

Visual C#

经典游戏编程开发

陈 锐 李 欣 夏敏捷 编著



- ◆以游戏开发**案例**为导向，以项目驱动，用耳熟能详的**经典游戏**使枯燥的语言学习充满乐趣。
- ◆分析经典游戏的**设计思路**和**难点**，让读者进入游戏设计的大门，真正了解游戏设计的秘密。
- ◆列出完整的**游戏代码**，对源代码进行了详细解说，并配有**视频光盘**。



科学出版社

Visual C# 经典游戏 编程开发

陈 锐 李 欣 夏敏捷 编著

科 学 出 版 社

北 京

内 容 简 介

本书是一本面向广大编程爱好者的游戏设计类图书。本书最大的特色在于通过具体案例讲解 C# 游戏开发,不仅可以加深知识点的掌握,更重要的是通过本书让读者学会如何开发游戏。本书涉及的游戏都是大家耳熟能详的,例如,推箱子、俄罗斯方块、象棋、坦克大战等。通过本书,让你对枯燥的计算机编程学习充满乐趣。对于初、中级的 C# 学习者来说,本书是一个很好的参考资料,我们不仅为您列出了完整的游戏代码,同时对所有的源代码进行了非常详细的解释,做到了通俗易懂、图文并茂。本书配有教学光盘,不仅有视频讲解,还有 PPT 教学课件以及程序源代码,方便读者使用。

本书可供游戏编程爱好者、程序设计人员和 C# 语言学习者参考阅读。

图书在版编目(CIP)数据

Visual C# 经典游戏编程开发/陈锐,李欣,夏敏捷编著. —北京:科学出版社,2011.8

ISBN 978-7-03-031883-1

I. V… II. ①陈…②李…③夏… III. 游戏-C 语言-程序设计
IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 143521 号

责任编辑:杨 凯 / 责任制作:董立颖 魏 谨

责任印制:赵德静 / 封面设计:郝建宝

北京东方科龙图文有限公司 制作

<http://www.okbook.com.cn>

科 学 出 版 社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

三河市李旗庄少明印装厂 印刷

科学出版社发行 各地新华书店经销

*

2011 年 11 月第 一 版 开本: B5(720×1000)

2011 年 11 月第一次印刷 印张: 28 3/4

印数: 1—5 000 字数: 458 000

定 价: 56.00 元

(如有印装质量问题,我社负责调换)

DotNet(.NET)是微软未来的技术发展方向,其强大的技术优势已经渐渐为人们所推崇,并且在全世界也掀起了学习 DotNet 技术的高潮。掌握 DotNet 技术,无异于掌握了参与目前激烈的就业竞争的利器。作为微软 DotNet 框架下的核心技术之一,Visual C#.NET 经过几年的发展,已经成为主流的开发语言。

本书作者长期从事 Visual C#.NET、Visual C#、Visual Basic 的教学与程序开发、自然语言处理的研究工作。在长期的工作与学习中,积累了丰富的教学和实践经验,同时参阅了大量的技术文档和参考资料,了解到学习编程时什么样的教材能够快速而有效地提高 C# 开发能力,以最少的时间投入得到最快的实际应用。

本书共 22 章,涵盖了益智、射击、棋牌、休闲、网络等游戏。其中,1~10 章为益智游戏篇;11~15 章为控制类游戏篇;16~18 章为棋牌类游戏篇,通过象棋、军棋和扑克牌实例介绍网络游戏的高级知识;19~22 章为其他类游戏篇。

本书内容丰富、全面,其中的通用代码可直接应用于一般的游戏。每款游戏实例均提供详细的设计思路、关键技术分析以及具体的解决步骤方案。每一个游戏实例都是活的、实用的 C# 编程实例。

如果你是一名 C# 初学者,或希望深入系统地掌握 C# 各种技术,那么这本书并不适合你,请将本书放回书架,并摆放在较为显眼的位置,在此表示感谢。本书适合具有一定 C# 基础,希望进一步提高 C# 技术并有意从事游戏开发的读者阅读。

需要说明的是学习编程是一个实践的过程,而不仅仅是看书、看资料的过程,亲自动手编写、调试程序才是至关重要的。通过实际的编程以及积极的思考,读者可以很快掌握很多编程技术,而且,在编程中读者会积累许多宝贵的编程经验。在当前的软件开发环境下,这种编程经验对开发者来说是不可或缺的。

本书由陈锐(国家高级程序员)、李欣(临沂大学)、夏敏捷(中原工学院)共同编写,其中,第1,3~7,10,16,17章由陈锐编写;第2,8,9,11~15章由李欣编写;第12,18~22章由夏敏捷编写。

同时参加编写的人员还有陈亚民、王桂英、王红、冉卫华、张小龙、郑春霞、郑立红、冯志祥、王小艳、冀小幸、卢香清、陈钰、米永刚、董改香、王金英、冯法成、扶晓、夏政伟、王胜军、王天东、王晓红、张思卿、张艺。

特别感谢科学出版社和希望科技图书工作室等帮助本书问世的所有人,尤其是杨凯先生,他十分看重本书的应用价值,在他的努力下,本书才得以顺利出版并早日与读者见面,对此,我们深怀感激。在本书的出版过程中,还得到了温县教育局、临沂大学和中原工学院的支持,在此表示衷心的感谢。

由于编者水平有限,书中难免有不足之处,敬请广大读者批评指正。

编者

2011年8月8日

第 1 篇 益智游戏

第 1 章 连连看游戏	2
1.1 连连看游戏介绍	2
1.2 程序设计的思路	3
1.2.1 动物方块布局	3
1.2.2 连通算法	4
1.2.3 智能查找功能的实现	14
1.3 程序设计的步骤	18
1.3.1 设计游戏界面窗体(Form1.cs)	18
1.3.2 编写程序代码	18
第 2 章 黑白棋游戏	27
2.1 黑白棋游戏介绍	27
2.2 程序设计的思路	28
2.2.1 棋子的显示	28
2.2.2 翻转对方的棋子	28
2.2.3 显示执棋方可落子位置	29
2.2.4 判断胜负功能	29
2.3 程序设计的步骤	29
2.3.1 设计游戏界面窗体(Form1.cs)	29
2.3.2 编写程序代码	29

第3章 汉诺塔游戏	42
3.1 汉诺塔游戏介绍	42
3.2 程序设计的关键技术	43
3.2.1 盘子图片的拖动	43
3.2.2 动态加载盘子图片	44
3.2.3 递归解决汉诺塔盘子移动	45
3.2.4 动画效果移动盘子	46
3.3 程序设计的思路	47
3.4 程序设计的步骤	47
3.4.1 设计游戏界面窗体(Form1.cs)	47
3.4.2 编写程序代码	48
第4章 推箱子游戏	58
4.1 推箱子游戏介绍	58
4.2 程序设计的思路	59
4.3 程序设计的步骤	61
4.3.1 设计游戏窗体(Form1.cs)	61
4.3.2 编写程序代码	62
4.3.3 推箱子游戏关卡地图编辑器(FrmConfig.cs)	70
第5章 扫雷游戏	76
5.1 扫雷游戏介绍	76
5.2 程序设计的思路	77
5.3 程序设计的关键技术	77
5.3.1 动态添加雷块按钮控件	77
5.3.2 无雷方块拓展(对于周围无雷的空白块)	78
5.4 程序设计的步骤	79
5.4.1 设计游戏窗体(Form1.cs)	79
5.4.2 编写程序代码	79

第 6 章 七巧板游戏	87
6.1 七巧板游戏简介	87
6.2 程序设计的关键技术	88
6.2.1 几何图形的绘制	88
6.2.2 几何图形的平移旋转	92
6.2.3 给 GraphicsPath 包含的图形填充颜色	95
6.3 程序设计的步骤	96
6.3.1 设计拼块类(CChip.cs)	96
6.3.2 设计窗体类(Form1.cs)	101
第 7 章 21 点扑克牌游戏	106
7.1 21 点扑克牌游戏介绍	106
7.2 程序设计的关键技术	107
7.2.1 扑克牌面绘制	107
7.2.2 游戏规则的算法实现	108
7.3 程序设计的步骤	110
7.3.1 设计游戏窗体(Form1.cs)	110
7.3.2 编写程序代码	110
第 8 章 人物拼图游戏(一)	117
8.1 人物拼图游戏介绍	117
8.2 程序设计的思路	118
8.3 程序设计的步骤	119
8.3.1 设计游戏窗体(Form1.cs)	119
8.3.2 播放声音类设计	125
第 9 章 人物拼图游戏(二)	126
9.1 人物拼图游戏介绍	126
9.2 程序设计的思路	127
9.3 程序设计的步骤	127
9.3.1 设计游戏窗体(Form1.cs)	127

9.3.2 编写程序代码	128
第 10 章 停车场游戏	134
10.1 停车场游戏介绍	134
10.2 程序设计的思路	135
10.2.1 停车位和汽车颜色编号	135
10.2.2 游戏中的相关数据结构	136
10.2.3 实现汽车移动	136
10.2.4 实现“路线提示”功能	137
10.3 程序设计的步骤	137
10.3.1 设计游戏窗体(Form1.cs)	137
10.3.2 编写程序代码	137

第 2 篇 控制类游戏

第 11 章 坦克大战游戏	148
11.1 坦克大战游戏介绍	148
11.2 程序设计的思路	149
11.3 程序设计的步骤	150
11.3.1 设计坦克类	150
11.3.2 设计子弹类	156
11.3.3 设计播放声音类	159
11.3.4 设计游戏窗体类	160
第 12 章 俄罗斯方块	168
12.1 俄罗斯方块游戏介绍	168
12.2 程序设计的思路	169
12.3 程序设计的步骤	170
12.3.1 设计方块类(Block.cs)	170
12.3.2 设计游戏类(Game.cs)	174
12.3.3 设计游戏窗体(Form1.cs)	183

第 13 章 贪吃蛇游戏	188
13.1 贪吃蛇游戏介绍	188
13.2 程序设计的思路	189
13.3 程序设计的步骤	189
13.3.1 设计豆类	189
13.3.2 设计块类	190
13.3.3 设计蛇类	193
13.3.4 设计场地类	201
13.3.5 设计窗体类	206
第 14 章 华容道游戏	211
14.1 华容道游戏介绍	211
14.2 程序设计的思路	212
14.2.1 数据结构	212
14.2.2 内部逻辑	213
14.3 程序设计的步骤	214
14.3.1 创建游戏界面窗体	214
14.3.2 编写程序代码	215
第 15 章 打字游戏	227
15.1 打字游戏介绍	227
15.2 程序设计的思路	227
15.2.1 字母的产生和下移	227
15.2.2 字母的消除	228
15.3 程序设计的步骤	228
15.3.1 设计游戏界面窗体(Form1.cs)	228
15.3.2 产生随机字符类(myRandom.cs)	233

第 3 篇 棋牌类游戏

第 16 章 网络中国象棋	236
16.1 中国象棋介绍	236
16.1.1 棋 盘	236
16.1.2 棋 子	236
16.1.3 各棋子的走法说明	237
16.1.4 关于胜、负、和	238
16.2 程序设计的关键技术	238
16.2.1 P2P 知识	238
16.2.2 UdpClient 类	239
16.2.3 UdpClient 类开发 UDP 程序的过程	240
16.3 程序设计的思路	241
16.3.1 棋盘表示	241
16.3.2 棋子表示	243
16.3.3 走棋规则	245
16.3.4 坐标转换	247
16.3.5 通信协议设计	247
16.3.6 网络通信传递棋子信息	248
16.3.7 生成棋谱信息	250
16.3.8 实现复盘	251
16.4 程序设计的步骤	252
16.4.1 设计棋子类 (Chess.cs)	252
16.4.2 设计棋盘类	256
16.4.3 设计中国象棋游戏窗体	270
16.4.4 复盘窗体	285
第 17 章 两人对战网络军棋	304
17.1 军棋游戏介绍	304

17.2	程序设计的思路	305
17.2.1	界面设计	305
17.2.2	棋盘数据结构	306
17.2.3	通信协议设计	307
17.2.4	走棋规则设计	309
17.2.5	布阵规则设计	311
17.2.6	通信过程	311
17.3	程序设计的步骤	314
第 18 章	拱猪扑克牌游戏	344
18.1	拱猪扑克牌游戏介绍	344
18.2	程序设计的思路	347
18.2.1	牌手牌的储存	347
18.2.2	记录牌手出牌	347
18.2.3	识别分牌	348
18.2.4	在游戏中出牌顺序逻辑控制	348
18.3	程序设计的步骤	349
18.3.1	Card 类和 Poke 类	349
18.3.2	设计游戏界面窗体(MainForm.cs)	354

第 4 篇 其他类游戏

第 19 章	涂鸦游戏软件	380
19.1	涂鸦游戏软件简介	380
19.2	程序设计的关键技术	381
19.2.1	基本图形绘制	381
19.2.2	绘图轮廓(轨迹)的消除	383
19.2.3	绘图工具箱实现	384
19.3	程序设计的步骤	385
19.3.1	设计绘图工具类(DrawTools.cs)	385

19.3.2	设计涂鸦游戏窗体(Form1.cs)	392
第 20 章	智力问答游戏	400
20.1	智力问答游戏简介	400
20.2	程序设计的关键技术	401
20.2.1	在保持连接的方式下进行数据操作	401
20.2.2	在无状态方式下进行数据操作	401
20.2.3	DataSet 和 DataReader	402
20.3	程序设计的思路	403
20.4	程序设计的步骤	403
20.4.1	设计游戏窗体(Form1.cs)	403
20.4.2	编写程序代码	403
第 21 章	对对碰游戏	411
21.1	对对碰游戏介绍	411
21.2	程序设计的思路	413
21.3	程序设计的步骤	414
21.3.1	设计方块类(Block.cs)	414
21.3.2	设计游戏场景类(GameField.cs)	416
21.3.3	设计游戏窗体(Form1.cs)	423
21.3.4	设计帮助窗体(Help.cs)	432
第 22 章	百变方块游戏	433
22.1	百变方块游戏介绍	433
22.2	程序设计的思路	434
22.3	程序设计的步骤	434
22.3.1	设计拼块类(CCChip.cs)	434
22.3.2	设计窗体类(Form1.cs)	440

Visual

第1篇

益智游戏

井
经
典
游
戏
编
程
开
发

- ★ 第1章 连连看游戏
- ★ 第2章 黑白棋游戏
- ★ 第3章 汉诺塔游戏
- ★ 第4章 推箱子游戏
- ★ 第5章 扫雷游戏
- ★ 第6章 七巧板游戏
- ★ 第7章 21点扑克牌游戏
- ★ 第8章 人物拼图游戏 (一)
- ★ 第9章 人物拼图游戏 (二)
- ★ 第10章 停车场游戏

第 1 章

连连看游戏

1.1 连连看游戏介绍

“连连看”是源自台湾的桌面小游戏，自从传入大陆以来风靡一时，也吸引了众多程序员开发出多种版本的“连连看”。“连连看”考验的是各位的眼力，在有限的时间内，只要把所有能够连接的相同图案，一对一对地找出来，每找出一对，它们就会自动消失，将所有的图案全部消完即可获得胜利。所谓能够连接指的是：无论横向或者纵向，从一个图案到另一个图案之间的连线不能超过两个弯，其中，连线不能从尚未消去的图案上经过。

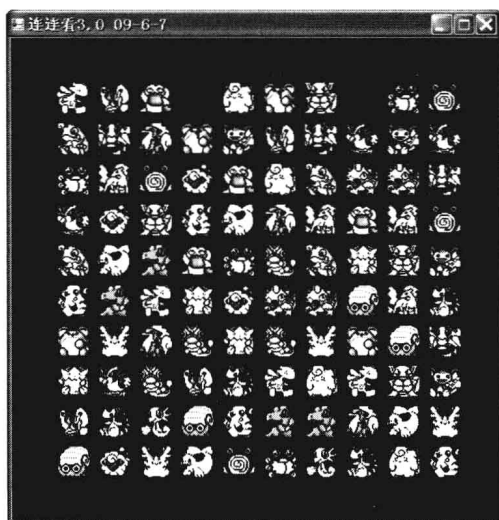


图 1.1 连连看运行界面

连连看游戏的规则总结如下：

- (1) 两个选中的图案是相同的。
- (2) 两个选中的图案之间连接线的折点不超过两个（连接线由 x 轴和 y 轴的平行线组成）。

本章将介绍如何开发连连看游戏，游戏效果如图 1.1 所示。

本游戏增加智能查找功能，当玩家自己无法找到时，可以右击画面，提示可以消去的两个图案（被加上蓝色边框线）。

1.2 程序设计的思路

1.2.1 动物方块布局

首先,我们知道每种方块有 4 个,可以先按顺序把每种动物方块排好放入 ArrayList 列表 tmpMap(临时地图)中,然后再随机从 tmpMap 中取一个动物方块放入地图 m_map 中。实际上程序内部是不需要认识动物方块的图像的,只需要用一个 ID 来表示,运行界面上画出来的动物图形是根据地图中的 ID 获取资源里的图片的。如果 ID 值为 -1(BLANK_STATE),则说明此处已经被消除了。

```
using System.Collections;    //ArrayList 命名空间
private void StartNewGame()
{
    //初始化地图,将地图中所有方块区域位置置为空方块状态
    for(int iNum = 0; iNum < (m_nCol * m_nRow); iNum++)
    {
        m_map[iNum] = BLANK_STATE;
    }
    Random r = new Random();
    //生成随机地图
    //将所有匹配成对的动物物种放进一个临时地图中
    ArrayList tmpMap = new ArrayList ();
    for(int i = 0; i < (m_nCol * m_nRow)/4; i++)
        for(int j = 0; j < 4; j++)
            tmpMap.Add(i);
    //每次从上面的临时地图中取走(并在获取后从临时地图中删除)一个动物,
    //放到地图的一个空方块上
    for (int i = 0; i < m_nRow * m_nCol; i++)
    {
        //随机挑选一个位置
        int nIndex = r.Next() % tmpMap.Count ;
```

```
//获取该选定物件放到地图的空方块上
```

```
m_map[i] = (int)tmpMap[nIndex];
```

```
//在临时地图中除去该动物
```

```
tmpMap.RemoveAt(nIndex);
```

```
}
```

```
}
```

1.2.2 连通算法

连接一般分三种,如图 1.2 所示。

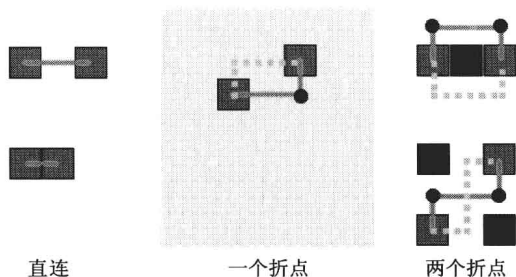


图 1.2 两个选中的方块之间的连接线示意图

1. 直连方式

在直连方式中,要求两个选中的方块 x 或 y 相同,即在一条直线上,并且之间没有其他任何图案的方块,是三种连接方式中最简单的一种。

2. 一个折点

其实相当于两个方块划出一个矩形,这两个方块是一对对角顶点,另外两个顶点中某个顶点(即折点),如果可以同时和这两个方块直连,那就说明可以“一折连通”。

3. 两个折点

这种方式的两个折点($z1, z2$)必定在两个目标点(两个选中的方块) $p1, p2$ 所在的 x 方向或 y 方向的直线上。

按 $p1(x1, y1)$ 点向四个方向探测。例如,向右探测,每次 $x1+1$,判断 $z1(x1+1, y1)$ 与 $p2(x2, y2)$ 点可否形成一个折点连通。如果可以形成连通,则两个折点连通,否则直到超过图形右边界区域。如果超过图形右边界区域,则需判