

# 大学物理实验操作预习系统的仿真实现

简宇虹,甘维达

(广西民族大学 物理与电子工程学院,广西 南宁 530006)

**摘 要:**介绍了“大学物理实验操作预习系统”,并以系统中的静电场描绘实验为例,详细讲解了如何用 flash 软件开发仿真的大学物理虚拟实验课件。

**关键词:**大学物理实验;虚拟实验;仿真;FLASH 软件开发

**中图分类号:** G434 **文献标识码:** A **文章编号:** 1007-0311(2006)04-0090-06

为了适应开放大学实验室,让大学生自主进行实验这一教学形式的需要,我们制作了《大学物理实验预习操作系统》多媒体网络课件。一方面,使学生能够在实验前,通过该系统直观地了解和熟悉实验的步骤与操作方法,以达到充分预习的目的;另一方面,为学生在开放实验室中进行自主实验时,提供了一个“电子指导教师”的作用。下面将介绍该系统的仿真性能,并以静电场描绘实验中“自己动手”环节为例,详细介绍其仿真的实现。

## 1 大学物理实验操作预习系统简介

大学物理实验操作预习系统,属于桌面式的虚拟实验系统。该系统采用目前常用的网络动画设计软件和图像处理软件 FLASH、PHOTOSHOP 制作而成。

系统中每一实验内容,包括有目的要求、实验原理、仪器介绍、操作演示和自己动手等五个项目环节,而其中的核心部分是“操作演示”和“自己动手”这是两个层次不同的仿真实验环节。“操作演示”环节,是学生在文字、声音以及引导线的引导下,通过使用鼠

标点击仪器中需调节的部位,进行观看实验步骤的动画演示,其实验演示的操作自主性强。“自己动手”环节,则是在没有提示的情况下,学生独立思考,通过鼠标的点击、拖动等操作,完成对虚拟的实验仪器进行调节,其操作效果犹如身临实验现场。

## 2 大学物理实验操作预习系统的仿真性能

该课件制作实现了实验仪器、实验现象、仪器操作、测量和读数的仿真。

### 2.1 实验仪器的仿真

系统中用到的实验仪器来源于真实仪器的数码照片,利用 PHOTOSHOP 处理后既美观又不失真。根据实验需求,同一仪器拍有多个角度的图片,使学生可从不同角度了解仪器形体外观,以最大的可能和限度,缩短了从模拟实验到实际实验的认知距离。

### 2.2 实验现象的仿真

在虚拟实验中的实验现象,动画全都按照实际实验观察到的现象来进行制作,达到符合真实的实验情况。学生在虚拟实验中看到的实验现象能够完全与实

• 收稿日期:2006-07-10.

作者简介:简宇虹(1955-),女,湖南新宁人,广西民族大学物理与电子工程学院大学物理教研室主任,副教授;甘维达(1982-),男,广西南宁人,广西民族大学2002级物理本科班学生。

际实验时的效果一致。

### 2.3 仪器操作的仿真

大学物理实验操作预习系统具有很强的互动性,在每个操作页面中都有直观、方便的按钮与用户互动。学生可以自主地根据自己的熟悉程度来控制实验进度,充分发挥学生的学习积极主动性。其中“自己动手”具有强大的互动性和仿真性。在这一环节,学生不仅可以自己做出“拿”、“旋转”、“拨动”、“打点”等实际操作仪器的动作,而且还可获得虚拟实验系统的反馈。例如,静电场描绘实验中,实验者可以通过鼠标拖动导线进行连接线路。实验者还可以用鼠标拨动电压表,电源的旋钮,拖动滑线变阻器上的滑块等具体操作。在用探针打点时,用鼠标把探针移到“纸上”,每单击一次都会在探针处打一个点,同时随着探针在纸上的位置变化,电压表的读数也是在跟随着变化的。

### 2.4 测量和读数的仿真

当实验中遇到需要测量和读数时,要求学生输入读数,系统可以自动判断出读数的正确与否。系统中专门设计有训练学生练习读数的环节。

## 3 静电场描绘实验“自己动手”环节的仿真实现

下面着重介绍静电场描绘实验“自己动手”环节中有代表性的步骤,以及实现仪器仿真的制作方法。

### 3.1 “自己动手”的 Flash 的结构安排

静电场描绘实验“自己动手”的层次安排为:代码—文字—帮助—按钮—动画—背景。注意,这不是指 FLASH 中的具体层,具体制作时根据需要还可以继续细分每层结构,即不一定只有 6 层。图 1 是静电场描绘实验 FLASH 中的具体图层和时间轴上的安排。时间轴上基本按实验步骤划分到每一帧,代码层上主要写好该步要调用的函数和初始化数据的代码。动作多的步骤可以多分几帧,这样安排可以方便地实现上一步,下一步的跳转,同时日后如发现某步骤有错,或需在中间添加步骤都能够方便地删除,插入。

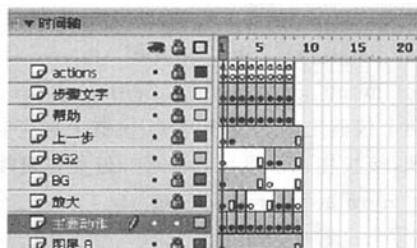


图 1 静电场描绘实验 FLASH 中的图层和时间轴结构

### 3.2 实验操作和各仪器元件的制作

#### 3.2.1 接线场景的制作

接线是电学实验的重要步骤。在课件中要用鼠标拖动导线连接到接线柱上,模仿出用手拿导线连接仪器的过程。为了更进一步仿真,随机给出导线的颜色不同,接好后的线路要与使用导线颜色一致才为正确。全部连接正确才能进入下一步。

首先准备好这一步用到的接线仪器元件。将仪器图片合理的摆放到舞台上。在用一个 Movieclip(实例名为 jietou\_mc)专门绘制连接上仪器的导线(实例名为 tt1\_mc、tt2\_mc……tt10\_mc)并在接线柱的位置放置热区,热区实例名为 zhu1, zhu2……zhu8。用到的所有导线接头结构分为两层,第一层用来放接头图形,铁片部分和裹胶部分要分离,裹胶部分的部分转化为 Movieclip,为后面用改变颜色准备。下一层放置接头的热区,并调为透明。在主场景上专门用一层放置连好的导线。

第一帧中实现连线功能的代码如下:“//”后为讲解,解释的语句。

```
var pass, tt; //tt 表示被选中的接头代号,0 表示没有接头被选中。pass 为判断此步是否完成的变量
```

```
color _ array = new Array ( 0xFF0000, 0x0000FF, 0x339933);  
//定义一个数组用来准备导线颜色,这里使用红,蓝,绿三种颜色
```

```
_root.line0.rope0.ropecolor = new Color(_root.line0.rope0);
```

```
for (k=1; k<=8; k++) {
```

```
target = _root.jietou_mc["tt" + k + "_mc"] ["rope" + k];
```

```
target.ropecolor = new Color(target);
```

```
target = _root["line" + k];
```

```
target.lcolor = new Color(target);
```

```
} //这一段代码为每条导线声明 Color 对象
```

```
startint(); //初始化仪器状态,变量数据
```

```
//下面是这一步接线用到的自定义函数
```

```
//初始化函数
```

```
function startint() {
```

```
pass = 0;
```

```
tt = 0; //记录 line0 的原 X,Y 坐标
```

```
line0x = line0._x;
```

```
line0y = line0._y; //静电场描绘仪上带有
```

的两根导线一开始将不被隐藏

```

_root.jietou_mc.tt9_mc._visible = 1;
_root.jietou_mc.tt10_mc._visible = 1;
_root.line0._visible = 1;
_root.line0.rope0.ropecolor.setRGB(color_array[0]);

```

//显示桌面上提供连接的导线 line0 并把颜色设为黑色

```
hideline();
```

```
hidejiet();
```

hidezhu(); //调用函数隐藏导线、接头、接线柱,把它们都设为不可见

```
}
```

//接着定义要用到的函数

//判断是否全部接好线的函数

```

function alllineok() {
    for (ii=1; ii<=6; ii++) {
        if (_root["line"+ii]._visible == 1) {
            if (ii == 6) {
                pass = 1;
                _root.line0._visible = 0;
            }
        } else {
            break;
            pass = 0;
        }
    }
}

```

//如果所有连接好的导线\_visible 属性为 1 即为连接好线路 pass=1

//隐藏导线函数和接线柱的函数

```
function hideline() {
```

```

    for (i=1; i<=6; i++) {_root["line"+i]._visible = 0;}
}

```

```
function hidezhu() {
```

```

    for (i=1; i<=8; i++) {
        _root.jietou_mc["zhu"+i]._visible = 0;
    }
}

```

```

    _root.jietou_mc["zhu"+i].use = 1;
}

```

//接线柱下的 use 变量表明该接线柱是否还可以连接导线.

```
}
```

//隐藏接头函数

```
function hidejiet() {
```

```

    for(i=1;i<=8; i++) {

```

```

        _root.jietou_mc["tt"+i+"_mc"]._visible = 0;
    }
}

```

//把接头连上仪器的函数,将 line0 的接头碰上接线柱热区并释放时调用.

```
function line0relase(thistarget) {
```

```

    for (i=1; i<=8; i++) {

```

```

        if (thistarget.hitTest(_root.jietou_mc["zhu"+i])&&

```

```

            _root.jietou_mc["zhu"+i].use == 1) {

```

```

                nowcolor = _root.line0.rope0.ropecolor.getRGB();

```

```

                _root.jietou_mc["tt"+i+"_mc"][_root.jietou_mc["zhu"+i].ropecolor.setRGB(nowcolor);

```

```

                _root.jietou_mc["tt"+i+"_mc"]._visible = 1;
            }
        }
    }
}

```

//line0 的接头碰上接线柱热区就把连上仪器的导线显示出来. nowcolor 获得 line0 的颜色并把它赋给连上仪器的导线,使得连上仪器的导线与 line0 的颜色一致

//判断连线的函数, targetnum 为要连接的接线柱代号, rl 为连接好的导线代号.

```
function rightline(targetnum, rl) {
```

```

    ztarget = _root.jietou_mc["zhu"+targetnum];

```

```

    if (ztarget.hitTest(_level0._xmouse, _level0._ymouse)) {

```

```

        _root.jietou_mc["tt"+_root.tt+"_mc"]._visible = 0;

```

```

        _root.jietou_mc["tt"+targetnum+"_mc"]._visible = 0;

```

```

        _root.jietou_mc["zhu"+_root.tt].use = 0;

```

```

        _root.jietou_mc["zhu"+targetnum].use = 0;

```

//两个仪器之间的线连好了,仪器上相应的接头不会出现,接线柱也不可用, use=0. 避免重复接线.

```

        myvalue = _root.jietou_mc["tt"+_root.tt+"_mc"][_root.jietou_mc["zhu"+targetnum].ropecolor.getRGB();
    }
}

```

```

        .ropecolor.getRGB();
    }
}

```

```

        if(r1! =1 && r1 ! = 5) {
            _root["line"+r1].lcolor.setRGB(my-
value);
        }
        _root["line"+r1]._visible = 1;
    } //如果是正确的接线,对应的连好的导线显示
    出来,并且设置它的颜色与所用来连线的接线头一
    致.
    } //line0 换色函数
    function changecolor() {
        _root.line0._x = _root.line0x;
        _root.line0._y = _root.line0y;
        _root.line0.rope0.ropecolor.setRGB(col-
or_array[random(3)])
        // line0 回到原位并随机设置颜色为红,蓝,绿
        的其中一种
    }

```

至此,所有自定义函数都定义完了.接头处按住鼠标移动至目标接线柱时的代码略.

### 3.2.2 仿真圆形按钮的制作

在实验中经常要调节仪器旋钮.下面介绍一个典型的圆形按钮地制作——电源箱上调节输出电压的旋钮.

旋钮要能够被鼠标拨动旋转,鼠标放开或离开按钮,按钮不动.另外由于输出电压有最大最小值,所以旋钮不能 360 度无限旋转,实际中这个旋钮只能旋转 200°左右.电源面板上的指针与旋钮关联,随着旋钮的调节改变.注意,旋钮一开始放置时旋钮所对的位置为-x 轴方向,Movieclip 的注册点为旋钮中心,这是 FLASH 中 0°的位置,否则需在代码中修正偏差度数.在转动旋钮 Movieclip 上添加上代码:

```

onClipEvent (load) {
    minR = 65; //设置旋转的最小度数
    maxR = 265; //设置旋转的最大度数
    _rotation = minR; //设置旋转的开始旋
    转的角度为最小度数
    //定义 ang() 函数用来取得鼠标到旋钮中心连
    线的倾斜角度

```

```

    function ang() {
        xdif = _parent._xmouse-_x;
        ydif = _parent._ymouse-_y;
        angle = ((Math.atan2(ydif, xdif)) *
180)/Math.PI;

```

```

        //通过计算 tan 值来取得倾斜弧度并转为角度
        angle += 180;
        //flash 中规定角度范围为-180°~ +180°这里
        把它化为 0°~ 360°
        return angle;
    } //定义 rot() 函数用来确定旋钮旋转的度
    数
    function rot() {
        _rotation = ang()-curangle;
        g = _rotation;
        if (g<0) {
            g = g+360;
        }
        if (g>maxR) {
            rotate = false;
            _rotation = maxR;
        }
        if (g<minR) {
            rotate = false;
            _rotation = minR;
        }
    }
}

```

旋转的角度=当前鼠标距中心倾斜角度-按下鼠标时鼠标距中心倾斜角度与按下时旋钮的旋转角度差值 curangle. 这样在按钮旋转时,总是保持与鼠标按下时的角度差,不会发生跳变.变量 g 是用来判断旋转的角度是否超出了规定的旋转角度范围.g 有可能小于 0,需要把它转为正值;如果 g 大于旋转的最大度数,则 Movieclip 旋转角度为 maxR;如果 g 小于旋转的最小度数,则 Movieclip 旋转角度为 minR,并停止旋转(rotate = false).

### 3.2.3 静电场描绘仪的仿真

静电场实验最重要的,当然是静电场描绘仪的仿真.根据静电场描绘仪分上下两层的特点,画面安排如图 2.静电场描绘仪 Movieclip(实例名为 bb)如图 3 所示,包含探针(实例名为 l)和上层白纸(实例名为 paper)、下层导电胶三个主要元件.

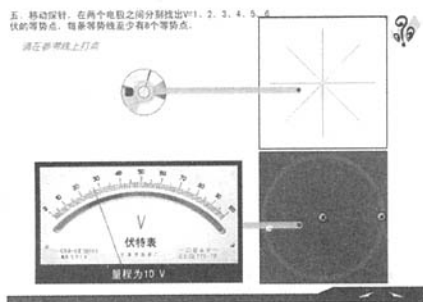


图 2 静电场描绘仪操作画面

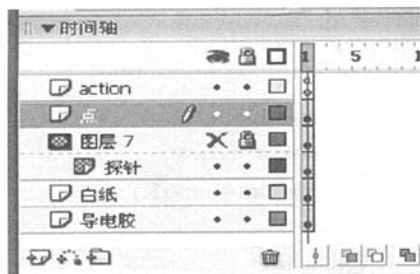


图 3 静电场描绘仪 Movieclip(bb)结构

首先制作探针元件. 探针打点的动作, 用鼠标单击来模拟, 探针头跟随鼠标位置移动. 而鼠标除了模拟打点外, 应该还要能点击舞台上其他按钮(如上一步、下一步按钮).

探针上的代码如下:

```
onClipEvent (mouseMove) {
    px = _parent. paper. _xmouse;
    py = _parent. paper. _ymouse; //储存鼠标在
    纸上的 x,y 坐标
    a = 5; b = 120; k = b/120;
```

根据距中心  $r$  处的电势公式  $U'_r = U_a \frac{\ln \frac{b}{r}}{\ln \frac{b}{a}}$  先定义  $a$ 、 $b$ 、 $k$  几个变量.  $a$  表示中心小圆电极半径,  $b$  表示大圆电极半径. 这里以像素为单位. 所以根据画面大小用  $k$  变量转化一下, 这里经几次测试后, 发现  $k$  使用  $b/120$  比较合适.

$b$ 、 $k$  几个变量.  $a$  表示中心小圆电极半径,  $b$  表示大圆电极半径. 这里以像素为单位. 所以根据画面大小用  $k$  变量转化一下, 这里经几次测试后, 发现  $k$  使用  $b/120$  比较合适.

```
if (Math. abs (_parent. paper. _xmouse) <=
b && Math. abs (_parent. paper. _ymouse) <= b)
{ //判断鼠标是否在白纸上
    Mouse. hide(); //隐藏鼠标
    this. _x = px;
    this. _y = py; //让探针在纸上跟着鼠标移动
```

```
xr = Math. sqrt(px * px + py * py); //用两点
间距离公式计算鼠标离白纸中心的距离
```

```
r = k * xr; //xr 是像素单位, 乘于 k 转化一下
接着通过公式关联伏特表面板(实例名为 vb)
```

```
y = (8 * Math. log(b/r))/Math. log(b/a);
```

```
v = Math. round(y * 10);
```

```
if (v > 80) { v = 80; }
```

```
_root. vb. gotoAndStop(v);
```

```
updateAfterEvent();
```

```
} else { //如果鼠标不在纸上, 将鼠标显示出来,
方便其他操作
```

```
Mouse. show();
```

```
}
```

白纸上模拟打点的制作. 课件中打出的点实际上有两种, 一为原来就预先放在画面上的点; 二为用 `duplicateMovieClip` 函数复制出来的点. 用来复制的点藏在右上角线框边. 白纸中间预先放好的点是按照实际实验正确打点位置放置的, 见图 4 所示. 如果模拟打点时, 点在正确的位置上就让预先放置的正确的点显示出来; 如果不是在正确的点则复制右上角的点到打点的位置.

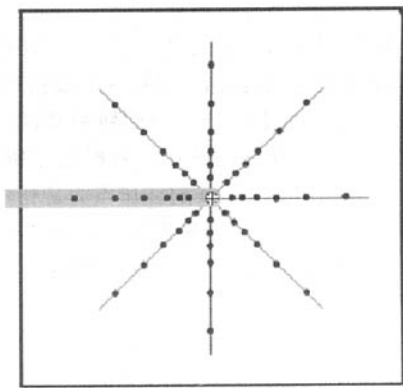


图 4 白纸上的点预先放置的位置

要复制的点实例名为 `dot`, 放好的点实例名为 `rightdot1`、`rightdot2`、

.....`rightdot48`. `bb` 上有如下代码:

```
this. paper. i = 0; //用 i 来记录当前的打点数
```

```
this. l. swapDepths(1000);
```

```
//把探针的深度设为 1000, 保证它在所有点上面. 当
鼠标按下时, 就打点.
```

```

在白纸 paper 上添加 on (press)事件代码:
on (press) {
    if (_root. bb. paper. i>180) {_root. huanzhi. _
visible = 1;}
    else {i++;//每打一点,打点记数加 1
        _parent. dot. duplicateMovieClip("dot"+i,
i);
        _parent["dot"+i]. _x = _xmouse;
        _parent["dot"+i]. _y = _ymouse; //把点
复制到探针(鼠标)位置
        for (k=1; k<=48; k++) {
            if(_root. bb["rightdot"+k]. hitTest(_
level0. bb["dot"+i]. _x+595.5, _level0. bb["dot"
+i]. _y+136, true)) {
                _root. bb["rightdot"+k]. _alpha =
100;
                _root. bb["dot"+i]. removeMovieClip
();
            }
        }
    }
}

```

## 5 结语

“大学物理实验操作预习系统”做到了很好的交互、仿真效果,部分实验课件已在实际教学中试用过,并取得了良好的效果.该系统于 2005 年 12 月获得了第五届广西高等教育教学软件大赛一等奖.2006 年 10 月获得了教育部举办的第六届全国多媒体课件大赛二等奖.

## [参 考 文 献]

- [1]简宇虹,罗颖彬.大学物理仿真实验的制作[J].广西民族学院学报(自然科学版),2003,9(4):24-28.
- [2]向东,何毅,郭萍,等.基于 BS 模式的大学物理仿真实验系统开发[J].数理医药学杂志,2004,17(2):183-184.
- [3]谭守标,王晓蒲,霍剑青.大学物理仿真实验教学系统中霍尔效应虚拟实验课件的设计实现[J].计算机工程与应用,2003,4:42-44.

[责任编辑 黄祖宾]

[责任校对 苏 琴]

## The University Physics Experiment Operation System for Students Preparing their Course

——the Simulation Realization of Electrostatic Field Description Experiment

JIAN Yu-hong, GAN Wei-da

(College of Physics and Electron Engineering,

Guangxi University for Nationalities, Nanning 530006, China)

**Abstract:** This article has introduced "The university physics experiment operation system for students preparing their course". It has taken the electrostatic field description experiment in the system as the example, and has explained how to use flash software to develop the university physics hypothesized experiment courseware in detail.

**Key Words:** University physics experiment; Virtual experiment; Simulation; FLASH software development