

;http://www.melon64.com/forum/viewtopic.php?t=342

; SYS 4096

* = \$1000

; Double buffered scroll with colour ram routines

; screenone is located at \$c000 and screentwo is located at \$c400

; colourram is located at \$d800

screenone = \$c000

screentwo = \$c400

colourmap = \$d800

_stacka = \$a0

_stackx = \$a1

_stacky = \$a2

; setup irq

```
_start      sei
            lda #$35
            sta $01
            lda #$7f
            sta $dc0d
            sta $dd0d
            lda #$01
            sta $d01a
            ldx #$fe
            sta $d012
            lda #$1b
            sta $d011
            lda #<irq
            sta $fffe
            lda #>irq
            sta $ffff
            lda #<@dummy
            sta $fffa
            lda #>@dummy
            sta $ffffb
            lda #$00
            sta $d020
            sta $d021
            lda #$ff
```

```

        sta $d015
        lda #100
        sta $d000
        sta $d001
        lda #%00000000 ; select $c000-$ffff for vic
bank
        sta $dd00
        lda #%00000000 ; character memory $c800 (vic
bank + $800)
        sta $d018
        cli
@here   jmp @here
@dummy  rti

irq     sta _stacka
        stx _stackx
        sty _stacky
        lda #$ff
        sta $d019
        lda #150
        sta $d012
        inc $d020
        jsr read_joystick
        dec $d020
        ldy _stacky
        ldx _stackx
        lda _stacka
        rti

; read joystick

read_joystick
@rj     ldx #$08
        lda $dc00
        and joystick_table,x
        beq @rj_self_mod
        dex
        bpl @rj
        rts
@rj_self_mod  lda joystick_action_high,x
        sta @rj_jsr+2
        lda joystick_action_low,x
        sta @rj_jsr+1
@rj_jsr  jsr joystick_fire
        rts

joystick_table  byte $1,$2,$4,$8,$5,$9,$a,$6,$10

```

```
joystick_action_high
    byte >joystick_up
    byte >joystick_down
    byte >joystick_left
    byte >joystick_right
    byte >joystick_up_left
    byte >joystick_up_right
    byte >joystick_down_right
    byte >joystick_down_left
    byte >joystick_fire
```

```
joystick_action_low
    byte <joystick_up
    byte <joystick_down
    byte <joystick_left
    byte <joystick_right
    byte <joystick_up_left
    byte <joystick_up_right
    byte <joystick_down_right
    byte <joystick_down_left
    byte <joystick_fire
```

```
joystick_up
    lda #2
    sta $d020
    dec $d001
    rts
```

```
joystick_down
    lda #7
    sta $d020
    inc $d001
    ldx #$27
```

```
loop1
    LDA $1296,X
    STA $4000,X
    LDA $12BE,X
    STA $4028,X
    LDA $12E6,X
    STA $CD40,X
    LDA $130E,X
    STA $CD68,X
    LDA $1336,X
    STA $CD90,X
    LDA $135E,X
    STA $CDB8,X
```

```
LDA $1386,X
STA $CE08,X
LDA $13AE,X
STA $CE58,X
LDA $13D6,X
STA $CE80,X
DEX
BPL loop1
;RTS
```

```
rts
```

```
joystick_left
```

```
lda #4
sta $d020
dec $d000
rts
```

```
joystick_right
```

```
lda #8
sta $d020
inc $d000
rts
```

```
joystick_up_left
```

```
lda #1
sta $d020
dec $d000
dec $d001
rts
```

```
joystick_up_right
```

```
lda #5
sta $d020
inc $d000
dec $d001
rts
```

```
joystick_down_left
```

```
lda #6
sta $d020
dec $d000
inc $d001
rts
```

```
joystick_down_right
    lda #3
    sta $d020
    inc $d000
    inc $d001
    rts
```

```
joystick_fire
    inc $d020
    rts
```

*; copy from screenone to screentwo, left one column
(line/character)*

```
S1S2L   LDX #$00
@loop   LDA screenone+$1,X
        STA screentwo,X
        LDA screenone+$29,X
        STA screentwo+$28,X
        LDA screenone+$51,X
        STA screentwo+$50,X
        LDA screenone+$79,X
        STA screentwo+$78,X
        LDA screenone+$A1,X
        STA screentwo+$A0,X
        LDA screenone+$C9,X
        STA screentwo+$C8,X
        LDA screenone+$F1,X
        STA screentwo+$F0,X
        LDA screenone+$119,X
        STA screentwo+$118,X
        LDA screenone+$141,X
        STA screentwo+$140,X
        LDA screenone+$169,X
        STA screentwo+$168,X
        LDA screenone+$191,X
        STA screentwo+$190,X
        LDA screenone+$1B9,X
        STA screentwo+$1B8,X
        LDA screenone+$1E1,X
        STA screentwo+$1E0,X
        LDA screenone+$209,X
        STA screentwo+$208,X
        LDA screenone+$231,X
        STA screentwo+$230,X
        LDA screenone+$259,X
        STA screentwo+$258,X
```

```

LDA screenone+$281,X
STA screentwo+$280,X
LDA screenone+$2A9,X
STA screentwo+$2A8,X
LDA screenone+$2D1,X
STA screentwo+$2D0,X
LDA screenone+$2F9,X
STA screentwo+$2F8,X
LDA screenone+$321,X
STA screentwo+$320,X
INX
CPX #26
BEQ @end
JMP @loop
@end RTS

```

*; copy from screentwo to screenone, left one column
(line/character)*

```

S2S1L LDX #00
@loop LDA screentwo+$01,x
      STA screenone,x
      LDA screentwo+$29,x
      STA screenone+$28,x
      LDA screentwo+$51,x
      STA screenone+$50,x
      LDA screentwo+$79,x
      STA screenone+$78,x
      LDA screentwo+$A1,x
      STA screenone+$a0,x
      LDA screentwo+$C9,x
      STA screenone+$c8,x
      LDA screentwo+$F1,x
      STA screenone+$f0,x
      LDA screentwo+$119,x
      STA screenone+$118,x
      LDA screentwo+$141,x
      STA screenone+$140,x
      LDA screentwo+$169,x
      STA screenone+$168,x
      LDA screentwo+$191,x
      STA screenone+$190,x
      LDA screentwo+$1b9,x
      STA screenone+$1b8,x
      LDA screentwo+$1e1,x
      STA screenone+$1e0,x
      LDA screentwo+$209,x

```

```

        STA screenone+$208,x
        LDA screentwo+$231,x
        STA screenone+$230,x
        LDA screentwo+$259,x
        STA screenone+$258,x
        LDA screentwo+$281,x
        STA screenone+$280,x
        LDA screentwo+$2a9,x
        STA screenone+$2a8,x
        LDA screentwo+$2d1,x
        STA screenone+$2d0,x
        LDA screentwo+$2f9,x
        STA screenone+$2f8,x
        LDA screentwo+$321,x
        STA screenone+$320,x
        INX
        CPX # $26
        BEQ @end
        JMP @loop
@end    RTS

```

*; copy from screenone to screentwo, right one column
(line/character)*

```

S1S2R  LDX # $25
@loop  LDA screenone,x
        STA screentwo+$01,x
        LDA screenone+$28,x
        STA screentwo+$29,x
        LDA screenone+$50,x
        STA screentwo+$51,x
        LDA screenone+$78,x
        STA screentwo+$79,x
        LDA screenone+$a0,x
        STA screentwo+$A1,x
        LDA screenone+$c8,x
        STA screentwo+$C9,x
        LDA screenone+$f0,x
        STA screentwo+$F1,x
        LDA screenone+$118,x
        STA screentwo+$119,x
        LDA screenone+$140,x
        STA screentwo+$141,x
        LDA screenone+$168,x
        STA screentwo+$169,x
        LDA screenone+$190,x
        STA screentwo+$191,x

```

```

LDA screenone+$1b8,x
STA screentwo+$1b9,x
LDA screenone+$1e0,x
STA screentwo+$1e1,x
LDA screenone+$208,x
STA screentwo+$209,x
LDA screenone+$230,x
STA screentwo+$231,x
LDA screenone+$258,x
STA screentwo+$259,x
LDA screenone+$280,x
STA screentwo+$281,x
LDA screenone+$2a8,x
STA screentwo+$2a9,x
LDA screenone+$2d0,x
STA screentwo+$2d1,x
LDA screenone+$2f8,x
STA screentwo+$2f9,x
LDA screenone+$320,x
STA screentwo+$321,x
DEX
BMI @end
JMP @loop
@end RTS

```

*; copy from screentwo to screenone, right one column
(line/character)*

```

S2S1R LDX #25
@loop LDA screentwo,x
      STA screenone+$1,x
      LDA screentwo+$28,x
      STA screenone+$29,x
      LDA screentwo+$50,x
      STA screenone+$51,x
      LDA screentwo+$78,x
      STA screenone+$79,x
      LDA screentwo+$a0,x
      STA screenone+$a1,x
      LDA screentwo+$c8,x
      STA screenone+$c9,x
      LDA screentwo+$f0,x
      STA screenone+$f1,x
      LDA screentwo+$118,x
      STA screenone+$119,x
      LDA screentwo+$140,x
      STA screenone+$141,x

```



```

LDA screentwo+$168,x
STA screenone+$169,x
LDA screentwo+$190,x
STA screenone+$191,x
LDA screentwo+$1b8,x
STA screenone+$1b9,x
LDA screentwo+$1e0,x
STA screenone+$1e1,x
LDA screentwo+$208,x
STA screenone+$209,x
LDA screentwo+$230,x
STA screenone+$231,x
LDA screentwo+$258,x
STA screenone+$259,x
LDA screentwo+$280,x
STA screenone+$281,x
LDA screentwo+$2a8,x
STA screenone+$2a9,x
LDA screentwo+$2d0,x
STA screenone+$2d1,x
LDA screentwo+$2f8,x
STA screenone+$2f9,x
LDA screentwo+$320,x
STA screenone+$321,x
DEX
BMI @end
JMP @loop
@end RTS

; copy from screenone to screentwo, up one row (line/character)

S1S2U LDX #$00
@loop LDA screenone+$28,x
      STA screentwo,x
      LDA screenone+$50,x
      STA screentwo+$28,x
      LDA screenone+$78,x
      STA screentwo+$50,x
      LDA screenone+$a0,x
      STA screentwo+$78,x
      LDA screenone+$c8,x
      STA screentwo+$a0,x
      LDA screenone+$f0,x
      STA screentwo+$c8,x
      LDA screenone+$118,x
      STA screentwo+$f0,x
      LDA screenone+$140,x

```

```

STA screentwo+$118,x
LDA screenone+$168,x
STA screentwo+$140,x
LDA screenone+$190,x
STA screentwo+$168,x
LDA screenone+$1b8,x
STA screentwo+$190,x
LDA screenone+$1e0,x
STA screentwo+$1b8,x
LDA screenone+$208,x
STA screentwo+$1e0,x
LDA screenone+$230,x
STA screentwo+$208,x
LDA screenone+$258,x
STA screentwo+$230,x
LDA screenone+$280,x
STA screentwo+$258,x
LDA screenone+$2a8,x
STA screentwo+$280,x
LDA screenone+$2d0,x
STA screentwo+$2a8,x
LDA screenone+$2f8,x
STA screentwo+$2d0,x
LDA screenone+$320,x
STA screentwo+$2f8,x
INX
CPX #27
BEQ @end
JMP @loop
@end
RTS

```

; copy from screentwo to screenone, up one row (line/character)

```

S2S1U   LDX #00
@loop   LDA screentwo+$28,x
        STA screenone,x
        LDA screentwo+$50,x
        STA screenone+$28,x
        LDA screentwo+$78,x
        STA screenone+$50,x
        LDA screentwo+$a0,x
        STA screenone+$78,x
        LDA screentwo+$c8,x
        STA screenone+$a0,x
        LDA screentwo+$f0,x
        STA screenone+$c8,x
        LDA screentwo+$118,x

```

```

STA screenone+$f0,x
LDA screentwo+$140,x
STA screenone+$118,x
LDA screentwo+$168,x
STA screenone+$140,x
LDA screentwo+$190,x
STA screenone+$168,x
LDA screentwo+$1b8,x
STA screenone+$190,x
LDA screentwo+$1e0,x
STA screenone+$1b8,x
LDA screentwo+$208,x
STA screenone+$1e0,x
LDA screentwo+$230,x
STA screenone+$208,x
LDA screentwo+$258,x
STA screenone+$230,x
LDA screentwo+$280,x
STA screenone+$258,x
LDA screentwo+$2a8,x
STA screenone+$280,x
LDA screentwo+$2d0,x
STA screenone+$2a8,x
LDA screentwo+$2f8,x
STA screenone+$2d0,x
LDA screentwo+$320,x
STA screenone+$2f8,x
INX
CPX # $27
BEQ @end
JMP @loop
@end RTS

; copy from screenone to screentwo, down one row
(line/character)

S1S2D LDX # $26
@loop LDA screenone+$2f8,x
STA screentwo+$320,x
LDA screenone+$2d0,x
STA screentwo+$2f8,x
LDA screenone+$2a8,x
STA screentwo+$2d0,x
LDA screenone+$280,x
STA screentwo+$2a8,x
LDA screenone+$258,x
STA screentwo+$280,x

```

```

LDA screenone+$230,x
STA screentwo+$258,x
LDA screenone+$208,x
STA screentwo+$230,x
LDA screenone+$1e0,x
STA screentwo+$208,x
LDA screenone+$1b8,x
STA screentwo+$1e0,x
LDA screenone+$190,x
STA screentwo+$1b8,x
LDA screenone+$168,x
STA screentwo+$190,x
LDA screenone+$140,x
STA screentwo+$168,x
LDA screenone+$118,x
STA screentwo+$140,x
LDA screenone+$f0,x
STA screentwo+$118,x
LDA screenone+$c8,x
STA screentwo+$f0,x
LDA screenone+$a0,x
STA screentwo+$c8,x
LDA screenone+$78,x
STA screentwo+$a0,x
LDA screenone+$50,x
STA screentwo+$78,x
LDA screenone+$28,x
STA screentwo+$50,x
LDA screenone,x
STA screentwo+$28,x
DEX
BMI @end
JMP @loop
@end
RTS

; copy from screentwo to screenone, down one row
(line/character)

S2S1D LDX #26
@loop LDA screentwo+$2f8,x
STA screenone+$320,x
LDA screentwo+$2d0,x
STA screenone+$2f8,x
LDA screentwo+$2a8,x
STA screenone+$2d0,x
LDA screentwo+$280,x
STA screenone+$2a8,x

```

```

LDA screentwo+$258,x
STA screenone+$280,x
LDA screentwo+$230,x
STA screenone+$258,x
LDA screentwo+$208,x
STA screenone+$230,x
LDA screentwo+$1e0,x
STA screenone+$208,x
LDA screentwo+$1b8,x
STA screenone+$1e0,x
LDA screentwo+$190,x
STA screenone+$1b8,x
LDA screentwo+$168,x
STA screenone+$190,x
LDA screentwo+$140,x
STA screenone+$168,x
LDA screentwo+$118,x
STA screenone+$140,x
LDA screentwo+$f0,x
STA screenone+$118,x
LDA screentwo+$c8,x
STA screenone+$f0,x
LDA screentwo+$a0,x
STA screenone+$c8,x
LDA screentwo+$78,x
STA screenone+$a0,x
LDA screentwo+$50,x
STA screenone+$78,x
LDA screentwo+$28,x
STA screenone+$50,x
LDA screentwo,x
STA screenone+$28,x
DEX
BMI @end
JMP @loop
@end RTS

```

; copy from screenone to screentwo, up one row and left one column (line/character)

```

S1S2UL LDX #$00
@loop LDA screenone+$29,x
      STA screentwo,x
      LDA screenone+$51,x
      STA screentwo+$28,x
      LDA screenone+$79,x
      STA screentwo+$50,x

```

```

LDA screenone+$a1,x
STA screentwo+$78,x
LDA screenone+$c9,x
STA screentwo+$a0,x
LDA screenone+$f1,x
STA screentwo+$c8,x
LDA screenone+$119,x
STA screentwo+$f0,x
LDA screenone+$141,x
STA screentwo+$118,x
LDA screenone+$169,x
STA screentwo+$140,x
LDA screenone+$191,x
STA screentwo+$168,x
LDA screenone+$1b9,x
STA screentwo+$190,x
LDA screenone+$1e1,x
STA screentwo+$1b8,x
LDA screenone+$209,x
STA screentwo+$1e0,x
LDA screenone+$231,x
STA screentwo+$208,x
LDA screenone+$259,x
STA screentwo+$230,x
LDA screenone+$281,x
STA screentwo+$258,x
LDA screenone+$2a9,x
STA screentwo+$280,x
LDA screenone+$2d1,x
STA screentwo+$2a8,x
LDA screenone+$2f9,x
STA screentwo+$2d0,x
LDA screenone+$321,x
STA screentwo+$2f8,x
INX
CPX #26
BEQ @end
JMP @loop
@end RTS

```

; copy from screentwo to screenone, up one row and left one column (line/character)

```

S2S1UL LDX #00
@loop LDA screentwo+$29,x
      STA screenone,x
      LDA screentwo+$51,x

```

```

STA screenone+$28,x
LDA screentwo+$79,x
STA screenone+$50,x
LDA screentwo+$a1,x
STA screenone+$78,x
LDA screentwo+$c9,x
STA screenone+$a0,x
LDA screentwo+$f1,x
STA screenone+$c8,x
LDA screentwo+$119,x
STA screenone+$f0,x
LDA screentwo+$141,x
STA screenone+$118,x
LDA screentwo+$169,x
STA screenone+$140,x
LDA screentwo+$191,x
STA screenone+$168,x
LDA screentwo+$1b9,x
STA screenone+$190,x
LDA screentwo+$1e1,x
STA screenone+$1b8,x
LDA screentwo+$209,x
STA screenone+$1e0,x
LDA screentwo+$231,x
STA screenone+$208,x
LDA screentwo+$259,x
STA screenone+$230,x
LDA screentwo+$281,x
STA screenone+$258,x
LDA screentwo+$2a9,x
STA screenone+$280,x
LDA screentwo+$2d1,x
STA screenone+$2a8,x
LDA screentwo+$2f9,x
STA screenone+$2d0,x
LDA screentwo+$321,x
STA screenone+$2f8,x
INX
CPX #26
BEQ @end
JMP @loop
@end
RTS

```

; copy from screenone to screentwo, up one row and right one column (line/character)

```
S1S2UR LDX #25
```

```

@loop    LDA screenone+$28,x
         STA screentwo+$01,x
         LDA screenone+$50,x
         STA screentwo+$29,x
         LDA screenone+$78,x
         STA screentwo+$51,x
         LDA screenone+$a0,x
         STA screentwo+$79,x
         LDA screenone+$c8,x
         STA screentwo+$A1,x
         LDA screenone+$f0,x
         STA screentwo+$C9,x
         LDA screenone+$118,x
         STA screentwo+$F1,x
         LDA screenone+$140,x
         STA screentwo+$119,x
         LDA screenone+$168,x
         STA screentwo+$141,x
         LDA screenone+$190,x
         STA screentwo+$169,x
         LDA screenone+$1b8,x
         STA screentwo+$191,x
         LDA screenone+$1e0,x
         STA screentwo+$1b9,x
         LDA screenone+$208,x
         STA screentwo+$1e1,x
         LDA screenone+$230,x
         STA screentwo+$209,x
         LDA screenone+$258,x
         STA screentwo+$231,x
         LDA screenone+$280,x
         STA screentwo+$259,x
         LDA screenone+$2a8,x
         STA screentwo+$281,x
         LDA screenone+$2d0,x
         STA screentwo+$2a9,x
         LDA screenone+$2f8,x
         STA screentwo+$2d1,x
         LDA screenone+$320,x
         STA screentwo+$2f9,x
         DEX
         BMI @end
         JMP @loop
@end     RTS

```

; copy from screentwo to screenone, up one row and right one column (line/character)


```

S2S1UR  LDX  #$25
@loop   LDA  screentwo+$28,x
        STA  screenone+$1,x
        LDA  screentwo+$50,x
        STA  screenone+$29,x
        LDA  screentwo+$78,x
        STA  screenone+$51,x
        LDA  screentwo+$a0,x
        STA  screenone+$79,x
        LDA  screentwo+$c8,x
        STA  screenone+$a1,x
        LDA  screentwo+$f0,x
        STA  screenone+$c9,x
        LDA  screentwo+$118,x
        STA  screenone+$f1,x
        LDA  screentwo+$140,x
        STA  screenone+$119,x
        LDA  screentwo+$168,x
        STA  screenone+$141,x
        LDA  screentwo+$190,x
        STA  screenone+$169,x
        LDA  screentwo+$1b8,x
        STA  screenone+$191,x
        LDA  screentwo+$1e0,x
        STA  screenone+$1b9,x
        LDA  screentwo+$208,x
        STA  screenone+$1e1,x
        LDA  screentwo+$230,x
        STA  screenone+$209,x
        LDA  screentwo+$258,x
        STA  screenone+$231,x
        LDA  screentwo+$280,x
        STA  screenone+$259,x
        LDA  screentwo+$2a8,x
        STA  screenone+$281,x
        LDA  screentwo+$2d0,x
        STA  screenone+$2a9,x
        LDA  screentwo+$2f8,x
        STA  screenone+$2d1,x
        LDA  screentwo+$320,x
        STA  screenone+$2f9,x
        DEX
        BMI  @end
        JMP  @loop
@end    RTS

```

; copy from screenone to screentwo, down one row and right one column (line/character)

```
S1S2DR  LDX #25
@loop   LDA screenone+2f8,x
        STA screentwo+321,x
        LDA screenone+2d0,x
        STA screentwo+2f9,x
        LDA screenone+2a8,x
        STA screentwo+2d1,x
        LDA screenone+280,x
        STA screentwo+2a9,x
        LDA screenone+258,x
        STA screentwo+281,x
        LDA screenone+230,x
        STA screentwo+259,x
        LDA screenone+208,x
        STA screentwo+231,x
        LDA screenone+1e0,x
        STA screentwo+209,x
        LDA screenone+1b8,x
        STA screentwo+1e1,x
        LDA screenone+190,x
        STA screentwo+1b9,x
        LDA screenone+168,x
        STA screentwo+191,x
        LDA screenone+140,x
        STA screentwo+169,x
        LDA screenone+118,x
        STA screentwo+141,x
        LDA screenone+f0,x
        STA screentwo+119,x
        LDA screenone+c8,x
        STA screentwo+f1,x
        LDA screenone+a0,x
        STA screentwo+c9,x
        LDA screenone+78,x
        STA screentwo+a1,x
        LDA screenone+50,x
        STA screentwo+79,x
        LDA screenone+28,x
        STA screentwo+51,x
        LDA screenone,x
        STA screentwo+29,x
        DEX
        BMI @end
        JMP @loop
```

@end RTS

; copy from screentwo to screenone, down one row and right one column (line/character)

```
S2S1DR   LDX  #$25
@loop    LDA  screentwo+$2f8,x
          STA  screenone+$321,x
          LDA  screentwo+$2d0,x
          STA  screenone+$2f9,x
          LDA  screentwo+$2a8,x
          STA  screenone+$2d1,x
          LDA  screentwo+$280,x
          STA  screenone+$2a9,x
          LDA  screentwo+$258,x
          STA  screenone+$281,x
          LDA  screentwo+$230,x
          STA  screenone+$259,x
          LDA  screentwo+$208,x
          STA  screenone+$231,x
          LDA  screentwo+$1e0,x
          STA  screenone+$209,x
          LDA  screentwo+$1b8,x
          STA  screenone+$1e1,x
          LDA  screentwo+$190,x
          STA  screenone+$1b9,x
          LDA  screentwo+$168,x
          STA  screenone+$191,x
          LDA  screentwo+$140,x
          STA  screenone+$169,x
          LDA  screentwo+$118,x
          STA  screenone+$141,x
          LDA  screentwo+$f0,x
          STA  screenone+$119,x
          LDA  screentwo+$c8,x
          STA  screenone+$f1,x
          LDA  screentwo+$a0,x
          STA  screenone+$c9,x
          LDA  screentwo+$78,x
          STA  screenone+$a1,x
          LDA  screentwo+$50,x
          STA  screenone+$79,x
          LDA  screentwo+$28,x
          STA  screenone+$51,x
          LDA  screentwo,x
          STA  screenone+$29,x
          DEX
```

```
        BMI @end
        JMP @loop
@end    RTS
```

; copy from screenone to screentwo, down one row and left one column (line/character)

```
S1S2DL LDX #$00
@loop  LDA screenone+$2f9,x
        STA screentwo+$320,x
        LDA screenone+$2d1,x
        STA screentwo+$2f8,x
        LDA screenone+$2a9,x
        STA screentwo+$2d0,x
        LDA screenone+$281,x
        STA screentwo+$2a8,x
        LDA screenone+$259,x
        STA screentwo+$280,x
        LDA screenone+$231,x
        STA screentwo+$258,x
        LDA screenone+$209,x
        STA screentwo+$230,x
        LDA screenone+$1e1,x
        STA screentwo+$208,x
        LDA screenone+$1b9,x
        STA screentwo+$1e0,x
        LDA screenone+$191,x
        STA screentwo+$1b8,x
        LDA screenone+$169,x
        STA screentwo+$190,x
        LDA screenone+$141,x
        STA screentwo+$168,x
        LDA screenone+$119,x
        STA screentwo+$140,x
        LDA screenone+$f1,x
        STA screentwo+$118,x
        LDA screenone+$c9,x
        STA screentwo+$f0,x
        LDA screenone+$a1,x
        STA screentwo+$c8,x
        LDA screenone+$79,x
        STA screentwo+$a0,x
        LDA screenone+$51,x
        STA screentwo+$78,x
        LDA screenone+$29,x
        STA screentwo+$50,x
        LDA screenone+$1,x
```

```

        STA screentwo+$28,x
        INX
        CPX #$26
        BEQ @end
        JMP @loop
@end    RTS

```

; copy from screentwo to screenone, down one row and left one column (line/character)

```

S2S1DL LDX #$00
@loop  LDA screentwo+$2f9,x
        STA screenone+$320,x
        LDA screentwo+$2d1,x
        STA screenone+$2f8,x
        LDA screentwo+$2a9,x
        STA screenone+$2d0,x
        LDA screentwo+$281,x
        STA screenone+$2a8,x
        LDA screentwo+$259,x
        STA screenone+$280,x
        LDA screentwo+$231,x
        STA screenone+$258,x
        LDA screentwo+$209,x
        STA screenone+$230,x
        LDA screentwo+$1e1,x
        STA screenone+$208,x
        LDA screentwo+$1b9,x
        STA screenone+$1e0,x
        LDA screentwo+$191,x
        STA screenone+$1b8,x
        LDA screentwo+$169,x
        STA screenone+$190,x
        LDA screentwo+$141,x
        STA screenone+$168,x
        LDA screentwo+$119,x
        STA screenone+$140,x
        LDA screentwo+$F1,x
        STA screenone+$118,x
        LDA screentwo+$C9,x
        STA screenone+$f0,x
        LDA screentwo+$A1,x
        STA screenone+$c8,x
        LDA screentwo+$79,x
        STA screenone+$a0,x
        LDA screentwo+$51,x
        STA screenone+$78,x

```

```

        LDA screentwo+$29,x
        STA screenone+$50,x
        LDA screentwo+$01,x
        STA screenone+$28,x
        INX
        CPX #26
        BEQ @end
        JMP @loop
@end    RTS

```

```

; fill colour map from screenone using colourmap_data lookup
table
; LDX to start loop, LDY to get character, LDA to get
corresponding character colourmap_data,
; STA to store character colour to colourmap
; done in 2 loops

```

```

S1CM    LDX #26
@loop1  LDY screenone,x
        LDA colourmap_data,Y
        STA colourmap,x
        LDY screenone+$28,x
        LDA colourmap_data,Y
        STA colourmap+$28,x
        LDY screenone+$50,x
        LDA colourmap_data,Y
        STA colourmap+$50,x
        LDY screenone+$78,x
        LDA colourmap_data,Y
        STA colourmap+$78,x
        LDY screenone+$a0,x
        LDA colourmap_data,Y
        STA colourmap+$a0,x
        LDY screenone+$c8,x
        LDA colourmap_data,Y
        STA colourmap+$c8,x
        LDY screenone+$f0,x
        LDA colourmap_data,Y
        STA colourmap+$f0,x
        LDY screenone+$118,x
        LDA colourmap_data,Y
        STA colourmap+$118,x
        LDY screenone+$140,x
        LDA colourmap_data,Y
        STA colourmap+$140,x
        DEX
        BPL @loop1

```

```

LDX #$26
@loop2 LDY screenone+$168,x
LDA colourmap_data,Y
STA colourmap+$168,x
LDY screenone+$190,x
LDA colourmap_data,Y
STA colourmap+$190,x
LDY screenone+$1b8,x
LDA colourmap_data,Y
STA colourmap+$1b8,x
LDY screenone+$1e0,x
LDA colourmap_data,Y
STA colourmap+$1e0,x
LDY screenone+$208,x
LDA colourmap_data,Y
STA colourmap+$208,x
LDY screenone+$230,x
LDA colourmap_data,Y
STA colourmap+$230,x
LDY screenone+$258,x
LDA colourmap_data,Y
STA colourmap+$258,x
LDY screenone+$280,x
LDA colourmap_data,Y
STA colourmap+$280,x
LDY screenone+$2a8,x
LDA colourmap_data,Y
STA colourmap+$2a8,x
LDY screenone+$2d0,x
LDA colourmap_data,Y
STA colourmap+$2d0,x
LDY screenone+$2f8,x
LDA colourmap_data,Y
STA colourmap+$2f8,x
LDY screenone+$320,x
LDA colourmap_data,Y
STA colourmap+$320,x
DEX
BPL @loop2
RTS

```

```

; fill colour map from screentwo using colourmap_data lookup
table
; LDX to start loop, LDY to get character, LDA to get
corresponding character colourmap_data,
; STA to store character colour to colourmap
; done in 2 loops

```

```

S2CM      LDX  #$26
@loop1    LDY  screentwo,x
          LDA  colourmap_data,Y
          STA  colourmap,x
          LDY  screentwo+$28,x
          LDA  colourmap_data,Y
          STA  colourmap+$28,x
          LDY  screentwo+$50,x
          LDA  colourmap_data,Y
          STA  colourmap+$50,x
          LDY  screentwo+$78,x
          LDA  colourmap_data,Y
          STA  colourmap+$78,x
          LDY  screentwo+$a0,x
          LDA  colourmap_data,Y
          STA  colourmap+$a0,x
          LDY  screentwo+$c8,x
          LDA  colourmap_data,Y
          STA  colourmap+$c8,x
          LDY  screentwo+$f0,x
          LDA  colourmap_data,Y
          STA  colourmap+$f0,x
          LDY  screentwo+$118,x
          LDA  colourmap_data,Y
          STA  colourmap+$118,x
          LDY  screentwo+$140,x
          LDA  colourmap_data,Y
          STA  colourmap+$140,x
          DEX
          BPL  @loop1
          LDX  #$26
@loop2    LDY  screentwo+$168,x
          LDA  colourmap_data,Y
          STA  colourmap+$168,x
          LDY  screentwo+$190,x
          LDA  colourmap_data,Y
          STA  colourmap+$190,x
          LDY  screentwo+$1b8,x
          LDA  colourmap_data,Y
          STA  colourmap+$1b8,x
          LDY  screentwo+$1e0,x
          LDA  colourmap_data,Y
          STA  colourmap+$1e0,x
          LDY  screentwo+$208,x
          LDA  colourmap_data,Y
          STA  colourmap+$208,x

```


