

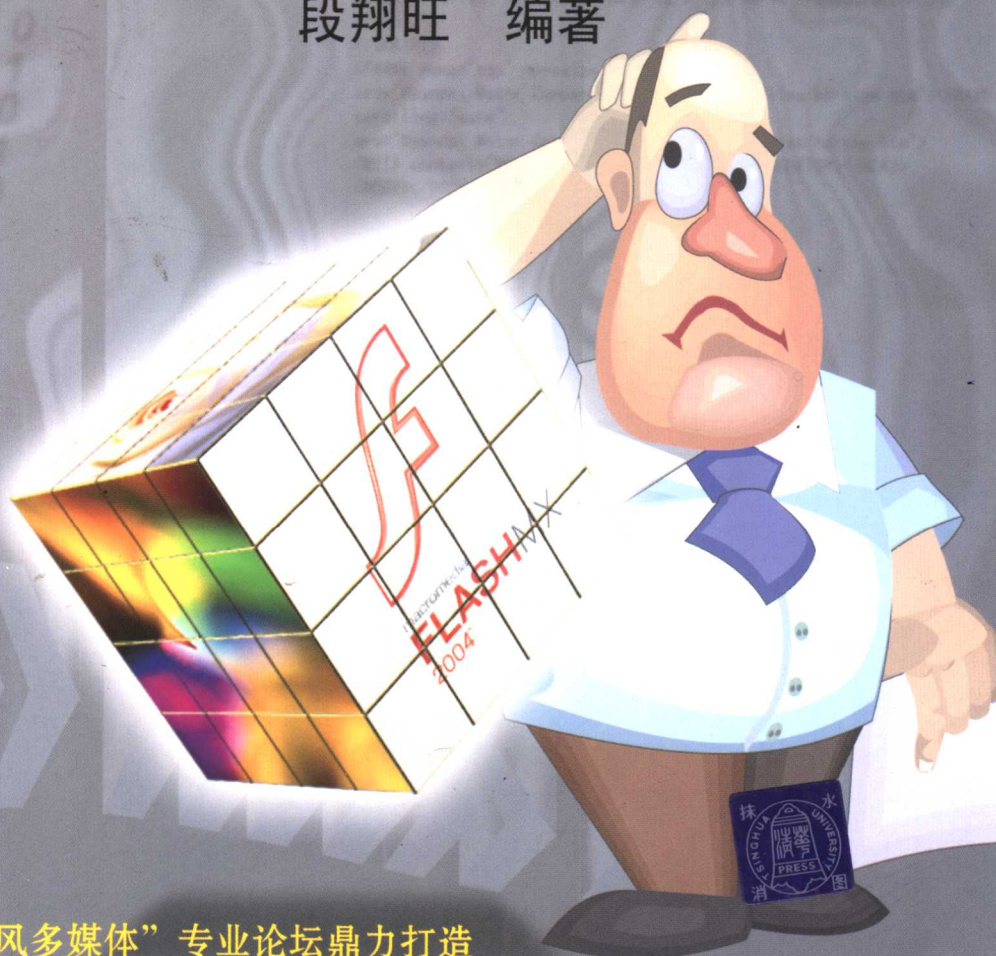
实例通系列



# Flash MX 2004

## 互动游戏设计教程

段翔旺 编著



“宇风多媒体”专业论坛鼎力打造  
提供全部Flash游戏实例的源程序代码

清华大学出版社

实例通系列

# Flash MX 2004 互动游戏设计教程

段翔旺 编著

清华大学出版社

北 京

## 内 容 简 介

本书重点介绍 Flash MX 2004 游戏设计的方法和技巧,通过大量的精彩实例讲解,让读者能够用较短的时间熟悉 Flash 游戏设计流程并提高设计技能。全书共分为 9 章,第 1 章介绍了 Flash 游戏设计流程与游戏设计中应注意的事项;第 2~9 章通过经典实例介绍了游戏设计技巧,并重点讲解在游戏设计中经常使用的重要技术与原理,包括三维原理、本地高分排行榜、游戏的存取等内容。

本书附带光盘的内容丰富且非常实用,包括本书所有实例的演示以及源文件。

本书面向初、中级用户,内容深入浅出、简洁易懂、各种知识技巧与实例紧密结合,富于启发性。非常适合广大 Flash 游戏制作爱好者以及相关设计人员阅读。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

Flash MX 2004 互动游戏设计教程/段翔旺编著. —北京:清华大学出版社,2005.5  
(实例通系列)

ISBN 7-302-10737-8

I. F… II. 段… III. ①动画 - 设计 - 图形软件, Flash MX 2004 - 教材 ②游戏 - 应用程序 - 程序设计 - 教材 IV. ①TP391.41 ②G899

中国版本图书馆 CIP 数据核字(2005)第 025997 号

出 版 者:清华大学出版社  
<http://www.tup.com.cn>  
社 总 机:010-62770175

地 址:北京清华大学学研大厦  
邮 编:100084  
客户服务:010-62776969

责任编辑:魏江江

封面设计:杨 兮

印 刷 者:北京市清华园胶印厂

装 订 者:三河市金元装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:22.75 字数:546 千字

版 次:2005 年 5 月第 1 版 2005 年 5 月第 1 次印刷

书 号:ISBN 7-302-10737-8/TP·7149

印 数:1~3000

定 价:38.00 元(含光盘)

# 前 言

Flash 是美国著名的 Macromedia 公司推出的动画制作与网络开发软件, Flash MX 2004 是目前发布的最新版本。Flash 从刚出现就受到了广大设计师的青睐, 经过 Macromedia 公司设计人员的不断努力, 现在 Flash 已经是一个独立的、综合的、基于矢量的设计和开发软件。Flash 正被全球数以百万计的开发人员和设计人员使用。

本书着眼于实际应用, 密切结合设计实例讲解 Flash MX 2004 中文版在游戏设计方面的应用方法与技巧。全书共分 9 章, 其中第 1 章介绍了 Flash 游戏设计流程与设计过程中的注意事项; 第 2 章介绍了方块原理, 颜色控制以及时间限制的实现; 第 3 章讲述了千里马游戏的制作, 重点介绍了三维原理的背景滚动; 第 4 章介绍了穿越峡谷游戏的制作, 并且着重讲解了重力控制及导弹的发射; 第 5 章介绍了信鸽游戏的制作, 着重讲解了相对运动与高分排行榜的实现; 第 6 章介绍了谁动了我的奶酪的制作, 着重讲解了房间编辑、转换原理以及游戏的存取; 第 7 章介绍了暴力摩托游戏的实现, 着重讲解了图片分割原理以及三维的动态控制; 第 8 章介绍了经典游戏超级玛利奥的实现, 着重讲述了平台类游戏的关卡编辑以及平台引擎的开发; 第 9 章为吞食鱼游戏的实现, 着重讲解了目标运动原理以及鼠标的交互控制。针对每章介绍的实例以及重点技术原理的讲解都在光盘中提供了源文件, 大家在阅读过程中, 最好能对照源程序测试学习。

书中主要内容来自于作者几年来使用 Flash 进行游戏设计的经验总结, 读者在使用本书的过程中若有相关问题或意见、建议, 可以访问作者个人网站 <http://www.xw18.com> 或宇风多媒体网站 (<http://www.yfdmt.com/bbs/>) 与作者联系。

IT 书吧 (<http://www.itbook8.com>) 提供本书的相关图书咨询以及相关资料下载。

本书在写作与修订过程中, 获得了宇风多媒体网站负责人程勇的大力支持。在此表示衷心的感谢!

编 者

# 目 录

|                          |    |
|--------------------------|----|
| 第 1 章 Flash 游戏设计流程 ..... | 1  |
| 1.1 定义游戏规则 .....         | 1  |
| 1.2 游戏素材准备 .....         | 1  |
| 1.3 开发游戏引擎 .....         | 2  |
| 1.4 设定主角与敌人 .....        | 2  |
| 1.5 优化图形界面 .....         | 3  |
| 1.6 添加音乐音效 .....         | 7  |
| 1.7 设置游戏重置 .....         | 9  |
| 1.8 游戏测试 .....           | 10 |
| 1.9 本章小结 .....           | 21 |
| 第 2 章 滑块小子 .....         | 22 |
| 2.1 游戏规则 .....           | 22 |
| 2.2 素材准备 .....           | 22 |
| 2.3 方块原理 .....           | 24 |
| 2.4 颜色控制 .....           | 27 |
| 2.5 设定主角与敌人 .....        | 32 |
| 2.6 实现时间限制 .....         | 40 |
| 2.7 添加音乐音效 .....         | 45 |
| 2.8 设置游戏重置 .....         | 46 |
| 2.9 游戏的网上发布 .....        | 55 |
| 2.10 本章小结 .....          | 58 |
| 第 3 章 千里马 .....          | 59 |
| 3.1 游戏规则 .....           | 59 |
| 3.2 素材准备 .....           | 59 |
| 3.3 三维原理 .....           | 64 |
| 3.4 设定主角 .....           | 68 |
| 3.5 设定障碍物 .....          | 70 |
| 3.6 设定奖励元素 .....         | 72 |
| 3.7 关编辑 .....            | 76 |
| 3.8 设置游戏重置 .....         | 79 |

---

|                    |            |
|--------------------|------------|
| 3.9 游戏的网上发布        | 92         |
| 3.10 本章小结          | 93         |
| <b>第4章 穿越峡谷</b>    | <b>94</b>  |
| 4.1 游戏规则           | 94         |
| 4.2 素材准备           | 94         |
| 4.3 重力控制           | 99         |
| 4.4 设定主角           | 103        |
| 4.5 导弹发射           | 108        |
| 4.6 设置奖励元素         | 113        |
| 4.7 关编辑            | 116        |
| 4.8 设置游戏重置         | 117        |
| 4.9 游戏的网上发布        | 124        |
| 4.10 本章小结          | 126        |
| <b>第5章 信鸽</b>      | <b>127</b> |
| 5.1 游戏规则           | 127        |
| 5.2 素材准备           | 127        |
| 5.3 设定主角           | 140        |
| 5.4 相对运动           | 144        |
| 5.5 设置能量柱          | 146        |
| 5.6 暂时保存最好分数       | 147        |
| 5.7 设置游戏重置         | 150        |
| 5.8 本地高分排行榜        | 157        |
| 5.9 游戏的网上发布        | 162        |
| 5.10 本章小结          | 163        |
| <b>第6章 谁动了我的奶酪</b> | <b>164</b> |
| 6.1 游戏规则           | 164        |
| 6.2 素材准备           | 165        |
| 6.3 房间转换原理         | 195        |
| 6.4 房间编辑原理         | 197        |
| 6.5 寻找奶酪           | 201        |
| 6.6 游戏的存取          | 211        |
| 6.7 多主角切换控制        | 214        |
| 6.8 寻找地图           | 229        |
| 6.9 设置游戏重置         | 247        |
| 6.10 本章小结          | 253        |

|                  |     |
|------------------|-----|
| <b>第7章 暴力摩托</b>  | 254 |
| 7.1 游戏规则         | 254 |
| 7.2 素材准备         | 254 |
| 7.3 图片分割原理       | 257 |
| 7.4 三维控制原理       | 262 |
| 7.5 设定主角         | 266 |
| 7.6 计时及最好成绩      | 269 |
| 7.7 设置游戏重置       | 272 |
| 7.8 游戏的网上发布      | 281 |
| 7.9 本章小结         | 282 |
| <b>第8章 超级玛利奥</b> | 283 |
| 8.1 游戏规则         | 283 |
| 8.2 素材准备         | 283 |
| 8.3 关图的编辑        | 289 |
| 8.4 设定主角         | 295 |
| 8.5 平台引擎开发       | 298 |
| 8.6 添加敌人元素       | 302 |
| 8.7 Tween 类      | 304 |
| 8.8 设置游戏重置       | 306 |
| 8.9 游戏的网上发布      | 318 |
| 8.10 本章小结        | 319 |
| <b>第9章 吞食鱼</b>   | 320 |
| 9.1 游戏规则         | 320 |
| 9.2 素材准备         | 320 |
| 9.3 鼠标交互控制       | 325 |
| 9.4 设定主角         | 326 |
| 9.5 设定敌人智能       | 331 |
| 9.6 目标运动原理       | 336 |
| 9.7 设置游戏重置       | 342 |
| 9.8 游戏的网上发布      | 352 |
| 9.9 本章小结         | 353 |

# 第1章 Flash 游戏设计流程

在使用 Flash 设计游戏前，并不是有了想法就可以马上坐到计算机前，打开 Flash 开始编制繁杂代码的。如果那样的话，你的游戏也许在没有出世前，就因为很多新问题的出现使得你无所适从以致放弃了最初的想法。所以在有了一个游戏想法后，必须设定好一个完整的游戏规划。最好能找出一支笔和一张纸，画出整个游戏的草稿。这样一个游戏的设计就需要遵循以下几个步骤，才能做到事半功倍。

## 1.1 定义游戏规则

游戏规则是整个游戏的灵魂思想，它决定着游戏的可玩性以及在设计游戏时所设定的游戏元素必须符合的某种规则。只要设定了游戏规则，你就可以在口头上向人们谈起这个游戏了。

玩麻将时大家都清楚怎样才能和牌，怎样才会赢得更多的筹码，而同时规则又禁止你去做任何影响麻将公平的作弊活动。关于这些规则的定义，或者是大家所共识的；或者是参与者共同定义的，这就导致了不同地方不同时间对于一个游戏会有不同的玩法。比如扑克，可以升级，也可以斗地主，还有更多。对于计算机游戏设计来说，在设计一个游戏之前，首先要考虑的便是游戏的规则。具体应当包括游戏的目的（比如玩计算机麻将游戏中的目的就是取得更多分数）、游戏与玩家的交互（主要是键盘或者鼠标）、游戏的奖励及惩罚措施等。这些在游戏设计分工时，是属于企划的工作，有了这些，程序员和美工才能开始设计代码和游戏图形。

相对于现实生活而言，在游戏设计中，游戏规则是由游戏设计者决定的。玩家在玩这个游戏时，始终遵循着设计者所设定的游戏规则，这也是游戏设计一个很迷人的地方。但是需要注意的是，设计出来的游戏，最终是为玩家服务的。如果游戏规则定义得过于复杂，玩家必须花很多时间去理解你的游戏规则，那么这个游戏肯定是失败的。所以，一个简单易懂的游戏规则，有助于这个游戏的广泛传播，比如经典的俄罗斯方块、打砖块等。

## 1.2 游戏素材准备

在构思好游戏规则后，这时就需要通过不同的途径去取得所需要的素材。一般的游戏素材是图形和声音，更进一步便是将所需要的图形剪辑、影片剪辑和按钮等元素都制作齐全。

### 1.2.1 图形素材

对于图形素材，如果你有一定的美术基础，可以通过一些专业的图像处理软件（如 Photoshop）或者直接通过 Flash 绘制。这样也有助于展现你的个性与整个游戏图形界面的统一。当然，如果你比较擅长程序设计而美术偏弱，则可以通过网络下载或者从购买的素材光盘中将现有的素材进行稍微的改动。

对于真正的游戏设计，大多会进行分工，比如各有专人负责游戏的程序、美工或音效，包括最开始的游戏策划人员。但是在 Flash 游戏设计中，在没有团队合作时，通常这些都需要游戏设计者独自完成。不过由于 Flash 的易于上手和操作简单，使得程序和美工的结合并不困难。这些也决定了 Flash 游戏具备开发周期短以及动感十足等特点。

### 1.2.2 声音素材

同样地，对于声音素材，也可以通过 CoolEdit 等专业音频处理软件进行录制和编制。不过，如果你并不需要某个特定的音效，通过网络与光盘资源取得是最便利的途径了。

## 1.3 开发游戏引擎

游戏的引擎与汽车的引擎有着相似之处，就像汽车引擎影响着汽车的性能、速度以及操作等很多因素一样，游戏引擎也是一个游戏的大脑，它决定着游戏的关卡和剧情等多项因素。每个游戏都有自己的引擎，而且要想让一个游戏的引擎成为标准并不是一件容易的事。迄今为止，能成为标准的游戏引擎并不多。在引擎的进化过程中，肯·西尔弗曼 1994 年为 3D Realms 公司开发的 Build 引擎是一个重要的里程碑，它的游戏作品便是《毁灭公爵》。《毁灭公爵》已经具备了第一人称射击游戏的所有标准内容，如跳跃、360 度环视以及下蹲等特性。这也说明了一个同样的游戏引擎可以开发出很多款不同的游戏，也直接导致了引擎授权模式的出现。

或许你不需要了解具体什么是引擎，而只要清楚一个游戏需要一个大脑似的系统，它赋予游戏众多元素。比如在打水漂游戏中，首先需要考虑的东西便是石子始终受到的重力作用，还要考虑石子在与水碰撞后所受到的浮力、弹力以及摩擦力等因素。当然，还可以具体考虑水流速度和水流方向等。总之，引擎赋予了整个游戏的一个大致框架。当引擎设置完毕，就只需要往框架上添加具体元素了。

## 1.4 设定主角与敌人

主角在游戏中是处于中心地位的，并且是一条贯穿游戏始终的线索。而且，主角的很多特性也决定着游戏的可玩性，这便是主角激励。主角激励指的是主角的品行、事业给玩

家带来的激励效果。首先,主角的品行和事业决定了玩家是否将主角当作自己在游戏中的化身。没有人想成为一个龌龊、弱智的角色,所以,游戏主角要么是经过修炼成为武功盖世的侠客,要么能演绎一段风花雪月的浪漫,或者整个世界陷入黑暗之中,等待玩家所控制的救世主的出现。这些,往往是玩家在现实生活中不能实现的梦想,而在游戏中只需要经过一定的努力就可以完成。

另外一个重点就是敌人人工智能的设置,这的确是个比较为难的工作,如果敌人人工智能过于复杂,对于玩家来说,难以上手,刚开始接触游戏就想放弃;而如果过于简单,缺乏挑战性,同样会使玩家因为无聊而中途弃手。所以对于敌人的设置,似乎都遵循从易到难的路线。在简单的时候,让玩家易于上手,上手之后,随着玩家的深入,提升游戏难度,这样便促使玩家在存着希望的情况下,不断地去玩这个游戏。敌人设置的难易程度主要体现在敌人生命、速度等属性上和敌人的攻击复杂性。

游戏中的主要亮点便是主角与敌人,所以还需要将他们具体规划好,包括它们的具体属性以及所具有的各种功能。比如打水漂游戏中,主角属性包括石子的质量、形状、旋转次数、转动惯量、入水角度和速度等。你还可以进一步在水中设置很多木桩,如果石子与木桩碰撞,则游戏结束。这些木桩便是游戏的敌人元素,如果赋予它一定的智能,那它就并不是静止不动了,它可以根据玩家石子的速度与方向做出相应的反应去阻止石子继续漂飞。同时也可以设置很多非常有意思的奖励元素,比如小滑板,要求玩家掌握好石子的速度和方向去碰撞它,石子便相应飞得更高更远。这样的话,对于玩家技巧性的要求也提高了,玩家会通过反复玩这个游戏来提高自己的技能,取得更好的成绩。

## 1.5 优化图形界面

图形界面取决于游戏的类型。比如益智类游戏(大富翁系列),需要一个轻快的暖色调界面。对于恐怖游戏(比如生化危机),则需要一个阴沉的冷色调界面。如果你不注意这些,那就会出现图形界面与类型和整个游戏的故事背景非常不协调的情形。而且,在游戏中,应该注意游戏图形界面的多样性和变化性,这样才会给予玩家新鲜感。

应该说,近年来游戏产业在技术方面整体上取得了飞跃发展。现在的游戏大都拥有实时三维效果,而且物体的表面都具有非常逼真的纹理,城墙、草木都非常真实。即使在高分辨率下运行,速度都可以达到60帧/秒。但是,对于Flash来说,在现时还不能做到这一点。所以,设计Flash游戏时,复杂的图形即使非常漂亮,也并不是好事,因为它会严重影响游戏性能。有很多游戏设计者都有过这方面的教训。有些游戏主角通过几千条曲线来创建,同时,拥有复杂华丽的滚动背景,使得游戏运行速度只有几帧每秒。这样的游戏肯定不会流行起来的。虽然现在计算机的发展非常迅速,但是,在设计精美的图形界面时必须考虑游戏性能和计算机配置要求。所以在设计游戏时需要考虑图形要素对游戏的影响,包括像素、矢量、位图、文字、透明、渐变及代码等方面。

### 1.5.1 像素

读者可以发现,在很多图形设计软件中都有像素这个单位,因为整个显示屏就是由一组横纵排列的像素组成的。所以,图形的渲染速度和像素的多少会有很大关系。一组固定数量的像素是按照从左到右,从上到下的顺序排列的。分辨率越高,屏幕上的像素就会越多,所渲染的时间就要越长。

### 1.5.2 矢量

这一要素大家肯定很熟悉,Flash 就是基于矢量的。而与矢量相对的,便是位图。它们最大的区别是位图在放大后会出现锯齿,而矢量图不管放大多少倍,都是非常光滑的。但是在游戏设计方面,考虑的重点应该是它们所占文件大小和渲染时间的长短。以一个边长为 100 像素的蓝色正方形为例,对于位图而言,它基于像素,即一个像素拥有颜色和透明值等属性,假设这些属性占有 4 个字节,那么整个正方形需要 40KB 的空间。所以,后来的程序员就想通过一种简单的方式去实现。比如上述的正方形,不考虑它的每一个像素,而是将矩形作为一个整体考虑。即只需考虑它的边长、颜色和透明值等属性。对于边长的储存只需要 8 个字节就可以了,而颜色和透明值等属性占有 4 个字节,一共才 12 个字节。文件大小相差巨大,这也是 Flash 动画、游戏能在网络上迅速流行的原因。但是,不要因为这点而对位图有所偏见。上述正方形的计算是一个非常规则图形的计算例子,假如是长方形,长宽就都需要 8 个字节了。而如果图形很不规则,并且拥有大量曲线,那矢量图的渲染则需要非常大的计算量,这时使用位图效果会更好。

### 1.5.3 位图

从上面的矢量介绍,你应该明白,位图和矢量除了在体积上的差距,在渲染方面也是很不相同的。因为位图基于像素,所以在渲染时,不需要太复杂的计算。而矢量不同,一般来讲,矢量图形中的曲线数量不应该超过相同位图中像素数量的十分之一,否则,就应该选择使用位图了。当然,这也不是绝对的,你可以结合自己游戏的特点做出选择,因为矢量图可以创建非常光滑的图形,给人一种非常干净的感觉。

### 1.5.4 文字

Flash 中文字的渲染同样需要占用很多的 CPU (中央处理器) 资源。因为文字不仅包含了许多复杂的曲线,而且它的许多边界都需要消除锯齿。所以,在设计游戏时,应该尽量选择笔画较直或者非常平滑,而且棱角分明的字体。如果不得不选择一些特殊字体,那就应该通过(修改—打散)命令将它们打散,然后通过(修改—形状—优化)命令将它们优化。有些人认为打散后会使得体积增大,实际上大家可以自己测试一下,如果是多个字,将

它们打散成单独的文本时，体积会比它们在一个文本框大，而将它们彻底打散体积又会减小。如果再进行优化，体积会更小。

### 1.5.5 透明

有时候使用透明会产生非常奇妙的效果，比如形成一种朦胧感，也会使一些比较粗糙的画面看起来比较光滑。但是当你了解渲染透明的工作原理后，你就需要考虑是否值得选用这种奇特效果。在渲染透明颜色时，它必须执行一种颜色的渗透效果，也就是普通的纯色，在渲染时只需要通过像素的写操作就能实现，但如果是透明效果，它就需要在绘制颜色时读取下面的像素颜色，并最后将两种颜色混合绘出。所以，绘制透明效果的图像需要花费相当于纯色图像 3 倍长的时间。故读者在游戏设计时，对使用透明的尺度要把握好。

### 1.5.6 渐变

渐变同样能使图像产生非常漂亮的效果。但是它的渲染也会耗费大量的 CPU 资源。所以，对于它的使用，你同样需要把握好用量。而且，越复杂的渐变所需渲染的时间越长，越影响游戏性能。通常表现在以下几个方面。

- 渐变中颜色数量越多，渲染时间越长。
- 渐变形状越复杂，渲染时间越长。比如径向渐变比线形渐变渲染时间长。
- 渐变面积越大，渲染时间越长。
- 使用透明颜色的渐变，会使得渲染需要执行渐变和透明两种效果的渲染，所以需要慎重使用。

### 1.5.7 代码设置

由于玩家计算机配置各不相同，对于图像和性能之间的比例关系的要求也各不一样，所以通常游戏会要求玩家设置游戏的图像质量。当然，图像质量越高，游戏性能就会相对弱一点。这需要用到全局属性 `_quality`。需要注意的是设备字体始终是带有锯齿的，所以并不受这个属性影响。

这个属性可以设置下面 4 个值。

- “LOW” 呈现低品质。将不消除图形的锯齿，也不对位图进行平滑处理。
- “MEDIUM” 呈现中等品质。使用  $2 \times 2$  网格（以像素为单位）消除图形锯齿，但不对位图进行平滑处理。适用于不包含文本的影片。
- “HIGH” 呈现高品质。使用  $4 \times 4$  网格（以像素为单位）消除图形锯齿，如果影片是静态的，则对位图进行平滑处理。这是 Flash 使用的默认呈现品质设置。
- “BEST” 呈现极高品质。使用  $4 \times 4$  网格（以像素为单位）消除图形锯齿，并且始终对位图进行平滑处理。

下面来看看代码设置图像质量的实例，具体操作步骤如下。

① 新建 Flash 文档，舞台大小、颜色及帧频率默认。保存为“质量设置.fla”。

② 按 Ctrl+F8 组合键新建一个按钮，在“点击”（Hit）帧添加一个方块，即隐形按钮。在舞台上写上低、中、高，分别在上面拖入一个隐形按钮并依次将实例名称命名为 di, zhong, gao。在旁边绘制一条宽度为 10 的线条，并导入一张位图，如图 1-1 所示。

③ 在舞台时间轴第 1 帧动作面板上添加代码，如图 1-2 所示。

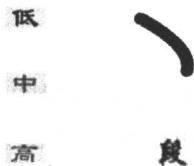


图 1-1

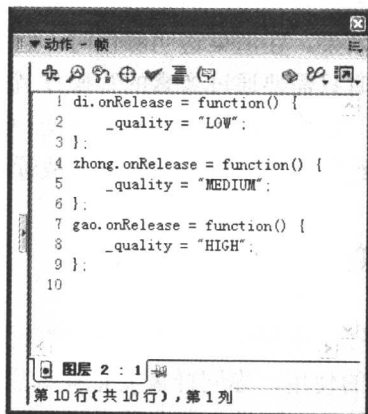


图 1-2

```
di.onRelease = function() {
    _quality = "LOW";
    //如果触发 di 按钮，将影片质量设置为 LOW
};
zhong.onRelease = function() {
    _quality = "MEDIUM";
};
gao.onRelease = function() {
    _quality = "HIGH";
};
```

//同样，如果触发 zhong, gao 按钮，将影片质量分别设置为 MEDIUM, HIGH

④ 一般影片质量设置都是高或低，最终效果的对比如图 1-3 和图 1-4 所示。

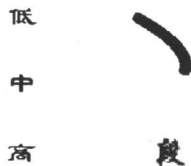


图 1-3

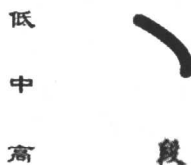


图 1-4

## 1.6 添加音乐音效

### 1.6.1 音乐音效

音乐音效同样取决于游戏的类型。针对暖色调界面，需要欢快的背景音乐，而对于恐怖游戏，背景音乐当然是越深沉越好。而且，在增加游戏的交互性方面，声音和音乐能起到很好的提示效果。

声音是游戏中使用的另一个非常重要的元素。在现实生活中，给人以强大冲击力的无非是视觉和听觉。所以，在游戏中，声音的使用同样能达到一定的震撼效果。而且，一些故事情节的体现，一些游戏信号的传递，使用声音都会达到意想不到的效果。每一种音乐音效都体现着当时游戏应该表达的心情，然后传递给玩家。当然，不同的游戏，音乐音效的使用也是各不相同，比如激烈的游戏会使用令人紧张的音乐，而欢快的游戏如果使用恐怖片中的怪叫就极不协调了。所以，声音的使用也是关系一个游戏好坏的主要因素。下面具体介绍音乐音效所包含的一些对于游戏有较大影响的要素。

音效一般非常简短，所以适合在一个很小的事件上使用，以便在听觉上给予玩家刚才发生了该事件的信号。比如：

- 空战中的射击以及射中敌机。
- 两个影片剪辑的碰撞。
- 某些奖励元素的取得。
- 某些惩罚措施的实施。

而音乐往往用于某个关卡或整个游戏中，这些都需要把握音乐的一些具体要素。

- 节奏。决定音乐类型的主要因素就是节奏，即一首曲子的快慢。如果音乐节奏很快，则会给人紧张感；而如果节奏较慢，则会使人感觉到轻松柔和；如果很慢，也许会使人有种沉闷的感觉，以至于有可能出现莫名的惊恐感。
- 旋律。音乐的节奏可以相似甚至相同，但是旋律总会相差甚远，一种音乐能否带给人美感，很大成分就取决于它的旋律。这也就出现了一些背景音乐反复播放但玩家并不厌烦，那便是好的旋律。相反，有些音乐玩家听上一两遍就失去兴趣了。当然，每个人喜欢的音乐风格均不相同，但是对于好的旋律，大家肯定不会反对的。因为音乐需要占较大空间，所以在 Flash 游戏中，往往都是采取一段音乐的循环播放。这样你就需要在游戏里添加一个音乐开关，即如果玩家觉得背景音乐烦人便可以将它关闭。也可以提供不同的几段音乐让玩家自己选择轮流播放。音乐开关的具体制作，在 1.6.2 节会具体介绍。
- 伴奏。单调的旋律总会使音乐显得非常单薄，所以需要添加不同的伴奏。这需要一些乐理方面的知识，但如果你稍微了解，在使用别人的音乐时，也会做出很好的选择。伴奏一般会用到和弦，比如大家都听说过的大小三和弦。不同的和弦，会决定音调高低的走势。但是和弦与和弦之间的衔接是否协调是非常重要的。通

常它们之间相差 4 度是最理想的状态,比如 A—D—G。如果相差了 6 度或者 7 度,无疑是失败的。当然,大家最需要了解的是不同的和弦会表现出什么样的风格。比如大三和弦会体现欢快,而小三和弦便代表了悲伤。

一个好的游戏,必定会有一段精彩的音乐旋律。就像平常影片中时常出现的精彩旋律一样,比如经典射雕里的那段。这样的旋律往往会给玩家留下深刻的印象,所以在玩游戏过后,当听到哪里播放这段旋律,或旁边有人哼上这段旋律时,玩家都会产生共鸣。因为它本身就像一首很受欢迎的歌曲。

### 1.6.2 声音开关

具体操作步骤如下。

- ① 新建 Flash 文档,舞台大小,颜色及帧频率默认。保存为“声音开关.flas”。
- ② 在舞台上绘制一个红色方块,按 F8 键将其转换为按钮,再按 F8 键转换为影片剪辑。进入影片剪辑编辑状态,在影片剪辑第 2 帧插入关键帧,分别在第 1、第 2 帧的红色按钮上添加“关”,“开”字样,如图 1-5 所示。
- ③ 导入一个声音文件,在库中右击它打开其“链接属性”对话框,将标识符命名为 yinyue。
- ④ 在舞台时间轴上新增图层 2,打开图层 2 第 1 帧动作面板添加声音播放代码,如图 1-6 所示。

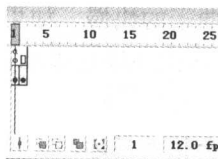


图 1-5



图 1-6

```
mysound = new Sound();
mysound.attachSound("yinyue");
//附加声音文件于舞台上
mysound.start(0, 999);
//循环播放
```

- ⑤ 接着需要在声音开关里,通过开关两个按钮的执行代码去实现声音文件的播放与停止。在主场景中,双击开关影片剪辑进入其编辑状态。在第 1 帧“关”按钮动作面板上添加代码,如图 1-7 所示。

```
on (release) {  
    _root.mysound.stop();  
    //声音文件的播放停止  
    gotoAndStop(2);  
    //跳转到第2帧，即转换到开按钮  
}
```

⑥ 同样在“开”按钮上添加代码，如图 1-8 所示。



图 1-7



图 1-8

```
on (release) {  
    _root.mysound.start(0, 999);  
    gotoAndStop(1);  
    //循环播放声音文件并跳转到关按钮  
}
```

在很多设置单一循环背景音乐的游戏，都需要设置一个音乐开关以满足不同的玩家，当然如果影片大小允许，还可以设置几首不同的背景音乐由玩家自己选择播放。

## 1.7 设置游戏重置

上面步骤完成了游戏的主体制作部分，但是在玩家完成任务或者游戏结束后，也许要重新玩这个游戏，这样便需要添加游戏重置来实现游戏的重复可玩性。

当玩家完成任务或游戏结束，界面通常会跳转到 win 或者 gameover 帧，要实现游戏的再玩，首先得使游戏回到开始界面。所以需要在 win, gameover 帧添加一个 replay 按钮以实现玩家重新回到游戏界面。另外，还需要在跳回开始界面前将某些变量恢复到初始状态，之所以将游戏的重置单独作为一节来讲，是想让读者形成在游戏重置时注意游戏中变量恢复的概念。同时，这也是游戏设计流程的一个重要部分。

## 1.8 游戏测试

再严谨的程序员,也不可能一次性完美地完成一个游戏,他需要反复测试,以调试 Bug。在 Flash 中按 Ctrl+Enter 组合键即可测试,还可以通过监视去查出程序中的问题。在你觉得游戏问题不大时,就应该多让朋友一起测试,这样发现的问题才会全面。

除了游戏的 Bug,另外比较重要的测试便是游戏性能及游戏控制的测试。

### 1.8.1 游戏性能测试

在介绍游戏性能测试实例前,先来了解一下有哪些影响游戏性能的具体因素。

一般在舞台中,元件个数的多少影响着游戏的性能,尤其是运动着的影片剪辑。所以,在游戏设计中,应尽可能让较少的影片剪辑在舞台上运动着。也就是,如果需要某些对象不停的运动,最好设置它在进入玩家视野时才开始运动,而离开玩家视野(即离开舞台)则停止运动或删除。

另一种,影响游戏性能的主要因素就是图形了。因为游戏在运行中,游戏界面是需要不停渲染的,这很耗 CPU 资源,所以图形因素也在于它的被渲染速度。通常,越简单的图形,使用色彩越少,渲染速度越快。当然,另外也还有下面一些影响渲染的因素。

- 尽量使用简单的单色彩图形。也就是在渲染方面,如果是使用渐变色彩,或者透明色彩填充,将严重影响渲染速度。
- 尽量少用线条,尤其是曲线。线条的渲染同样非常耗费 CPU 资源。
- 如果是非常复杂的图像,或者还需要它运动着,应尽量使用位图代替矢量图。也许大家对于位图有一定偏见,实质上,对于复杂的矢量图,尤其是运动着的矢量图,渲染速度会慢很多的。
- 尽量使得各图形保持较小的尺寸。因为尺寸越大,需要渲染的像素会越多。

所以,当一些精美图片影响着游戏性能而使得游戏不够顺畅时,就应该考虑舍弃那些精美图片来提高游戏的性能了。另一个影响游戏性能的主要因素,就是程序的执行效率。这需要了解代码的优化。

那么,又是怎样去测试已经完成的游戏性能呢?最直接的表现方式便是游戏的帧频率测试。下面来具体讲讲怎样制作一个帧频率测试剪辑。游戏性能测试的影片剪辑在制作上非常简单,只需要比较在帧频率下的数值递增与系统时间递增的关系。具体操作步骤如下。

- ① 新建 Flash 文档,舞台大小、颜色默认,帧频率设置为 30,保存为“FPS fla”。
- ② 新建 fps 影片剪辑,在其编辑状态下添加一个动态文本框,并在属性栏将变量命名为 fps。在动态文本框前面添加 fps 文字,并在图层 1 添加 3 帧普通帧。新增图层 2,用来添加测试性能代码。整个影片剪辑布置如图 1-9 所示。
- ③ 在影片剪辑内部时间轴第 1 帧动作面板上添加代码,如图 1-10 所示。