doi:10.16104/j.issn.1673-1891.2018.04.019

# 基于OpenGL的交互式科普平台设计研究

程 飞
(安徽电子信息职业技术学院电子工程系,安徽 蚌埠 233000)

**摘要**:分析了当前科普平台的特点,讨论了交互式科普实验平台的关键技术.结合教育心理学和软件工程的原理,以外摆线和等速降线为例,给出了交互式科普实验平台开发实例。

**关键词**:科普;实验平台
**中图分类号**:TP311.1 **文献标志码**:A **文章编号**:1673-1891(2018)04-0078-04

# Research on the Design of Interactive Popular Science Platform Based on OpenGL

CHENG Fei
(Department of Electronic Engineering, Anhui Vocational College of Electronics & Information Technology, Bengbu, Anhui 233000, China)

**Abstract**: The characteristics of current science popularization platform are analyzed, and the key technologies of the interactive science popularization experimental platform are discussed. Based on the principles of educational psychology and software engineering, an example of interactive experimental platform for science popularization(for example, the outer cycloid and the isokinetic line) is given.

**Keywords**: science popularization;experiment platform

## 1 问题的提出

当前,科技类场馆开展科普工作主要通过三种手段:(1)实物模拟,(2)采用多媒体技术,利用声光电营造场馆环境,(3)微视频展示科学原理,(4)利用虚拟现实技术。对于第一类,观众一般不能触摸实物模型,第二类技术对于传播科技知识作用有限,而对于第三类,观众一般情况下仅能通过视频了解科学原理和科技知识,被动接受得多,主动参与得少。第四类则依赖于硬件设备(如VR头盔等)。总体看来,科普平台的展示功能较多,而具有交互功能的科普平台较为少见。如果结合计算机图形学技术,开发具有交互功能的科普虚拟平台,可以有效提高科普效果,促进科学知识的传播。本文结合中国科协科普大赛的命题《轮旋线》为例,结合软件工程理论,详述交互式科普平台的设计方法。

## 2 轮旋线原理和特性

一个动圆(圆心为$O$,半径为$r$)在一条定曲线$L$上滚动时,与动圆圆心$O$距离为定值c的点$P$的轨迹称为轮旋线[1],又称摆线。$OP$称为摆轴,且$c=|OP|$。一般L为直线或者圆。如果$L$为直线,称为普通摆线(如图1所示)。如果L为定圆,分为两种情况,称为内圆摆线(如图2所示)和外圆摆线(如图3所示)。若$c>r$,称为长轴摆线,若$c<r$,称为短轴摆线。图4从左至右显示了$c>r$、$c=r$、$c<r$三种内圆摆线的形成过程。

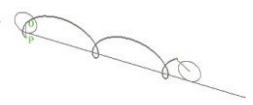

图1 普通摆线

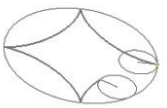

图2 内圆摆线

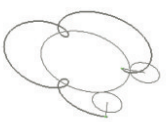

图3 外圆摆线

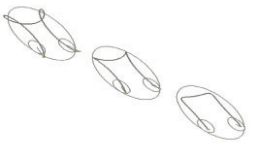

图4 三种内圆摆线

## 3 虚拟平台总体设计

### 3.1 需求分析

从交互式科普角度考虑,虚拟平台应能够动

**收稿日期**:2018-07-17
**作者简介**:程飞(1970— ),男,江苏江都人,副教授,硕士,研究方向:CAD/计算机仿真。

态、交互表达:(1)摆线的形成过程(2)摆线的数学表达式(3)摆线的特殊性质(4)摆线的简史(5)摆线的应用(6)摆线的绘制。由于科普平台的作用是激发科学研究的兴趣,对于公式的推导和理论一般不做展开。

### 3.2 场景设计

基于教育心理学的理论,人类视觉对于运动的物体比静止的物体敏感;此外,在立体的场景中多视图观察形体,所获得的信息量远大于固定视角所获得的信息量。因而,将摆线的动态形成过程置于三维场景中,场景的视角可以变换,其效果如图5所示。这里视角矢量为(1,1,1),即轴侧观察。该图显示了各种摆线的形成过程。

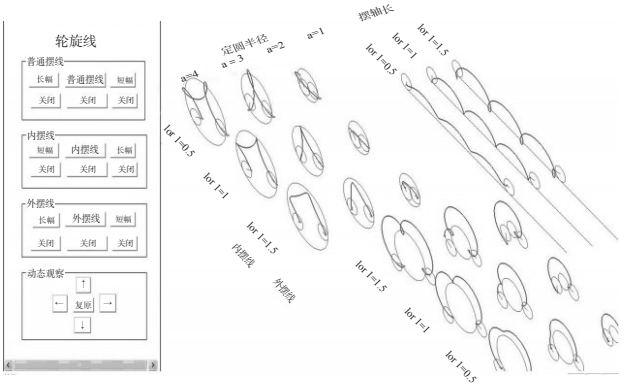


图5　各种摆线的形成过程

### 3.3 开发引擎选择

这里采用底层引擎OpenGL作为开发工具[2]。这是因为:(1)OpenGL具有良好的浮点计算能力,特别适用于数学函数处理;(2)其动态刷新功能可以保证动画效果的实现;(3)其显示表功能可以实现场景中的中英文字标注,还可以方便载入图片说明;(4)合理调用VC++的sleep函数,可以实现场景的自动变换,可以以微视频的形式输出;(5)基于VC++的一档多视技术[3],可以增加表单视(FormView),易于实现设计交互式界面,如图5左半部分所示。

## 4 详细设计

限于篇幅,本文仅以(1)外摆线的形成(2)等时降线(3)最速降线为例说明详细设计的思路和方法。

### 4.1 外摆线的形成过程详细设计

外摆线如图3所示,其方程为[1]:

$$\begin{cases} x=(a+b)\cos\theta-c\cos\dfrac{a+b}{b}\theta \\ y=(a+b)\sin\theta-c\sin\dfrac{a+b}{b}\theta \end{cases} \quad (1)$$

其中,$a$为定圆半径,$b$为动圆半径,$c$为动圆圆心$O$到点$P$的距离,即$c=|OP|$,$\theta$对应于动圆的自转角度。给定$a$,$b$(一般情况下$a$、$b$取有理数),如果$a/b$可以化为最简分式$m/n$($m$、$n$为互质整数),则动圆将会绕定圆公转$m$圈,同时自转$n$圈(即共计自转$n$*2π弧度),回到起点[4]。每自转一圈,形成摆线的一瓣。如图3所示,$a=3$,$b=1$,$c=1.5$,动圆公转1圈,同时完成三次自转,形成摆线的3瓣。OpenGL下,摆线的动画实现步骤如下:

(1)计算n.给定$a$,$b$,计算$a/b$,化为最简分式$m/n$,得到$n$;

(2)计算曲线上的各个数据点。因为$\theta\in[0, n*2\pi]$,将$[0, n*2\pi]$均分1 000份,令$\theta_i=i\dfrac{2n\pi}{1\,000}, i=0,1,...,1000$,代入(1)式,计算各个$\theta_i$所对应的$p_i(x_i,y_i)$。

(3)绘制图形,OpenGL用多段直线描绘曲线。给定一个整数$t$,绘制直线$P_0P_1,\cdots,P_{t-1}P_t$。$t$初始化为0,每次OpenGL刷新重绘的时候$t$增加1,即每次刷新图形多绘制一段直线,从而实现摆线绘制动画。当$t=1\,000$时,置$t=0$,重新绘制。关键代码如下:

```
for(int  k=0;k<=1000;k++)//计算数据
{theta1[k]=k*2*3.14/1000;}
……..
for( k=0;k<=1000;k++)//画摆线
{x[k]=(a+b)*cos(theta1[k])-c*cos((float(a+b)/b)
*theta1[k]);
y[k] =(a + b)*sin(theta1[k])-c*sin((float(a + b)/b)
*theta1[k]);}
……
for(  k=1;k<=t;k++)//每次重绘摆线,长度增加
{glLineWidth(2.0f);
glBegin(GL_LINES);
glVertex3f(x[k],y[k],0);
glVertex3f(x[k-1],y[k-1],0);
glEnd();}
```

为了增加动画效果,定圆绘制完成后,还要绘制动圆以及动圆上的摆轴$OP$的运动,如图3所示。定圆和动圆圆周的绘制方法参考相关文献[5]。只要知道动圆的圆心,就可以通过每次刷新重绘动圆,表现动圆沿定圆圆周滚动。采用下面的语句移动动圆的圆心:

glRotatef(t*0.36*m,0,0,1);//配合公转,刷新1 000次公转角度360*m

进一步通过动圆的自转实现摆轴$OP$的运动效果,关键语句如下:

glRotatef(n*0.36*t,0,0,1);//配合自转，刷新 1 000 次，自转 360*n 角度

绘制效果如图 3 所示，由图可见，$P$ 点的运动形成了外摆线。对于内摆线和普通摆线的情形，用同样的方法绘制，不再赘述。图 1、图 2、图 3 显示了摆轴 $OP$ 的旋转以及 $P$ 点的运动产生摆线的过程。

### 4.2 等时降线的动画设计

多个小球从摆线的不同高度同时沿摆线下落，会同时到达最低点，这就是摆线的等时降线特性[4]。图 6(a)显示了位于摆线上不同高度的小球，按下“启动”按钮后，各个小球同时沿摆线下滑，图 6(b)显示了各点同时到达最低点。

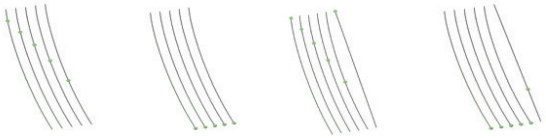


**图 6　等时降线和最速降线**

为了表现具有真实感的小球下滑过程，需要依据物理学的原理计算小球沿等时降线下滑的相关位置数据。思路如下：假设从初始点下滑到最低点的总时间为 $nt$，小球在 $it$ 时刻的位置即为 $H^{(i)}$，($i$=0，1，2，3，…，$n$；特别的，$i$=0 时记为 H，$i$=1 时记为 H’，$i$=2 时记为 H”)。第 $i$ 次刷新的时候，在位置 $H(i)$ 绘制小球。每次刷新重绘的时间间隔都是相同的，但是小球移动的距离 $d^{(i)}=|H^{(i)}-H^{(i-1)}|$ 不同，这样就表现出真实感的动画效果。下面计算 $H^{(i)}$($i$=1，2，3，…，$n$)，为简化问题，这里仅计算 H，H’以及 H”。

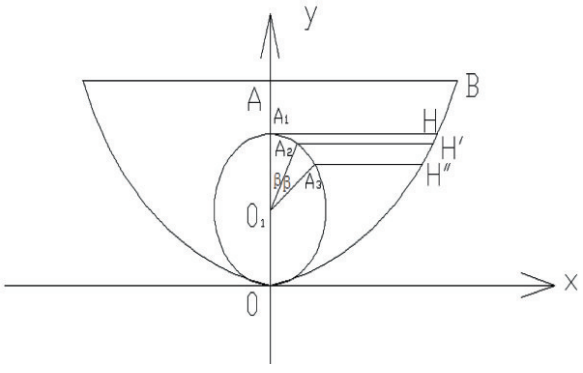


**图 7　小球沿摆线下滑位置计算**

如图 7 所示，$|AB|=\alpha$，摆线的方程为：

$$\begin{cases} x = a(\theta - \sin\theta) - \pi a \\ y = a(\cos\theta + 1) \end{cases} \quad \theta \in [0, 2\pi] \qquad (2)$$

小球初始位置位于 $H$，坐标表示为 $H$(m，h)，t 时刻位于 $H$’(m’，h’)，$2t$ 时刻位于 $H$”(m”，h”)。过 $H$ 作 $x$ 轴的平行线，交 $y$ 轴于 $A_1$，以 $OA_1$ 为直径作辅助圆 $O_1$，过 $H$’、$H$” 作 $x$ 轴的平行线，交辅助圆于 $A_2$、$A_3$，则 $\angle A_1O_1A_2 = \angle A_2O_1A_3=\beta$。即重力作用下小球下落时各个位置点的纵坐标的值等于一个小球沿辅助圆 $O_1$ 匀速转动(角速度为 $\sqrt{\frac{g}{a}}$，g 为重力加速度)时相应位置点的纵坐标的值。这样，$H$’点的纵坐标就可以表示为：

$$h' = \frac{H}{2} + \frac{H}{2}\sin(\frac{\pi}{2} - \beta) \qquad (3)$$

代入(2)式的第 2 式，有

$$\frac{H}{2} + \frac{H}{2}\sin(\frac{\pi}{2} - \beta) = a(\cos\theta + 1), \theta \in [-\frac{\pi}{2}, \frac{\pi}{2}] \qquad (4)$$

求得

$$\theta = \arccos((\frac{H}{2} + \frac{H}{2}\sin(\frac{\pi}{2} - \beta))/a - 1) \qquad (5)$$

再将(5)式代入(2)式的第 1 式，求出 $H$ 点的横坐标。

而 $H$”点的纵坐标就可以表示为：

$$h'' = \frac{H}{2} + \frac{H}{2}\sin(\frac{\pi}{2} - 2\beta) \qquad (6)$$

采用和上面相同的方法，可以求出 $H$”点的横坐标，以及所有关键点的位置坐标。

可以证明，小球下落的时间与初始位置无关[4](证明请参阅相关文献)。小球初始位置不同，沿摆线下滑，最终同时下落到最低点。等时降线详细设计的思路如下：给定初始位置 $H$’(m’，h’)，首先在初始位置绘制小球；OpenGL 刷新一次(对应于经过 $t$ 时刻)，在 $H$”(m”，h”)位置小球；以此类推，计算其余关键点的位置数据。小球沿摆线下滑的动画效果如图 6(a)、(b)所示。小球到达最低点以后，由于其具有水平速度，会沿摆线的左半部分继续上升，即动画的效果是小球沿摆线来回震荡。关键代码如下：

```
float baiqiushujux[201], baiqiushujuy[201];//存放小球位置数据
void baiqiushuju( float a)
{for( i=0;i<=100;i++)//右半部分
{float du = (baixianbaix[i] + 2*pi*a)/a-sqrt(1-( baixianbaiy[i]/a-1)* ( baixianbaiy[i]/a-1));//计算θ
baiqiushujux[i] =baixianbaix[i] + 2*a*sin(du);//计算横坐标
baiqiushujuy[i]=baixianbaiy[i]+ 2*a*sin(du) *sin (du)/(cos(du)-1);}//计算纵坐标
```