



完全基于互联网丛书



本书所用技术：
Java、Flash

笑傲江湖 在线游戏大放送



机械工业出版社
China Machine Press

完全基于互联网丛书

笑傲江湖——在线游戏大放送

摆渡人工作室 编著

机械工业出版社

网络游戏是网上常见的娱乐形式，尤其是联机游戏，更发挥了网络独有的优势，为众多网友带来无穷的乐趣。

本书是完全基于互联网丛书的第 9 本，主要介绍如何开发网络游戏，主要是用 14 个精彩实例页面全面介绍如何开发制作网络小游戏。

本书主要使用了 Java 和 Flash，制作的游戏效果十分精彩。

本书适合网络爱好者和网络开发人员使用。

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：王听讲 余茂祚 封面设计：艾迪艾藤

责任印制：郭景龙

北京京丰印刷厂印刷·新华书店北京发行所发行

2001 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·24.5 印张·602 千字

0 001—6000 册

定价：42.00 元（1CD，含配套书）

ISBN 7-900043-91-8/TP·87

E-Mail: sbs@mail.machineinfo.gov.cn

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话（010）68993821、68326677-2527

前 言

从 20 世纪 60 年代网络雏形的发展到目前全球化 Internet 的广泛应用, 互联网正逐渐改变我们的生活方式。我国自 20 世纪 90 年代初才开始接入 Internet, 而仅仅过了 10 年左右的时间, 我们周围的生活就显现出了“Internet 化”——从网上资讯浏览、网上购物到网上聊天, 我国网民人数正以日逾万人的速度增长着。

近几年, WWW(World Wide Web)的应用日益成为 Internet 发展中的重点, 不但各种类型的网站应运而生, 而且 WWW 的设计技术也层出不穷。大型 ICP 为了吸引网民的注意力, 不断扩大麾下的个人主页空间, 这为网络爱好者提供了一个可以展示个人风采的大舞台。

因为 WWW 具有易于浏览、可操作性强、使用环境简单、开发容易的特点, 我们可以预见, 未来 WWW 的创意、设计和编程人员将越来越多, 人才需求越来越大, 应用范围也会越来越广, 传统的应用编程也向 WWW 靠拢。

完全基于互联网丛书 (Fully Based on Internet, 简称 FBI) 主要就是针对 Internet 的发展倾向, 面向广大读者, 从 WWW 的创意、设计和编程的各个方面进行说明。FBI 丛书的主要特点可以用“一个中心, 两个基本点”来概括。

♣ 一个中心——完全基于互联网的发展技术为中心。丛书中包括目前几乎所有互联网所涉及到的方方面面, 从网站的整体规划到最细微方面的设计, 囊括了 WWW 领域内时兴的应用点, 如典型的计数器、BBS、聊天室、数据查询与搜索、Web 数据库、各种网页特技、安全控制、电子商务等。换句话说, “网上有什么, FBI 中就讲述什么”。

♣ 两个基本点——便于阅读和便于交流

(1) 按效果 (功能) 划分, 方便了读者阅读 在 FBI 丛书的每一本书中都会向读者讲解几种相关软件的应用技术, 便于读者阅读和应用。为了实现某一种效果 (功能), 往往会涉及到很多种应用工具软件, 如在制作网页时, 不但要熟悉 FrontPage、Dreamweaver、Flash、Fireworks、Photoshop 等软件的不同特点, 而且还要学会各种编程技术, 如 Java、ASP (JavaScript/VBScript)、PHP、Perl、SQL/MySQL 等, 这样才能使您制作的网页更加精彩。在同一本书中您可以看到各种不同的软件和语言为了实现同一个效果 (功能) 协同作用。而采用更有针对性的页面形式编排, 融技术于具体应用, 读者只要按照书中的步骤就可以完成预定的效果 (功能)。这是目前市面上仅有的按照效果 (功能) 划分并以网页形式编排的系列图书。

(2) 完善的售后服务体系, 便于读者与作者之间的交流 为了方便广大读者与本丛书作者的相互沟通, 我们特为本套丛书建立了专用网站。请记住我们的网址 <http://www.buydoren.com>, 欢迎您指正 FBI 丛书存在的问题, 或者就您感兴趣的问题和我们探讨。我们也将定期对新出现的计算机技术在网上进行实例讲解。

我们都是 Internet 的爱好者, 相信通过我们的共同努力, 可以为互联网在我国的普及尽一点微薄之力。

摆渡人工作室

www.buydoren.com

关于本书

本书是完全基于互联网丛书的第 9 本，主要介绍如何开发网络游戏，主要是网络小游戏，可以放置在网页上作为闲暇娱乐。

在众多的 Web 编程当中，最适合编制游戏的语言是 Java，因为 Java 语言拥有类似于 C/C++、Pascal 等这些高级语言的控制和运行能力，可以完成复杂的运行效果，同时既可独立运行，也可放置在 Web 页中运行；而 ASP 等语言主要是针对 Web 开发的，不适合作为复杂计算和界面控制的编程语言，即灵活性不够。因此本书中的游戏实例也是以 Java 语言为主。近两年，Flash 应用越来越多，Micromedia 提供的 Flash 不但可以为主页提供更加丰富的动画效果，也可以用来设计一些简单游戏、幽默漫画故事等，本书中的页面 13 就是一个用 Flash 制作的 21 点游戏。

Java 语言写的 13 个游戏，在编程上有相似之处，但更多的是编程控制、实现方法、技术要领上的不同，读者完全可以从这 13 个 Java 游戏上深刻体会利用 Java 编写游戏的技术要领，从而达到从小游戏学习编写大游戏的目的。

Flash 实例在完全基于互联网丛书的其他书籍当中也有，主要以实现各种动画效果为主，这里提供的唯一一个应用于游戏开发的实例却很具有代表性，因为 Flash 比较容易，多少有点编程经验（例如 VB 编程经验）的读者就可以看出其间的思路技巧来。

本书所选择的 14 个游戏都非常具有代表性，而且许多是大家熟知的优秀游戏，例如扫雷、弹球、迷宫等。

本书适合网络爱好者和网络开发人员使用。

关于我们——摆渡人工作室

“摆渡人工作室”的前身是在两年前成立的“清汉计算机工作室”。

两年多来，我们和机械工业出版社通力合作，精心策划、编辑、出版了《计算机开发与制作实例丛书》（以下简称《实例》）。这套丛书共计 26 本，覆盖了从 Microsoft Office 到应用程序和数据库开发，以及网页制作和多媒体等多方面实用技术和技巧，以其独特的魅力在同类计算机图书中发展成了一个大家族。

我们立足清华大学，广泛了解并研究计算机发展方向，认真进行计算机图书的策划、组织和编辑工作。

我们的努力得到了应有的回报！2000 年我们的实例丛书获得了巨大的成功，得到了读者和出版社的广泛好评。其中《FrontPage 2000 网页制作实例》和《Visual Basic 6.0 数据库开发实例》一年内加印了 4 次，累计销量达到 16000 册，丛书平均销量也达到了近 10000 册。

在实例丛书的具体运作中我们体验了创业的艰辛，成功的喜悦，在成功后我们开始在更深的层次上思考读者真正需要什么。

我们开始思考两个问题：

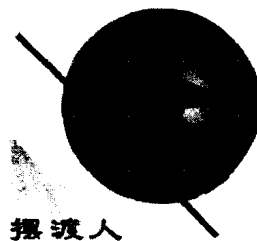
◆ 我们何不推出更具针对性的实例呢？

我们发现实例丛书中的实例虽然是我们精心设计的，但很多却不是“现存的”，也就是说这些实例不是在网上存在的，而是我们为了讲解某些知识专门设计的。这对需要解决实际问题的读者来说还需一个消化吸收的过程。

◆ 何不建立一个更完善的图书售后服务体系呢？

有些读者购买了我们的实例丛书后，针对书中的某些问题或打电话或发 Email 和我们探讨，我们为读者对我们的厚爱感到由衷的高兴，但同时也感到交流的不顺畅阻碍了我们和读者进一步的探讨。读者买到图书，正是学习的开始，我们的工作应该延伸到读者的学习过程中，而不是售出书就万事大吉！

带着这些急需解决的问题，我们考虑了现在最便利的联络方式——互联网，在对现在全球互联网技术现状和未来发展走势的精心调查研究下，我们隆重推出了《完全基于互联网丛书》（Fully Based on Internet，简称 FBI）。这套丛书继承了《计算机开发和制作实例丛书》的特点，以页面的形式讲解当今互联网业涉及的几乎所有的技术。从申请域名建立网站，到实现人机交互和建立自己的电子商务网站等全过程。读者只需要按照我们的讲解，按部就班地学习丛书中介绍的各种技术，不仅可以建立展现自我风采的个人网站，还可以成功地建立完善的商业网站。同时这套丛书还具有“一个中心+两个基本点”的特征。关于这方面的内容您可以从本丛书的前言中看到。相信在您认真阅读了这套丛书后，会对这些特点有更深刻的理解。



随着 FBI 丛书的发展,我们相信一定会与您成为好朋友,因为我们和您一样都是爱好新鲜事物的年轻人,充满着对未来的憧憬。我们努力、创造,我们自信、把握,我们相信机会在自己的手中。

您有什么好的建议或者对我们的丛书有什么意见,请访问我们的网站:

www.buydoren.com。

关注生活,关注我们的图书!

附录:关于《实例》丛书

《实例》丛书的特点:

- (1) 通过一个个精心设计的实例,用户可以轻松地逐步掌握一种软件。
- (2) 书中每一个实例的主题对应该软件的一个主要功能,通过所有实例的学习,您可以全面掌握该软件的精髓。
- (3) 虽然《实例》丛书采用一个个相对独立的实例,但我们在丛书的策划中考虑到知识的连贯性和统一性,丛书中的“归纳和注释”可以帮助读者举一反三地掌握书中介绍的内容。使用附光盘的书籍更可以使您的学习和开发如虎添翼。
- (4) 随书光盘中还包含和该书籍相关的软件及网络资源,使您在阅读该书的同时还可以涉猎其他相近的领域。

《实例》丛书书目:

1. Word 2000 中文版实例
2. Excel 2000 中文版实例
3. PowerPoint 2000 中文版实例
4. Access 2000 中文版实例
5. Delphi 5.0 开发实例 (附 CD)
6. VBScript 开发实例 (附 CD)
7. ASP 开发实例 (附 CD)
8. C++ Builder 网络开发实例 (附 CD)
9. Flash 5.0 网页动画制作实例 (附 CD)
10. Photoshop 6.0 图像制作实例 (附 CD)
11. FrontPage 2000 网页制作实例 (附 CD)
12. CorelDRAW 9.0 图画制作实例 (附 CD)
13. Photoshop 5.5 图像制作实例 (附 CD)
14. 3D Studio MAX R3.0 动画制作实例 (附 CD)
15. Authorware 5.0 制作实例 (附 CD)
16. Director 7.0 多媒体制作实例 (附 CD)
17. Visual Basic 6.0 数据库开发实例 (附 CD)
18. Visual Basic 6.0 多媒体开发实例 (附 CD)
19. Visual Basic 6.0 网络开发实例 (附 CD)
20. Visual FoxPro 6.0 开发实例 (附 CD)
21. Flash 4.0 动画、网页制作实例 (附 CD)
22. AutoCAD 2000 中文版综合应用实例 (附 CD)
23. Visual C++ 6.0 数据库与网络开发实例 (附 CD)
24. Visual C++ 6.0 多媒体开发实例 (附 CD)
25. Dreamweaver 3.0 网页制作实例 (附 CD)
26. Fireworks 3.0 网页素材制作实例 (附 CD)

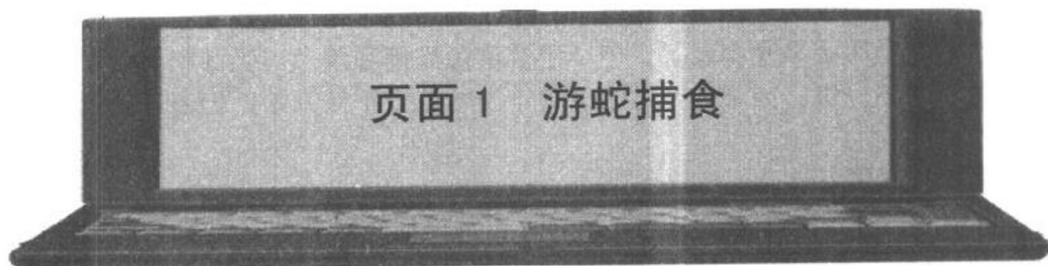
目 录

前言

关于本书

关于我们——摆渡人工作室

页面 1	游蛇捕食	1
页面 2	飞碟大战	30
页面 3	星球大战	53
页面 4	击球游戏	85
页面 5	吃虫子游戏	99
页面 6	蠕虫游戏	139
页面 7	扫雷	181
页面 8	汉诺塔	195
页面 9	拼图游戏	206
页面 10	赛车	222
页面 11	弹球游戏(1).....	248
页面 12	弹球游戏(2).....	262
页面 13	纸牌游戏 21 点.....	291
页面 14	迷宫	304



<http://www.brainjar.com/Java/snakepit/index.html>

页面效果和技术概要

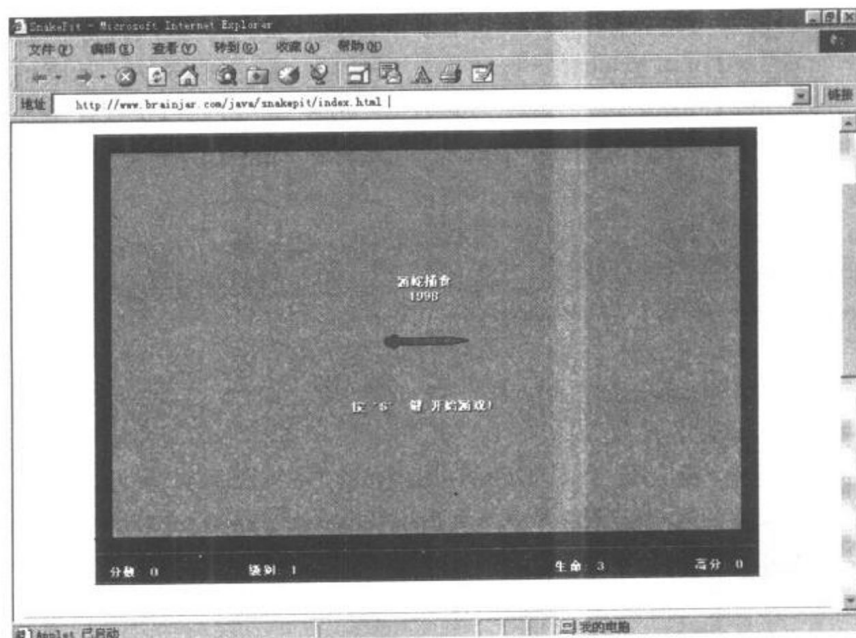


图 1-1 游戏“游蛇捕食”

游戏介绍和使用技巧

① “游蛇捕食”游戏 是一个简单而又有趣的游戏。玩家用方向键控制一条蛇来吃掉在屏幕上游动的老鼠。此游戏简单而又具有耐玩性和刺激性，它有五种级别和对应的游戏速度。游戏在开始时，要装入画面和声音，请稍等片刻。装入操作完成后，会在画面上提示：请按“S”键开始游戏。此游戏可在大多数的平台上运行。页面效果见图 1-1。

② 游戏操作方法 “S”键——开始游戏，“P”键——暂停游戏，“M”键——关掉声音。方向键控制“蛇”上下左右移动。

③ 评分标准 吃掉一只老鼠得 10 分，晋升一级奖励 200 分。每得 500 分，就会增加一条生命。

④ 注意事项 如果游戏运行较慢，请关掉声音。这并不影响游戏的运行。如果游戏

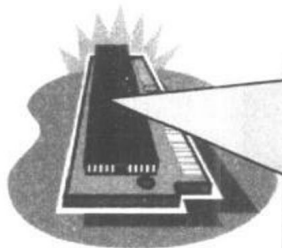
对按键不响应, 请用鼠标在游戏区域点击一下, 然后利用方向键控制“蛇”去吃掉周围的老鼠。做游戏时要尽量多吃掉老鼠, 但一定要避免蛇撞在墙上。当红色的钥匙出现时, 吃掉它, 就会进入下一级。如果连续过了 5 关, 游戏就会重新开始, 但游戏的速度会比以前快一点。

本页面的技术要点

🌐 **Java 包** Java 引进包的概念是为了实现访问控制和解决类名空间的冲突。包是由 Java 接口和类组合成的, 通过包可以访问 Java 的 API 及应用程序接口。将类放到包中以免包外知道类的存在。一个类在自己的包中是独一无二的, 从而避免了类名的冲突。Java 程序的第一条语句, 是用 `package` 语句来指定程序文件中定义的类和接口所属的包。然后是用 `import` 语句说明要被类和接口程序引用的包。

🌐 **线程** 在 Java 中, 线程的产生和运行都是通过 `Thread` 和 `ThreadGroup` 类来完成的。线程从产生到消失以前, 一直处于产生、运行、暂停和消失四种状态之中。线程的产生是由 `new` 语句完成的, 线程的运行是由线程的 `start` 方法来实现的。当线程在运行过程中用 `suspend`、`sleep` 等方法或遇到了一些 I/O 的堵塞, 线程会进入暂停状态。在暂停状态中, 线程不会运行, 但可以返回可运行状态。如使用 `resume` 方法, 睡眠时间结束或 I/O 操作完成时, 线程又返回运行状态。当线程调用 `stop` 方法后, 线程消失。每一个线程都有自己的优先级。当多线程对同样的资源出现竞争时, Java 是通过查看线程优先级的大小来解决的。

🌐 **Java 类** 类是组成 Java 程序的基本要素。它封装了一类对象的状态方法。它能派生出其他的类, 新派生出的类能从父类中继承已定义的数据和方法。一个完整的类包括类的声明, 类体, 成员变量和成员方法。



重点记忆:

一、Java.lang 包

包 `Java.lang` 提供了构成 Java 语言和虚拟机核心的类和接口, 其中最重要的类是 `Object` 类, 它是 Java 类继承关系的根, 所有的 API 中的类都是由它生成的。

二、Java.util 包

包 `Java.util` 包含了各种各样有效的类和接口, 包括产生各种数据结构的类。



实现步骤和技术详解



Java Applet 程序的建立

- (1) 启动 Microsoft Visual J++ 6.0。
- (2) 选择菜单“File”, 然后选择“New Project...”中的“Web Pages”-->“Applet on

HTML”。

(3) 选择存储程序的合适的位置，本程序存于 E:\Java\Project1。

(4) 单击“OK”按钮，完成程序的建立。



定义游戏运行中需要的变量

在程序中，要用到一些必不可少的类，而这些类都包括在包（package）中。引进包的概念是为了实现访问控制和解决类名空间的冲突。包是由 Java 接口和类组合而成的，通过包可以访问 Java 的 API 及应用程序接口。

在程序的开头，类似于 C/C++ 语言的头文件，需要在 Java 程序中引用 Java 包，通常使用 import 语句说明要被类和接口程序引用的包，如下所示：

```
import Java.awt.*;
import Java.net.*;
import Java.util.*;
import Java.applet.Applet;
import Java.applet.AudioClip;
```

Java 的数据有常量和变量之分。语句 final int MAX_DELAY=75，定义了一个整形常量。变量是 Java 程序中的基本存储单元。Java 的数据类型包括以下几种：

integer(整数)。

floating(浮点数)。

point(指针)。

Boolean(布尔变量)。

Character or String(字符或字符串)。

在游戏开始时，需要定义一些变量。这些变量定义了屏幕刷新的速度、游戏区域的大小、游戏的级别数目、在蛇周围游动的老鼠的数量、游戏状态（初始化、运行、级别、结束）、玩家得分、蛇的形状和字体等等。

为了实现上述目的，对 Java 自动生成的类进行了修改，其中，“SnakePit”是类名。

改后代码如下：

```
public class SnakePit extends Applet implements Runnable {
    //线程控制变量
    Thread loopThread;
    Thread loadThread;
    //常量
    static final int MAX_DELAY = 75;    //屏幕更新的间隔时间
    static final int MIN_DELAY = 50;
    static final int DELAY_INCR = 5;
    static final int GRID_WIDTH = 39;  //游戏区域的大小
    static final int GRID_HEIGHT = 25;
    static final int GRID_SIZE = 16;
```

```
static final int NUM_LEVELS = 5;    //游戏的级别
```



注意问题:

常量不能再赋新值

与 C/C++ 不同, Java 不能通过 `#define` 命令把一个标识符定义为常量。Java 使用关键字 `final` 来实现常量的定义的。如 `final double PI=3.14159` 把 `PI` 定义为一个浮点型常量, 它的值为 3.14159。常量只能声明一次, 而且在声明以后, 其值再也不能改变了。

```
static final int NUM_MICE    = 6;    //活动的老鼠的数目
static final int NUM_LIVES  = 3;    //游戏开始时的生命数
//游戏状态常量
static final int INIT    = 1;
static final int PLAY    = 2;
static final int LEVEL   = 3;
static final int END      = 4;
static final int OVER    = 5;
static final int END_COUNT= 30;
static final int LEVEL_COUNT = 40;
static final int TYPE_MASK = 0x00FF0000;
static final int EMPTY     = 0x00000000;
static final int BLOCK     = 0x00010000;
static final int SNAKE     = 0x00020000;
static final int MOUSE     = 0x00030000;
static final int KEY       = 0x00040000;
static final int DIR_MASK  = 0x0000FF00;
static final int NONE      = 0x00000000;
static final int LEFT      = 0x00000100;
static final int RIGHT     = 0x00000200;
static final int UP        = 0x00000300;
static final int DOWN      = 0x00000400;
static final int SHAPE_MASK= 0x000000FF;
static final int SQUARE   = 0x00000000;
static final int SNAKEHEAD = 0x00000001;
static final int SNAKEBODY = 0x00000002;
static final int SNAKEELB1 = 0x00000003;
static final int SNAKEELB2 = 0x00000004;
static final int SNAKETAIL = 0x00000005;
```

```
static final int MOUSEBODY = 0x00000006;
static final int KEYSHAPE = 0x00000007;
//分数值常量
static final int MOUSE_POINTS = 10;
static final int LEVEL_POINTS = 200;
static final int EXTRA_LIFE = 500;
//尺寸变量
int width;
int height;
//游戏数据变量
int score; //分数
int highScore;
int lives;
int extraLife;
int level; //级别
int levelTotal;
int miceNeeded;
int miceEaten;
int delay;
int[][] grid; //游戏区域
Point[] snake; //有关蛇的数据
int headPtr;
int tailPtr;
int direction;
int lastDirection;
boolean lockKeys;
Point[] mouse; //有关老鼠的数据
Point key; //按键数据
boolean keyActive;
int gameState; //游戏状态
int levelCounter;
int endCounter;
boolean paused;
//屏幕颜色
Color bgColor = Color.black;
Color fgColor = Color.white;
Color blockColor = new Color(0, 0, 153);
Color fieldColor = new Color(204, 153, 102);
Color snakeColor = new Color(0, 153, 0);
Color mouseColor = Color.gray;
```

```

Color keyColor    = new Color(204, 102, 102);
//基本形状
Polygon snakeHead;
Polygon snakeBody;
Polygon snakeTail;
Polygon snakeElb1;
Polygon snakeElb2;
Polygon mouseBody;
Polygon keyShape;
//声音数据
boolean    loaded = false;
boolean    sound;
AudioClip bonkSound;
AudioClip munchSound;
AudioClip squeakSound;
AudioClip chimeSound;
AudioClip advanceSound;
Dimension offDimension;
Image      offImage;
Graphics offGraphics;
//字体数据
Font font = new Font("Helvetica", Font.BOLD, 12);
FontMetrics fm;
int fontWidth;
int fontHeight;
//Java 程序信息
public String getAppletInfo() {
    return("游蛇捕食, 1998 ");
}

```

以上是此程序中用到的所有的变量，由于 Java 是面向对象的语言，因此所有的变量只能在类中定义，也就是只能是类的成员变量，这是和 C/C++ 等语言的不同之处。

读者若是对游戏的设置不太满意的话，可以按照上面的注释对游戏进行适当的修改，如颜色、字体、分数和速度等。



初始化游戏

在完成了变量的定义之后，下面来进行游戏的初始化。进行游戏的初始化，需要用到 `init()` 方法。`init()` 方法是 Java 程序的初始化所必需的方法。因此，要完成游戏的初始化操作，需要修改 `init()` 方法。在修改后的 `init()` 方法中，包括在屏幕上输出游戏的名称和字体、设置字体、画出蛇和老鼠的形状、设定生命数、老鼠的数目、游戏的状态等。在 `SnakePit`

类中建立以下方法:

```
public void init();           //初始化方法
public void initGame();       //设定游戏状态
public void endGame();        //结束游戏
public void initLevel();      //设定游戏级别
public void endLevel();       //终止游戏级别
public void initLife();       //在屏幕上重新画蛇
public void endLife();        //结束生命
public void initBlocks();     //初始化游戏区域
public void initSnake();      //在游戏区域的中心画蛇
public void killSnake();      //在游戏区域中去掉蛇
public void initMouse(int n); //在游戏区域画 n 个老鼠
public void killMouse(int n); //当老鼠被蛇吃掉后, 清除此老鼠
public void initKey();        //初始化键盘信息
public void moveMouse(int n); //移动老鼠
```

具体程序代码如下, 并作详细的注释:

```
public void init() {
    Graphics g;
    int i;
    //游戏名称和日期
    System.out.println("游蛇捕食, 1998 ");
    //设置字体
    g = getGraphics();
    g.setFont(font);
    fm = g.getFontMetrics();
    fontWidth = fm.getMaxAdvance();
    fontHeight = fm.getHeight();
    grid = new int[GRID_WIDTH][GRID_HEIGHT];
    //画出蛇和老鼠的形状
    //画出蛇的头部
    snakeHead = new Polygon();
    snakeHead.addPoint(4, 15);
    snakeHead.addPoint(3, 14);
    snakeHead.addPoint(1, 12);
    snakeHead.addPoint(1, 8);
    snakeHead.addPoint(3, 6);
    snakeHead.addPoint(3, 4);
    snakeHead.addPoint(6, 1);
    snakeHead.addPoint(9, 1);
    snakeHead.addPoint(12, 4);
}
```

```
snakeHead.addPoint(12, 6);
snakeHead.addPoint(14, 8);
snakeHead.addPoint(14, 12);
snakeHead.addPoint(12, 14);
snakeHead.addPoint(11, 15);
//画出蛇的躯干
snakeBody = new Polygon();
snakeBody.addPoint(11, 0);
snakeBody.addPoint(11, 15);
snakeBody.addPoint(4, 15);
snakeBody.addPoint(4, 0);
snakeElb1 = new Polygon();
snakeElb1.addPoint(11, 0);
snakeElb1.addPoint(11, 4);
snakeElb1.addPoint(15, 4);
snakeElb1.addPoint(15, 11);
snakeElb1.addPoint(7, 11);
snakeElb1.addPoint(4, 8);
snakeElb1.addPoint(4, 0);
snakeElb2 = mirror(snakeElb1);
//画出蛇的尾部
snakeTail = new Polygon();
snakeTail.addPoint(11, 0);
snakeTail.addPoint(8, 15);
snakeTail.addPoint(7, 15);
snakeTail.addPoint(4, 0);
//画出老鼠
mouseBody = new Polygon();
mouseBody.addPoint(8, 1);
mouseBody.addPoint(12, 5);
mouseBody.addPoint(12, 6);
mouseBody.addPoint(10, 7);
mouseBody.addPoint(12, 9);
mouseBody.addPoint(12, 12);
mouseBody.addPoint(10, 14);
mouseBody.addPoint(5, 14);
mouseBody.addPoint(3, 12);
mouseBody.addPoint(3, 9);
mouseBody.addPoint(5, 7);
mouseBody.addPoint(3, 6);
```

```

mouseBody.addPoint(3, 5);
mouseBody.addPoint(7, 1);
//画出钥匙
keyShape = new Polygon();
keyShape.addPoint(1, 6);
keyShape.addPoint(7, 6);
keyShape.addPoint(9, 4);
keyShape.addPoint(13, 4);
keyShape.addPoint(15, 6);
keyShape.addPoint(15, 10);
keyShape.addPoint(13, 12);
keyShape.addPoint(9, 12);
keyShape.addPoint(7, 10);
keyShape.addPoint(3, 10);
keyShape.addPoint(3, 11);
keyShape.addPoint(0, 11);
keyShape.addPoint(0, 7);
//设定信息
highScore = 0;//得分
sound = true;
snake = new Point[GRID_WIDTH * GRID_HEIGHT];
//生成一条蛇
for (i = 0; i < snake.length; i++)
    snake[i] = new Point(-1, -1);
//生成老鼠
mouse = new Point[NUM_MICE];
for (i = 0; i < NUM_MICE; i++)
    mouse[i] = new Point(-1, -1);
key = new Point(-1, -1);
lockKeys = false;
//初始化游戏
initGame();
endGame();
for (i = 0; i < NUM_MICE; i++)
    killMouse(i);
//设定游戏状态
gameState = INIT;
}

```