WindowScrollUp

BLWindowScrollUp.bas

This subroutine specified rectangle of screen and scrolls it up by a character. You might be able to use it for games (though there are probably faster scrolly routines for that); but the aim here is to be able to scroll up part of the screen, so that you can split between text on a rectangle area and other information elsewhere - e.g. graphic adventures.

```
SUB BLWindowScrollUp(X as uByte, Y as uByte, Width as uByte, Height as uByte)
ASM
;Routine for printing and scrolling text in any
;window, anywhere on the screen.

;Main scrolling routine
BLWindowScrollUp:
EX AF, AF'
LD   A,(IX+11)  ;(ROWS) Store # of Lines in A'
EX AF,AF'

LD   HL,BLWindowScrollUpScreenTable    ;Start of address table
LD   C,(IX+7)   ;(Y) Move the "pointer" to the
LD   B,0         ;appropriate position in the table
ADD  HL,BC       ;and store it in (BLWindowScrollPOINT)
ADD  HL,BC
LD   (BLWindowScrollUpPOINT),HL  ;Pointer position stored
BLWindowScrollUpLOOP:

    LD   HL,(BLWindowScrollUpPOINT)
    LD   E,(HL)
    INC  HL
    LD   D,(HL)

    ;Address of start of screen line now in DE
    LD A,(IX+5) ;(X)
    ADD A,E
    LD E,A

    ;Address of left-hand side of window now in DE
    EX AF,AF'
    DEC  A
    JP   Z,BLWindowScrollUpBLANK     ;Quit this loop if we have
    EX AF,AF'
    INC  HL          ;Move the pointer to the next item
    LD   (BLWindowScrollUpPOINT),HL  ;in the table. Save position
    LD   C,(HL)
    INC  HL
    LD   B,(HL)      ;Start of next line down in BC
    LD   L,(IX+5)    ; (X)
    LD   H,0
    ADD  HL,BC
    ;HL now points to the screen address 8 pixels below
    ;the one held in DE
    LD   B,8         ;8 pixel lines to be transferred
    ;Now move 8 pixel lines up the screen by 8 pixels

    BLWindowScrollUpTRANS:
        LD A,B ; Save B
        LD   C,(IX+9) ;(Cols)
        LD   B,0
        PUSH HL
        PUSH DE          ;Save all registers
        LDIR             ;Transfer the line of pixels
        POP  DE
        POP  HL

        ;Move HL and DE down one pixel
        INC D
        INC H

        LD B,A ; Recover B

    DJNZ BLWindowScrollUpTRANS
    ;One line of characters has now been transferred

JP   BLWindowScrollUpLOOP        ;Back for next line of characters

;Scrolling finished. Now erase last character line
BLWindowScrollUpBLANK:
LD   C,8
LD   L,(IX+9) ; (COLS)
```

```
BLWindowScrollUpLOOP2:

    PUSH DE
    LD   B,L ;(IX+11) - Cols
    XOR  A
    BLWindowScrollUpLOOP3:
        LD   (DE),A
        INC E
    DJNZ BLWindowScrollUpLOOP3
    POP  DE
    INC D
    DEC C
    JR NZ, BLWindowScrollUpLOOP2

;DJNZ BLWindowScrollUpLOOP2
JP BLWindowScrollEnd

BLWindowScrollUpPOINT:  DEFW 0

BLWindowScrollUpScreenTable:
       DEFW 16384
       DEFW 16416
       DEFW 16448
       DEFW 16480
       DEFW 16512
       DEFW 16544
       DEFW 16576
       DEFW 16608
       DEFW 18432
       DEFW 18464
       DEFW 18496
       DEFW 18528
       DEFW 18560
       DEFW 18592
       DEFW 18624
       DEFW 18656
       DEFW 20480
       DEFW 20512
       DEFW 20544
       DEFW 20576
       DEFW 20608
       DEFW 20640
       DEFW 20672
       DEFW 20704

BLWindowScrollEnd:
END ASM
END SUB
```

Usage

```
BLWindowScrollUp(TopLeftXCoordinate, TopLeftYCoordinate, WidthInCharacters, HeightInCharacters)
```

The parameters are the X,Y print coordinates of the Top Left corner, width in characters, and height in characters.

Example in use:

```
REM Quick routine to fill the screen with crap so we can demonstrate scrolling.
SUB fillRubbish()
ASM
    LD DE,16384
    LD HL,0
    LD BC,6144
    LDIR
END ASM
END SUB

fillRubbish() ' Fill the screen with stuff.

'actual use demo here:

FOR n=1 to 10
BLWindowScrollUp(3,3,8,15)
BLWindowScrollUp(28,10,3,8)
PAUSE 10
NEXT n
```