PixelScroll

pixelScroll.bas

Does what it says on the tin! Scrolls the screen by the set number of pixels, leaving blank pixel rows beyond it. Attributes are untouched.

It's fast - it uses the screen tables method. You'll need the relevant screen tables files - https://dl.dropbox.com/u/4903664/ScreenTables.7z

```
SUB PixelScrollUp(numOfLines as uByte)
ASM
; BLPixelTable is where the table starts

AND A ; Flags off A
JP Z, BLPixelScrollUpEnd ; We were asked to scroll zero. Quit!
CP 192
JP NC, BLPixelScrollUpEnd ; We can't scroll more than 191 lines up. Quit!

LD C,A ; Current Line
LD B,A ; Jump
PUSH BC ; Save Line count.

BLPixelScrollUpMainLoop:

; screen address routine
LD H,BLPixelTable/256
LD L,C
LD D,(HL)
INC H
LD E,(HL)   ; DE is source address

;LD H,BLPixelTable/256
DEC H ; get H back to pixeltable.

LD A,C
SUB B  ; A is now destination line number
LD L,A
LD A,(HL)
INC H
LD L,(HL) ; HL: is destination line address
LD H,A    ;
EX DE,HL  ; Swap! ; HL=Source Address. DE=Dest address.


```; A small version has these two lines instead of the pile of LDI:
;   ld bc,32 ; 32 bytes to transfer
;   ldir

;(A very small version would calculate screen addresses, instead of use the table!)

; A fast version has these 32 LDIs instead: (About 27% faster) - but uses up 28 bytes more.
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
LDI
```
```

```
LDI
LDI

POP BC
INC C
LD A,C
CP 192
PUSH BC ; Save count again.

JP C, BLPixelScrollUpMainLoop ; Not carry? then We hit the bottom of the screen. Need zeroes.

; blank remaining rows
POP BC ; Balance Stack
LD C,B ; Push diff into C
LD A,192
SUB C  ; A now shows row num of the top row to clear.
CP 192 ; are we done
JP Z,BLPixelScrollUpEnd
LD D,0

BLPixelScrollUpClearBigLoop:

LD H,BLPixelTable/256
LD L,A
LD C,(HL)
INC H
LD L,(HL)
LD H,C      ; HL is current row
LD B,32 ; 32 bytes
BLPixelScrollUpClearLoop:
LD (HL),D
INC L
DJNZ BLPixelScrollUpClearLoop

INC A
CP 192

JP C, BLPixelScrollUpClearBigLoop
JP BLPixelScrollUpEnd

END ASM
#include once "ScreenTables.bas"
ASM

BLPixelScrollUpEnd:
END ASM
END SUB
```

Usage

Example:

```
PixelScrollUp(2)
```

Will scroll the screen up by 2 pixels.