

本章知识

- ◆ 结构化、模块化课件的设计方法
- ◆ 场景的管理和应用
- ◆ 用loadMovieNum()函数实现SWF影片的互相调用
- ◆ Loading（动画预载画面）的制作

前面 4 章讨论和制作的是一些内容少、结构简单的课件，这类课件通常在一个 Flash 场景中就可以制作完成。但是对于一些内容多、结构复杂的课件，例如要将某课程的一章教学内容制作成一个课件，则利用前面的制作思路和方法就行不通了。

对于这类内容多、结构复杂的大型课件，初学者往往感到无从下手，这是因为没有掌握一种系统的、科学的设计方法。本章要研究的多场景导航课件就是专门用来处理容量大、结构复杂的教学内容的。

具体来说，使用 Flash 制作多场景导航课件的方法有以下几种。

1. 模块化课件设计的方法

在规划多场景导航课件时主要使用的是结构化、模块化的程序设计方法，具体设计方法是根据课件的内容将其分解为一个课件主控模块和几个课件功能模块，如果有需要，将课件功能模块再细化为几个功能子模块。课件主控模块用来控制和调度各个课件功能模块的播放，各个课件功能模块具体实现相应课件内容的展示，如图 5-1 所示。

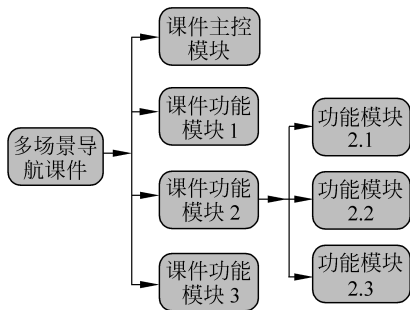


图 5-1 模块化课件设计

2. 典型多场景导航课件制作方法

前面都是在一个场景中制作 Flash 课件，其实 Flash 支持多场景影片制作。在一个 Flash 文档中可以创建多个影片场景，并且在每个场景中独立地制作属于本场景的动画内容，这样利用 Flash 的多场景技术可以很好地实现模块化课件设计的思想。

3. 网络型多模块导航课件制作方法

Flash 提供了一个 loadMovieNum()函数，利用这个函数可以调用外部 SWF 影片，用户可以将每一个课件模块制作成独立的 SWF 影片，然后利用 loadMovieNum ()函数进行相互调用，这样也可以很好地实现模块化课件设计的思想，并且制作出的课件更适合在网络上

播放。

4. Loading（加载影片等待画面）的制作方法

本章第3节还要讨论一下 Loading（加载影片等待画面）的制作方法。在当前的网络环境中播放大型多媒体课件，Loading 是课件必不可少的一个组成部分，它可以使制作的多媒体课件更专业、更适合在网络上播放。

5.1 典型多场景导航课件——变阻器



视频讲解

Flash 动画文件的层次结构一般是这样的：一个 Flash 动画文件可以包含几个场景，每个场景中又包含若干个层和帧。每个场景上的内容通常是某个相同主题的广告，Flash 利用不同的场景组织不同的动画主题。

在播放一个多场景的动画时，将按照它们在“场景”面板中的排列顺序逐个播放，添加了多个场景的“场景”面板如图 5-1-1 所示。

专家点拨：选择“窗口”|“其他面板”|“场景”命令或按 Shift+F2 快捷键可以打开“场景”面板，在该面板中可以对场景进行编辑。

在 Flash 中实现图 5-1 所示的模块化设计思想的典型方法是利用场景组织课件中的各个课件模块，将不同的课件模块放在不同的场景中实现。

一般情况下，多场景导航课件不一定按照“场景”面板中的顺序逐个播放。它需要单独设计一个导航场景，用来控制整个课件的交互播放，这个场景实际上就是课件主控模块。图 5-1-2 所示的是典型的多场景导航课件设计方法示意图。



图 5-1-1 “场景”面板

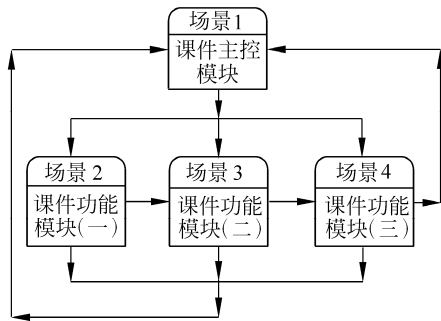


图 5-1-2 多场景导航课件设计方法

本节通过一个课件范例——变阻器的制作过程介绍利用多场景动画实现多场景导航课件的方法。



课件简介

本范例是中学物理中的“变阻器”多场景导航课件，这个课件共包括 5 个模块，其中有一个主控导航模块、4 个功能模块。4 个功能模块包括变阻器的结构原理、变阻器的工作

原理、变阻器的接线方法、变阻器的使用方法。在主控导航模块中可以通过导航按钮在 4 个功能模块间任意切换，反复播放以加深学生对所学知识的理解。

课件运行时先显示一个主控界面，如图 5-1-3 所示。在课件主控界面下面有 4 个导航按钮，单击按钮可以进入到相应的课件功能模块进行播放，图 5-1-4 所示的是“变阻器的结构原理”模块中的一个画面。



图 5-1-3 课件主控界面

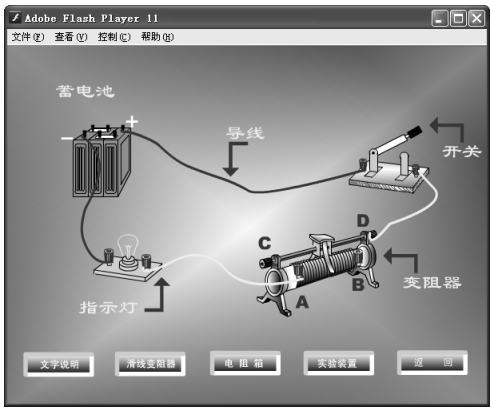


图 5-1-4 课件运行中的一个画面



知识要点

- ◆ 多场景动画的制作方法
- ◆ 多场景课件导航结构的实现过程
- ◆ 在 Photoshop 中创建主控界面图像的方法
- ◆ 隐形按钮的制作方法



制作步骤

5.1.1 整体规划多场景课件的模块结构

首先通过对教学内容和表现形式的分析以模块化设计方法为指导思想对课件结构做整体规划。

通过分析，本课件分为 5 个模块，即一个主控导航模块和 4 个功能模块。主控导航模块在单独的场景中实现，其余 4 个功能模块——“变阻器的结构原理”“变阻器的工作原理”“变阻器的接线方法”“变阻器的使用方法”分别在 4 个单独的场景中实现，具体的模块结构如图 5-1-5 所示。

用户可以在主控导航场景中单击不同的按钮进入相应的场景，在每个场景中也定义了一个按钮用来返回到主控导航场景。主场景中的按钮和各个功能场景中的返回按钮的动作脚本一般用下面的形式实现：

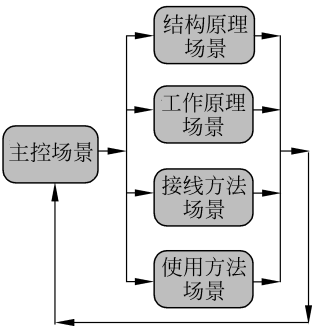


图 5-1-5 课件模块结构示意图

```
on (press) {
    gotoAndPlay("场景名", 1);
}
```

其含义为当单击按钮时课件转到相应的场景中，播放该场景的第1帧。动作脚本中的“场景名”参数决定具体转到的场景。

5.1.2 在 Photoshop 中制作课件主控界面图像

根据教学目标制作课件的主控界面效果，或以艺术性为主，或以突出教学内容为主，其目的都是增强课件的感染力。本课件的主控界面以突出教学内容为主，它利用了一些物理元件图像素材在 Photoshop 中进行制作，效果如图 5-1-6 所示。

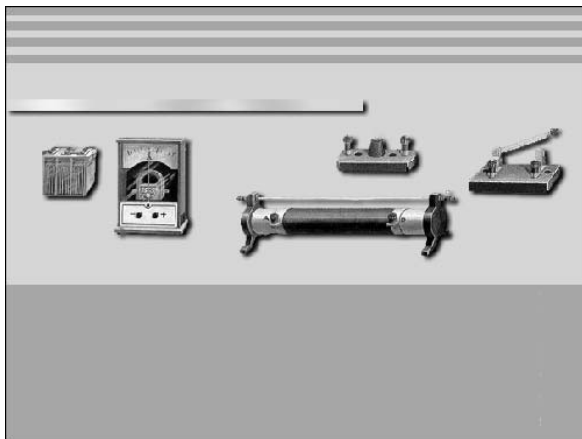


图 5-1-6 在 Photoshop 中制作的课件主控界面图像

通过前面章节的学习，读者对在 Photoshop 中处理位图的方法有了一定的了解在下面的制作过程中将进一步学习在 Photoshop 中处理位图、制作图像背景的方法。

1. 处理图像素材

(1) 运行 Photoshop CS6，选择“文件”|“打开”命令，在配套素材中找到所需的图像素材（文件路径：素材\part5\物理设备.jpg），将其打开。然后新建“未标题-1.psd”文档，设置“尺寸”为 600×448 像素、“背景颜色”为 #ACDFFA。

(2) 单击“物理设备.jpg”窗口，使之成为活动窗口，然后在“图层”面板中双击“背景”，将“背景”图层修改为“图层 0”，并用“魔棒工具”去掉背景色。选择“矩形选框工具”，将鼠标指针移动到图片上拖动鼠标，将变阻器图像用虚线框住，按 Ctrl+C 快捷键将变阻器图像复制。

(3) 单击“未标题-1.psd”，使之成为活动窗口，按 Ctrl+V 快捷键将变阻器图形粘贴到新建文档的画布中，此时在“图层”面板中可以看到新增加了一个图层，如图 5-1-7 所示。

专家点拨：实际上，在“选框工具”组中包括了“矩形选框工具”“椭圆选框工具”



“单行选框工具”和“单列选框工具”。“矩形选框工具”可以在图像中选取矩形区域，按下 Shift 键可以选择正方形区域；“椭圆选框工具”可以在图像中选取椭圆形区域，按下 Shift 键可以选择圆形区域；“单行选框工具”和“单列选框工具”只能在图像中建立一个像素宽的横线区域或竖线区域。在选取的区域中可以进行各种操作，选框以外的区域是不可编辑的。

(4) 选择“编辑”|“自由变换”命令，在图形四周将出现 8 个控制点，将鼠标指针放在控制点上，光标变成双箭头，拖动鼠标调整图形的大小，如图 5-1-8 所示。

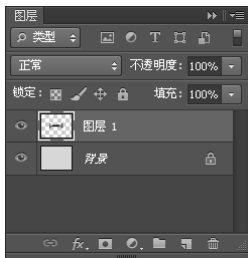


图 5-1-7 新增图层

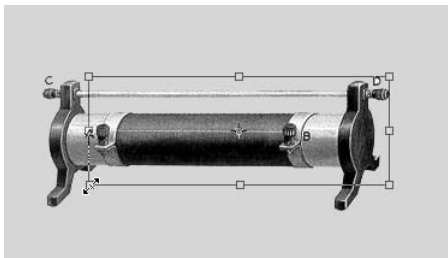


图 5-1-8 调整变阻器图形的大小

专家点拨：除了可以利用“自由变换工具”以外，还可以利用“变换工具”对图像进行缩放、旋转、斜切、透视等一系列的操作。

(5) 调整图形的大小合适后，按 Enter 键取消对图形的选择，用“移动工具”调整图形位置。

(6) 用同样的方法将其他电器元件全部复制粘贴到“未标题-1.psd”中，并调整好大小和位置，整个文档的图层结构如图 5-1-9 所示。

专家点拨：对图层可以进行新建、复制、删除、移动等操作，并可以对图层的不透明度、显示模式等进行设置，还可以对图层进行投影、斜面和浮雕等效果编辑。另外，所有的操作只对活动图层起作用，不会影响到其他图层。活动图层在“图层”面板上高亮显示。

(7) 单击“背景”图层前的眼睛图标，隐藏背景图层，然后单击“图层”面板右上角的小三角按钮，在弹出的快捷菜单中选择“合并可见图层”命令，所有可见图层将全部合并到一个图层上，图层结构如图 5-1-10 所示。



图 5-1-9 “图层”面板

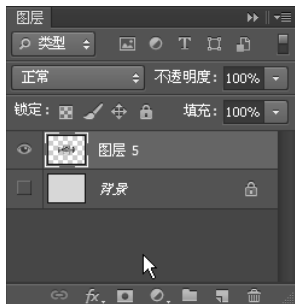


图 5-1-10 合并可见图层后的图层结构

2. 绘制背景

(1) 单击“背景”图层前的眼睛图标, 解除对“背景”的隐藏, 然后隐藏“图层 5”。在“图层”面板的下方单击“创建新图层”按钮, 增加一个新图层, 然后选择工具箱中的“矩形选框工具”, 拉出一个矩形虚线框, 再按住 Shift 键在画布上拉出 4 个矩形虚线框, 如图 5-1-11 所示。

(2) 选择工具栏上的“油漆桶工具”, 单击“设置前景色”按钮, 弹出“拾色器”对话框, 将鼠标指针放在“选择前景色”区域中单击选取颜色, 然后单击“好”按钮, 前景色变为刚拾取的颜色, 将鼠标指针放在选取的区域上单击, 为矩形选区填充颜色, 如图 5-1-12 所示。



图 5-1-11 用“矩形选框工具”选取填充区域

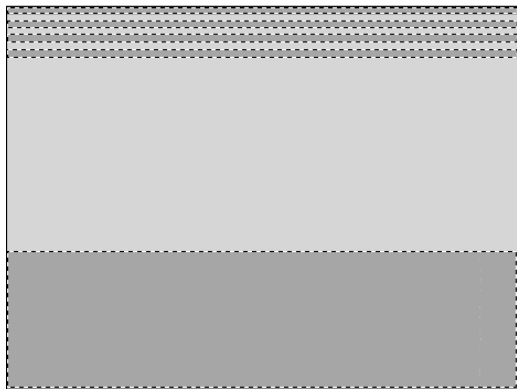


图 5-1-12 填充颜色

专家点拨:“油漆桶工具”组中包含了“油漆桶工具”和“渐变工具”, 利用“油漆桶工具”只能对图像进行单颜色填充, 利用“渐变工具”可以在图像中填充层次连续变化的颜色, 从而达到色彩渐变的图像效果。

(3) 解除对“图层 5”的隐藏, 图像的效果如图 5-1-13 所示。

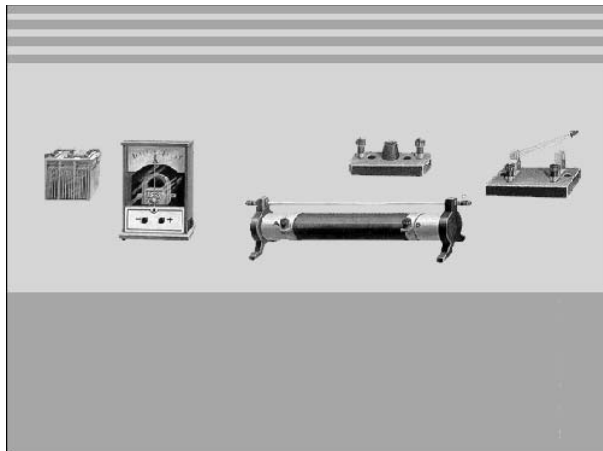


图 5-1-13 绘制背景



(4) 在“图层”面板上单击“图层 5”，使之成为活动图层，然后选择工具箱中的“矩形选框工具”，在画布左上方画一个细长的矩形虚线框，再选择“渐变工具”，在主菜单下方出现“渐变选项”选项栏，如图 5-1-14 所示。



图 5-1-14 “渐变选项”面板

(5) 在“渐变选项”选项栏中单击 ，弹出“渐变编辑器”对话框。

(6) 单击“预设”列表框中的小三角形按钮，在弹出的快捷菜单中选择“色谱”选项，在“预设”列表框中可以看到系统默认的几种色谱渐变色，选择其中的“色谱”选项，如图 5-1-15 所示，单击“确定”按钮。

专家点拨：在“渐变编辑器”对话框中不仅可以选择各种渐变色进行填充，用户还可以自己编辑渐变色。

(7) 将鼠标指针放在虚框选区左边，光标变成小十字形状，水平向右拖动鼠标，释放后可以看到如图 5-1-16 所示的效果。



图 5-1-15 “渐变编辑器”对话框

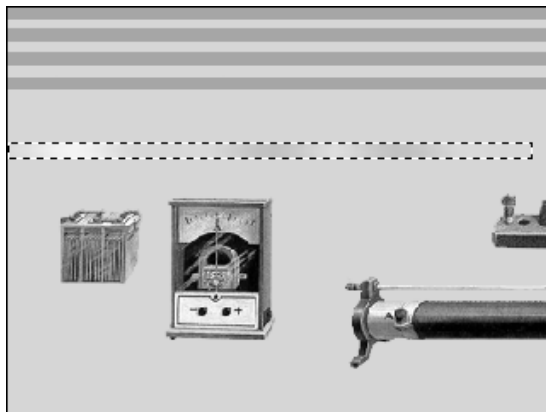


图 5-1-16 填充渐变色

专家点拨：渐变色的填充样式除了线性渐变以外还有径向渐变、角度渐变、对称渐变和菱形渐变等。

(8) 保持“图层 5”处于活动图层，选择“图层”|“图层样式”|“投影”命令，弹出“图层样式”对话框，如图 5-1-17 所示。

专家点拨：在“图层”面板中双击图层也可以弹出“图层样式”对话框。

(9) 在该对话框中设置“不透明度”为 100%，调整角度为 130°，其余参数保持不变，然后单击“确定”按钮，得到如图 5-1-18 所示的效果。



图 5-1-17 “图层样式”对话框

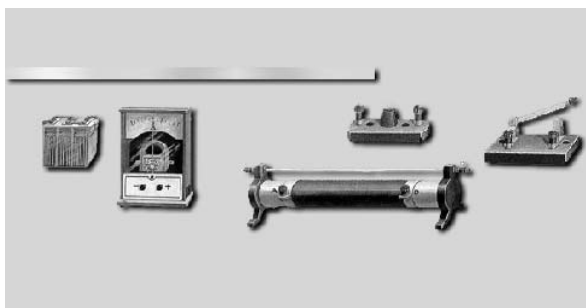


图 5-1-18 为图层增加投影效果

专家点拨：在“图层样式”对话框中还可以为图层增加描边、内发光、外发光、斜面和浮雕等一系列效果，这些效果大部分作用于图层画面的边缘，从如图 5-1-18 所示的画面效果就容易看到。如果画面覆盖了整个图层，则很多效果不能显示出来。

3. 输入标题

(1) 选择“横排文字工具”，然后展开“字符”面板，如图 5-1-19 所示。

专家点拨：用户也可以选择“窗口”|“字符”命令打开“字符”面板。

(2) 在“字符”面板中设置文字的大小、字体、颜色，然后在画布上单击得到一个新图层，在图层中输入标题，并用“移动工具”调整好位置。单击“图层”面板上的小三角按钮，在弹出的快捷菜单中选择“合并图层”命令，将所有图层拼合在一起，整个主控界面制作完成，如图 5-1-20 所示。

专家点拨：用户也可以选择“图层”|“合并图层”命令进行拼合。



图 5-1-19 “字符”面板



图 5-1-20 制作完成的主控界面

(3) 选择“文件”|“存储为 Web 所用格式”命令，将编辑好的图像优化输出。

5.1.3 在 Flash 中创建多场景文档

(1) 新建一个 Flash 文件 (ActionScript 2.0)，设置“尺寸”为 600×448 像素、“背景颜色”为淡紫色 (#979AF3)，其他参数保持默认。

(2) 选择“窗口”|“其他面板”|“场景”命令或按 Shift+F2 快捷键打开“场景”面板。在该面板中有一个默认的“场景 1”，单击“添加场景”按钮  为影片添加 4 个场景。双击场景名，分别将场景命名为“主控界面”“结构原理”“工作原理”“接线方法”和“使用方法”，如图 5-1-21 所示。

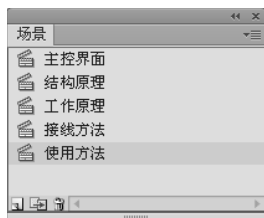


图 5-1-21 添加场景

添加完场景后关闭“场景”面板，接下来在各场景中进行模块功能的制作。

5.1.4 制作分场景动画

1. 制作“主控界面”场景

(1) 单击“编辑场景”按钮  弹出下拉菜单，如图 5-1-22 所示，在菜单中选择“主控界面”，编辑环境便转换到“主控界面”场景中，在舞台的左上角相应地显示出场景的名称，如图 5-1-23 所示。



图 5-1-22 选择编辑场景



图 5-1-23 “主控界面”场景

(2) 选择“文件”|“导入”|“导入到舞台”命令，把在 Photoshop 中制作的主控界面图像导入场景中（文件路径：素材\part5\主控界面.jpg），并用“选择工具”调整好位置，将图层重命名为“背景”。

(3) 新建一个名称为“按钮 1”的按钮元件，在按钮元件编辑场景中制作一个按钮。在“库”中右击“按钮 1”，在弹出的快捷菜单中选择“直接复制”命令，复制 3 个按钮，并分别命名为“按钮 2”“按钮 3”和“按钮 4”。在各个按钮的编辑场景中只要将文本更改成对应的主题即可，这样得到了 4 个导航按钮，如图 5-1-24 所示。

(4) 在“主控界面”场景中插入一个新图层并命名为“导航按钮”，将准备好的按钮从“库”面板中拖放到场景中，并调整好位置，主控界面场景布局完成，效果如图 5-1-25 所示。对于导航按钮的脚本定义将在后面进行介绍。

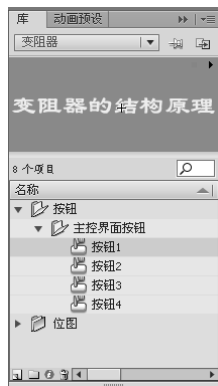


图 5-1-24 “库”中的导航按钮



图 5-1-25 主控界面场景效果

2. 制作“结构原理”功能模块场景

在本课件的 4 个功能模块中，“结构原理”是制作相对比较复杂的一个模块，它由按钮来实现多模块交互演示过程，其中包括“文字说明”“滑线变阻器结构”“电阻箱”和“实验装置图”4 个影片剪辑元件。由于很多制作方法在前面已经学习过，在这里不再重复叙述，只对其中几种重要的制作方法进行讲解。

(1) “文字说明”影片剪辑元件是一个打字效果动画，下面介绍一下其制作方法。切换到“结构原理”场景，按 Ctrl+F8 快捷键新建一个名为“文字说明”的影片剪辑元件。将“图层 1”命名为“文本”，用“文本工具”输入文本，并把“文本”图层延伸至 180 帧。插入新图层，命名为“遮罩”，在“遮罩”图层的第 1 帧用“刷子工具”将文本的第 1 个字涂上颜色。在第 3 帧插入一个关键帧，用“刷子工具”再将文本的第 2 个字涂上颜色，依照这种方法操作，将所有的字均涂上颜色。右击“遮罩”图层，在弹出的快捷菜单中选择“遮罩层”命令创建遮罩动画，按 Enter 键预览观察动画效果，可以看到打字的动画效果。

专家点拨：为了使动画更逼真，在“遮罩”图层不要逐帧插入关键帧进行涂色，而要根据写字的快慢灵活地插入关键帧。



(2) 介绍滑线变阻器的结构是本模块的一个重要内容, 在这里将其创建成一个名为“滑线变阻器结构”的影片剪辑元件, 这个元件的图层结构如图 5-1-26 所示。

在“变阻器”图层的第 1 帧放置一个滑线变阻器的静止图片, 从第 2~7 帧分别放置 6 个影片剪辑元件, 每个元件介绍相对应的部件的组成情况, 事先制作好的元件如图 5-1-27 所示。



图 5-1-26 “滑线变阻器结构”影片剪辑元件



图 5-1-27 介绍每个部件的影片剪辑元件

“按钮文本”图层用“文本工具”输入了一些文字, 效果如图 5-1-28 所示。“按钮”图层中放置了 6 个隐形按钮覆盖在相应的文字上, 效果如图 5-1-29 所示。隐形按钮在场景中为方便编辑用蓝色显示, 在播放动画时是不显示的。

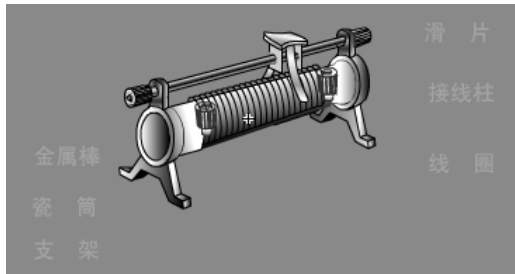


图 5-1-28 按钮文本

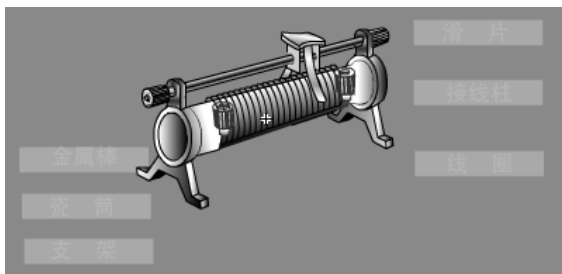


图 5-1-29 添加了阴影按钮

专家点拨: 当按钮的“弹起”帧无任何内容时, 该按钮称为隐形按钮。其特点是便于隐藏, 它可以放置在动画的任意位置而不会显示, 只有当将鼠标指针移动到响应区域上时才能触发按钮动作, 并产生相应效果。这种利用隐形按钮实现很多特殊效果的方法在课件制作中很实用。

(3) 将“结构原理”场景中的内容全部制作成影片剪辑元件, 由按钮来调用, 这样既简化图层结构又便于操作, 其图层结构如图 5-1-30 所示。

3. 制作其他功能模块场景

“使用方法”场景的制作方法与“结构原理”场景的相同, 其他两个模块也是用简单的

传统补间来实现，具体的制作请参考课件源文件。

由于本课件的元素很多，为了便于管理，按元件类型建立了不同文件夹，在文件夹中又按场景建立了分文件夹，将元件分别放在不同的文件夹中，这样使“库”面板的结构条理清楚、内容一目了然，如图 5-1-31 所示。



图 5-1-30 “结构原理”场景的图层结构



图 5-1-31 建立库文件夹

5.1.5 定义动作脚本实现多场景导航

(1) 切换到“主控界面”场景，单击“变阻器的结构原理”按钮，在“动作”面板中定义这个按钮的动作脚本如下：

```
on (press) {
    gotoAndPlay("结构原理", 1);
}
```

这段动作脚本的含义是当单击“变阻器的结构原理”按钮时，播放头跳转到“结构原理”场景的第 1 帧开始播放，这就实现了多场景的自由跳转。

(2) 分别选择其他 3 个导航按钮，在“动作”面板中定义其脚本。

“变阻器的工作原理”按钮的动作脚本如下：

```
on (press) {
    gotoAndPlay("工作原理", 1);
}
//单击按钮，跳转到“工作原理”场景的第 1 帧开始播放
```

“变阻器的接线方法”按钮的动作脚本如下：

```
on (press) {
    gotoAndPlay("接线方法", 1);
}
```



//单击按钮，跳转到“接线方法”场景的第1帧开始播放

“变阻器的使用方法”按钮的动作脚本如下：

```
on (press) {  
    gotoAndPlay("使用方法", 1);  
}
```

//单击按钮，跳转到“使用方法”场景的第1帧开始播放

(3) 在每个功能模块的场景中都放置了一个“返回”按钮，以便于播放完该场景内容后返回到“主控界面”场景，其动作脚本如下：

```
on (press) {  
    gotoAndPlay("主控界面", 1);  
}
```

//单击按钮，跳转到“主控界面”场景的第1帧开始播放

以上是“主控界面”场景的动作脚本，其他各功能模块场景的动作脚本定义方法与前面学习的相同，读者可参考课件源文件。

(4) 测试并保存文档。由于本课件内容较多、结构复杂，在制作完成后应该反复进行测试，直到课件导航功能达到要求，再将课件保存并导出。

选择“控制”|“测试场景”命令，可以测试当前编辑场景。一般先将制作好的场景测试正确，再选择“控制”|“测试影片”命令测试整个影片。

5.2 网络型多模块导航课件——金属的物理性质



视频讲解

多媒体课件的重要特征是具有多媒体性和交互性。多媒体性指课件内容是由文字、图片、动画、音频、视频等多种媒体整合而成的，具有丰富的感染力；交互性指用户可以控制课件的播放，可以实现用户和计算机的相互交流，从而达到控制学习过程的目的。多媒体课件的这些特征使传统教学方式受到挑战，也使多媒体课件具备了强大的生命力和应用需求。

随着网络技术的飞速发展，特别是宽带技术的普及应用，多媒体课件的网络化应用发展趋势越来越明显。这里所说的多媒体课件的网络化应用是指在 Internet 网络中用户可以实时下载播放多媒体课件。那么怎样既能保证多媒体课件的多媒体性和交互性的特征，又能保证用户在 Internet 网络中实时、稳定地下载播放多媒体课件内容呢？这是多媒体课件网络化亟待解决的问题。多媒体性和交互性使得课件体积（文件大小）相对较大，特别是对于复杂的大型多媒体课件更是这样，而体积较大的文件必然会影响网络中的下载、播放速度。这是一个十分突出的矛盾，解决这个矛盾应该是解决多媒体课件网络化的关键。

牺牲多媒体课件的多媒体性和交互性，以减少课件文件的体积，肯定是不可取的方法。在保证多媒体课件原有的多媒体性和交互性的基础上实现多媒体课件的网络化应用是解决问题的方向。多媒体课件的体积大了，网络化应用就难以实现。从多媒体课件的体积入手，“化大为小，分而治之”是一种有效的解决方法，具体思路如下。

1. “化大为小”——将多媒体课件模块化

多媒体课件的规模是难以控制的。如果一味地控制课件的规模,虽然可以起到控制课件体积的作用,但必然会影响课件的效果和完整性。将多媒体课件模块化,“化大为小”的思路是比较明智的选择。根据多媒体课件的内容,采取自顶向下,逐步细化、模块化的方法,将一个大型的课件分解为由若干课件模块组成。这样,一个大型的多媒体课件就被分解成若干模块,每个模块就是一个“小多媒体课件”,它们的体积比较小,易于在网络中实时下载、播放。

2. “分而治之”——在交互控制下各模块分别下载播放

按照“化大为小”的方法,多媒体课件被分解成若干课件模块,通常包括一个主控模块和若干功能模块,所有功能模块合在一起实际上就是多媒体课件的全部内容。通过主控模块可以交互控制功能模块的播放。“分而治之”的思路是当网络用户提出多媒体课件应用需求时不是将多媒体课件的全部内容一次下载到用户本地的计算机上,而是将主控模块先下载到用户本地的计算机上,在主控模块的控制下,用户可以根据需要再分别下载功能模块进行播放。这样就减轻了网络压力,能取得比较好的应用效果。

能制作多媒体课件的软件有很多,但最适合实现多媒体课件网络化应用的软件非 Flash 莫属,原因如下:

- ◆ Flash 是目前最流行的动画制作软件,它自身不仅具有图形图像处理能力,还支持多种媒体(图像、音频、视频等)的导入和编辑。另外,利用 Flash 的脚本语言可以实现强大的交互功能。因此用 Flash 制作的课件具有较强的多媒体性和交互性。
- ◆ Flash 主要用于网络应用程序的开发,它采用矢量图形技术,制作的 SWF 动画文件非常小,因此 Flash 制作的多媒体课件非常适合在网络上播放。另外,Flash 动画目前是 Internet 事实上的动画标准,具有强大的兼容性和广阔的发展空间。

- ◆ Flash 提供了一个 loadMovieNum() 函数(和 loadMovie() 函数),可以调用外部 SWF 影片文件,能较好地实现模块化程序调用的功能。在设计时把课件分割为若干个模块后,将每个课件模块制作成独立的 Flash 影片,然后利用 loadMovieNum() 函数(或者 loadMovie() 函数)实现各个课件模块间的相互调用,如图 5-2-1 所示。

利用这种设计思想制作出来的课件既较好地实现了模块化功能,又使课件具备了更强的网络特性,因为这种类型的课件在播放时并不需要先把全部的课件模块都装载到计算机的内存中,只需要先装载课件的主控模块,需要时在课件主控界面上单击控制按钮,再把其他

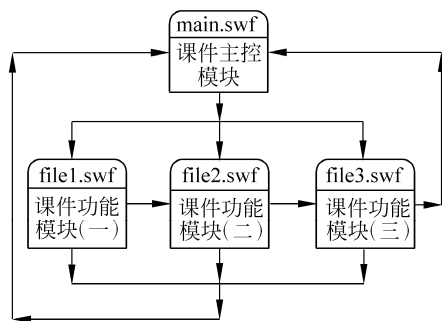


图 5-2-1 loadMovieNum ()函数实现模块化课件设计



的课件模块装载运行。



课件简介

本范例是高中化学中的“金属的物理性质”网络型多模块导航课件。这个课件共包括6个模块，分别是一个课件片头、一个主控导航模块、4个功能模块。其中4个功能模块包括金属的内部结构、金属的导热性、金属的导电性、金属的延展性。

这个课件的特点是动画十分精彩，通过动画直观模拟了金属物质微观世界的奥秘，这是传统的教学模式很难实现的。课件运行时，先播放一个课件片头，如图5-2-2所示。



图 5-2-2 课件片头

在这个课件片头播完以后，影片将自动跳转到课件的主控导航界面，如图5-2-3所示。在主导导航界面下面有4个导航按钮，单击它们可以分别调用独立的影片文件，以打开相应的课件功能模块进行播放。图5-2-4所示的是“金属的导电性”功能模块播放时的一个画面。



图 5-2-3 课件的主控导航界面



图 5-2-4 课件播放时的一个画面



知识要点

- ◆ 利用 Flash 制作多模块导航课件主界面的方法

- ◆ 课件片头在多模块课件结构中的实现方法
- ◆ 利用 loadMovieNum () 函数实现网络型多模块导航课件的方法
- ◆ 调用外部 SWF 文件的方法



制作步骤

5.2.1 整体规划网络型导航课件的模块结构

1. 规划课件内容和结构

在制作课件之前先进行模块化课件设计。通过分析,本课件共划分为6个模块,即一个课件片头模块、一个主控导航模块、4个功能模块。课件片头模块和主控导航模块分别用一个独立的SWF影片文件实现,另外的4个功能模块对应演示课件的具体内容,它们分别用4个独立的SWF影片文件实现,最后用loadMovieNum ()函数实现整个课件的导航控制功能。

这6个课件模块对应的SWF影片文件名如下:

课件片头模块——片头.swf。

主控导航模块——主控界面.swf。

功能模块1(金属的内部结构)——内部结构.swf。

功能模块2(金属的导热性)——导热性.swf。

功能模块3(金属的导电性)——导电性.swf。

功能模块4(金属的延展性)——延展性.swf。

需要特别注意的是,6个SWF影片文件必须放在同一个文件夹下。

2. 绘制课件层次结构图

为了更清楚地表达上面规划的课件结构,先绘制一个课件层次结构图,以方便在设计制作之前做到有的放矢、心中有数,如图5-2-5所示。

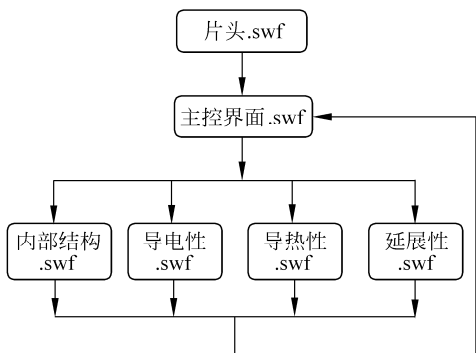


图 5-2-5 课件结构层次图

5.2.2 制作课件片头影片

(1) 新建一个Flash文件(ActionScript 2.0), 设置“尺寸”为600×448像素、“背景颜色”为蓝色(#000099), 其他参数保持默认。

(2) 在主场景中创建一个课件片头动画效果, 如图5-2-6所示。在本课件的片头播放时, 滚动的小球、不断变换的字母条、颜色各异的金属符号加上音乐, 极富感染力, 为课件增加了效果。

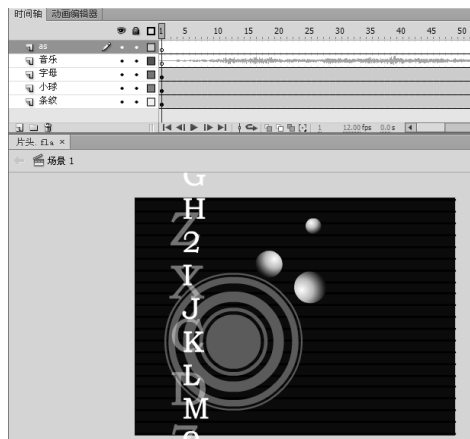


图 5-2-6 课件片头

5.2.3 制作主控导航影片

主控导航影片文档属性和片头影片一样。在舞台上绘制一个渐变色矩形，覆盖整个舞台，这样得到一个渐变色的背景效果。另外，导航条和课件标题都用影片剪辑元件，它们将整个界面动态地表现出来，增加了课件的感染力，如图 5-2-7 所示。



图 5-2-7 主控界面效果

在主控界面的下边放置了 4 个导航按钮，它们是主控导航界面上最重要的对象，分别定义它们的动作脚本就可以实现对其他课件功能模块的调用。这 4 个导航按钮被制作成了一个独立的影片剪辑元件，名字叫“导航条”。这里将 4 个导航按钮放置在一个影片剪辑元件中就是为了让它们呈现一种飘动的效果。

5.2.4 制作其他功能模块影片

本课件中有 4 个功能模块对应 4 个独立的 SWF 影片文件，当单击主控界面的导航按钮时就会调出相应的 SWF 影片文件。这 4 个 SWF 影片通过交互动画分别演示金属的不同

特性，下面分别介绍4个功能模块的制作思路。

1. 功能模块1——内部结构.swf

本课件将金属的内部结构设计成一个功能模块，用独立的Flash影片（内部结构.swf）来实现这个功能模块的内容。打开影片源文件“内部结构.fla”，图层结构如图5-2-8所示。

这个影片共有5个图层，“标题文本”图层放置的是标题和文字说明；“晶体MC”图层放置的是“金属离子”“自由电子”和“正电荷”影片剪辑；“典型结构MC”图层的第2帧上放置了“体心立方”“面心立方”和“密排六方”3个影片剪辑，这是课件的主要演示内容；“按钮”图层上有两个关键帧，第1个关键帧上是实现跳转到“典型结构MC”图层的第2帧进行播放的按钮，第2个关键帧上是实现返回主控界面的按钮；as图层上的两个空白关键帧上分别定义了停止动作，用来配合按钮控制动画。



图 5-2-8 内部结构.swf 的图层结构

运行“内部结构.swf”，先播放影片第1帧并停止，屏幕上显示标题、文字说明以及“金属离子”“自由电子”“正电荷”影片剪辑元件的组合动画效果。

单击屏幕右下角的“点击观看”按钮跳转到“典型结构MC”图层的第2帧进行播放，由于“典型结构MC”图层的第2帧放置了“体心立方”“面心立方”和“密排六方”3个影片剪辑，所以这时屏幕上显示这3个影片剪辑实例。

“体心立方”“面心立方”和“密排六方”3个影片剪辑元件分别用动画展示3种典型金属晶体结构，在每个影片剪辑元件中通过定义一些传统补间动画来模拟金属晶体的内容结构，十分直观。

“密排六方”影片剪辑元件的图层结构如图5-2-9所示。“密排六方”影片剪辑元件的动画效果如图5-2-10所示。单击“密排六方”按钮就可以开始播放动画，从而模拟展示这种类型的金属晶体内部结构。

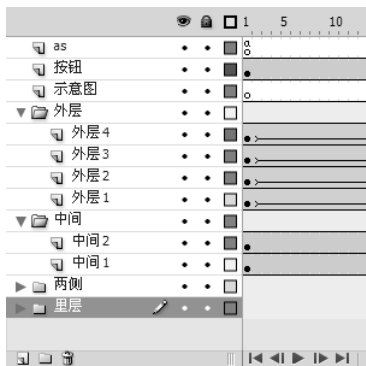


图 5-2-9 “密排六方”元件的图层结构



图 5-2-10 “密排六方”的动画效果

专家点拨：对于图层特别多的元件或场景，在制作中可分类合理设置图层文件夹，然后将相应的图层移动至图层文件夹中，这样可以给课件制作带来极大的便利。



2. 功能模块 2——导热性.swf

本课件将金属的导热性设计成一个功能模块，用独立的 Flash 影片（导热性.swf）来实现这个功能模块的内容。

（1）播放“导热性.swf”影片，单击屏幕上的“开始”按钮后动画演示点燃酒精灯给金属加热，可以看到金属晶体慢慢从右到左变红，最后整块金属达到同样的温度。整个动画过程只包含了传统补间动画，它是利用色调的变化来实现这个动画过程的。

（2）打开影片源文件“导热性.fla”，图层结构如图 5-2-11 所示。

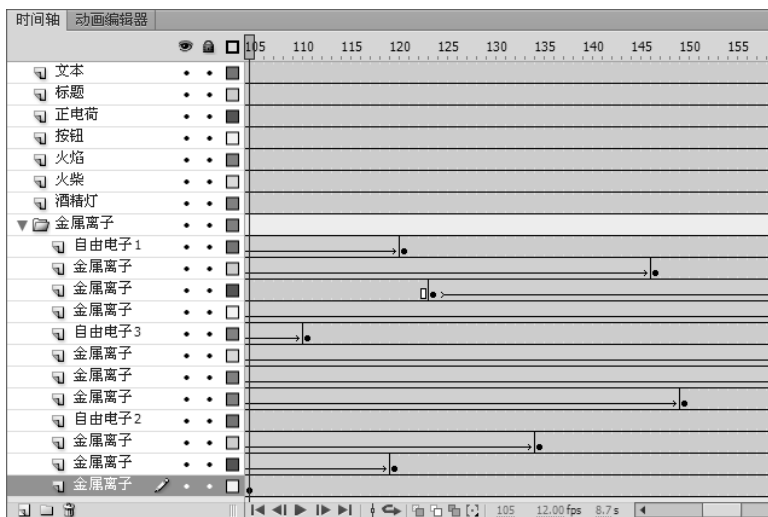


图 5-2-11 图层结构

从图层结构可以看出，这个影片主要是由“金属离子”和“自由电子”的传统补间动画实现的。进一步研究可以发现，每个传统补间动画都是利用对象的色调变化制作完成的。

例如最下边的“金属离子”图层是一个从第 25 帧至第 105 帧的传统补间动画。具体制作步骤是在这个图层的第 25 帧放置一个“金属离子”影片剪辑，在第 105 帧插入关键帧，选择第 25 帧创建传统补间动画。单击第 105 帧上的实例，打开“属性”面板，在“色彩效果”栏的“样式”下拉列表中选择“色调”选项，将红、绿、蓝值分别设置为 255、90、0，将“色调”设置为 60%，如图 5-2-12 所示。



图 5-2-12 设置色调

3. 另外两个功能模块

本课件用独立的 Flash 影片（导电性.swf）来实现金属的导电性这个功能模块的内容，用独立的 Flash 影片（延展性.swf）来实现金属的延展性这个功能模块的内容，具体制作步骤这里不再详述，读者可以参考课件源文件。

5.2.5 定义动作脚本以实现网络型课件的交互导航控制

1. 定义主控界面上的导航按钮动作

(1) 打开“主控界面.fla”文件，在“库”面板中双击打开“导航条”影片剪辑元件，在这个元件的编辑场景中，在“内部结构”图层的第11帧上选择“内部结构性”按钮，在“动作”面板中定义这个按钮的动作脚本如下：

```
on(release) {
    loadMovieNum("内部结构.swf", 0);
}
```

这段动作脚本的功能是单击并释放按钮后加载名字为“内部结构.swf”的影片文件，级别为0。

专家点拨：在“动作工具箱”中展开“全局函数”|“浏览器/网络函数”类别，可以看到loadMovieNum()函数。

(2) 按照同样的方法为其他3个按钮定义动作脚本。

“导热性”按钮的动作脚本如下：

```
on(release) {
    loadMovieNum("导热性.swf", 0);
}
//单击并释放按钮后，加载名字为“导热性.swf”的影片文件，级别为0
```

“导电性”按钮的动作脚本如下：

```
on(release) {
    loadMovieNum("导电性.swf", 0);
}
//单击并释放按钮后，加载名字为“导电性.swf”的影片文件，级别为0
```

“延展性”按钮的动作脚本如下：

```
on(release) {
    loadMovieNum("延展性.swf", 0);
}
//单击并释放按钮后，加载名字为“延展性.swf”的影片文件，级别为0
```

这4个按钮的动作脚本主要使用的就是loadMovieNum()函数，功能是加载课件的4个功能模块对应的SWF影片文件。从脚本中可以看出，影片文档全部被设置加载在0级上，这样后面被加载的动画总会取代原来0级别上的动画文件，不会出现重叠的问题。

2. 定义返回主控界面的动作

从4个功能模块SWF影片返回到主控界面SWF影片是通过返回按钮来实现的，返回按钮的脚本如下：



```
on(release) {  
    loadMovieNum("主控界面.swf", 0);  
}  
//单击并释放按钮后，加载名字为“主控界面.swf”的影片文件，级别为0
```

这个动作脚本的功能是通过 loadMovieNum()函数调用主控界面 SWF 影片文件，从而实现返回主控界面的目的。

3. 定义课件片头到主控导航界面的连接动作

“片头.swf”是一个独立的影片文件，它也是通过 loadMovieNum()函数调用“主控界面.swf”影片文件来实现自然连接至主控界面的，它的动作脚本被定义在片头动画播放完的最后一帧上：

```
loadMovieNum("主控界面.swf", 0);
```

当片头动画播放完就开始执行这段动作脚本，也就是立即调用“主控界面.swf”文件，进入主控导航界面进行播放。

4. 测试、保存文档

至此整个课件的导航结构制作完成，这时可以检查一下，6 个课件模块对应的 SWF 影片是否放在同一个文件夹里，如果不在同一个文件夹里，调用就不能实现。检查完后再进行课件的测试，测试成功后保存并导出。

5.2.6 ActionScript 补习班——加载函数详解

在制作 Flash 课件时，由于交互的需要，常常在当前影片播放不停止的情况下播放另外一个影片，或者是在多个影片间自由切换，这时就会用到 loadMovieNum()函数或者 loadMovie()函数。loadMovieNum()函数是加载一个影片或图片到一定的级别（_level），而 loadMovie()函数是加载一个影片或图片到一个目标路径（target），例如场景中的一个空影片剪辑实例。

1. loadMovieNum()函数

如果要将 SWF 文件或 JPEG 文件加载到特定的级别，就要使用 loadMovieNum()函数。loadMovieNum()函数的作用是在播放原来加载的 SWF 文件的同时将 SWF 文件或 JPEG 文件加载到 Flash Player 中的某个级别。

loadMovieNum()函数的一般形式如下：

```
loadMovieNum("URL", level);
```

URL 设置加载的 SWF 文件的绝对路径或相对路径。一般这里都是使用相对路径，也就是直接用 SWF 文件名代替这个参数。这时，必须让多个 SWF 文件（主控模块和若干功能模块）都应该存放在相同的文件夹下。

level (级别) 用于设置将动画加载到哪一级界面上。在 Flash 播放器中, 按照加载的顺序, 影片文件被编上了号。第一个加载的影片将被放在最底层 (0 级界面), 以后载入的影片将被放在 0 级以上的界面中。例如, 在一个主影片中利用下面的程序代码加载另一个影片:

```
loadMovieNum("概述.swf", 0);
```

这个函数将要加载的“概述.swf”的级别设置为 0, 由于主影片默认的也是在 0 级别上, 所以被加载的影片将取代原来 0 级别上的主影片文件。如果把加载的影片的级别定义为 1:

```
loadMovieNum("概述.swf", 1);
```

则 0 级别上的主影片不会被加载的影片取代, 两个影片会同时存在, 一个在 0 级别上, 另一个在 1 级别上。当不想让加载进来的影片将主控影片中的导航菜单覆盖时, 可以采用这种级别设置。

loadMovieNum() 函数中的 level 参数对设计课件导航结构很重要, 用户在实际工作中要根据具体的导航实现思路来设置 level 参数, 从而确定被加载的影片是否覆盖原有的影片。

2. loadMovie() 函数

如果要将影片或者图片加载到某一个目标路径, 例如一个影片剪辑中, 那么就要使用 loadMovie() 函数。

loadMovie() 函数使用的一般形式如下:

```
loadMovie(影片或图片, 目标路径);
```

其中影片或图片表示要加载的 SWF 文件或者 JPEG 文件的绝对或相对 URL 地址。该 URL 必须和当前影片处于相同的子域中。如果要使用 loadMovie() 函数, 则所有的 SWF 文件都必须存储在相同的文件夹中, 文件名前面不能有文件名称或磁盘标识符。

目标路径表示目标影片剪辑的路径, 目标影片剪辑将被加载的影片或图像所代替。

下面通过一个小范例讲解一下 loadMovie() 函数的用法。

(1) 提前准备两个外部文件, 一个是 JPEG 文件, 另一个是 SWF 文件。

(2) 新建一个 Flash 影片, 设置“尺寸”为 400×300 像素, 其他文件属性保持默认, 将文件保存在和两个外部文件同一个目录下。

(3) 新建一个名为“空 mc”的影片剪辑元件, 在这个元件的编辑场景中不做任何操作, 这样就创建了一个空影片剪辑元件, 将来会把外部的 SWF 文件或 JPEG 文件加载到这个空影片剪辑中。

(4) 返回主场景, 将“图层 1”重命名为“加载 mc”, 把“库”面板中的“空 mc”影片剪辑元件拖放到场景的左上角。在“属性”面板中定义实例名为 mc, 用来加载外部的图片或者影片, 如图 5-2-13 所示。

(5) 新插入一个图层, 并重命名为 as。单击 as 图层的第 1 帧, 打开“动作”面板, 输入动作脚本如下:

```
loadMovie("car.jpg", mc);  
//导入当前目录下的 car.jpg 到主场景中的空影片剪辑实例 mc 中
```



(6) 测试影片，如图 5-2-14 所示。

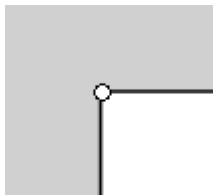


图 5-2-13 空影片剪辑实例



图 5-2-14 从外部导入的图片

(7) 把 as 图层的第 1 帧上的代码改为：

```
loadMovie("test.swf", mc);
//加载当前目录下的影片 test.swf
```

(8) 测试影片，可以看到加载到 mc 中的影片。

因为 loadMovie()函数可以将外部的图片或者影片加载到某个影片剪辑中，所以利用 loadMovie()函数加载的影片或图片可以进行控制。例如，要使从外部导入的图片位于场景的中央，只需在 as 图层的第 1 帧上的代码后加上：

```
mc._x = (400-mc._width)/2;
//图片的 x 坐标等于场景的宽度减去图片的宽度之差除以 2
mc._y = (300-mc._height)/2;
//图片的 y 坐标等于场景的高度减去图片的高度之差除以 2
```

另外，loadMovie()函数加载影片，而 unloadMovie()函数可以卸载由 loadMovie()函数加载的影片。例如：

```
on(release) {
    unloadMovie(mc);
}
```

这段动作脚本的功能是单击并释放按钮后卸载 mc 中的影片。

5.3 网络课件中 Loading 的制作



视频讲解

对于比较大的 Flash 课件，如果想在 Internet 上播放，最好在课件的开始处加入预载画面，等到整个课件都下载完成后再播放。其目的就是使用户避免盲目等待，让课件能流畅播放，这个预载画面就是 Loading。

在 Internet 上观看 Flash 课件时，有时由于影片文件太大或者是网速限制需要装载一段时间才能播放，但由于这个装载时间对于观看者来说是

未知的,所以在 Flash 影片装载过程中如果没有任何提示,多数用户不会有足够的耐心在面对空白的网页许久的情况下仍继续等待。制作 Loading 就是要告诉用户目前 Flash 电影的装载情况,即使制作的 Loading 只是简单的一个小动画,也会起到很好的效果。

本节要介绍两种 Loading 的制作方法,一种是没有下载进度提示的简单 Loading,另一种是显示下载进度的精确 Loading。

课件简介

本范例是一个带 Loading 的中学数学课件,它适合在 Internet 上播放,播放时先出现一个 Loading 画面,提示用户影片加载的进度,如图 5-3-1 所示。



图 5-3-1 课件中的精确 Loading 画面

当影片全部加载以后,影片自动跳转到课件内容画面,展示中学数学“椭圆标准方程推导”的动画演示过程,如图 5-3-2 所示。

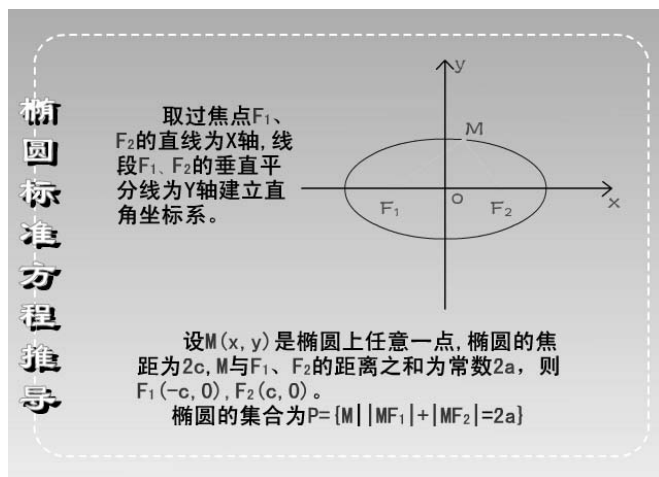


图 5-3-2 课件主动画



知识要点

- ◆ 简单 Loading 的制作方法
- ◆ 精确 Loading 的制作方法



制作步骤

5.3.1 制作课件主场景动画

(1) 新建一个 Flash 文件 (ActionScript 2.0)，设置“尺寸”为 660×480 像素，其他参数保持默认。

(2) 选择“窗口”|“其他面板”|“场景”命令或按 Shift+F2 快捷键，打开“场景”面板，在其中将“场景 1”重命名为“课件”，如图 5-3-3 所示。

(3) 在“课件”场景中创建课件动画。本课件的主要内容是一个演示椭圆标准方程推导过程的动画，此动画效果的制作主要应用传统补间动画，对于具体制作过程这里不再详述，请读者参考课件源文件。



图 5-3-3 场景面板

5.3.2 制作简单 Loading

1. 新建 Loading 场景

(1) 选择“窗口”|“其他面板”|“场景”命令或按 Shift+F2 快捷键，打开“场景”面板，在其中新建一个场景，并重命名为 Loading，如图 5-3-4 所示。

(2) 在一般情况下，预载动画都是放在一个单独的场景中，这样便于编写动作脚本控制影片，也可以使影片结构更加清晰。接下来将在这个 Loading 场景中制作影片预载动画。带 Loading 的 Flash 课件的场景布局一般如图 5-3-5 所示。

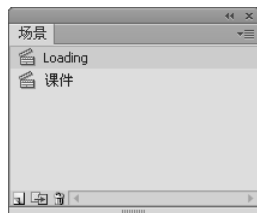


图 5-3-4 新建 Loading 场景

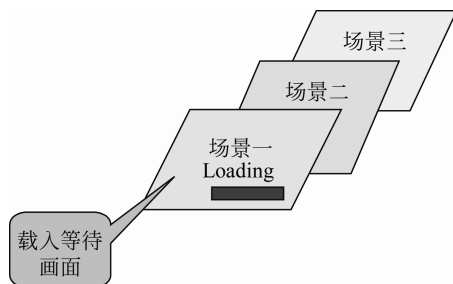


图 5-3-5 带 Loading 的影片场景布局

2. 制作预载画面

(1) 新建一个影片剪辑元件，并命名为“预载”。

(2) 在这个元件的编辑场景中制作下载时的动态画面,画面效果和图层结构分别如图 5-3-6 和图 5-3-7 所示。

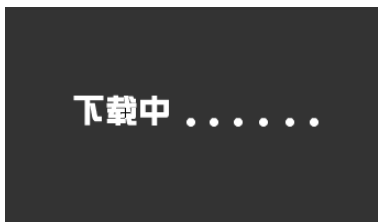


图 5-3-6 “预载”影片剪辑画面



图 5-3-7 “预载”影片剪辑元件的图层结构

3. 布局场景并定义帧动作

(1) 在 Loading 场景中新建一个图层,将两个图层分别重命名为 Loading 和 action,然后在 Loading 图层上将“预载”影片剪辑元件从“库”面板中拖放到场景中。

(2) 在 action 图层的第 2、3 帧分别按 F7 键添加一个空白关键帧。然后单击 action 图层的第 2 帧,定义它的动作脚本如下:

```
if(_framesloaded<_totalframes) {
    gotoAndPlay(2);
} else {
    gotoAndPlay("课件",1);
}
```

这段动作脚本使用了 `_framesloaded` 和 `_totalframes` 两个属性,其中, `_framesloaded` 表示影片文件当前已经加载的帧数, `_totalframes` 表示影片文件包含的总帧数。

在动作脚本中还使用了 `if` 语句,它被称为条件判断语句,是基于某些条件的评估判断来控制脚本流程的。下面详细解释这段脚本的含义。

```
if(_framesloaded<_totalframes){
    gotoAndPlay(2);
//先判断影片加载的帧数是否超过了影片的总帧数,如果没有超过则说明影片没有加载完成
//就要执行 gotoAndPlay(2),也就是继续执行时间轴的第 2 帧
}else{
    gotoAndPlay("课件",1);
//如果超过了,说明影片已经加载完,因此执行 gotoAndPlay("课件",1)
//也就是跳转到“课件”场景的第 1 帧开始播放影片
```

(3) 选择 action 图层的第 3 帧,定义它的动作脚本如下:

```
gotoAndPlay(2);
```

这个帧动作主要是使程序形成一个循环,不断地执行时间轴的第 2 帧来判断影片加载的帧数是否超过了影片的总帧数。

Loading 场景的图层结构如图 5-3-8 所示。



5.3.3 制作精确 Loading

精确 Loading 能显示已装载帧数与总帧数的百分比，
可以让人一目了然，便于用户把握时间。下面在前面制作
的基础上制作一个带精确 Loading 的课件范例。这个课件
和前面制作的带简单 Loading 的课件实例的场景结构一样，也包括“课件”场景和 Loading
场景，同样的步骤这里不再重复叙述，下面重点讲解一下 Loading 场景的制作方法。

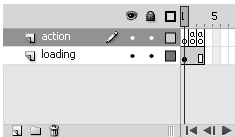


图 5-3-8 Loading 场景的图层结构

1. 制作精确预载画面

(1) 在 Loading 场景中将“图层 1”重命名为“背景”，然后从“库”面板中将事先做好的背景拖放到场景中。

(2) 在“背景”图层上方插入一个图层，并重命名为“文本对象”。然后在这个图层上用“文本工具”创建 4 个动态文本对象和一些辅助说明的静态文本，如图 5-3-9 所示。

专家点拨：在 Flash 文档中使用“文本工具”可以创建 3 种不同类型的传统文本，即静态文本、动态文本和输入文本，在前面几章中创建的文本均是静态文本。动态文本是可以实时显示动态更新的文本，一般在动作脚本中赋予它变化的内容。输入文本则用于在程序运行时由用户输入文本。



图 5-3-9 文本对象

在图 5-3-9 中，从上到下 4 个动态文本框分别用来显示课件文件的总字节数、总 KB 数、加载字节数、进度百分比。这 4 个动态文本框在“属性”面板中的设置如图 5-3-10~图 5-3-13 所示。

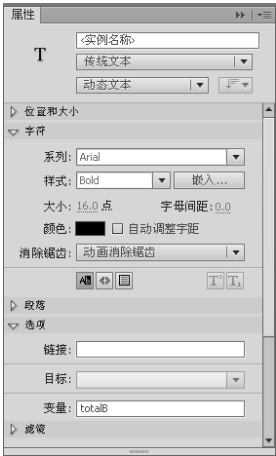


图 5-3-10 “总字节数”动态文本

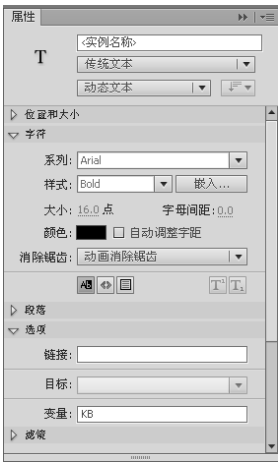


图 5-3-11 “总 KB 数”动态文本

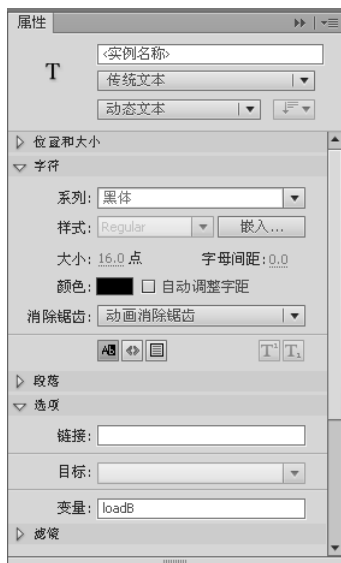


图 5-3-12 “加载字节数”动态文本

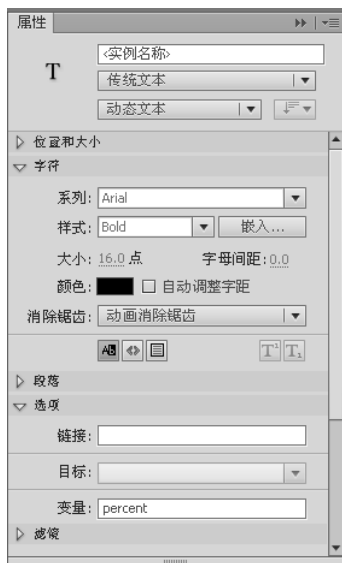


图 5-3-13 “进度百分比”动态文本

(3) 在“文本对象”图层上方新插入一个图层，并将它重命名为“进度条”。然后在该图层中画一个矩形，“笔触颜色”为无、“填充色”为任意合适的颜色，并将其转换为影片剪辑元件，如图 5-3-14 所示。

2. 定义帧动作

(1) 在“进度条”图层上方插入图层，并重命名为 as，然后分别在该图层的第 2、3 帧插入空白关键帧，图层结构如图 5-3-15 所示。

实现预载进度条随着预载内容的增多而逐渐延伸的效果必须应用到影片剪辑元件的一个属性_xscale，即水平缩放值。课件运行时，当预载的百分比为 30%时，应将进度条的（水平缩放）_xscale 属性设置为 30；当预载的百分比为 85%时，应将进度条的_xscale 属性设置为 85%，以此类推，即可实现延伸效果。

(2) 选择进度条影片剪辑实例，在“属性”面板中定义它的“实例名称”为 bar，如图 5-3-16 所示。在动作脚本中通过设定影片剪辑 bar 的_xscale 属性就可以达到预载进度条随着预载内容增多而逐渐延伸的目的。



图 5-3-14 进度条



图 5-3-15 图层结构

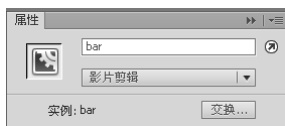


图 5-3-16 设置实例名称



专家点拨：为了在动作脚本中引用实例，就必须要为舞台上的按钮和影片剪辑实例定义实例名称（不能为图形元件实例定义实例名称）。这是一种良好的习惯，因为动作脚本是靠实例名称来标识对象的。

（3）在“动作”面板中分别定义前 3 帧的动作脚本，第 1 帧的动作脚本如下：

```
bar._xscale=0;
//初始化进度条影片剪辑实例 bar 的水平缩放属性值为 0
totalB = this.getBytesTotal();
//将影片文件的字节大小数值存入 totalB 变量中
KB = Int(totalB / 1024);
//把 Byte 转换为 KB 单位显示
```

这段动作脚本主要使用了影片剪辑（MovieClip）对象的 `getBytesTotal()` 方法，它可以获取影片剪辑的字节大小。由于本课件要检测的对象不是特定的影片剪辑而是电影文件，所以将影片剪辑的名称部分写成了 `this`，也可以写成 `_level0` 或者 `_root`。

第 2 帧的动作脚本如下：

```
loadB = this.getBytesLoaded();
//取得目前已经加载的字节数
percent = int((loadB/totalB)*100);
//转换成百分比
bar._xscale = percent;
//设置进度条影片剪辑实例的水平缩放属性值为 percent 变量的值
```

这段动作脚本主要使用了影片剪辑（MovieClip）对象的 `getBytesLoaded()` 方法，它可以获取已经加载的影片剪辑的字节大小。另外，还使用下面的公式计算了加载进度的百分比数值：加载进度的百分比=（目前加载的字节数/总字节数）×100。

第 3 帧的动作脚本如下：

```
if(percent < 100) {
    gotoAndPlay(2);
} else {
    gotoAndPlay("课件", 1);
}
```

这段动作脚本的功能是当影片预载不到 100% 时重复执行第 2 帧，而当影片全部预载完后跳转到“课件”场景的第 1 帧并开始播放。

3. 测试预载画面

在本地计算机上制作 Loading 时，课件不管有多大，装载不需要多少时间，所以 Loading 的动画往往还没来得及显示就进入主动画开始播放了，那么怎么来测试制作 Loading 的效果呢？

可以通过 Flash 中的“显示数据流”的方法来模拟从网络装载页面的情况。具体做法很简单，在动画全部做好后按 `Ctrl+Enter` 快捷键测试效果，在测试画面菜单中选择“视图”

菜单下的“带宽设置”命令，然后选择“视图”|“模拟下载”命令，或再次按 Ctrl+Enter 快捷键就可以看到装载画面了，如图 5-3-17 所示。

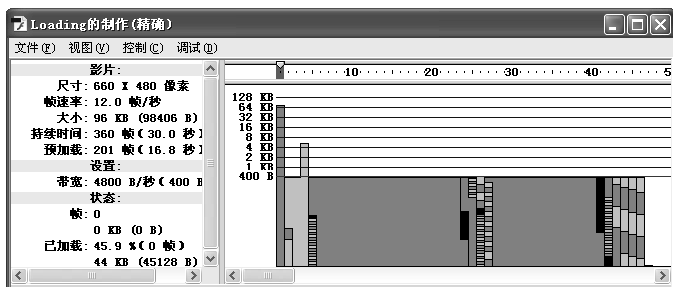


图 5-3-17 模拟下载效果

如果用户仍觉得装载得太快，看不清 Loading 效果，则可以在测试画面菜单中选择“视图”|“下载设置”命令，在其中选择合适的速率，例如 14.4kb/s，也可以自己定义合适的速率，这样就可以对 Loading 在不同的网络模拟速度下进行调试了。

5.4 本章习题

一、填空题

1. 在规划多场景导航课件时主要使用的是_____、_____的程序设计方法，具体设计方法是根据课件的内容将其分解为一个_____和几个_____，前者用来控制和调度各个课件功能模块的播放，后者具体实现相应课件内容的展示。
2. 对于比较大的 Flash 课件，如果想在 Internet 上播放，最好在课件的开始处加入_____，等到整个课件都下载完成后播放。其目的就是使用户避免盲目等待，使课件能流畅播放，这个画面也就是_____。
3. 在制作预载画面时，动作脚本中常常要使用 `_framesloaded` 和 `_totalframes` 两个属性，其中，`_framesloaded` 表示_____，`_totalframes` 表示_____。
4. 在 Flash 文档中使用“文本工具”可以创建 3 种不同类型的传统文本，其中，_____可以实时显示动态更新的文本，一般在动作脚本赋予它变化的内容；_____则用于在程序运行时由用户输入文本。

二、选择题

1. 在 Flash 中实现模块化设计可采用的最好的技术方法是（ ）。
A. 多场景 B. 多图层 C. 多元件 D. 多图片
2. Flash 可以调用外部 SWF 影片文件，在设计时把课件分割为若干个模块后将每个课件模块制作成独立的 Flash 影片，然后利用（ ）函数实现各个课件模块间的相互调用。
A. `gotoAndPlay` B. `loadMovieNum`



图 5-5-2 白蝴蝶之恋

练习3 课件 Loading

针对练习1的课件分别制作一个简单 Loading 和一个精确 Loading。