

滚动条的实现（功能不完整）

仅供参考

说明：此文件只是一个不完整的文件，只是画一个可移动的窗口，没有任何与列表内容有关联，要实现这点，还有许多的地方需要修改。正在阅读 uc/gui3.24 的源码，学习它们是如何做的。

同时，感觉这个文件有问题，希望能与各位多多讨论！

另：新建了一个 WINDML 讨论群：42093219，有兴趣的话，可以加入来共同讨论！

谢谢！



Much 2008-7-9

```
/*
 *test.c
 *
 *实现一个滚动条的窗口，只是一个滚动条的窗口，还没有与外部要显示的内容的关联之处，还有待完善！
 *
 * 此文件先到此结束，另起文件，考虑显示内容关联的问题！ 2008-7-8 17:38:56 much
 *
 *08.07-07 much 创建类滚动条窗口
 */
```

```
#include <stdio.h>
#include <ugl/uglWin.h>
#include <ugl/winMgr/wwm/wwm.h>
#include <ugl/winMgr/wwm/wwmButton.h>
```

```
WIN_APP_ID appId;
UGL_RECT      winTestRect;    /*滑块的尺寸*/
UGL_RECT      RollRect;      /*滑块活动的范围*/
WIN_ID        winRollButtonId;
int           winTestIsRectSet = 0;
int           winRollButtonLenght = 10;    /*每次移动的距离*/
```

```
UGL_LOCAL UGL_STATUS cbAppCreate
(
    WIN_APP_ID appId,
```

```
        WIN_MSG * pMsg,  
        void * pData,  
        void * pParam  
    );
```

UGL_LOCAL UGL_STATUS cbWinTestDraw

```
(  
    WIN_ID winId,  
    WIN_MSG * pMsg,  
    void * pData,  
    void * pParam  
);
```

UGL_LOCAL UGL_STATUS cbWinRollDraw

```
(  
    WIN_ID winId,  
    WIN_MSG * pMsg,  
    void * pData,  
    void * pParam  
);
```

UGL_LOCAL UGL_STATUS cbWinRollButtonRectSet

```
(  
    WIN_ID winId,  
    WIN_MSG * pMsg,  
    void * pData,  
    void * pParam  
);
```

UGL_LOCAL UGL_STATUS cbWinRollButtonSlippage

```
(  
    WIN_ID winId,  
    WIN_MSG * pMsg,  
    void * pData,  
    void * pParam  
);
```

UGL_LOCAL UGL_STATUS cbWinRollUpSelect

```
(  
    WIN_ID winId,  
    WIN_MSG * pMsg,  
    void * Pdata,  
    void * pParam  
);
```

```
UGL_LOCAL UGL_STATUS cbWinRollDownSelect
```

```
(
    WIN_ID    winId,
    WIN_MSG   *   pMsg,
    void      *   Pdata,
    void *     pParam
);
```

```
void test()
```

```
{
    appId = winAppCreate("test",0,0,0,UGL_NULL);
    winAppCbAdd(appId,  MSG_APP_CREATE,0,cbAppCreate, UGL_NULL);
}
```

```
UGL_LOCAL UGL_STATUS cbAppCreate
```

```
(
    WIN_APP_ID appId,
    WIN_MSG   *   pMsg,
    void *     pData,
    void *     pParam
)
{
```

```
    WIN_ID    winTestId;
    WIN_ID    winRollId;
    WIN_ID    winRollUpId;
```

```
    WIN_ID    winRollDownId;
```

```
    RollRect.left = 0;
    RollRect.top = 20;
    RollRect.right = 19;
    RollRect.bottom = 479;
```

```
    winTestId = winCreate(appId,UGL_NULL,WIN_ATTRIB_FRAMED
WIN_ATTRIB_VISIBLE,0,0,797,560,UGL_NULL,0,UGL_NULL);
```

```
    winCbAdd(winTestId,MSG_DRAW,0,(WIN_CB *)cbWinTestDraw,UGL_NULL);
    winAttach(winTestId,UGL_NULL_ID,UGL_NULL_ID);
```

```
/*滚动条窗口*/
```

```
winRollId
```

```
=
```

```

winCreate(appId,UGL_NULL,WIN_ATTRIB_VISIBLE,300,20,20,500,UGL_NULL,0,UGL_NULL);
    winCbAdd(winRollId,MSG_DRAW,0,(WIN_CB *)cbWinRollDraw,UGL_NULL);

    winCbAdd(winRollId,MSG_PTR_BTN1_DOWN,0,(WIN_CB*)cbWinRollButtonSlippage,UGL_NULL);
    winAttach(winRollId,winTestId,UGL_NULL_ID);

    /*向上的按钮*/
    winRollUpId = wwmButtonCreate(appId," ↑ ",0,0,20,20,UGL_NULL,0,UGL_NULL);
    winCbAdd
        (winRollUpId,MSG_BUTTON_SELECT,0,(WIN_CB
*)cbWinRollUpSelect,UGL_NULL);
    winAttach(winRollUpId,winRollId,UGL_NULL_ID);

    /*滑块*/
    winRollButtonId = wwmButtonCreate(appId,"",0,20,20,460,UGL_NULL,0,UGL_NULL);

    winCbAdd(winRollButtonId,MSG_RECT_CHANGING,0,(WIN_CB*)cbWinRollButtonRectSet,UGL_NULL);

    winAttach(winRollButtonId,winRollId,UGL_NULL_ID);

    /*向下的按钮*/
    winRollDownId = wwmButtonCreate(appId," ↓ ",0,480,20,20,UGL_NULL,0,UGL_NULL);
    winCbAdd
        (winRollDownId,MSG_BUTTON_SELECT,0,(WIN_CB
*)cbWinRollDownSelect,UGL_NULL);
    winAttach(winRollDownId,winRollId,UGL_NULL_ID);

}

```

```

UGL_LOCAL UGL_STATUS cbWinTestDraw

```

```

(
    WIN_ID    winId,
    WIN_MSG   *   pMsg,
    void *    pData,
    void *    pParam
)
{
    UGL_GC_ID gcId = pMsg->data.draw.gcId;
    UGL_RECT  rect = pMsg->data.draw.rect;
    uglLineWidthSet(gcId,0);
    uglBackgroundColorSet(gcId,WIN_LIGHTBLUE);
    uglRectangle(gcId,rect.left,rect.top,rect.right,rect.bottom);
}

```

```

UGL_LOCAL UGL_STATUS  cbWinRollDraw
(
    WIN_ID    winId,
    WIN_MSG   *   pMsg,
    void *    pData,
    void *    pParam
)
{
    UGL_GC_ID gcId = pMsg->data.draw.gcId;
    UGL_RECT  rect = pMsg->data.draw.rect;
    uglLineWidthSet(gcId,0);
    uglBackgroundColorSet(gcId,WIN_GREEN);
    uglRectangle(gcId,rect.left,rect.top,rect.right,rect.bottom);
}

```

/*移动滑块*/

```

UGL_LOCAL UGL_STATUS cbWinRollButtonSlippage
(
    WIN_ID    winId,
    WIN_MSG   *   pMsg,
    void *    pData,
    void *    pParam
)
{
    UGL_RECT rect;
    int height;
    winRectGet(winRollButtonId, &rect);
    height = UGL_RECT_HEIGHT(rect);

    if (UGL_POINT_IN_RECT (pMsg->data.ptr.position,RollRect)) /*在滚动条窗口内*/
    {
        if (UGL_FALSE ==  UGL_POINT_IN_RECT (pMsg->data.ptr.position,winTestRect))
/*判断有没有在滑块上*/
        {
            if ((pMsg->data.ptr.position.y < winTestRect.top) &&
(pMsg->data.ptr.position.y >20)) /*判断使其只响应在滑块上方位置时的点击*/
            {
                UGL_RECT_MOVE(winTestRect,0,-height);

                if (winTestRect.top <= 20)
                {
                    UGL_RECT_MOVE_TO(winTestRect,0, 20 );

```

```

        }
    }
    else if ((pMsg->data.ptr.position.y > winTestRect.bottom) &&
(pMsg->data.ptr.position.y < 480))/*判读使其只响应在滑块下方位置时的点击*/
    {
        UGL_RECT_MOVE(winTestRect,0,height);
        if (winTestRect.bottom >= 480)
        {
            UGL_RECT_MOVE_TO(winTestRect,0, 480 - height );
        }
    }
}

}
winRectSet(winRollButtonId,UGL_NULL);
}

```

```

UGL_LOCAL UGL_STATUS cbWinRollButtonRectSet

```

```

(
    WIN_ID    winId,
    WIN_MSG   *   pMsg,
    void *    pData,
    void *    pParam
)
{
    WIN_ID winPId;

    winPId = winParent(winId);
    /*只是在第一次画的时候不使用新设置的尺寸*/
    if (winTestIsRectSet == 1)
    {
        pMsg->data.rect.newRect.top = winTestRect.top;
        pMsg->data.rect.newRect.bottom = winTestRect.bottom;
        winRectInvalidate(winId,UGL_NULL);
    }
    winTestIsRectSet = 1;
}

```

```
UGL_LOCAL UGL_STATUS cbWinRollUpSelect
(
    WIN_ID    winId,
    WIN_MSG   *   pMsg,
    void      *   Pdata,
    void *     pParam
)
{
    UGL_RECT rect;
    int height;
    winRectGet(winRollButtonId, &rect);
    height = UGL_RECT_HEIGHT(rect);

    if (rect.top >= 20)
    {
        rect.top = rect.top - winRollButtonLenght;
        rect.bottom = rect.bottom - winRollButtonLenght;
        if (rect.top <=20)
        {
            rect.top = 20;
            rect.bottom = rect.top + height - 1 ;

            if (rect.bottom >= 479)
            {
                rect.bottom = 479;
            }
        }
        winTestRect.top = rect.top;
        winTestRect.bottom = rect.bottom;
    }
    winRectSet(winRollButtonId,UGL_NULL);
}
```

```
UGL_LOCAL UGL_STATUS cbWinRollDownSelect
(
    WIN_ID    winId,
    WIN_MSG   *   pMsg,
    void      *   Pdata,
    void *     pParam
```

```

    )
    {
        UGL_RECT rect;
        int height;
        winRectGet(winRollButtonId, &rect);
        height = UGL_RECT_HEIGHT(rect);
        if (rect.bottom <= 479)
            {
                rect.top = rect.top + winRollButtonLenght;
                rect.bottom = rect.bottom + winRollButtonLenght;

                if (rect.bottom >=479)
                    {
                        rect.bottom = 479;
                        rect.top = rect.bottom - height + 1 ;

                        if (rect.top <= 20)
                            {
                                rect.top = 20;
                            }
                    }

                winTestRect.top = rect.top;
                winTestRect.bottom = rect.bottom;
            }
        winRectSet(winRollButtonId,UGL_NULL);
    }

```

/*手动改变滑块的大小*/

```

UGL_STATUS kt(int x,int y)
{
    if ((x < y) && (x > 20) && (y < 460)) /*判断参数的范围*/
        {
            if (x <= 20)
                {
                    winTestRect.top = 20;
                }
            else
                {
                    winTestRect.top = x;
                }
        }
}

```

```
    if (y >= 460)
    {
        winTestRect.bottom = 479;
    }
    else
    {
        winTestRect.bottom = y;
    }

    winRectSet(winRollButtonId,UGL_NULL);
}
else
    return (UGL_STATUS_ERROR);
}
```

希望能与大家一起讨论，共同提高！

BLOG: www.rtoslab.net