小球碰撞检测的算法设计和台球游戏开发

韩绍兵[1] 朱元忠[1,2]
(1.厦门大学,福建 361005;
2.北京工业职业技术学院,北京 100042)

摘 要:从台球游戏的设计实现出发,分析了小球碰撞的物理过程,给出了基于一定时间间隔的碰撞算法设计,对小球在台球桌面上的滚动与碰撞过程进行了较为真实地模拟,并在程序设计上介绍了游戏过程中的声音与动画的实现方法。

关键词:碰撞检测;算法;游戏
中图分类号:TP311.52 **文献标识码**:B **文章编号**:1671-6558(2004)04-57-06

Collision Dectecion and Response Algorithm for Billiards Game Delelopment

Han Shaobing[1] Zhu Yuanzhong[1,2]
(1.Department of Computer Science, Xiamen University, Xiamen 361005, China;
2.Beijing Vocational & Technical Institute of Industry, Beijing 100042, China)

Abstract: This paper discusses the collision detection and response algorithm for developing billiards game. The algorithm can detect collisions in a fairly fast and precise way. Also the paper describes some way to handle multimedia and spirit animation in a computer game.

Key words: collision detection; algorithm; game

0 引言

游戏程序的设计和开发是当前程序设计的一个热点,台球游戏程序综合运用了物理运动分析、碰撞算法设计和计算机多媒体技术,其可以作为多媒体软件设计教学的一个范例,能较好地启迪学生的思维并锻炼其软件设计的实践能力。本文在碰撞检测(预测)算法的基础上给出了一个台球游戏的初步实现。

1 小球碰撞的运动分析

两球在碰撞前的相对速度不沿两球球心连线的碰撞叫"斜碰",又称"非对心碰撞"。台球的碰撞运动属于典型的斜碰问题。斜碰也可分为弹性碰撞和非弹性碰撞两类,具体到台球的碰撞,可近似地处理为大小和质量均相同的两个小球的刚性碰撞,即弹性碰撞。

对于斜碰的分析,采用正交分解法较为方便。图1为两小球碰撞瞬间的示意图,两球的球心连线为nn′该线倾角为α,过碰撞点的切线tt′,两球的运动速度分别为v_{1x},v_{1y}和v_{2x},v_{2y}。

倾角α的正、余弦为:

$$sin\alpha = \frac{y_1 - y_2}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}},$$
$$cos\alpha = \frac{x_1 - x_2}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \quad (1)$$

收稿日期:2004-09-20
作者简介:韩绍兵(1974-),男,湖北汽车工业学院,机械制造专业毕业,厦门大学在读硕士研究生,讲师。

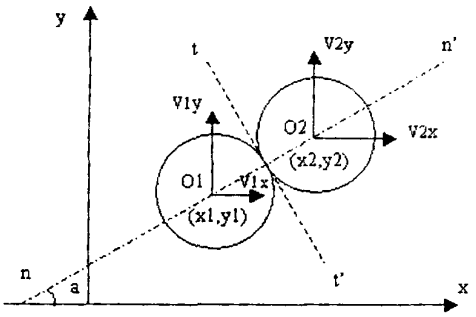

图1 小球斜碰示意图

根据弹性碰撞的运动规律(可由动能和动量守恒定律分析可得,此处从略),两球碰撞后球心连接线方向的速度将相互交换,而碰撞点切线方向的速度保持不变。

碰撞前,两球的速度为:

$$\begin{cases} v_{1n} = v_{1x}\cos\alpha + v_{1y}\sin\alpha \\ v_{1t} = -v_{1x}\sin\alpha + v_{1y}\cos\alpha \end{cases},$$

$$\begin{cases} v_{2n} = v_{2x}\cos\alpha + v_{2y}\sin\alpha \\ v_{2t} = v_{2x}\sin\alpha + v_{2y}\cos\alpha \end{cases} \tag{2}$$

碰撞后两球 nn′方向的速度相互交换,速度变为:

$$\begin{cases} v'_{1n} = v_{2n} \\ v'_{1t} = v_{1t} \end{cases}, \begin{cases} v'_{2n} = v_{1n} \\ v'_{2t} = v_{2t} \end{cases} \tag{3}$$

再将 nn′,tt′方向的速度分别向 x,y 方向投影:

$$\begin{cases} v'_{1x} = v'_{1n}\cos\alpha - v'_{1t}\sin\alpha \\ v'_{1y} = v'_{1n}\sin\alpha + v'_{1t}\cos\alpha \end{cases},$$

$$\begin{cases} v'_{2x} = v'_{2n}\cos\alpha - v'_{2t}\sin\alpha \\ v'_{2y} = v'_{2n}\sin\alpha + v'_{2t}\cos\alpha \end{cases} \tag{4}$$

将式(1)、(2)、(3)代入式(4),经计算,可知两球 x,y 方向的碰后速度为:

$$\begin{cases} v'_{1x} = v_{1x} - \triangle v_x \\ v'_{1y} = v_{1y} - \triangle v_y \end{cases}, \begin{cases} v'_{2x} = v_{2x} + \triangle v_x \\ v'_{2y} = v_{2y} + \triangle v_y \end{cases} \tag{5}$$

上式中:

$$\triangle v_x = \frac{(v_{1x} - v_{2x})(x_1 - x_2)^2 + (v_{1y} - v_{2y})(x_1 - x_2)(y_1 - y_2)}{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

$$\triangle v_y = \frac{(v_{1y} - v_{2y})(y_1 - y_2)^2 + (v_{1x} - v_{2x})(x_1 - x_2) + (y_1 - y_2)}{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2 小球碰撞检测(预测)算法分析

所谓碰撞检测,就是对产生碰撞的对象和碰撞所发生的时刻进行确定。由上面的分析可知,当碰撞发生后,小球的运动速度将随之改变,因此在程序设计时,必须预先计算出碰撞所发生的时刻,并在该时刻后将小球的速度赋上新值。

2.1 小球的碰撞时刻

Windows 程序设计中,有关时间问题的处理一般采用 Timer 控件实现,该控制的功能是每经过一个指定的时间间隔,就会自动触发先前由用户定义好的程序,这一在时间间隔结束之时被执行的程序称为事件处理程序。

但是,实际中小球的碰撞并非一定发生在每个时间间隔的开始或是结束之时,而可能是发生在该时间间隔内的某个时刻,也就是说,在中间某个时刻发生了碰撞,小球在该时刻前后的运动速度是不同的。因此,为得到准确的运动轨迹,必须预先计算出小球发生碰撞的时刻,并在随后的事件处理程序中,对小球的运动分为前后两个时间段来进行处理。

2.2 单位时间间隔内两球碰撞时刻的计算

下面,对小球发生碰撞的时刻进行求解。为方便计算,定义时间间隔长度为 1,即 1 个时间单位,每个单位的具体时间长度在程序中另行指定。

在某时间间隔的开始时刻,两小球的状态如图 2 所示。o_1o_2 为两球的球心连线,r 为球半径,v_x,v_y 为球 1 相对于球 2 的运动速度,o_1l 为球 1 相于球 2 的运动轨迹。以球 2 的球心 o_2 为中心,2r 为半径作圆,见图中虚线所示。若 o_1l 与虚线圆相交,则说明两球会发生碰撞,因为球 1 运动到交点 A、B 之间时,两球的球心距离小于两球的半径和 2r。显然,碰撞发生的位置为 A 点,B 点发生的碰撞无意义。记碰撞发生的时刻为 t。

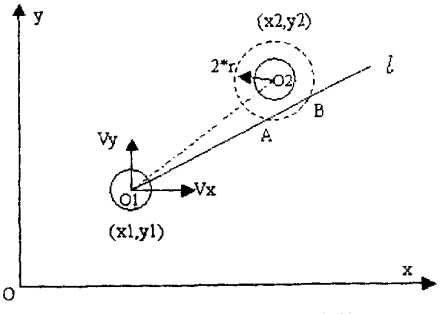

图2 小球碰撞时刻计算

球 1 对球 2 的相对速度为:

$$\begin{cases} v_x = v_{1x} - v_{2x} \\ v_y = v_{1y} - v_{2y} \end{cases} \tag{6}$$

根据 $\vec{O_1A} + \vec{AO_2} = \vec{O_1O_2}$，在 x，y 方向上分别有：

$$v_x \cdot t + (x_2 - x_A) = x_2 - x_1 \quad (7)$$

$$v_y \cdot t + (y_2 - y_A) = y_2 - y_1 \quad (8)$$

联立(6)、(7)、(8)三式，得：

$$A \cdot t^2 + B \cdot t + C = 0 \quad (9)$$

上式中：

$$A = (v_{1x} - v_{2x})^2 + (v_{1y} - v_{2y})^2 \quad (A > 0)$$

$$B = 2 \cdot [(x_1 - x_2)(v_{1x} - v_{2x}) + (y_1 - y_2)(v_{1y} - v_{2y})]$$

$$C = [(x_1 - x_2)^2 + (y_1 - y_2)^2] - 4 \cdot r^2 \quad (C \geq 0)$$

若发生碰撞，则必有 $A > 0$（$A = 0$ 意味着两球相对静止，碰撞不可能发生），$C \geq 0$（$C < 0$ 意味初始时刻两球重叠，现实情况不存在。显然，在 $A > 0, C \geq 0$ 的情况下，根据式(9)，发生碰撞时须要有 $B < 0$。

因此，在 $A > 0, C \geq 0, B < 0$ 和 $B^2 - 4AC \geq 0$ 成立时，可求得碰撞时刻为：

$$t = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad (10)$$

根据前面所述，碰撞时刻发生在单位时间内，因此若 $t < 0$ 或 $t > 1$，则意味着本时间间隔内碰撞不会发生。当 $C = 0$ 时，意味着初始时刻两球刚好处于接触的状态，即表示若有碰撞发生，其时刻为 $t = 0$

3 小球运动动画的实现

要体现台球游戏的真实性，必须实现小球在桌面上的滚动效果。分析小球滚动的物理过程，其包含两个方面的运动，一是小球在桌面上的移动，另一是小球的转动。小球的滚动是通过动画实现的，下面介绍在本游戏设计中所采用的动画技术。

3.1 精灵(Spirit)动画的实现技术

精灵，简单的说是一个可以在背景图片上四处移动的不规则图形，先将精灵图形画在背景图片上，然后将所画的上一个精灵从背景上抹去，再将精灵图画在背景的另一个地方，依次类推，就可使精灵在背景上动了起来，从视觉上就形成了动画。精灵动画在多媒体及游戏软件中有着广泛的应用，其可以降低系统的开销、减少屏幕的闪烁感和防止背景图片被破坏，达到高效、流畅、实用的效果。

但是，计算机处理的图片一般是规则的矩形区域，对于不规则的精灵图形该如何进行处理呢？这里需要用到图像合成技术。图 3、图 4 分别是动画中的前景和背景图片。在实现精灵动画前，需先将前景图进行处理得到两幅图：一副是“精灵图”，其是把前景图中不需要在背景中显示的部分，即要求透明的部分用黑色（黑色的像素值为 0）填充而成，如图 5 所示；另一副是“精灵遮罩图”，是把前景中需要显示的部分用黑色填充，而不需要显示的部分用白色（白色的像素值为 1）填充而成，如图 6 所示。

图 3　前景

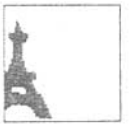

图 4　背景

图 5　精灵图

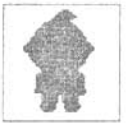

图 6　精灵遮罩图

要正确地把精灵图形显示在背景上，需要通过两步来实现。第一步，将“精灵遮罩图”按逻辑“与”的方法画在背景图上，即两图的像素值按逻辑“与”的方式进行合成，这相当于在背景上挖了一个与精灵外形相等的洞，如图 7 所示。第二步，是将第一步所合成的图片再与“精灵图”按逻辑“或”的方式进行合成，这样就正确的将精灵图形画在了背景上，如图 7 所示。

图 7　背景挖洞图

图 8　正确的合成图

3.2 小球滚动的实现

先前分析过，小球的滚动包含移动和自转两部分，要实现小球的滚动，只需在精灵动画的基础上加上小球的自转便能实现，具体的做法就是每隔一定的时间间隔，变换“精灵图”即可，图 9 是由不同方位

的小球所做成“精灵图”。

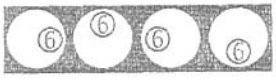

图9 不同方位的小球

4 游戏功能的实现与程序的设计

游戏要实现的功能主要有击球、球的滚动、球与球的碰撞、球与边的碰撞、球落袋以及发声等等。下面通过简单易用的 Visual Basic 6.0 来编制程序实现上述相关功能。

4.1 球的类型及变量定义

程序中,对小球运动的处理涉及到球的位置、速度、方位以及转动状况等等。为了便于描述,定义了球的类型和变量如下:

```
Private Type BallType'定义“球”的数据类型
    Newx As Integer'球中心新的位置:X、Y 坐标
    Newy As Integer
    Oldx As Integer'球中心原来位置:X、Y 坐标
    Oldy As Integer
    Vx As Single'球的速度:X、Y 方向分量
    Vy As Single
    CelNo As Integer'球的转动方位
    MoveFlag As Boolean'球转动的标志
End Type
.....
Dim BALL(0 To BallNum)As BallType'定义球的变量
```

4.2 主要事件处理程序

4.2.1 击球事件

台球游戏中击球的作用是使白球获得一定的速度。本例中,可通过一个按钮事件进行处理。

```
Private Sub Start-Click()
    Timer1.Enabled = True'启动时间控件事件
    BALL(0).Vx = 10'设置白球的速度,本例中只是简单地赋值,实际情况要复杂许多
    BALL(0).Vy = 15'
    BALL(0).MoveFlag = True'设置白球的转动标志
End Sub
```

4.2.2 时间控件事件

游戏中需要不停地处理球的各种运动,包括滚动、碰撞、减速、落袋等等。本例中,对小球的运动处理,均在一个 Timer 控件事件中来完成,相关程序代码如下:

```
'时间事件,触发球的“滚动”(平移和转动)
Private Sub Timer1-Timer()
    Dim i As Integer
    For i = 0 To BallNum
      '若球的移动标志为“真”,则开始运动
      If BALL(i).MoveFlag Then
        '更新球自转时的方位
        BALL(i).CelNo = (BALL(i).CelNo + 1) Mod 4
        MoveBall i'球运动到下一个方位
        SpeedDown i'减速处理
    End If
    Next i
End Sub
```

基于模块化设计的思路,本例中对球的运动处理均放在过程 MoveBall 中,包括球滚动时位置的变化,碰撞的处理等等,详见下面程序片断:

```
Private Sub MoveBall(ByVal i As Long)
    Dim j As Integer
    Dim IsCollide As Boolean
    ClearBall i'从前一位置抹去球,恢复背景
    '球运动到新的位置
    BALL(i).Newx = BALL(i).01dX + BALL(i).Vx
    BALL(i).Newy = BALL(i).01dY + BALL(i).Vy
    For j = 0 To BallNum'球之间的碰撞处理,含有球速与位置的修正
      If i <> j Then
        IsCollide = CollideBall(i,j)
      End If
    Next j
    Collide Table i'球边之间的碰撞,含有球速与位置的修正
    PutBall i'在新的位置将球画在背景上
    '完成球的位置更新
    BALL(i).01dX = BALL(i).NewX
    BALL(i).01dY = BALL(i).NewY
    Table.Refresh
End Sub
```

因碰撞处理程序篇幅较长,读者可根据前面的算法分析来完成程序的编制,笔者不再赘述。

4.3 有关技术处理

4.3.1 资源文件的处理

因程序中要用到较多的图片和声音文件，直接嵌入一个个 Picturebox 控件（或 Image 控件），或是在程序程序运动中装载图片、声音文件，会使程序的开发和运行效率大大降低。

可以通过引入资源文件，将程序中用到的图片、声音全部装入其中，然后在窗体的 Load 事件中进行集中加载，有关资源文件见图 10 所示。对资源的加载这里用到了控件数组的动态扩充功能，相关程序代码如下：

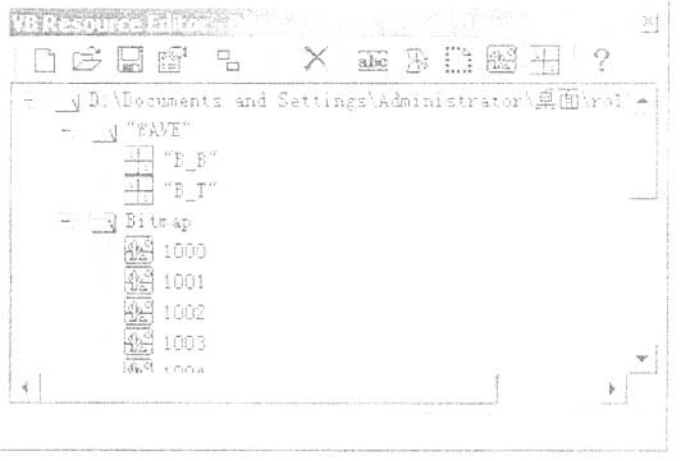

图 10　资源文件的使用

```
Private Sub From-Load()
Dim i As Integer
For i=1 To 15'控件数组的动态扩充
  Load ballpic(i)
  Load BackPic(i)
Next i
For i=0 To 15'从资源文件中加载图片
  ballpic(i).Picture = Load ResPicture(1000 + i,vbResBitmap)
  ballpic(i).Visible = False
  BackPic(i).Visible = False
Next i
……
End Sub
```

4.3.2 位块传送实现图像合成

程序在运行时对系统的资源的占用主要消耗在动画的实现上，本例中通过直接采用 Windows GDI 的 API 函数 BitBlt 进行图像合成，提高了运行速度，下面是在背景上放置球的程序代码。

```
Private Sub PutBall(ByVal i As Long)'将球放置在新的位置
'将"精灵遮罩图"以"与"方式画在背景上
BitBlt Table.hDC, BALL(i).NewX - BallRadius,BALL(i).NewY - BallRadius,
    CelWidth, CelHeight, MaskPic, hDC, 0, 0, SRCAND
'将"精灵图"以"或"方式画在背景上
BitBlt Table.hDC, BALL(i).NewX - BallRadius,BALL(i).NewY - BallRadius,
    CelWidth,CelHeight,ballpic(i).hDC,BALL(i).CelNo * CelWidth,0,
    SRCPAINT
End Sub
```

4.3.3 游戏发声程序

游戏中的声音来自球之间的碰撞、球与边的碰撞、击球以及球落袋等。本例中对声音的处理是通过调用 Windows API 函数来播放资源中的 WAVE 文件来实现的，具体代码如下：

```
Private Sub PlayWave(ByVal sSoundName As String)
Dim 1Ret As Long
'调用 Windows API 函数
1Ret = PlaySound(sSoundName, App.hInstance,SND-RESOURCE + SND-ASYNC + SND-NODEFAULT)
End Sub
```

注意，上面程序代码在 Visual Basic 的调试运行环境内并不会发出声音，需要程序编译为可执行文

件后才有。

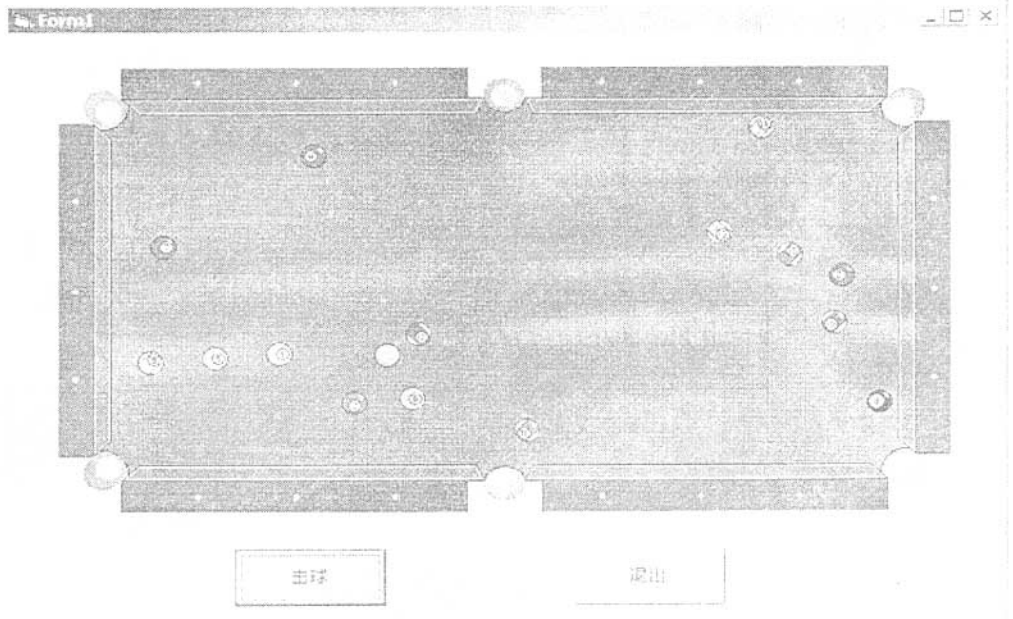

图 11 台球游戏界面

5 结束语

本文提出了基于一定时间间隔的小球碰撞检测算法，解决了当多个小球同时运动时，碰撞时刻难于准确计算的问题。其对于碰撞的检测、以及碰撞事件的响应处理都具有很好的参考意义，可以作为多媒体软件和游戏软件开发的学习范例。图 11 是本例实现的台球游戏界面，继续对本例程序在击球、落袋、积分等方面加以完善，就可以开发出一个功能完善的台球游戏程序。

参考文献：

[1]陈启安，刘志镜. 多媒体软件设计技术[M]. 西安：西安电子科技大学出版社，1999

[2]抖书书屋. Visual Basic 6.0 多媒体实用编程技术[M]. 北京：中国水利水电出版社，1999

[3]David Jung，等. 前导工作室译. Visual Basic 6.0 开发人员参考手册[Z]. 北京：机械工业出版社，2000

（责任编辑：冯玉山）