# 滚动控件初步分析

## 引言

此窗口控件主要用于方便,快捷的编辑文本内容和图片内容.

## 需求分析:

功能需求:

1. 控件以单元格作单位显示其中内容.内容种类包括文本,图片
2. 当控件内容超出控件本身大小时,自动显示滚动条,用于浏览超出控件大小的内容
3. 支持动态布局,控件里的内容布局随控件尺寸变化等比例变化
4. //每个单元格均可进行编辑操作 (暂不实现)

## 总体设计:

1. 控件主窗口-滚动窗口类-ScrollWindow
2. 滚动条- 滚动条类 -ScrollBar
3. 单元格 单元格类 – UnitWindow

数据结构及api:

Class ScrollWindow{

//Vector<UnitWindow\*> units;

Vector<CWnd\*> units;

ScrollBar\* scrollbar;

Int width;

Int height;

Int x;

Int y;

COLORREF bgcolor;

Int row\_max

Int col\_max;

Int unit\_width;

Int unit\_height;

Public:

Bool addWidget(int row, int col, CWnd\* wnd);

Bool removeWidget(int row, int col);

Bool updateWidget(int row, int col,CWnd\* wnd);

Void displayWIndow();

Void setWidth(int width);

Void setHeight(int height);

Void setColor(bgColor);

Void setScrollBar(Scrollbar\* scrollBar);

Void onResizeEvent();

Void styDisplayCalculate(); // 一种计算页面滚动的方法

}

/\*Class UnitWindow{

CPixmap\* pixmap;

CString \*str;

Int width;

Int height;

COLORREF bgcolor

COLORREF textcolor;

}\*/

Class Scrollbar{

Int wnd\_width;

Int wnd\_height;

Int \_width

Int \_height;

COLORREF bgcolor;

COLORREF barcolor;

CButton\* up\_arrow\_btn;

CButton \* down\_arrow\_btn;

Public:

Void setWidth();

Void setHeight();

Void setWndHeight();

Void setWndWidth();

Void setBgColor();

Void setTxColor();

Void setUpArrowbtn();

Void setDownArrowBtn();

}

## 详细设计:

接口设计:

1. ScrollWnd类根据自身内容尺寸的变化调整滚动条的尺寸

由于ScrollWnd为主要内容,因此在ScrollWnd中添加

UpdateChild方法用于改变滚动条的尺寸及位置.

功能实现:

.1.ScrollWnd类

1. 当类中添加的内容超出控件尺寸时,调用显示进度条方法ShowBar();

(2) 类中接收鼠标滚轮事件, 当鼠标滚轮滚动时,

规定鼠标滚轮每次滚动内容整体移动scroll\_heihgt = 10pixel

滚轮向后滚动页面向下,滚动,滚轮向前滚动页面向上滚动

详细滚动算法见1.1..

ScrollWnd类调用UpdateChild方法, 调整滚动条尺寸等信息.

1. addWidget 方法

类添加一行, 用于装载添加的控件,初始化控件坐标后直接显示,并对滚动条大小等信息作出修改

1. removeWidget方法

类删除此控件,并对滚动条大小等信息作出修改

1. updateWidget()方法

更新参数中指定位置的控件信息,并删除原控件

# 滚动控件问题总结

## 问题1: 显示算法显示画面闪烁问题

难以解决显示窗口滚动闪烁问题. 因为滚动控件显示的内容并非由滚动控件绘制,因此无法使用双缓冲技术.

改进措施: 可以在当前控件已有的以单元格为单位绘制的基础上, 将所有绘制操作都交给WM\_PAINT消息,由此可以利用双缓冲技术解决闪烁问题,并能够支持内容滚动., 但缺点在于无法支持滚动显示CWnd类, 为了显示更多类型的内容,如图片,彩色字体等,只能为滚动条扩充新的绘图接口,用于设置每个单元格对应的内容种类.最后统一交给paint函数绘制

具体猜想步骤:

用户以单元格为单位指定每个单元格的内容.指定完成后,在滚动窗口时,paint函数根据页面滚动算法计算出的数据直接绘制内容.问题在于如何局部显示单元格内容,将要显示的全部内容在内存DC中绘制成一张位图,将该位图按照显示在屏幕的部分进行坐标设置

**如此,我们的控件即可支持两种内容显示,只是对CWnd类的支持还有一些问题**

# 备注

## 1.

页面滚动实际上就是改变页面内容中显示的控件的坐标.

由上, 只需在页面滚动信号发出后,将页面中所有控件的坐标均作出相应修改即可实现滚动页面的效果.

滚动页面内容之后,滚动条中滚珠额位置应当有相应变化.

为了计算滚动条中滚珠的位置和滚动页面当前显示内容之间的相对关系.

假设需要滚动内容总高度为ContentHeight, 滚动页面高度为:PageHeight

滚动条总高度 ScrollHeight, 则设滚珠高度为BallHeight()

滚动页面在滚动内容中所占的比例与滚珠在滚动条中所占的比例相同,据此可得滚珠高度:

BallHeight / ScrollHeight = PageHeight / ContentHeight

接下来计算滚珠在滚动条的起始位置BallStart

滚动页面高度

滚动内容总高度

滚动条总高度

滚球

为了便于计算滚珠的起始位置,将所有添加到滚动页面的内容按照其高度进行累加统一组织起来形成vector数组, 使用虚拟头指针Ptr指向滚动页面当前显示内容的首个元素在滚动内容中的位置cur\_head\_y.

如此可得到

Ballstart / ScrollHeight = cur\_head\_h / ContentHeight

为了实现滚动页面的效果,建立每一行和其位置信息的映射关系.map.

显示时,设此时cur\_head\_y值为y,

则此时应显示的起始行应为:

Row\_start = y / row\_height

该行显示的起始位置应为

Y\_start = y % row\_height

显示的结束行应为:

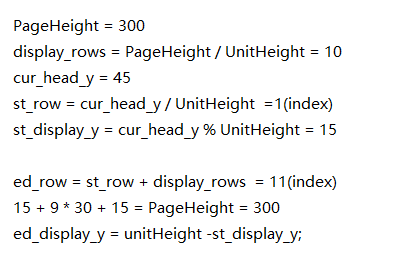
Row\_end = row\_start + Page\_Height / row\_height

**综上**

**通过cur\_head\_y控制整个页面内容显示的起始位置,如公式 1, 当要显示滚动页面时,首先根据公式的计算结果找到对应行号之间的全部内容, 根据st\_display\_y计算结果进行内容显示.**

**即设置该区域的控件坐标.. 建立一个cur\_show\_controls数组进行保存当前显示的所有控件,依据st\_display\_y修改该数组所有控件的坐标.**

**亦即在styDisplayCalculate方法中, 计算好需要的数据后,在units中找到对应区域的控件,设置该区域范围控件的坐标,每次滚动前,先隐藏之前显示的所有控件,之后重新设置坐标,再进行显示**



公式 1