

# 用Visual Basic进行动画程序设计

钱雪凤(慈溪职高)

动画是一张张静止的图像连续快速播放形成的。制作动画就是根据这个原理先绘制相近但又细微变化的静止图片(即动画帧),然后每隔一段很短的时间就依次显示其中一帧,由于人眼视觉暂留效应,就能看到连贯的动画效果。Visual Basic实现动画的原理也如此:在Visual Basic中利用工具箱中图像控件(Image)和定时器控件(Timer)可以很容易地实现动画效果。我们只要准备好一帧帧动画单幅图像文件(BMP或JPG),把它们依次放入图像控件数组中去,便可用定时器控件的Interval属性来控制每幅图片的显示间隔,在Timer事件过程中,把动画显示的位置确定好,按时间间隔依次从图像控件数组中取出一帧图像进行显示。如此反复交替显示,便可产生动画效果。

VB实现动画的方法主要有三种

一、利用控件的移动:

采用控件的移动技术可实现动画效果。在VB中,控件移动方式可分为两种:一是在程序运行过程中,随时更改控件的位置坐标Left, top属性,使控件出现动态效果;二是对控件

对象调用Move方法,产生移动的效果。这里的控件可以是命令按钮、文本框、图形框、图像框、标签等。例如文字的水平移动,只要在程序中不断的改变Label对象的Left坐标属性,就达到了文字水平移动的效果;如果还想同时有垂直方向的移动,只要同时改变Label对象的top坐标即可实现。而气球的膨胀或缩小,则可用move方法, Move方法的一般格式:对象.move left,top 其中left为对象左边界的水平坐标(x轴),top为对象上边界的垂直坐标(y轴)。文字水平移动程序代码如下:

```
Private Sub Timer1_Timer()
    If Label1.Left <= (Label1.Width) Then
        Label1.Left = Me.Width
    End If
    Label1.Left = Label1.Left - 50
End Sub
```

气球同心缩放程序代码如下:

```
Option Explicit
Dim Isplaying As Boolean '动画是否启动
Dim Width_Org%, Height_Org% '气球原来
```

小

```
Dim Incx%, Incy% '沿x、Y轴的缩放增量
Private Sub Form_Load()
    Isplaying = False
    Image1.ToolTipText = "开始缩放"
    Width_Org% = Image1.Width
    Height_Org% = Image1.Height
    Incx% = 100
    Incy% = 100
End Sub
```

```
Private Sub Image1_Click()
    If Isplaying Then
        Isplaying = False
        Timer1.Enabled = Isplaying
        Image1.ToolTipText = "开始缩放"
    Else
        Isplaying = True
        Timer1.Enabled = Isplaying
        Image1.ToolTipText = "停止缩放"
    End If
End Sub
```

步开发此Agent系统,增强其功能。如图中所示,在第一次迭代中已经实现了Agent 1。第二次迭代实现功能2需要Agent 1提供对系统功能2的支持,所以第二次迭代中需要进一步对Agent 1进行迭代。

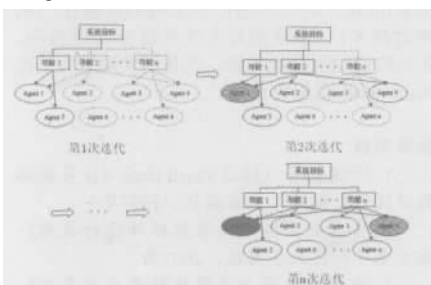


图2 按照系统功能迭代的增量式迭代方式

按照系统功能的迭代开发不需要一次性实现系统所有功能,而是在每次迭代中实现或是增强一些系统功能。这样在每次迭代完成时可以验证此次迭代成功与否,系统总体设计是否合理,提前规避了系统开发风险。如果发现问题,可以及时修改,如果有必要甚至需要重新设计系统框架,这样可以防止到整个多Agent系统都设计实现以后才发现错误,避免了更大的风险。

系统功能与Agent子系统是多对多的关系,因为一项系统功能必然需要多个Agent子系统协助完成,而一个Agent子系统也可以参与多个系统功能。由于整个系统的开发是按照系统功能进行一步步迭代的,所以我们实现某个Agent子系统时只需要让该Agent子系统能够支持此次迭代的系统功能即可,不需要实现对其它功能的支持,这样做可以进一步减小一次迭代中的开发代价,进而可以进一步降低系统开发风险。这样就会出现某些Agent子系统在迭代过程中反复开发,如上图中Agent 1。本文下面部分将对用迭代开发方法开发Agent子系

统的可行性进行分析,并探讨单个Agent进行迭代的特点。

2.2 单个Agent内部结构的迭代

Agent系统与用其它方法所构建的程序的最大区别在于系统的内部结构。通常一个Agent系统主要由两大部件组成,即用于封装的外壳和内部执行的内核。构建Agent系统的模板可以定义为一个二元组: Agent 外壳, Agent 内核, 其中外壳的主要作用是负责控制Agent的运行行为,协调Agent与用户、其它Agent及环境的相互作用,它是一种桥梁,是Agent的外部和社会的能力。Agent内核所反映的是Agent内部和领域问题的处理能力。Agent 外壳与内核的组合是可选的,用户可根据实际要求自由组装。这符合软件工程中松散耦合的思想。这种结构使单个的Agent可以独立地开发,然后加到系统中。Agent子系统也可以采用迭代式的方法。

迭代开发Agent子系统建议采用演进式迭代策略。每个Agent都有自己的特点,不同Agent应该有自己的合理的内部结构,而且系统整体行为只有在运行时才能体现。演进式迭代策略正是通过多次迭代的探索,逐步演化建立起系统的架构的。

在车间管控多Agent系统中, AgentX是控制X型号自动化设备的Agent。下面表中显示了车间管控多Agent系统的迭代过程中, AgentX的演进过程。

表1 车间管控多Agent系统的迭代过程中 AgentX的演进

系统功能	第一次系统迭代	第二次系统迭代	第三次系统迭代	第四次系统迭代
AgentX子系统的功能	系统功能管理 设备号初始化 设备号维护 设备号删除	系统功能管理 设备号初始化 设备号维护 设备号删除	系统功能管理 设备号初始化 设备号维护 设备号删除	系统功能管理 设备号初始化 设备号维护 设备号删除
AgentX子系统的接口	设备号初始化 设备号维护 设备号删除 设备号查询	设备号初始化 设备号维护 设备号删除 设备号查询	设备号初始化 设备号维护 设备号删除 设备号查询	设备号初始化 设备号维护 设备号删除 设备号查询

在整个系统的迭代过程中,每次AgentX的迭代都需要对AgentX的内核进行迭代,到系统迭代完毕才能最终确定AgentX的内部结构。

3 结语

本文针对多Agent系统开发难度大、风险高的问题,分析了多Agent系统的特点和迭代式开发方法可以提前规避系统开发风险的长处。提出将迭代开发方法的思想引入多Agent系统的开发中,并指出了基于系统功能的迭代和单个Agent子系统内部迭代两种迭代方式。多Agent系统一般用于解决复杂的分布式问题,要在实际项目中运用迭代方法开发多Agent系统需要根据具体情况灵活运用,迭代化开发不是一种高深的软件工程理论,它所提供的是控制项目风险的非常有效的机制,一种复杂问题分解成小问题的思想。

参考文献:

1. Michael Wooldridge, 《An Introduction to Multi Agent Systems》, 2002
2. IBM, 迭代化软件开发技术IBM Rational 技术白皮书(版本 1.0), 2004
3. 刘大有、杨鲲、陈建中, Agent研究现状与发展趋势, 2000
4. Jennings NRM. Wooldridge, agent oriented software engineering [A]. in: J Bradshaw (Ed.). Handbook of agent Technology[C]. 2000
5. Wooldridge, Agent oriented software engineering[J]. IEE Proc. Software engineering, 1997
6. 饶运清、刘世平、张洁、李培根, 基于MAS的车间管控系统中通用Agent建模与设计, 2002
7. 刘韶涛、王宁生, 软件体系结构的迭代增量勾践方法, 2002
8. 曹军海、张和明、熊光楞, AOSE中Agent行为的形式化定义与描述方法, 2004
9. 程明、程泽、刘豹, Agent-Based 软件工程方法, 2002
10. 陈建中, 刘大有, 唐海鹰. 智能Agent建模的一种模板结构[J]. 计算机研究与发展, 1999

```
Private Sub Timer1_Timer() '定义Timer 事件
'Image 对象调用move方法, 使图像同心
缩放
Image1.Move Image1.Left-Incx%/2,Image1.
Top-Incy%/2
Image1.Width=Image1.Width+Incx%
Image1.Height=Image1.Height+Incy%
'放大到碰到边界,则缩小
If Image1.Left+Image1.Width>=Form1.Width
Or
Image1.Top+Image1.Height>=Form1.Height
Or
Image1.Left<=0 Or Image1.Top<=0 Then
Incx%=- Incx%
Incy%=- Incy%
End If
'缩小到原大,则膨胀
If Image1.Width<=Width_Org% Or Image1.
Height<=Height_Org% Then
Incx%=- Incx%
Incy%=- Incy%
End If
End Sub
```

## 二、利用动画控件

VB的工具箱中专门提供了一个动画控件 (Desaware Animated Button Control) 进行动画设计, 该工具在 windows\system 子目录下以 Anibtn32.ocx 文件存放, 用时可加入项目文件中, 这种方法实现动画的过程与电影胶片的放映极为相似, 它将多幅图像装入内存, 并赋予序号通过定时或鼠标操作进行图像的切换, 通过这种方法可实现相对符号的动画。

## 三、利用图片剪贴控件Pictureclip

该控件提供了在一个控件上存储多幅图片的技术, 访问时不是依次读入多幅图片, 而是先将多幅图片放置在一个控件中, 然后在程序设计时利用选择控件中的区域, 将图片动态地剪切下来放置于图片框进行显示, 程序控件每隔一定时间剪切并显示一幅图片, 这样便可以产生动画效果。下面我们利用Pictureclip和Timer 控件来制作动画效果。先做准备工作: 绘制好动画的每帧画面 (注意每帧画面要一样大), 在图形处理软件里将它们按动画次序拼成一张大图, 如共有8帧图片, 可把它们排成2行4列。

然后在VB中, 右击工具箱, 点击“部件”, 在出现的对话框中选“Microsoft Pic-

tureclip Control6.0", 导入Pictureclip 控件。设计窗体: 在窗体中加入一个Timer控件 (名称为Timer1)、一个Pictureclip控件(名称Picture1) 和一个PictureBox(名称Picture1)控件。接下来设置各属性: 1、将Timer1的Enabled属性设为False (Timer对象的Enabled属性决定了Timer事件是否有效。设置Enabled属性为True将启动Timer事件; 为False则将使Timer事件失效。这里先设为False); Timer1.Interval属性设为56 (时间间隔属性, 决定Timer事件发生的频率。而机器系统的定时器时间间隔为1/18秒, 即56毫秒)。2、将Picture1的Picture属性设为预先准备好的图片 (注意仅能装入.Bmp文件); Picture1的Rows.Cols属性分别设为2和4 (Rows.Cols是根据拼成的大图设定控件的总的行列数, 这里为2行4列); 3、将Picture1的Appearance属性设为0 (Flat, 平面), Picture1.BackColor (背景色) 属性设为白色, Picture1.BorderStyle属性设为0(None, 无边框)。

接下来进行编程:

```
Dim Number As Integer '当前的图编号
Private Sub Form_Activate()
Label1.Visible=True
Number=0 '动画从第一帧开始
Timer1.Enabled=True '允许timer事件发生
End Sub
Private Sub Timer1_Timer()
Number=Number+1
If Number>7 Then
Number=0 '动画如到最后一帧, 则从头开始
End If
Picture1.Picture=Picture1.GraphicCell
(Number)
End Sub
```

注意这儿用的GraphicCell(N): 是一个一维数组, 用于访问picture属性装入图像中的第几个图像元素, 次序为从左到右, 从上而下。GraphicCell(rows\*cols-1)表示最后一帧图。

## 四、利用LoadPicture 函数

在Image控件或Picture控件的Picture属性中, 调用loadpicture函数来装载不同的图像, 将在对象中显示不同的图像, 也可以实现图像变化; 如翻书动画。这是一种典型的无位移动画, 实现的方法是, 设置好Image对象和Timer对象后, 在Timer事件过程中调用LoadPicture函

数来装载不同的图像, 并赋给Image对象的Picture属性, 使对象中显示不同的图像, 就实现了图像的变化。翻书动画程序如下:

```
Option Explicit
Dim Number% '当前的位图编号
Dim Isplaying As Boolean '动画是否启动
Dim aa%='1'翻页方向
Private Sub Form_Load()
Isplaying=False
Image1.ToolTipText="开始翻页" '鼠标移到书上时,出现提示
Number%=1
End Sub
Private Sub Image1_Click()
If Isplaying Then
Isplaying=False '动画已启动,则停止
Timer1.Enabled=Isplaying
Image1.ToolTipText="开始翻页"
Else
Isplaying=True
Timer1.Enabled=Isplaying
Image1.ToolTipText="停止翻页"
End If
End Sub
Private Sub Timer1_Timer()
Number%=Number%+Aa
If Number%>=4 or Number%<1 Then
Aa=- Aa
Image1.Picture=LoadPicture (App.Path &
"\Book"&Number%&"& ".Bmp")
'在Image对象中的Picture属性中,按时间间隔分别装载入图片帧,实现动画
End Sub
```

上面我们只谈了VB实现图形动画的基本方法。用VB来进行动画编程, 还可以实现许多动画效果, 比如我们可以利用Windows API 函数BitBlt, 对图形进行一系列的位操作 (从映射技术), 来使图片上某些部分实现透明。还可以做动画游戏等等。让我们共同努力, 让Visual Basic世界更美好。

## 参考书目:

1. 徐瑞奎著《精通Visual Basic 4.0 多媒体程序设计》机械工业出版社, 1997年
2. 陈启安主编《多媒体软件设计技术》西安电子科技大学出版, 2001年
3. 林东等编著《多媒体技术应用基础》电子科技大学出版社, 1998年

(上接第21页) 压安全防护都作了一些规定, 人们可以分析研究一下这些规定可否满足智能化建筑的要求, 满足不了时, 可进行修改或补充, 特别是对低压配电设计规范(GB50054-95)进行补充或修改比较合适。在进行修改时, 应注意到现在有些连接件或电子元器件本身已经有了能防止15KV过电压冲击的功能, 电压防护等级已经很高, 但能量吸收方面能否满足要求, 还要和有关配电设计规范进行协调。

接地系统设计一直是困扰着计算机控制与网络系统以及通信系统的一个较重要的问题, 这方面由于专业不同, 看问题的角度就不同, 认识也就很难统一。所以在工业与民用电力装置的设计规范(GBJ65-83)制定时, 没有明确计算机控制系统与通信系统的接地要求, 严格讲计算机控制与网络系统, 以及通信系统接地设计也属电气接地设计范畴, 在安全接地方面也必须遵守国家规定的工业与民用电力装置的

接地设计规范(GBJ65-83)。当前在计算机控制或网络系统, 以及通信系统的接地上还有一定分歧, 其他线除保护接地外, 还分为系统地、现场地、逻辑地、数字地、模拟地、信号地等, 这么多地线, 有无重复, 要由该专业方面的专家去认真分析研究, 取得统一之后, 才能对其如何接地作出合理的规定。接地方式有的提出要单独接地, 但国家有关规范提出集中一点联合接地, 这就发生了矛盾, 接地电阻有的提出要求小于1, 有的要求小于2, 国家有关规范规定要求小于4, 这里也发生了矛盾, 工作实践中曾遇到过有些计算机控制系统要求接地电阻小于0.5。国外的计算机控制系统的接地电阻要求各个厂家也不统一, 有的要求2, 有的要求4, 还有的要求10, 但没有遇到过小于1的要求。分析其原因主要与计算机控制系统本身的硬件设计, 产品加工制造质量有关, 硬件设计越先进, 产品加工

质量越高, 对接地电阻的要求就不那么小。

## 5、结束语

综上所述, 随着现代科技的迅速发展, 人们对智能化建筑的期望值也会增高, 有可能导致对建筑布局, 从形式到内容的变化。为使建筑物更适合用户需要, 建筑师在构思建筑方案之初, 就要考虑建筑智能化的问题, 智能化设计人员也要在方案阶段参与建筑设计, 只有建筑师与智能化设计人员的密切配合, 才能使建筑与智能化系统、设备设计更完善、更合理、以使用户满意。

## 参考文献:

1. 杨绍胤: 《智能建筑实用技术》, 机械工业出版社2003-02
2. 李石兵, 王天林: 《智能建筑一体化集成系统中LonWorks OPC服务器的开发》, 《智能建筑》2005年第8期