

CAI 中混屏显示技术

熊励 余智华 田瑞庭 (湖北汽车工业学院 442002)
毛法尧 (华中理工大学计算机系)

摘要:该系统对多媒体 CAI 中同一屏幕显示正文、关键概念、插图的混屏显示问题;界面设计中所设计到的窗口类问题实现问题进行了研究并付诸实践。

关键词:CAI 混屏显示 界面

随着计算机技术的发展,各类 CAI 软件的不涌现,这些软件中有许多涉及到在数据显示中除常规文本信息外,还需要图形信息。例如:在同一屏幕显示文本内容的同时还显示一些插图,特别对关键概念还需要有明显的标志和进一步解释加深,为此需要采用混屏显示。

1. 混屏显示所采取的措施

尽管实现图文混屏对开发者来说有着极大的吸引力,但在实现上却存在一定的困难,因为当用一屏同时显示图文时,所花的空间太大。而从目前使用 CAI 软件来看,基本上是将它们分开显示,也就是把屏幕分成几部分,规定某一部分显示图形另一部分显示文本;或以牺牲存储空间把文本当作图形来处理,无法满足图文的任意滚屏,即在课文的任何地方均可插入图形图像和弹出关键概念。

为此我们采用的方法是建立三个类库:概念类库;插图类库;课文类库。以课文为基类,关联概念和图形。为了紧凑地存储有效数据,同时不浪费额外空间,我们必须动态地为数据分配存储空间[1],因此,选用链式结构为基本存储结构。

2. 概念关键字的读取

对重点概念以节为单位存入相应概念库,概念的索引标志存放在课文库里,在读取节课文的同时检测是否有概念索引标志,若有则打开相应概念库。

由于概念信息具有暂时性和随时性,故在显示概念的处理上采用弹出式窗口实现,弹出时需要将原有屏幕的信息进行保存与恢复。

对于一屏上若有多个概念则应允许用户随意挑选活动概念,所以需处理活动与非活动概念。

(1)活动/非活动概念处理。对于课文讲解中一屏允许显示多个关键概念,为了满足这一要求,同时考虑到用户阅读概念一般是一个个地进行,即用户在某一时刻只

可能挑选某一个概念进行重点加深,为此,对出现的概念在字体颜色上进行了区别,当用户把光标移动到某一个概念,该概念即为活动概念,用户按“空格”键便弹出概念解释,否则,为非活动概念。

在活动概念与非活动概念的处理上,首先设置概念关键字状态;根据用户的选择来改变关键字显示状态;然后读取当前概念关键字;最后显示概念关键字。如下程序:

改变概念关键字显示状态:

```
for (i=0, firstkey = -1; i < num-key; i++)
    if ((concept[i] -> line < curline) || (concept[i] -> line > = curline + s-line))
        concept[i] -> status = 0;
    else
        { concept[i] -> status = 1;
          if (firstkey == -1)
              firstkey = i;
        };
    if ((curkey == -1) || (! concept[curkey] -> status))
        curkey = firstkey;
    if (curkey! = -1)
        concept[curkey] -> status = 2;
    mark-show = 1;
    .....
```

(2)弹出关键字窗口。近年来弹出式窗口在交换式图形环境中非常流行,原因在于它能把信息分区,加强程序的 I/O 形式。本文所提到的关键字窗口系统的独特特性是面向对象。系统的主体类是 MyConcept,它包含自己的窗口堆栈、一组变量和一组成员函数;堆栈决定如何弹出说明内容然后隐去,变量用于概念窗口所需的数据,

函数用来创建、显示和消除概念窗口,弹出说明内容然后隐去。类的作用是将与关键字窗口有关的操作全部封装在其中,如下表示:

```
class MyConcept
{ public:
    unsigned char * keyword; // 概念关键字
    int keylength; // 概念关键字长度
    unsigned char * sentence; // 概念说明内容
    int line, col; // 在正文中行列位置
    int status; // 状态 0:不显示/1:非活动/2:活动
    Text * intext; // 所属正文指针
public:
    MyConcept()
    { keyword = NULL;
      keylength = 0;
      sentence = NULL;
      line = 0;
      col = 0;
      status = 0; };
    void Show(); // 显示概念关键字
    void PopOut(); // 弹出说明内容然后隐去
};
```

封装的作用是可以有选择地将数据和代码隐藏起来,原则上其他对象不能直接修改该对象拥有的数据,而要通过该对象的成员函数来修改,这就避免了程序之间的相互干扰,提高了软件的安全性。

因为对每个概念的解释不可能是固定不变的,关键字有长有短,所以概念窗口的大小也应有所变化,考虑到屏幕的合理性,在处理弹出概念窗口时,仅以能容下概念解释为标准,即窗口大小随关键字的多少随意变动。当超出一行显示时,要处理跨行时有可能遇到的半个汉字显示问题,即:

```
if (* curpos >= 0xa0) // 汉字?
    hz = ! hz;
    curpos ++;
    n-col ++;
if (n-col >= num-char[n-line]) // 满一行?
    { if (hz) // 半个汉字?
      { num-char[n-line] --;
        n-line ++;
        n-col = 1; }
    else
```

```
    { n-line ++;
```

```
      n-col = 0; };
```

概念窗口的主要数据结构是用来保存每个窗口将要覆盖的屏幕内容。在这里堆栈被设成一个结构,结构中的每个变量存储着一个关键字窗口的信息,堆栈指针是指向堆栈中下一个要用到的存储单元,每次当一个新的窗口被压入堆栈,指针的数值就会增加。成员函数 `TextView::Hide()` 被用来存储可能被弹出窗口所遮盖的内容,也就是隐去课文显示但不清屏。

3. 文本滚动/翻页

由于在用户区窗口中,不能显示大量的文本和图形,而用户则经常要查看各种信息,并且需要对其进行自由浏览,也就是说必须通过上下滚动或翻页来显示更多的信息,目前所开发的一些应用软件中,一般在 DOS 环境下均使用键盘上的键,进行显示窗口的移动;在 Windows 环境中用鼠标进行用户窗口的移动。此处主要处理课文上下滚动和翻页并刷新显示。

(1)滚动/翻页类的定义。考虑到 CAI 的特点,用户在阅读课文时一般是一行一行地读或一页一页地翻,而习题则是一道一道地做,为了使该系统具有独立性,为此,借鉴 Windows 的优点,又能在 DOS 环境下使用,我们采用的文本滚动/翻页显示部件由两部分组成:一是课文显示区域(按钮),二是控制条(滚动条)。有了课文显示区域就应当有其坐标;有了控制条就可以直观的标明当前显示区所处的位置[2]。所以我们定义的类为:

```
class Text
{ protected:
    int left, right, top, bottom; // 课文显示区域(含边界)
    int curline; // 当前正文显示起始行
    unsigned char * curpos; // 当前正文显示起始地址
    unsigned char * content; // 正文内容
    unsigned char * str-concept; // 概念说明存储区
    MyConcept * concept[MAXCONS]; // 概念简介
    struct
    { int line-pic; // 插图在正文中的位置(行)
      char picname[40]; // 插图文件名,若引用已有插图则为其下标
      Graph * picture; // 插图
    } pic[MAXPICS];
public:
```

// 课文上下滚动/翻页并刷新显示, 返回 1: 需重新显示/0: 当前状态未变

```
int ScrollUp();
int ScrollDown();
int PageUp();
int PageDown();
```

(2) 部件的屏幕创建。一个滚动/翻页类应有三个部分, 一是滚屏创建部分, 它的作用是将一个用户定义的类型创建并在屏幕上实现; 二是部件的检测部分, 它将处理用户对部件的操作; 三是部件的关闭部分, 它用来将部件关闭并释放内存。

首先, 要对文本显示框在屏幕上显示的内容所需工作进行整理。

① 按类所定义的坐标和背景色在屏幕上开辟一个矩形区域, 作为课文显示区。

② 打开所定义的文本文件, 并跳至所定义的行, 并将其显示在区域中。

③ 按照文本文件的总行数及当前行创建滚动条。

在总结完所需工作后, 即用程序去实现。

```
textscroll.v = text ->GetCurLine();
```

```
CREATE-BOY-SCROLL(&textscroll); // 在屏幕上创建出滚动条
```

(3) 新状态的检测。当翻页或滚行时, 部件检测函数均为两大部分, 一是对鼠标的检测, 二是对键盘的检测。部件的检测方式是这样的: 通过 Test(TEST * keys) 函数获取系统事件, 并存放在结构中, 然后, 再把事件结构依次传给各个部件, 各部件如果因此事件激活则将此事件截取, 否则, 将事件退还系统。

```
case PGUP; // 向前翻一页
    if (text ->PageUp())
        Show();
    if (BOY-SCROLL-TEST(&textscroll, keys)) // 滚动条变动?
        if (textscroll.v - text ->GetCurLine() == 1)
            text ->ScrollDown(); // 下滚一行
        else
            text ->PageDown(); // 下滚一页
```

文本翻页/滚屏采用了键盘、按钮、滚动条相结合的处理方法, 使得在混屏方式下的显示内容可任意移动 [3]。

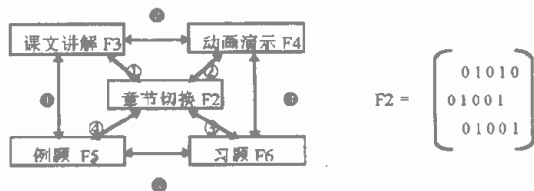
4. 界面设计特点

考虑到该系统的独立性, 可在 DOS 环境下使用, 也

可在 Windows 环境下使用, 在界面设计上充分利用功能键来激活各类事件, 也就是尽可能的把系统所能提供的功能, 在界面上用功能键来定义它, 减少用户的输入, 避免由用户不正确的操作而影响系统运行。

(1) 章节切换的实现。现有的 CAI 软件基本是采用类似树形目录的数据结构, 用户要想从当前章节转入其他章节学习必须返回主菜单再选, 灵活性差, 这无形给用户带来不便, 为克服这些缺点我们采用了网状数据结构, 用户无论在课文讲解、动画演示、例题还是习题模块均用功能键选择从当前章节跳转到所需要的章节, 灵活性强, 使用方便。在此介绍它的实现过程。

CAI 不同于其他应用软件, 它必须考虑用户的学习规律, 用户在学习课文的同时, 可观看动画的表示, 分析例题, 然后通过习题测试, 检查自己的学习成绩。功能模块之间跳转用下图来表示:



上图模拟了不同章节之间的逻辑联系, ①~④表示章与章之间、节与节之间、章与节之间和节与章之间进行跳转必须通过章节切换模块; ①~④表示同一章节之间跳转不需要通过章节切换模块。其实现过程为:

① 首先确定章节切换类, 在此类中主要含有装入章节信息函数和检测系统事件函数, 即:

```
class ChapterMenu: public Box
public:
    void LoadChapter(char * filename);
    int Test( TEST * keys);
```

② 装入章节信息, 在装入章信息时, 同时根据章消息装入节信息。

```
void LoadSection();
```

③ 切换实现

```
if (res)
    for (i=0; i<3; i++)
        caista.cur[i] = 0;
    clabel[ caista.zhang - 1 ] ->SetStatus(2);
    clabel[ caista.zhang - 1 ] ->Show();
```

(2) 窗口的简洁性。因用计算机软件辅助教学本身

就要考虑到用户使用的方便性,有可能用户是初次使用计算机,而用 CAI 课件进行教学,实际上是计算机屏幕直接和学生接触,因此在窗口设计时,屏幕的安排是非常重要的。屏幕设计不仅要能充分的表现内容,同时又要讲究美观,符合通常的习惯,所以必须设计简单明了的 CAI 窗口,让用户一看就会用,因此设计了按钮、滚动条、活动/非活动状态显示和帮助控件类,用户使用到哪提示跟到哪。下面介绍这些控件的创建、激活、响应事件处理方法上的特点。

窗口是应用程序和屏幕沟通的主要工具,窗口有三个维数:高度、宽度和深度,深度是它和其他窗口的关系。用户一进入系统,就处在课文讲解模块 F3 中,为了使子模块之间跳转相对容易,在主模块的前面定义了全局变量和全局类,即:

```
void mainwindow(void);
extern int chapterwindow(void); //章节切换子窗口,
返回 0:取消/1:切换 F2
extern int helpwindow(void); // 帮助子窗口;F1
extern int ltwindow(void); // 例题子窗口, 返回按
键编号 F5

extern int xtwindow(void); // 习题子窗口, 返回按
键编号 F6
extern int cjwindow(void); // 成绩显示子窗口, 返
回按键编号 F7
extern int cai-dh(char * fname); // 动画演示子窗
口, 返回按键编号 F4
.....
```

在处理上,首先通过内部键盘扫描码定义相应键值,例如:define F1 = 0x3b00;define F2 = 0x3c00,通过在主程序中的选择结构,驱动相应的子模块,并且隐含前一模块的各种功能,显示此模块相应的功能,这样处理不仅跳转方便,而且速度快,并且增删功能也容易。例如不管当前处在何种状态,F6 键按下,激活习题模块,把当前章节的习题库打开;习题界面调入;配合习题的正误判断、按钮、滚动条均处在活动状态,而原来的状态处在后台。

①按钮的实现。按钮控制是一类用于启动动作改变数据对象属性的控制,用户可以使用键盘或鼠标的点取对它们进行选择。CAI 功能的选择全部采用立体按钮菜单技术,界面友好,更紧接用户。在实现按钮技术中为了

产生逼真的立体感,用灰色按钮,当光从屏幕的左上角向屏幕的右下角照下来,一个凸出的按钮应是左上边呈白色,右下边呈深灰色;当按下时正好相反,产生凹入的感觉。

·按钮的创建

```
typedef struct -key3d {
    int x1,y1; /* 按键左上角坐标 */
    int x2,y2; /* 按键右下角坐标 */
    char ico[MaxPath]; /* 按键中图标文件名 */

    int enable; /* 是否允许激活 */
    int hotkey; /* 热键 */
    int index; /* 返回值 */
} KEY3D;
```

按钮控制信息是由应用程序使用函数发送给控制,用于获取或改变控制的状态,当选择一个按钮时,该控制就向它的父窗口发送一条消息,通知其父窗口在按钮时进行操作。

·按钮的检测 Test (TEST * keys);

·命令按钮的响应,每个按钮创建时必须指定唯一的一个 ID 值,根据 ID 值执行一定的操作。

②滚动杆。由于窗口用户区的限制,在窗口用户区中不能显示大量的文本或图形,必须使用滚动杆滚动用户区的限制,使原先在用户区之外的内容也能在用户区中显示出来。在此是把一个单元文件底定义为一个逻辑最大值,文件头定义为最小值,一屏为大改动步长,一行小改动步长。

参考文献

- [1] 崔屹编译.“数据结构与 C 语言程序设计”.北京 中科院希望电脑公司,1991
- [2] Kenneth E. Hay, Mark Gugdial, Shari Jackson, Robert A, Boyle and Elliot Soloway . Students as multimedia Composers. Computers Educ . 1994, Vol. 23, No. 4: pp. 301—317
- [3] 阮江,沈铁齐编译.“用 C 语言设计屏幕界面技术”.北京北京希望电脑公司,1991

(来稿时间:1997 年 12 月)