimes == 0) {

JOptionPane.*showMessageDialog*(**null**, "游戏开始？");

单击确定按钮进入游戏主界面，默认进入简单模式下的游戏主界面，如图5-3。

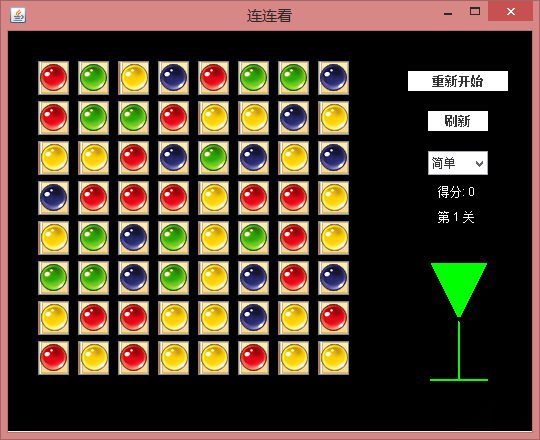


图 5-3　简单模式下的游戏界面图

Figure 5-3　Game Interface In Simple Mode

如上图所示，游戏主界面展示了本次连连看小游戏的主体布局界面。在界面的左侧是游戏主要操作和显示被操作图片的区域，界面的右侧展示了功能模块中几个功能组件。界面的右侧由上至下展示了“重新开始”按钮、“刷新”按钮、难度模式选择列表（默认是简单模式）、两个标签文本、一个沙漏状的倒计时动画（包括一个隐形的标签文本，当时间还剩五秒的时候，会显示倒计时信息）。

不同难度的游戏模式下，主要游戏界面展示略有不同，四种难度模式下的游戏界面图如图5-3、5-4、5-5、5-6。



Figure 54　Game Interface In Medium Mode

图 54　中等模式下的游戏界面图

Figure 55　Game Interface In Hard Mode

图 55　困难模式下的游戏界面图



图 5-6　变态模式下的游戏界面图

Figure 5-6　Game Interface In Hardest Mode

如上图所示，对游戏主界面没有任何操作时，四种难度模式下的游戏主界面最大的区别就是操作模块部分的图片排列，可以看出，难度越大图片种类越多，看起来越复杂，实现了前文对于难度变化提到的要求。每一种困难模式下的图片均是成对出现的，满足前文对于图片排列的要求。

整体界面以黑色为主，简洁大方，整体布局合理工整。左侧图片素材清晰、可爱，图片素材之间留有空隙，视觉上更舒服更易于区分，右侧功能键排列整齐，文字描述清晰。

布局部分核心代码如下：

Random random = **new** Random();

**int** temp = 0;

**for** (**int** i = 0; i < 32 - Kinds; i＋＋) {

temp = random.nextInt(Kinds) ＋ 1;

numbers.add(temp);

numbers.add(temp);

}

**for** (**int** i = 0; i < 8; i＋＋) {

**for** (**int** j = 0; j < 8; j＋＋) {

BlockButton[i][j] = **new** JButton();

add(BlockButton[i][j]);

BlockButton[i][j].addActionListener(**this**);

BlockButton[i][j].setBounds(30 ＋ j \* 40, 30 ＋ i \* 40, 31,34);

}

}

* + 1. 倒计时沙漏的设计

　　在游戏的界面上，还有一个特殊的面板，漏斗面板，漏斗面板上绘制了一个沙漏状的倒计时装置，该装置通过线程控制，时间变量每更新一次，漏斗面上重新绘制一次，每次绘制的线条通过判断条件设置摆放位置，随着时间的递增，沙漏状的倒计时装置就动态的开始倒计时，直至最后五秒，显示提示信息，倒计时结束后，弹出游戏失败提示框。沙漏示意图如下图5-7。

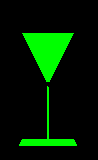


图 5-7　沙漏设计图

Figure 5-7　The Hourglass Design

此处沙漏使用Graphics类的paintComponet()方法，游戏开始自动调用，在线程中，当剩余时间参数改变时调用repaint()，重载paintComponet()，通过判断剩余时间，绘制线条，这样随着每一秒时间的流逝，上方的绿色线条会在视觉上像是流动到下方。当属于时间不足5秒时，JLabel显示提示信息。部分代码如下：

**super**.paintComponent(g);

g.setColor(Color.***green***);

**for** (**int** i = 0; i < 56; i＋＋) {

g.drawLine(x1 ＋ i / 2 ＋ 2, y1 ＋ i, x2 - i / 2 - 2, y1 ＋ i);

}

**if** (remainTimes < 55) {

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

g.drawLine(x1＋i/2＋2, y2-i-1,x2-i/2-2, y2-i-1);

}

g.drawLine((x1 ＋ x2) / 2, (y1 ＋ y2) / 2, (x1 ＋ x2) / 2, y2 - 2);

g.drawLine((x1＋x2)/2＋1, (y1＋y2)/2＋1, (x1＋x2)/2＋1, y2-2);

g.setColor(getBackground());

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

g.drawLine(x1 ＋ i / 2 ＋ 2, y1 ＋ i, x2 - i / 2 - 2, y1 ＋ i);

}

}

**if** (remainTimes >= 50 && remainTimes <= 55)

overJLabel.setText((55-remainTimes) ＋ "second");

**if** (remainTimes == 56)

overJLabel.setText("OVER");

remainTimes＋＋;

repaint();

倒计时结束时，意味着在规定时间内未完成游戏，判定游戏失败，此时应该有提示框，如图5-8。

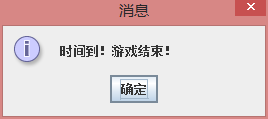


图 5-8　游戏结束对话框

Figure 5-8　The End Of Game Dialog

部分代码如下：

**if** (remainTimes == 56) {

JOptionPane.*showMessageDialog*(**null**, "时间到！游戏结束！");

}

* 1. 游戏执行的实现

　　因为每一次选择难度模式后、每通过一关游戏后都将开始一局新游戏，故一次游戏活动图如图5-9所示。最终状态有两种，游戏失败和游戏结束。

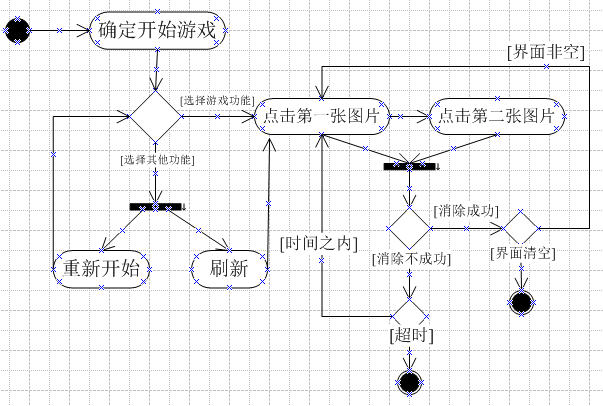


图 5-9　一次游戏活动图

Figure 5-9　Game Activity Diagram

　　一次游戏活动的顺序图如图5-10所示，一次游戏过程中，“点击图片”、“点击‘重新开始’按钮”、“点击‘刷新’按钮”操作上可不分时间先后顺序。每一次游戏结束后可以开始新一局游戏直至用户退出游戏。

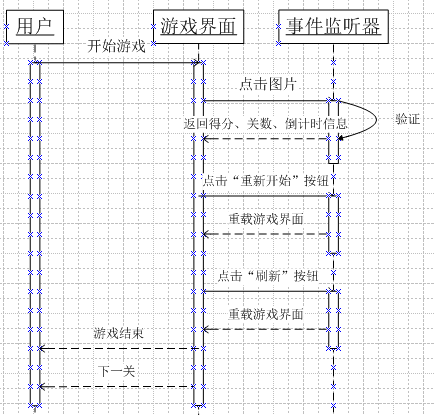


图 5-10　一次游戏顺序图

Figure 5-10　Game Sequence Diagram

* + 1. 点击图片的实现及其部分代码

前文提到，每选中一个图片应该区别于其他未被选中的图片。具体实现方式：当点击第一图片时，图片会变成激活状态，颜色变暗（用到了JButton的SetEnable方法）。当点击第二张图片时如果符合游戏规则，则两张图片消除，否则第一张图片变回默认状态，第二张图片变为激活状态；按钮状态图如图5-11所示，系统截图如图5-12。

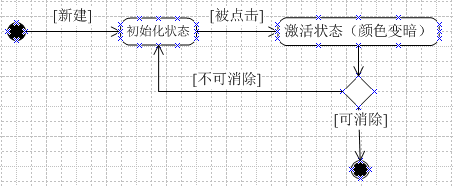


图 5-11　按钮状态图

Figure 5-11　Button State Diagram



图 5-12　系统截图

Figure 5-12　The System Screenshot

此功能实现代码：BlockButton[j][i].setEnabled(**false**);

* + 1. 记分、记关数的实现

在游戏过程中，需要对游戏当前状态进行记录，一次得分记录的过程活动图如图5-13，记录实例系统截图如下图5-14。

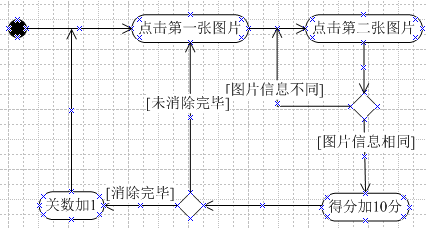


图 5-13　记录活动图

Figure 5-13　Scoring Activity Diagram



图 5-14　系统截图

Figure 5-14　The System Screenshot

游戏过程中，每消除一对相同图片，得分10，8×8格式的二维图片集，一共64张图片，共计32对，所以没过一关得分是320分，图5-9显示的是，消除34对后的得分。在每一难度下的每一关通过后，记分不清0，继续记分。但切换到不同难度时，得分会清0。

* + 1. 刷新功能的实现

游戏中每一关只有一次使用刷新功能的机会，此功能键触发的响应事件，会调用chongzai()方法，按照当前状态未消除的图片个数重新在二维按钮排列的其他位置排放其他图片。点“刷新”按钮的流程图如图5-15所示，前后变化如下图5-16、5-17所示。

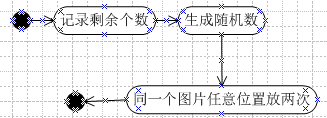


图 5-15　“刷新”流程图

Figure 5-15　“Refresh” Flow Diagram

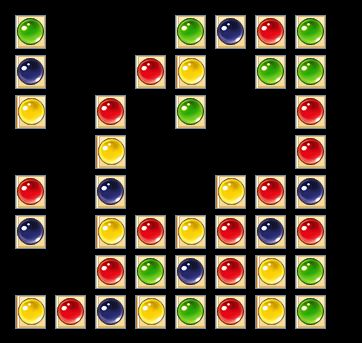
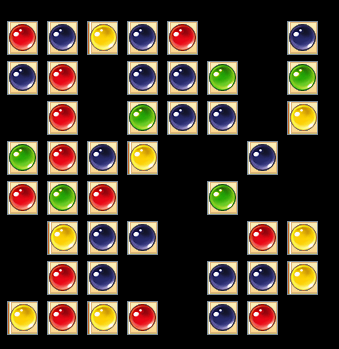


　　　　图 5-16刷新前系统截图　　　　　　　　　　　　图 5-17　刷新后系统截图

　　Figure 5-16　The System Screenshot　　　　　　　 Figure 5-17　 The System Screenshot

刷新功能一关游戏只能使用一次，不论是那一个难度下，一关内刷新功能只能使用一次，只有通过这一关，下一关中刷新功能才能使用。使用一次之后，按钮将不能点击，状态图如图5-18，系统截图如下图5-19所示。



图 5-18　刷新按钮状态图

Figure 5-18　“Refresh” Button State Diagram



图 5-19　点击一次之后系统截图

Figure 5-19　The System Screenshot

刷新功能部分代码如下：

**if** (e.getSource() == reLoad) {

chongzai();

reLoad.setEnabled(**false**);

chongzai()中部分代码如下：

**for** (**int** i = 0; i < jishushengyu / 2; i＋＋) {

kind = random.nextInt(Kinds) ＋ 1;

**do** {

randomx1 = random.nextInt(8);//0-8随机数

randomy1 = random.nextInt(8);

} **while** (map[randomy1][randomx1] > 0);

map[randomy1][randomx1] = kind;

**do** {

randomx = random.nextInt(8);

randomy = random.nextInt(8);

} **while** (map[randomy][randomx] > 0);

map[randomy][randomx] = kind;

}

* + 1. 重新开始游戏功能的实现

重新开始游戏功能实际也是开始新游戏功能，重新开始布局的过程。将图片成对的显示在场景中，并打乱调用的是newMap()方法。在该方法中，用了一个数组以数字标记的形式来记录加载的图片是哪种图片，在每次加载图片的时候是一张图片连续加载两次，以保证加载图片成对。然后用数组的随机方法将数组打乱。

* 1. 游戏的连接算法

连连看游戏中对于用户来说最简单的呈现，就是单击两个连在一起的相同的图像时，这两个图像就会消去，但是在Java代码中是如何实现的呢？连连看游戏的联通算法一般包括回溯法和堆栈实现的迷宫算法。

根据游戏规则能知道，即使不是紧邻的两个相同的图像块消去时也不能超过两次转弯。在本次开发的连连看中两个图像块可以消去的情况大致可以分为以下几种情况。

* + 1. 连连看连接方式的类型

连连看游戏的连接方式就是指两个图片之间连接的方式，一共可以分为以下这几种方法：

两点直线法示例如图5-20。



图 5-20　两点直线法示例图

Figure 5-20　Sample Graph

三点直角折线法示例如图5-21。

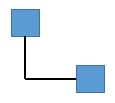


图 5-21　三点直线法示例图

Figure 5-21　Sample Graph

四点方形折线法示例如图5-22。

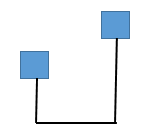


图 5-22　四点方形折线法示例图

Figure 5-22　Sample Graph

四点梯形折线法示例如图5-23。

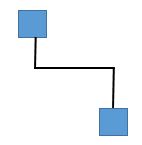


图 5-23　四点梯形折线法示例图

Figure 5-23　Sample Graph

下面对上面四种类型进行细致划分讲解：

两点直线法的连接类型包括：上下，左右

三点直角折线法包括：上左，上右，下左，下右，左上，右上，左下，右下

四点方形折线法包括：上左下，上右下，下左上，下右上，左上右，右上左，左下右，右下左

四点梯形折线法包括：上左上，下左下，上右上，下左下，左上左，右上右，左下左，右下右

* + 1. 连连看连接方式的分析

满足上述要求是第一步，即两个图片满足上次相连方式。第二步是满足图片相同。换言之，检验两个方块能否消掉，要让两个方块同时满足两个条件才行，就是两者配对并且连线成功，此处可分三种情况。从下面的这三种情况，我们可以知道，需要三个检测，这三个检测分别检测一条直路径。这样就会有三条路径。若这三条路径上都是空按钮，那么就刚好是三种直线（两个转弯点）把两个按钮连接起来了。

第一种情况：相邻。这种情况最简单。

第二种情况：不相邻。若不相邻，第一步在选中的第一个按钮的同行找到一个空按钮，此时分为两种情况：

情况A：找到空按钮后，再看选中的第二个按钮的横向到这个空按钮所在的列交叉处是否有按钮，若交叉处没有按钮，再从与第一个按钮到与它同行的那个空按钮之间是否有按钮，此时也没有按钮的话，路径通，反之则否。

情况B：第一步在选中的第一个按钮的同行找不到一个空按钮，此时在选中的第一个按钮的同列找一个空按钮。找到空按钮后，查看选中的第二个按钮竖向到这个空按钮所在行的交叉处是否有按钮。没有按钮的话，再查看选中的第一个按钮到与它同列的那个空按钮之间是否有按钮。没按钮的话，再判断与选中的第一个按钮同列的这个空按钮横向到与选中的第二个按钮同列有否有按钮，没有的话，路径通。

若满足以上情况的分析，也即路径通，选中的两个图片可消除。若以上情况分析均不满足，说明这两个按钮不可以消去。

具体判断流程图如图5-24。

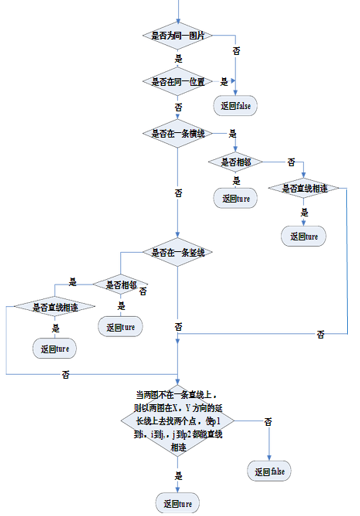


图 5-24　判断流程图

Figure 5-24　Flowchart

代码中定义的方法有：

判断在一列之内两图片之间是否全部是空白或直接相邻

判断在一行之内两图片之间是否全部是空白或直接相邻

是否可以一条直线相连

是否可以两条直线相连

是否可以三条直线相连

是否可以三条直线相连,似U形

是否可以三条直线相连,似之字形N

是否处于游戏区域的四条边的同一边上

* + 1. 部分核心代码

// 判断在一列之内两图片之间是否全部是空白或直接相邻

// 直接相连,因而不包含空白

**if** (Math.*abs*(posY1 - posY2) == 0) {

**return** **true**;

}

**int** a = posY1 < posY2 ? posY1 : posY2;

**int** b = posY1 < posY2 ? posY2 : posY1;

**for** (**int** j = a ＋ 1; j < b; j＋＋) {

**if** (map[posX1][j] != 0) {

**return** **false**;

}

}

// 是否可以一直线相连

**if** (posX1 != posX2 && posY1 != posY2) {

**return** **false**;

}

**if** (posX1 == posX2) {

**if** (containsAllOrNoneZeroInColumn(posX1, posY1, posX2, posY2)) {

**return** **true**;

}

}

**if** (posY1 == posY2) {

**if** (containsAllOrNoneZeroInRow(posX1, posY1, posX2, posY2)) {

**return** **true**;

}

}

// 是否可以三直线相连,似之字形N

**if** (isOnZigzagWith1Row2Cols(posX1, posY1, posX2, posY2)) {

**return** **true**;

}

**if** (isOnZigzagWith2Rows1Col(posX1, posY1, posX2, posY2)) {

**return** **true**;

}

// 是否处于游戏区域的4条边的同一边上

**if** ((posY1 == posY2 && posY2 == 0)

|| (posY1 == posY2 && posY2 == 8 - 1)

|| (posX1 == posX2 && posX2 == 0)

|| (posX1 == posX2 && posX2 == 8 - 1)) {

**return** **true**;

}

// 是否可以三直线相连,似之字形, 两行一列 Z

**int** moreX = posX1 < posX2 ? posX2 : posX1;

**int** lessX = posX1 < posX2 ? posX1 : posX2;

**for** (**int** i = lessX ＋ 1; i < moreX; i＋＋) {

**if** (containsAllOrNoneZeroInColumn(i, posY1, i, posY2)

&& containsAllOrNoneZeroInRow(i, posY1, posX1, posY1)

&& containsAllOrNoneZeroInRow(i, posY2, posX2, posY2)

&& map[i][posY1] == 0 && map[i][posY2] == 0) {

**return** **true**;

}

}

* 1. 本章小结

　　本章对连连看游戏进行了详细设计和实现，通过活动图和状态图，详细描述了操作的执行过程和一个特定对象的所有可能状态以及由于各种事件发生而引起的状态转换，为实现功能提供逻辑思路。

1. 系统测试

软件测试是保证软件质量的一种重要手段。

软件测试就是在正常条件与非正常条件下，分别对软件、应用程序进行操作或者设计出测试用例并执行测试用例，最后评价操作结果的过程。换言之，软件测试就是利用测试工具按照测试方案和流程对产品进行功能和性能测试，或者根据需要编写不同的测试用例，人为地设计和测试系统，对提出的测试方案中所有可能出现的问题进行分析和评估。

软件测试过程中应该故意地去促使错误的发生，也就是事情在不该出现的时候出现或者在应该出现的时候没有出现。从本质上说，软件测试是"找错"，在"找错"中发现软件的毛病。这也揭示了软件测试的真正意义。

* 1. 软件测试目的及意义

软件测试是指使用人工，或者自动的手段，来运行或测定某个软件产品系统的过程，其目的是检验是否满足规定的需求或者弄清预期的结果与实际结果之间的区别。测试包括找出破绽，但不包括跟踪错误和修正。

当然，软件测试也存在测试误区，在测试过程中，应尽量规避以下误区：

第一，测试并不仅仅是为了找出错误。通过分析错误产生的原因和错误的发生趋势，可以帮助项目管理者发现当前软件开发过程中的缺陷，以便及时改进；这种分析也有助于测试人员设计出有针对性的测试方法，改善测试的效率和有效性。

第二，没有发现错误的测试也是有价值的，完整的测试是评定软件质量的一种方法，应当把“尽早地和不断地进行软件测试”作为软件测试者的座右铭。

第三，完全测试是不可能的，测试需要终止，测试无法显示软件潜在的缺陷。

* + 1. 测试目的

Grenford J.Myers曾对软件测试的目的提出过以下观点：

第一，测试是为了发现程序中的错误而执行程序的过程；

第二，好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案；

第三，成功的测试是发现了至今为止尚未发现的错误的测试。

软件测试的目的决定了如何去组织测试。如果测试的目的是为了尽可能多地找出错误，那么测试就应该直接针对软件比较复杂的部分或是以前出错比较多的位置。如果测试目的是为了给最终用户提供具有一定可信度的质量评价，那么测试就应该直接针对在实际应用中会经常用到的商业假设。

在软件测试的实际应用中，我们对于系统性能方面的测试要多加重视。

系统在上线之后会有很多不能预知的性能问题，需要在上线之前实现进行模拟，以规避风险，包括大数据量访问，高并发数等等。当然也有很多应对手段，没有哪种手段可称为最完美，只有最合适的，需要灵活掌握，综合运用以达到最优程度，这是个很值得研究的领域。

针对连连看游戏的测试，分为两个方面，第一是游戏功能方面的测试，主要从界面布局、游戏响应两个方面进行。在游戏性能方面，从用户对本系统的满意度、易用性、稳定性着手，主要测试游戏过程的流畅程度。

界面布局方面，首先是在初始化游戏时，游戏界面的合理布局以及游戏所用到的图片是否能按要求完全显现。游戏开始后，玩家是否能对界面中的组件进行操作，其次，在响应事件结束后，界面能否按照要求反应变化。最后，游戏的开始和结束能否合理控制。

游戏响应方面，主要要求对用户操作组件的事件按需求正确处理，比如点击按钮时，应该怎样处理事件对象，选择游戏功能时，是否能完成游戏功能。

* + 1. 测试意义

软件测试的真正意义在于发现错误，而不在于验证软件是正确的，软件测试是发现软件中错误和缺陷的主要手段。在一般情况下，软件测试的过程与整个软件开发过程基本上是平行进行的。当然，测试计划应该在需求分析阶段就已经开始制定了。随后工作则是伴随着软件的开发过程逐步展开。由于缺陷在开发与维护的任何阶段都有可能发生，并有此产生一个或多个bug，包括错误、误解和冗余，有时候甚至会误导开发者。所以软件测试是发现并指出软件系统缺陷的过程。

软件测试是软件开发过程中的很重要的一项内容，是用来确认一个程序的品质或性能是否符合开发之前所提出的要求的过程。它不仅是软件开发阶段的有机组成部分，而且在整个软件工程(即软件定义、设计和开发过程)中占据相当大的比重。

* 1. 测试方法分类

　　从是否关心软件内部结构和具体实现的角度划分，可分为以下三类：

白盒测试：白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。它是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检测程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。

黑盒测试：黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能的前提下，通过测试来检测每一个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。黑盒测试方法主要有等价类划分、边值分析、因果图、错误推测等，主要用于软件确认测试。

灰盒测试：灰盒测试关注输出对于输入的正确性，同时也关注内部表现，但这种关注不像白盒那样详细、完整，只是通过一些表征性的现象、事件、标志来判断内部的运行状态，有时候输出是正确的，但内部其实已经错误了，这种情况非常多，如果每次都通过白盒测试来操作，效率会很低，因此需要采取这样的一种灰盒的方法。

从是否执行程序的角度，可以划分为两类：

静态测试：静态方法是指不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。对需求规格说明书、软件设计说明书、源程序做结构分析、流程图分析、符号执行来找错。静态方法通过程序静态特性的分析，找出欠缺和可疑之处，例如不匹配的参数、不适当的循环嵌套和分支嵌套、不允许的递归、未使用过的变量、空指针的引用和可疑的计算等。

动态测试：动态测试方法是指通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率、正确性和健壮性等性能。这种方法由三部分组成：构造测试用例、执行程序、分析程序的输出结果。

从软件开发的过程按阶段划分，可以划分为：单元测试、集成测试、确认系统、系统测试、验收测试、回归测试、Alpha测试、Beta测试

* 1. 测试步骤

此次测试过程按4个步骤来进行，即：单元测试、集成测试、确认测试和系统测试。

首先，进行的单元测试：单元测试就是对源代码实现的每一个程序单元的进行测试，检查每个程序模块是否能够正确地实现规定的功能。

之后，进行了集成测试：主要是根据软件体系结构，把已经测试过的单元模块组装起来，在组装过程中，检查程序结构组织是否正确。

接下来，进行确认测试：确认测试则是检查目前已实现的软件是否符合需求分析确定了的各种需求，以及软件配置是否完善、正确。

最后，是进行系统的测试：系统测试是把已经经过确认的软件放入实际运行环境中，通过与其他系统的成分组合在一起进行的测试。严格地说，系统测试已经超过了软件工程的范围。但在实际工作中这几个阶段却是相互交叉的。如图6-1所示。

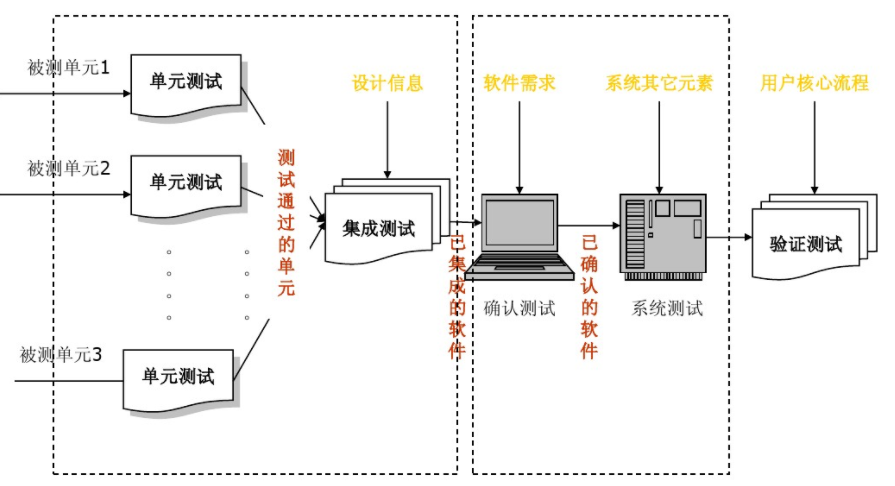


图 6-1　测试过程图

Figure 6-1　The Testing Process Diagram

单元测试：在代码级别上，针对构成软件的最小组件即单元进行测试，是测试过程中粒度最小的测试。其测试目的在于验证单元功能的正确性，与设计的符合程度。

集成测试：在单元测试的基础上，将所有经过单元测试的模块按照设计要求组装成子系统或者系统进行测试。其测试目的是为了关注各个模块之间的交互是否正确。

确认与系统测试：将已经集成好的软件系统，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合在一起。在实际运行（使用）环境下，对计算机系统进行一系列的组装测试和确认测试。其目的在于发现与系统定义不符或矛盾的地方以及系统各个部分是否可以协调工作。

* 1. 测试用例

游戏测试相当于是软件测试的子类，它继承了软件测试的特性，同时也衍生出了自己的特殊属性。测试的目的是发现软件中存在的缺陷。游戏测试主要有游戏情节的测试、游戏世界的平衡测试、游戏文化的测试三个内容。

由于游戏的特殊性，所以游戏测试则主要分为两部分组成，一是传统的软件测试，二游戏本身的测试，同时它又是游戏所以还涉及到娱乐性，可玩性等独有特性，所以测试的面相当的广。

要了解如何测试游戏必需了解如何做游戏，了解它的开发过程，才能真正的测好游戏。游戏要成功，其基本的必要条件有三。分别为Vision(设计)、technology(技术)和Process(过程)。 现针对游戏功能，设计如下几个测试用例，见表6-1测试用例表。

表 6-1　测试用例

Table 6-1　Test Case

| 测试项 | 用例编号 | 操作步骤 | 预期结果 | 测试结果 |
| --- | --- | --- | --- | --- |
| 游戏初始化界面显示 | 1 | 运行程序 | 打开连连游戏弹出“开始游戏确认”的对话框 | 多次试验与预期结果一致 |
| 2 | 单击对话框的“确认”按钮 | 开始游戏：初始化游戏界面、开始游戏倒计时 | 多次试验与预期结果一致 |
| 3 | 关闭对话框，单击对话框关闭按钮 | 开始游戏：初始化界面、开始倒计时 | 多次试验与预期结果一致 |
| 4 | 观察游戏开始后：游戏界面布局中的图片是否成对出现 | 游戏图片成对出现 | 多次试验与预期结果一致 |
| 游戏功能-困难选择功能 | 1 | 鼠标单击下拉列表，选择任意一种游戏难度 | 弹出“开始游戏确认”的对话框 | 多次试验与预期结果一致 |
| 2 | 选择“简单”模式 | 初始化游戏，游戏界面中的图片有四种，图片成对出现 | 多次试验与预期结果一致 |
| 3 | 选择“中等”模式 | 初始化游戏，游戏界面中的图片有八种，包含“简单”模式中的四种，且图片成对出现 | 多次试验与预期结果一致 |
| 4 | 选择“困难”模式 | 初始化游戏，游戏界面中的图片有十二种，包含“简单”、“中等”模式下的八种图片，且图片成对出现 | 多次试验与预期结果一致 |

续表6-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项 | 用例编号 | 操作步骤 | 预期结果 | 测试结果 |
| 游戏功能-困难选择功能 | 5 | 选择“变态”模式 | 初始化游戏，游戏界面中的　图片有十五种，包含“简单”、“中等”、“困难”模式的十二种，且图片成对出现 | 多次试验与预期结果一致 |
| 游戏功能-选中图片功能 | １ | 点击任意一张图片 | 图片被选中后，变暗，进入激活状态。 | 与预期结果一致 |
| 游戏功能-消除功能-一条直线路径 | 1 | 依次选中两张相邻且相同图片按钮 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 2 | 依次选中两张相邻但不相同图片按钮 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 游戏功能-消除功能-两条直线路径 | 1 | 依次选中两张图片相同且两张图片之间路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 2 | 依次选中两张图片相同且两张图片之间路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 3 | 依次选中两张图片相同且两张图片之间路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 4 | 依次选中两张图片相同且两张图片之间路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 5 | 依次选中两张相同图片且均不满足上述用例1、2、3、4的路径要求，即两条直线相连路径不通。 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |

续表6-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项 | 用例编号 | 操作步骤 | 预期结果 | 测试结果 |
|  | 6 | 依次选中两张不同图片且均不满足上述用例1、2、3、4的路径要求，即两条直线相连路径不通。 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 游戏功能-消除功能-两条直线路径 | 7 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 8 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 9 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 10 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 游戏功能-消除功能-三条直线路径 | 1 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 2 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 3 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 4 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 5 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |

续表6-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项 | 用例编号 | 操作步骤 | 预期结果 | 测试结果 |
| 游戏功能-消除功能-三条直线路径 | 6 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 7 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 游戏功能-消除功能-三条直线路径 | 8 | 依次选中两张相同图片，且连接路径满足 | 第一个选中的图片变暗，选中第二张图片的同时两张图片同时消除 | 多次试验与预期结果一致 |
| 9 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 10 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 11 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 12 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 13 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 14 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 15 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |

续表6-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项 | 用例编号 | 操作步骤 | 预期结果 | 测试结果 |
| 游戏功能-消除功能-三条直线路径 | 16 | 依次选中两张不同图片且满足 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 17 | 依次选中两张相同图片，且均不满足用例1、2、3、4、5、6、7、8的路径要求，即三条直线相连路径不通 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 18 | 依次选中两张不同图片，且均不满足用例1、2、3、4、5、6、7、8的路径要求，即三条直线相连路径不通 | 第一个选中的图片变暗，选中第二张图片的同时第一站图片变亮，第二张图片变暗，且两张图片没有消除 | 多次试验与预期结果一致 |
| 游戏功能-重新开始功能 | 1 | 不对图片做任何操作点击重新开始按钮 | 开始当前难度模式下的新一局游戏，重置记分、倒计时 | 多次试验与预期结果一致 |
| 2 | 对图片做任意操作后点击重新开始按钮 | 开始当前难度模式下的新一局游戏，重置记分、倒计时 | 多次试验中小概率出现缺少图片现象 |
| 游戏功能-记分、关数功能 | 1 | 正确消除一对图片后查看记分情况和关数情况 | 消除一对加10分，未消除完关数不变 | 多次试验与预期结果一致 |
| 2 | 正确消除所有图片 | 共得320分，且关数＋１ | 多次试验与预期结果一致 |
| 3 | 未消除一对图片 | 分数、关数均不变 | 多次试验与预期结果一致 |
| 游戏功能-倒计时功能 | 1 | 不做任何操作 | “流沙”从上到下流动，且最后五秒沙漏上方出现倒计时数秒字样，倒计时结束后，弹出对话框“时间到，游戏结束！” | 提示文字显示不全 |
| 2 | 规定时间内完成消除，开始下一关 | 倒计时装置重置 | 多次试验与预期结果一致 |
| 3 | 点击“重新开始”按钮或“刷新”按钮，或者更换难度 | 倒计时装置重置 | 多次试验与预期结果一致 |
| 游戏功能-刷新功能 |  | 点击刷新按钮 | 重新排列布局，显示操作前后个数相同种类不同的图片，图片种类符合当前难度要求，且图片成对出现，刷新按钮失效 | 多次试验与预期结果一致 |

* 1. 测试结果

本系统在测试和运行过程中，有以下几方面优点：界面友好，操作方便。系统界面简单，功能较完善，易上手，可以激发玩家的乐趣；通用性好，扩展性强。本系统可以适用于各种人群，具有优越的通用性能。但部分系统功能需完善，由于时间和技术因素，交互入口等方面还有待拓展。

第一，连连看基本游戏功能表现不错（即选择两个图形，判断是否满足一定条件，条件即两个图形一致且二者之间存在转弯少于3的路径。当判定满足这一条件后，两张图片被消除），在运行一次游戏过程中，点击两个不同的图片时，不会消除两个图片，在点击第一个图片时图片变暗，点击第二个图片时因为图片不匹配，第二个图片变暗，两张图片没有消失。在运行一次游戏过程中，点击满足存在多欲三个连接路径的两张图片，同样也不能消除。

第二，图片的排列正确。在开始新一局的游戏时，图片的排列清晰明了，满足随机的同时，成对出现。

第三，记分功能正确。没消除一对图片得分10。得分数在游戏界面上能明确标识，直观的显示出当前的游戏状态，包括游戏得分情况和闯关关数情况。

第四，沙漏计时功能有缺陷。沙漏状计时能正确，倒计时五秒时，提示问字有误，该显示的内容显示不全。

第五，游戏的开始和重新开始表现正确。在游戏开始前，弹出的提示框根据指示点击确定后，能够开始新游戏，即开始记分、计时。重新开始游戏功能也表现正确，能中断当前游戏，开始一局新的游戏，同样开始记分、计时，并且可以多次开始新游戏。

第六，游戏关闭和结束有缺陷。窗口游戏在关闭时，可以直接点击窗口的关闭按钮。游戏结束时，弹出的提速框表现正常，但是沙漏上方的JLabel表现异常，文字显示不全。

第七，游戏难度选择功能正常。游戏难度有4个等级，每个等级选择后，操作模块的界面能显示总体个数相同但不同种类的图片，难度越高，种类越多。在选择了一个难度模式之后，同时也开始一局新游戏。

第八，每一个难度下可以有很多关卡，每一个关游戏过关后可以直接进入下一关。

第九，刷新功能（游戏提示功能）正常。在游戏过程中只可以使用一次刷新功能，重新排列当前游戏未消除的图片个数的图片。

第十一，游戏要最大限度地实现了易维护性和易操作性，游戏运行稳定、安全可靠。

* 1. 本章小结

本章主要明确了测试的目的和测试的重要性，通过设计测试用例得出，游戏中的消除功能、计时功能、记录得分和关数功能、重新开始功能、刷新功能基本实现。

结 论

2017年二月底三月初确定了毕业设计课题至今，毕业设计总体完成情况良好，总体达到了毕设任务书上的要求。

我的毕业设计课题是连连看游戏设计与实现，该项目通过Java语言实现，为此我在着手之前简单了解了Java游戏开发的现状和走向，决定通过桌面窗体的形式完成项目。在本科学习大纲中，AWT体系结构、Swing体系结构并不是Java语言学习的重点和难点，但我们曾在软件开发工具的课程上简单学习过C#语言，在某种程度上来说，这两种面向对象的编程语言有很多相通的地方，这恰恰能帮助我更好更深入地理解Java Swing基本体系结构，也更容易上手去实践这个项目。所以说毕业设计是对本科学习过程的总结和实践，之于我而言，本科学习带给我的更多的是思想。

连连看游戏并不陌生，基本要求是要完成两个相同图片的消除。首先要考虑的就是项目要实现的功能了，我在基础要求上增加了计时功能，重置功能，和一个重新排列当前图片的功能，这些功能都是比较常见也是很容易理解的功能，增加功能主要是丰富项目操作的娱乐性，也使得项目内容更加丰满。其次设计游戏界面，用到最多的是JButton，游戏界面中还需要有提示或标识状态用的窗口或者字符串（JDiolo、JLable等），为实现计时功能绘制了漏斗。游戏界面做到了简洁大方一目了然，在有一定逻辑的基础上能够带领游戏用户体验游戏过程，也是吸引用户很直观的原因。最核心的部分是实现连接消除的功能，在本项目中，我将这种连接规则具体地分情况讨论，在将这些情况所涉及的判断选择封装成一个个的方法，每种情况调用一个或多个这样的方法来判断是否可以在三条路径之内完成消除。在最后的测试阶段，所要求的功能已完成，游戏整体的逻辑感良好，整个项目不论在功能方面还是界面方面都还有很大的进步空间。论文方面是一次新的尝试，不同于平时的实验报告，论文有更严密的格式要求，也更需要我对项目有很层次化地理解，在完成论文的过程中，总结和体会一个项目完成的过程和在项目实现的过程中学习到的点点滴滴，将这些分层次，分主次的记录下来是对我论文撰写能力的锻炼更是对知识体系的完善和总结。

毕业设计是个很好的机会用于自我总结和进步，单纯的理论知识学习早已不能满足当下社会对于人才的要求，实践能力是大学生优秀素质的体现，也是对大学毕业生是否合格毕业的检验标准。四年的本科学习提供了很多实践条件和机会，在老师和小组团队的合作帮助下完成一些项目，用过C/C＋＋、C#、Wab也用过Java。这一次的毕业设计是比较完整的一次独立实践的过程，过程中也遇到过很多问题，从不懂到懂，这些问题需要我自己去学习去领悟去应用，解决这些问题的过程让我受益匪浅，如果自我评价的话项目按预期完成了，但其实还有很多可以改进的地方，对于本科学习的编程思想掌握良好，对于项目设计过程熟悉应用，总体来说较为满意，希望在今后的学习实践中能有创造性的设计和发展。

致 谢

四载春秋如指尖缝隙里淌过的流水，稍纵即逝。

大三的那一年夏天，软件学院提前迎来了毕业季，我还记得在明向校区顶着烈日把校园装进相框的点滴。大四毕业在即，我在离家一千多公里的南方艳阳里格外想念明向校区的绿荫。至此，毕业论文也将画上句号，此刻，内心涌起的满满是对生活的感动和感激。

在此我要郑重感谢所有导师传道、授业、解惑，帮助我逐渐褪去曾经的青涩，走向稳重成熟。感谢每一位带来精彩课堂的老师，感谢老师在我完成毕业设计过程中的帮助，谢谢老师们不吝赐教、一丝不苟的认真态度！学生终身难忘！同时也要感谢著书前辈，感谢前辈的奋斗钻研和分享！

感谢各位同学的帮助与陪伴，谢谢宿舍小伙伴们每次课程设计中对我的严格要求，谢谢你们一次又一次在我烦躁不安垂头丧气时不懈地鼓励和信任！

最后我要感谢2013年的高考，这一机遇让我能以太原理工大学学子的身份踏入这所百年高校，并在此后，得以享受母校带给学子的便利，得以接受母校的谆谆教诲，得以抓住母校带来的人生机遇。感谢母校四年的培养，感谢母校带给我的荣誉！在母校的四年时光，因青春而美好，也因美好而青春。“求实、创新”的校训，见证着我的奋斗历程和心路成长，为我指明未来方向，我将带着母校的教导，回馈母校，为这个社会创造更多的价值。

祝同窗好友一帆风顺！祝老师们工作顺利、身体健康！祝母校的明天越来越好！

学生：贺洋

2017 年5月17日

参考文献

外文文献

**Java 2 Micro Edition and the World of Java**

**1 Introduction**

The computer revolution of the 1970s increased the demand for sophisticated computersoftware to take advantage of the ever-increasing capacity of computers to process data.The C programming language became the linchpin that enabled programmers to buildsoftware that was just as robust as the computer it ran on.

As the 1980s approached, programmers were witnessing another spurt in the evolutionof programming language. Computer technology advanced beyond the capabilities of the C programming language. The problem wasn’t new. It occurred previously and caused the demise of generations of programming languages. The problem was thatprograms were becoming too complicated to design, write, and manage to keep up with the capabilities of computers. It was around this time that a design concept based on Simula 67 and Smalltalk (from the late 1960s) moved programming to the next evolutionary step. This was the period when object-oriented programming (OOP), and with it a new programming language called C＋＋, took programmers by storm.

In 1979, Bjarne Stroustrup of Bell Laboratories in New Jersey enhanced the C programming language to include object-oriented features. He called the language C＋＋. (The ＋＋ is the incremental operator in the C programming language.) C＋＋ is truly an enhancement of the C programming language, and it began as a preprocessor language that was translated into C syntax before the program was processed by the compiler.

Stroustrup built on the concept of a class (taken from Simula 67 and Smalltalk), from which instances of objects are created. A class contains data members and member functions that define an object’s data and functionality. He also introduced the concept of inheritance, which enabled a class to inherit some or all data members and member functions from one or more other classes—all of which complements the concepts of object-oriented programming.

By 1988, ANSI officials standardized Stroustrup’s C＋＋ specification.

**2 Enter Java**

Just as C＋＋ was becoming the language of choice for building industrial-strength applications, another growth spurt in the evolution of programming language was budding, fertilized by the latest disruptive technology—the World Wide Web. The Internet had been a well-kept secret for decades before the National Science Foundation (who oversaw the Internet) removed barriers that prevented commercialization. Until 1991 when it was opened to commerce, the Internet was the almost exclusive domain of government agencies and the academic community. Once the barrier to commercialization was lifted, the World Wide Web—one of several services offered on the Internet— became a virtual community center where visitors could get free information about practically anything and browse through thousands of virtual stores.

Browsers power the World Wide Web.A browser translates ASCII text files written in HTML into an interactive display that can be interpreted on any machine. As long as the browser is compatible with the correct version of HTML and HTTP implementation, any computer running the browser can use the same HTML document without having to modify it for a particular type of computer, which was something unheard of at the time. Programs written in C or C＋＋ are machine dependent and cannot run on a different machine unless the program is recompiled.

The success of the Internet gave renewed focus to developing a machine-independent programming language. And the same year the Internet was commercialized, five technologists at Sun Microsystems set out to do just that. James Gosling, Patrick Naughton, ChrisWarth, Ed Frank, and Mike Sheridan spent 18 months developing the programming language they called Oak, which was renamed Java when this new language made its debut in 1995. Java went through numerous iterations between 1991 and 1995, during which time many other technologists at Sun made substantial contributions to the language. These included Bill Joy, Arthur van Hoff, Jonathan Payne, Frank Yelin, and Tim Lindholm.

Although Java is closely associated with the Internet, it was developed as a language for programming software that could be embedded into electronic devices regardless of the type of CPU used by the device. This is known as the EmbeddedJava platform and is in continuous use today for closed systems.

The Java team from Sun succeeded in creating a portable programming language, something that had eluded programmers since computers were first programmed. Their success, however, was far beyond their wildest dreams. The same concept used to make Java programs portable to electronic devices also could be used to make Java programs run on computers running Microsoft Windows, UNIX, and Macintosh. Timing was perfect. The Internet/intranet had whetted corporate America’s appetite for cost-effective, portable programs that could replace mission-critical applications within the corporation. And Java had proven itself as a programming language used to successfully develop machine-independent applications.

**3 Java Virtual Machine**

Writing Java programs is similar to writing C＋＋ programs in that the programmer writes source code that contains instructions into an editor, or in an integrated development environment, and then the source code is compiled. However, that’s where Java and C＋＋ part ways. The compiling and linking process of a C＋＋ program results in an executable that can be run on an appropriate machine. In contrast, the Java compiler converts Java source code into bytecode that is executed by the Java Virtual Machine (JVM).

Machine-specific instructions are not included in bytecode. Instead, they already reside in the JVM, which is machine specific. This means that the bytecode might contain fewer instructions that need to be translated than a comparable C＋＋ program.

A lthough the Java compiler generates bytecode that must be interpreted by the JVM at run time, the number of instructions that need translation are usually minimal and

have already been optimized by the Java compiler.

**4 Back to the Future: J2ME**

Remember that Java began as a programming language to create programs for embedded systems—microcomputers found in consumer and industrial products such as those used to control automobiles and appliances. The development team at Sun worked on Java in the early 1990s to address the programming needs of the fledgling embedded computer market, but that effort was sidetracked by more compelling opportunities presented by the Internet.

As those opportunities were addressed, a new breed of portable communications devices opened other opportunities at the turn of the century. Cell phones expanded J 2 M E : T h e C o m p l e t e R e f e r e n c e from voice communications devices to voice and text communications devices. Pocket electronic telephone directories evolved into personal digital assistants. Chipmakers were releasing new products at this time that were designed to transfer computing power from a desktop computer into mobile small computers that controlled gas pumps, cable television boxes, and an assortment of other appliances.

The time was right for the next evolution of Java. However, instead of beefing up Java with additional APIs, the team at Sun, along with the Java Community Process Program, dismantled both the Java programming language and the Java Virtual Machine. They

stripped down Java APIs and the JVM to the minimum coding required to provide intelligence to embedded systems and microcomputer devices. This was necessary

because of resource constraints imposed upon the hardware design of these devices. The result of their efforts is J2ME. J2ME is a reduced version of the Java API and Java Virtual Machine that is designed to operate within the sparse resources available in the new breed of embedded computers and microcomputers.

**5 How J2ME Is Organized**

Traditional computing devices use fairly standard hardware configurations such as a display, keyboard,mouse, and large amounts of memory and permanent storage. However, the new breed of computing devices lacks hardware configuration continuity among devices. Some devices don’t have a display, permanent storage, keyboard, or mouse. And memory availability is inconsistent among small computing devices. The lack of uniform hardware configuration among the small computing devices poses a formidable challenge for the Java Community Process Program, which is charged with developing standards for the JVM and the J2ME for small computing devices.

J2ME must service many different kinds of small computing devices, including screenphones, digital set-top boxes used for cable television, cell phones, and personal digital assistants. The challenge for the Java Community Process Program is to develop a Java standard that can be implemented on small computing devices that have nonstandard hardware configurations. The Java Community Process Program has used a twofold approach to addressing the needs of small computing devices. First, they defined the Java run-time environment and core classes that operate on each device. This is referred to as the configuration. A configuration defines the Java Virtual Machine for a particular small computing device. There are two configurations, one for handheld devices and the other for plug-in devices. Next, the Java Community Process Program defined a profile for categories of small computing devices. A profile consists of classes that enable developers to implement features found on a related group of small computing devices.

**6 J2ME configurations**

There are two configurations for J2ME as of this writing. These are Connected Limited Device Configuration (CLDC) and the Connected Device Configuration (CDC). The CLDC is designed for 16-bit or 32-bit small computing devices with limited amounts of memory.

CLDC devices usually have between 160KB and 512KB of available memory and are battery powered. They also use an inconsistent, small-bandwidth network wireless connection and may not have a user interface. CLDC devices use the KJava Virtual

Machine (KVM) implementation, which is a stripped-down version of the JVM. CLDC devices include pagers, personal digital assistants, cell phones, dedicated terminals, and

handheld consumer devices with between 128KB and 512KB of memory. CDC devices use a 32-bit architecture, have at least two megabytes of memory available, and implement a complete functional JVM. CDC devices include digital set-top boxes, home appliances, navigation systems, point-of-sale terminals, and smart phones.

**7 J2ME Profiles**

A profile consists of Java classes that enable implementation of features for either a particular small computing device or for a class of small computing devices. Small computing technology continues to evolve, and with that, there is an ongoing process of defining J2ME profiles. Seven profiles have been defined as of this writing. These are the Foundation Profile, Game Profile, Mobile Information Device Profile, PDA Profile, Personal Profile, Personal Basis Profile, and RMI Profile.

The Foundation Profile is used with the CDC configuration and is the core for nearly all other profiles used with the CDC configuration because the Foundation Profile contains core Java classes.

The Game Profile is also used with the CDC configuration and contains the necessary classes for developing game applications for any small computing device that uses the CDC configuration.

The Mobile Information Device Profile (MIDP) is used with the CLDC configuration and contains classes that provide local storage, a user interface, and networking capabilities to an application that runs on a mobile computing device such as Palm OS devices. MIDP is used with wireless Java applications.

The PDAProfile (PDAP) is used with the CLDC configuration and contains classes that utilize sophisticated resources found on personal digital assistants. These features include better displays and larger memory than similar resources found on MIDP mobile devices (such as cell phones).

The Personal Profile is used with the CDC configuration and the Foundation Profile and contains classes to implement a complex user interface. The Foundation Profile provides core classes, and the Personal Profiles provide classes to implement a sophisticated user interface, which is a user interface that is capable of displaying multiple windows at a time.

The Personal Basis Profile is similar to the Personal Profile in that it is used with the CDC configuration and the Foundation Profile. However, the Personal Basis Profile provides classes to implement a simple user interface, which is a user interface that is capable of displaying one window at a time.

The RMI Profile is used with the CDC configuration and the Foundation Profile to provide Remote Method Invocation classes to the core classes contained in the Foundation Profile.

There will likely be many profiles as the proliferation of small computing devices ontinues. Industry groups within the Java Community Process Program (java.sun.com/ boutjava/communityprocess) define profiles. Each group establishes the standard rofile used by small computing devices manufactured by that industry.

ACDC profile is defined by expanding upon core Java classes found in the Foundation rofile with classes specifically targeted to a class of small computing device. These device-specific classes are contained in a new profile that enables developers to create ndustrial-strength applications for those devices. However, if the Foundation Profile s specific to CDC, not all profiles are expanded upon the core classes found in the oundation Profile. eep in mind that applications can access a small computing device’s software and rdware features only if the necessary classes to do so are contained in the JVM and in the profile used by the developer.

**8 J2ME and Wireless Devices**

With the dramatic increase and sophistication of mobile communications devices such as cell phones came demand for applications that can run on those devices. Consumers and corporations want to expand mobile communications devices from voice communications to applications traditionally found on laptops and PCs. They want to send and receive

email, store and retrieve personal information, perform sophisticated calculations, and

play games.

Developers, mobile communications device manufacturers, and mobile network

p roviders are anxious to fill this need, but there is a serious hurdle: mobile communications devices utilize a number of different application platforms and operating systems.Without tweaking the code, an application written for one device cannot run on another device.

Mobile communications devices lack a standard application platform and operating system, which has made developing applications for mobile communications devices

a risky economic venture for developers.

The lack of standards is nothing new to computing or to any developing technology. Traditionally, manufacturers of hardware devices try to corner the market and enforce their own proprietary standard as the de facto standard for the industry. Usually one upstart succeeds, as in the case of Microsoft. Other times, industry leaders form a consortium, such as the Java Community Process Program, to collectively develop a standard.

The Wireless Application Protocol (WAP) forum became the initial industry group

that set out to create standards for wireless technology. Ericsson, Motorola, Nokia, and Unwired Planet formed theWAP forum in 1997, and it has since grown to include nearly all mobile device manufacturers, mobile network providers, and developers. The WAP forum created mobile communications device standards referred to as theWAP standard.

The WAP standard is an enhancement of HTML, XML, and TCP/IP. One element of this standard is the Wireless Markup Language specification, which consists of a blend of HTML and XML and is used by developers to create documents that can be displayed by a microbrowser. A microbrowser is a diminutive web browser that operates on a mobile communications device.The WAP standard also includes specifications for aWireless Telephony ApplicationInterface (WTAI) specification and the WMLScript specification. WTAI is used to create an interface for applications that run on a mobile communications device. WMLScript is a stripped-down version of JavaScript.

**9 J2ME applications**

J2ME applications referred to as a MIDlet can run on practically any mobile

communications device that implements a JVM and MIDP. This encourages developers to invest time and money in building applications for mobile communications devices without the risk that the application is device dependent. However, J2ME isn’t seen as a replacement for the WAP specification because both are complementary technologies. Developers whose applications are light-client based continue to use WML and

WMLScript. Developers turn to J2ME for heavier clients that require sophisticated

processing on the mobile communications device.

**J2ME和Java领域**

1、 介绍

20世纪70年代以来随着计算机革命的开始，对计算机先进软件的需求大大增加，从而可以充分利用功能日益增强的精密的计算机的处理数据的能力。C编程语言逐渐成为支柱，使程序员开发软件像计算机运行一样流畅。

80年代以来，程序员又目睹了编程语言领域的又一次变革的高潮。C语言的编程能力已经不能满足计算机的技术发展的需要。这问题并不是新问题。它造成了一代又一代的编程语言的新老更替。问题是，它使得程序设计过于复杂，从而使计算机软件的设计，编写和开发落后于硬件的发展。就是这个时候，两种基于设计概念的编程语言Simula 67和Smalltalk （从上世纪60年代末）带来了接近编程语言未来前景的循序渐进的步骤。这期间，当面向对象编程（ OOP ）， 与它一种新的编程语言，所谓的C ＋ ＋在程序员中掀起了一场风暴。

1979年， Bjarne Stroustrup的在新泽西州的贝尔实验室增强了C 语言，使其具有面向对象的特点即所谓的C ＋ ＋ 语言 （ ＋ ＋是C编程语言增强的承载符号 ）。 C ＋ ＋是一个真正的提高的C编程语言，它开始是一种前置语言，该计划最初是一种编译工具。 Stroustrup建立类的概念（借用了Simula 67和Smalltalk 中的概念） ，由类则可以创建实例对象。一个类包含数据成员和定义对象数据和功能的成员函数。他还介绍了继承的概念，使一类继承其他一个或多个类的部分或全部数据成员或成员函数，职能由一个或多个其他类别-所有这些概念就是面向对象的编程。

1988年由ANSI官员将C ＋ ＋标准化。

2、走进JAVA

正当C ＋ ＋作为企业开发所选择的语言时，由于最新的突破性技术-万维网的出现，别一种编程语言正在萌芽，并逐渐成熟。几十年来互联网对普通人来说还是个秘密，直到前美国国家科学基金会（监督互联网的机构）拆除了防止互联网商业化的壁垒。1991年当它向商贸开放时，互联网几乎是政府机构和学术团体的专利。一旦商业化的障碍被取消，万维网提供的几个服务，在互联网上成为虚拟社区中心，游客可以得到几乎任何的免费的资讯并浏览数以千计的虚拟商店。

浏览器加强了万维网。浏览器转换将用HTML编写的ASCII文本文件转化成为一个互动的显示功能，并可以任何计算机上编译运行。只要浏览器是和HTML和HTTP兼容的正确版本， 在任何电脑上运行的浏览器可以使用相同的HTML文件，而不是指定为某一特定类型的计算机，而在以前是闻所未闻的。而C或C ＋ ＋是依赖于机器的，并且不能运行于不同机器上，除非是重新编译。

互联网的成功使人们将重点放在发展独立于机器的编程语言上。和同年在互联网上被商品化，科技工作者SunMicrosystems詹姆斯戈斯林，帕特里克诺顿， chriswarth ，和麦克谢里登花了18个月发展规划，并在1995年向公众发布，他们所谓的Oak，后来改名为Java。Java 在1991年和1995年期间经历了无数次的变革。那个时候，在Sun公司的许多其他科技人员对该语言的发展做出了重大贡献。Joy，阿瑟范霍夫，乔纳森佩恩， Yelin ， 添佩尔都分享了成功的喜悦。

虽然Java是和互联网密切相关，但开始是为了开发一种可以嵌入到电子器件中去的软件的编程语言，而无论所用装置的CPU类型。这也是众所周知的所谓embedded java平台。

Java团队从Sun成功地建立了一种可移植性的编程语言，电脑程序员进行了首次编程，一些已经实现。他们的成功远远超出了他们的梦想。同样可以使Java程序进行在各种可移植的电子装置上，也可以使Java程序在安装了Microsoft Windows ， UNIX和Macintosh的计算机上运行。 这个时机很完美。因特网/内部网已使美国企业为符合公司成本效益，而开发应用于公司的可移植程序的兴趣。Java已被证明可以作为一种成功开发与机器无关的应用程序的编程语言。

3、Java虚拟机

写Java程序和写C ＋ ＋程序是相似的，这程序员可以编写： 包含指示的源代码，成为一个编辑器，或在一个综合发展环境中对源代码进行编译。不过，这也就是Java和C ＋ ＋ 一部分的方式。该编译和链接的过程中一个C ＋ ＋程序的结果可以运行在一个适当的可执行的机器上。与此相反，Java编译器则将Java 源代码转换成字节，即由Java虚拟机（ JVM ）执行 。

机器的具体指示，则不会包括在字节中。相反，他们已经包括在JVM的，这是属于具体机器的。这意味着该字节可能包含的C ＋ ＋

程序要翻译的指令要少。

4、未来的世界：J2ME

请记住， Java是在消费和工业产品中作为一种编程语言创建程序嵌入式系统微机发现的开始，如那些用来控制汽车和家用电器的微机。在20世纪90年代初，开发团队在Sun的工作利用Java以解决幼稚的嵌入式电脑市场的编程的需要，但这种努力是更侧重于来自互联网的吸引力的机会。

在世纪交替之年由于这些机会已得到解决，一种新的便携式通信带来了其他设备的机会。从语音通信设备的语音到文本通信设备。而便携式电子电话簿演变成个人数码助理。在这时候，芯片制造商所推出的新产品，其目的是为了从桌上型电脑转移计算能力到移动的小型电脑控制的燃气水泵，电缆电视盒，和其他组合的其他电器。

为下一步的演变的Java的时机是合理的 。不过，不是Java 与额外的空气污染指数加强，在Sun公司的该小组，即Java Community Process计划， 拆除均采用Java编程语言和Java虚拟机。他们从嵌入式系统和微机设备剥离下来的Java的API和JVM到最低限度编码须提供情报。由于这些设备资源方面的限制强加的硬件设计，这是必要的。他们的努力结果是J2ME的。 J2ME是减少了Java API和Java虚拟机是设计用来运作稀疏可用的资源在新的品种的嵌入式计算机和微机的版本。

5、组织J2ME

传统计算设备需要使用标准相当的硬件配置，如显示器，键盘，鼠标和大量的记忆和永久储存。不过，其中新一代的计算设备缺乏的硬件配置的连续性。有些设备不具有显示屏，永久储存，键盘， 或鼠标。其中小的计算设备的记忆提供情况是不一致的。缺乏统一标准的硬件配置之间的小型计算设备的构成是一项艰巨的挑战Java Community Process的计划则是为JVM和J2ME的小型计算设备制定标准。

许多不同种的小的J2ME的计算设备，包括screen phones ，数字机顶盒用于有线电视，手机和个人数字助理，所面临的挑战是Java Community Process的计划是建立一个可以对小的无标准的计算设备实施的Java标准。

6、J2ME Profile

一个配置文件构成的Java类，其执行的功能是为一特别小的计算装置或一类的小型计算设备的。小型计算机技术不断发展， J2ME的配置文件的发展也是一个持续的过程。 配置文件已被界定为编写文本。这些即是Foundation profile，Game Profile，MIDP， PDA personal profile， personal profile，personal basis profile ，和RMI personal profile。

基础配置文件是用来做CDC和配置的，而这种配置文件或CDC几乎是所有其他的配置文件与CDC的核心，因为配置的基础个人资料包含核心的Java类。   
 Game Profile也可以用来与CDC配置，并包括为开发游戏申请任何一个小的计算设备的使用，和CDC的配置所必要的类。  
 移动信息设备配置文件（ MIDP）是用来与cldc 配置和包含的类别提供本地的存储空间，用户界面，和联网能力，从而使应用程序运行在一个移动计算装置，例如Palm OS的设备。 MIDP即是使用无线Java应用程序。   
 pdap rofile （ pdap ）是用来实现cldc配置，并包含个人数字助理类所需要的先进的资源。相对于MIDP的移动设备（如手机），这些先进的功能包括更好的显示和较大的内存，比类似的资源。   
 Personal profile是用来供CDC的配置和foundation profile(基础配置文件)，并含有实现一项复杂的用户界面类。Personal profile提供的核心类，及personal提供一种类，用以实施一个先进的用户界面，也就是说一个用户界面，是能够显示多个窗口的。   
 personal basis profile是和personal profile类似的，因为它是用来为CDC配置的。然而，personal basis profile提供类所实施的一个简单的用户界面，这是在一段时间内用户接口所在地能够显示的一个窗口，。   
 RMI profile是用来与CDC的配置和basis profile提供以类为核心的包括foundation personal profile远程方法调用类。

随着小型计算设备的继续发展，有可能会出现更多的文件。业集团是Java Community Process计划（ java.sun.com / aboutjava / communityprocess ）定义的配置文件。该集团各组确立了标准配置文件所使用的小型计算设备制造行业的标准。 acdc的personal profile是指通过扩大后的核心Java类中发现的专门针对一类的小型计算设备basis profile与类。这些装置的具体类，装载于一个新的配置文件中，使开发人员能够利用这些设备开发一些应用程序。

但是，如果foundation profile 是特定CDC的，而不是所有的配置文件的，是在foundation personal profile是扩大后的核心类。请记住，应用程序可以访问的一小型电脑设备的软件和硬体功能，只有当必要的类，这样做是载于JVM和在配置文件所用的开发的需要。

7、J2ME和无线设备

随着手机等无线设备的大量出现和功能的逐渐完善，对运行在这样的设备上的软件的要求量也越来越大。消费者和企业都要扩大移动通信设备的功能，使其从传统的语音通信过渡到类似于笔记本电脑和个人电脑的功能，使他们可以传送和接收电子邮件，存储和检索的个人资料，执行精密的计算，和玩游戏。

开发商，移动通信设备制造商，和移动网络供应商现正急欲填补这方面的需求，但有是一个严重的障碍：移动通信设备利用许多不同的应用平台和经营调整代码，为一设备开发的代码将无法运行在另一个设备上。移动通信设备缺乏一个标准的应用平台及作业系统，这使得应用开发移动通信设备，促进经济发展有些冒险。

缺乏标准，就是没有新的计算或任何技术开发。 在传统意义上，硬件设备制造商的尝试产品遍及市场，并执行他们的自己的专有标准作为事实上的业界的标准。许多暴发户成功的情况和微软一样。其他时候，业界领袖则组成一个财团，例如Java Community Process计划，集体制定一种标准。

无线应用协议（ WAP ）行动网际网路论坛成为初步创建无线技术标准的产业集团。WAP 论坛创造了移动通信设备标准，被称为thewap标准。 thewap标准是一个提高的HTML ， XML的，和TCP / IP 的一个组成部分，这个标准是无线标记语言规范而组成的，融合了HTML和XML。 wmlscript 是一种剥夺式版本的JavaScript 。

8 、J2ME applications

J2ME的应用程序称为MIDlet几乎可以运行在任何实现了JVM和MIDP移动通信设备上。这鼓励开发商投资时间和金钱在建设移动通信设备上，而不需要担心该应用是设备依赖的风险。不过，J2ME是不会被视为更换为WAP的规格，因为两者都是互补的技术。开发商的申请是根据客户要求的基础继续使用wml和wmlscript 。开发商转向J2ME的较重的客户，需要对移动通信设备进行复杂的处理。

张红旭. 浅谈Java的发展及特点[J]. 硅谷, 2010(6):50-50.

Burns B. Darkstar: the java game server[J]. 2007.

吴越胜、李新磊.Eclipse 3.0程序开发技术详解[M]:清华大学出版社，2010.1~3

岷江. Java的运行原理[J]. 开放系统世界, 2002(5):85-86.

白亮. Java简介[J]. 厦门科技, 2005(1):27-29.

Piroumian V. Java GUI Development[C]:Sams, 1999.

李钟蔚. Java开发实战宝典[M]. 清华大学出版社, 2010

苏碧霞. 探析JAVA Swing工具包在图形用户界面设计中的应用[J]. 信息与电脑:理论版, 2013(5):143-144.

Hardy V J. Java 2D API Graphics[C]:Prentice Hall PTR, 2000.

明日科技. Java从入门到精通[M]. 清华大学出版社, 2012.

郑淑荣.Java的事件处理机制剖析.《办公自动化：综合月刊》，2011（10）：23-24.

胡家芬. Swing可视化组件多线程操作机制研究[J]. 电脑知识与技术,2012(11):7481-7482.

刘煜. 深入了解JAVA中的线程[J]. 中文信息, 2002(8):56-57.

王继成, 高珍. 软件需求分析的研究[J]. 计算机工程与设计, 2002, 23(8):18-21.

冯玲玲, 沈轶. 基于UML的需求分析与建模[J]. 科学技术与工程, 2005, 5(9):605-607.

Reenskaug T. The Model-View-Controller (MVC) Its Past and Present[J]. University of Oslo Draft, 2003.

朱亚玲, 张睿敏. 基于java的连连看游戏设计与实现[J]. 数字技术与应用, 2016(3):161-162.