

```

;-----
;
;   Proportional-Charset-Noter with Plasma Effect
;
;   Coding: Testicle/Payday
;   Logo: Zeitgeist/Civitas
;   Musik: Dalezy/Creators
;
;   Contact: daniel@popelganda.de
;   www.popelganda.de
;
;   For fast and easy navigation through the source,
;   make intensive use of the paragraph-function,
;   if you are using the Relaunch64 Crosseditor!
;
;   Get Relaunch64 at www.popelganda.de
;
;-----

;-----
;
;----- Paragraph @Globale Variablen@ -----
;
;-----

!to "noter.prg"

screen=$0400
pscreen=$05b8           ;screen-coordinates for plasma effect
charset=$4800           ;original-charset
bitmap=$2dc0            ;adress/offset of bitmap for prop-char, i.e. hi-res-
bmp for text
clearanz=$12            ;number * $100. shows how much of the bitmap
must be cleared.
chardest=$4000          ;prepeared charset
stext=$6000             ;adress/offset for note text
start=$c000             ;start of main code

plasma$inus=$9e00       ;plasma-data
plasma$colors=$9f00

pixelcounter=$40        ;counts, how many pixels of a char still have to
be "printed"
charcounter=$41         ;checks for new chars to be set
linecounter=$42         ;counts the lines already set
neuchar=$43
maxpage=$44            ;is set, when last page is reached
text=$50               ;text-counter in zeropage ($50/$51)
bmpoffset=$60          ;saves the offset of the bitmap in zeropage. tells,

```

```
where to print in the bitmap
z1=$52           ;variables for plasma effect
z2=$53

koalabmp=$2000   ;data for koala logo
koalascr=$4c00
koalacol=$4e00

soundinit=$1000
soundplay=$1003

TRUE=1
FALSE=0

AN=1
AUS=0

PZEILEN=14       ;number of lines of plasma effects
PSPALTEN=40      ;number of rows of plasma effects

;-----
;
;----- Paragraph @Includes@ -----
;
;-----

*= $0800
!byte $00,$0c,$08,$0a,$00,$9e,$34,$39,$31,$35,$32,$00,$00,$00,$00

*= charset
!bin "../sonstiges/prop-noter/char.bin"

*= koalabmp
!bin "../sonstiges/prop-noter/pay-bmp.bin"

*= koalascr
!bin "../sonstiges/prop-noter/pay-scr.bin"

*= koalacol
!bin "../sonstiges/prop-noter/pay-col.bin"

*= soundinit
!bin "../sonstiges/prop-noter/music.bin"
```

```

;-----
;
;   sourcecode start
;
;-----

;----- Paragraph @Start of Maiscode@ -----

*= start
!zone HAUPTSOURCE

    lda #0
    sta $d020
    sta $d021
    jsr $e536

    sei
    lda #<irq1
    sta $0314
    lda #>irq1
    sta $0315
    lda #0
    sta $dc0e
    jsr soundinit
    lda #1
    sta $d01a
    lda #$7f
    sta $dc0d
    lda #$2d
    sta $d012
    lda #$7b
    sta $d011

;-----

    ldx #0
.loop1   lda #$bf           ;hires-screen: $0b (darkgray) for textcolor,
    sta screen+$200,x       ;defined in screen-area.
    sta screen+$300,x
    inx
    bne .loop1
    ldx #$b8
    lda #$bf
.loop11  sta screen+$100,x
    inx
    bne .loop11

;----- Paragraph @Init: Koala-Logo@ -----

    ldx #0
.loopk1  lda koalascr,x

```

```
        sta $0400,x
        lda koalacol,x
        sta $d800,x
        inx
        bne .loopk1
.loopk2 lda koalascr+$100,x
        sta $0500,x
        lda koalacol+$100,x
        sta $d900,x
        inx
        cpx #104
        bne .loopk2
        ldx #79
        lda #20
.loopk3 sta $0568,x
        dex
        bpl .loopk3

;-----
;   "empty dummy screen" for fade in
;-----

        lda #20
        ldx #0
.clears  sta $0c00,x
        sta $0d00,x
        sta $0e00,x
        sta $0f00,x
        inx
        bne .clears

;-----

        jsr .clearbitmap

;-----

;-----
;   here the charset is being prepared.
;   the pixelrows of each char are not
;   stored in 8 succeeded on another byte
;   ($3000-$3008), but each at $3000, $3100,
;   $3200 to $3800. now you can more easily
;   read the single pixellines by simply storing
;   the char in the x-reg. and use
;
;   lda $3000,x
;   lda $3100,x
;   lda $3200,x
;   ...
```

```

;   lda $3700,x
;
;   for getting the 8 pixelows of a char.
;
;-----
;----- Paragraph @Prepare Charset@ -----

        ldx #0
        ldy #0
.loop4   lda charset,x
        sta chardest,y
        inx
        lda charset,x
        sta chardest+$100,y
        inx
        lda charset,x
        sta chardest+$200,y
        inx
        lda charset,x
        sta chardest+$300,y
        inx
        lda charset,x
        sta chardest+$400,y
        inx
        lda charset,x
        sta chardest+$500,y
        inx
        lda charset,x
        sta chardest+$600,y
        inx
        lda charset,x
        sta chardest+$700,y
        iny
        inx
        bne .loop4

;-----
.loop5   lda charset+$100,x
        sta chardest,y
        inx
        lda charset+$100,x
        sta chardest+$100,y
        inx
        lda charset+$100,x
        sta chardest+$200,y
        inx
        lda charset+$100,x
        sta chardest+$300,y
        inx
        lda charset+$100,x

```

```
    sta chardest+$400,y
    inx
    lda charset+$100,x
    sta chardest+$500,y
    inx
    lda charset+$100,x
    sta chardest+$600,y
    inx
    lda charset+$100,x
    sta chardest+$700,y
    iny
    inx
    bne .loop5
```

;-----

```
.loop6      lda charset+$200,x
            sta chardest,y
            inx
            lda charset+$200,x
            sta chardest+$100,y
            inx
            lda charset+$200,x
            sta chardest+$200,y
            inx
            lda charset+$200,x
            sta chardest+$300,y
            inx
            lda charset+$200,x
            sta chardest+$400,y
            inx
            lda charset+$200,x
            sta chardest+$500,y
            inx
            lda charset+$200,x
            sta chardest+$600,y
            inx
            lda charset+$200,x
            sta chardest+$700,y
            iny
            inx
            bne .loop6
```

;-----

; small break at the beginning

;-----

```
.p1      ldx #200
.p2      ldy #255
         bit $ea
         nop
```

```

    dey
    bne .p2
    dex
    bne .p1

;-----

    lda #0
    sta pixelcounter
    sta charcounter
    sta linecounter
    sta neuchar
    sta maxpage

    lda #<stext
    sta text
    lda #>stext
    sta text+1
    lda #<bitmap
    sta bmpoffset
    lda #>bitmap
    sta bmpoffset+1

    cli

;-----
;----- Paragraph @Wait for Keypress@ -----
;-----

WARTEN  lda #0
        beq WARTEN

;----- ende -----

;-----
;----- Paragraph @Read Char@ -----
;-----

.newpage  ldy #0
          lda (text),y          ;read text-char
          cmp #254
          bne .los              ;end-sign?
          lda #TRUE
          sta maxpage
          jmp .vortaste          ;if yes, finish

.los      cmp #$20                ;if space, override rol-code
          bne .next              ;and increase char- and pixelcounter
          ldx #$20                ;immediately

```

```
        jsr .bende
        jmp .space

.next      cmp #255                ;new line?
        bne .next2
        jsr .neuereihe            ;if yes, set new line and read
        jmp .newpage              ;new char

.next2     cmp #253                ;new page?
        bne .putit

        inc text
        bne .weiter
        inc text+1
.weiter    jmp .vortaste           ;if yes, goto waiting-loop

.putit     tax
        jsr .putchar              ;put char.

;-----

.space     lda neuchar             ;is a new char "started" in the bitmap?
        beq .weiter1
        lda #0                    ;if yes, reset variable
        sta neuchar
        lda bmpoffset             ;and set bitmap-vector to next char
        clc
        adc #8
        sta bmpoffset
        bcc .weiter1
        inc bmpoffset+1
.weiter1   inc text                ;textcounter to next char in the text
        lda text
        bne .weiter2
        inc text+1
.weiter2   jmp .newpage            ;and back to beginning

;-----

.vortaste  lda #AN
        sta PAUSE+1

TASTE     lda #0
        jsr plasma                 ;run plasma while in waiting-loop
        lda TASTE+1
        beq TASTE

        lda #AUS
        sta PAUSE+1                ;don't allow space-key
```



```

    lda #AUS
    sta TASTE+1           ;activate waiting-loop above

    lda maxpage           ;all pages shown?
    beq .weiter3

    lda #<stext           ;if yes, reset textvector
    sta text
    lda #>stext
    sta text+1
    lda #FALSE
    sta maxpage

.weiter3   jsr .neueseite
           jmp .newpage

;----- ende -----

;-----
;----- Paragraph @New Textrow@ -----
;-----

.neuereihe lda #<bitmap           ;first reset to original-bitmap-offsets
            sta bmpoffset
            lda #>bitmap
            sta bmpoffset+1

            ldx linecounter         ;how many lines do we have?
.matheloop inc bmpoffset+1         ;number of lines multiplied with 320
            lda bmpoffset           ;is the new offset for the bitmap
            clc
            adc #64
            sta bmpoffset
            bcc .mathe
            inc bmpoffset+1
.mathe     dex
            bpl .matheloop

            inc linecounter         ;increase linecounter

            lda #0                 ;re-initiate pixel- and charcounter-variables
            sta pixelcounter
            sta charcounter
            sta neuchar

            inc text               ;increase textcounter
            lda text
            bne .matheweiter
            inc text+1

```

```
.matheweiter      rts                ;go back

;----- ende -----

;-----
;----- Paragraph @New Textpage@ -----
;-----

.neueseite  lda #<bitmap                ;first reset to original-bitmap-offsets
            sta bmpoffset
            lda #>bitmap
            sta bmpoffset+1

            jsr .clearbitmap

            lda #0
            sta pixelcounter
            sta charcounter
            sta linecounter
            sta neuchar

            rts

;----- ende -----

;-----
;----- Paragraph @Clear Bitmap@ -----
;-----

.clearbitmap  lda #0                    ;clear bitmap
             ldx #0

.loop2
!set m=0
!do {
            sta bitmap+m*$100,x
!set m=m+1
} until m=clearanz

            inx
            bne .loop2
            ldx #0
.loopc1 lda #$bf                        ;hires-screen: $0b (darkgray) for textcolor
        sta screen+$200,x
        sta screen+$300,x
        inx
```

```

        bne .loopc1

        ldx #$b8
        lda #$bf
.loopc11  sta screen+$100,x
        inx
        bne .loopc11

        rts

;----- ende -----

;-----
;----- Paragraph @Put char@ -----
;-----

.putchar  lda chardest,x           ;get first pixelrow
          sta charbuffer+1         ;and buffer it
          lda chardest+$100,x      ;and so on...
          sta charbuffer+3
          lda chardest+$200,x
          sta charbuffer+5
          lda chardest+$300,x
          sta charbuffer+7
          lda chardest+$400,x
          sta charbuffer+9
          lda chardest+$500,x
          sta charbuffer+11
          lda chardest+$600,x
          sta charbuffer+13
          lda chardest+$700,x
          sta charbuffer+15

;-----
;----- Paragraph @Move char@ -----
;-----

        lda pixelcounter          ;load pixel-counter. this variable stores
the remaining amount of pixel
        and #7                    ;in the old char, so we know how many pixel
the new char must be moved
        eor #7
        tay
.nloop1  clc                      ;move char
        rol charbuffer+1
        rol charbuffer
        clc
        rol charbuffer+3
        rol charbuffer+2

```

```
clc
rol charbuffer+5
rol charbuffer+4
clc
rol charbuffer+7
rol charbuffer+6
clc
rol charbuffer+9
rol charbuffer+8
clc
rol charbuffer+11
rol charbuffer+10
clc
rol charbuffer+13
rol charbuffer+12
clc
rol charbuffer+15
rol charbuffer+14
dey
bpl .nloop1
```

```
ldy #0
```

```
;----- Paragraph @Char to Bitmap@ -----
```

```
lda (bmpoffset),y          ;and now copy the moved char with ORA
ora charbuffer             ;into the bitmap
sta (bmpoffset),y
iny
lda (bmpoffset),y
ora charbuffer+2
sta (bmpoffset),y
iny
lda (bmpoffset),y
ora charbuffer+4
sta (bmpoffset),y
iny
lda (bmpoffset),y
ora charbuffer+6
sta (bmpoffset),y
iny
lda (bmpoffset),y
ora charbuffer+8
sta (bmpoffset),y
iny
lda (bmpoffset),y
ora charbuffer+10
sta (bmpoffset),y
iny
```

```

        lda (bmpoffset),y
        ora charbuffer+12
        sta (bmpoffset),y
        iny
        lda (bmpoffset),y
        ora charbuffer+14
        sta (bmpoffset),y

        iny                ;from here the rest of the just copied chars
(i.e. the remaining pixels)
                        ;are being copied into the next char of the bitmap.
2x iny, because the counter
        lda charbuffer+1    ;is in pixel-row 8 at the moment, i.e. at
the "bottom". with the 2. iny the next offset
        sta (bmpoffset),y    ;starts at the top of the 1. pixelrow of
the next char in the bitmap
        iny
        lda charbuffer+3
        sta (bmpoffset),y
        iny
        lda charbuffer+5
        sta (bmpoffset),y
        iny
        lda charbuffer+7
        sta (bmpoffset),y
        iny
        lda charbuffer+9
        sta (bmpoffset),y
        iny
        lda charbuffer+11
        sta (bmpoffset),y
        iny
        lda charbuffer+13
        sta (bmpoffset),y
        iny
        lda charbuffer+15
        sta (bmpoffset),y

;----- Paragraph @End: Put char@ -----

.bende  lda pixelcounter        ;get pixelcounter
        clc
        adc chartable,x        ;add length of current char
        sta pixelcounter        ;and save it. the pixelcounter contains the
whole amount of pixels, which
                        ;already have been set in the current row.
        lda charcounter        ;update charcounter.
        clc
        adc chartable,x        ;add length of current char
        cmp #8                ;is a new char "started" in the bitmap?
        bcc .kein

```

```
        sbc #8
        sta charcounter

        lda #1                ;if yes, set remember-variable
        sta neuchar
        jmp .zurueck

.kein    sta charcounter
.zurueck ldx #17
        lda #0
.clearbuf sta charbuffer,x    ;clear buffer
        dex
        bpl .clearbuf
        rts
;----- ende -----

;-----
;
; here the chars are being set,
; which should be stored with ROL
; into the bitmap-matrix.
;
;-----

charbuffer
!fill 18,0

;----- ende -----

;-----
;
; char-table, storing the length
; of each char (measured in pixel)
;
;-----

;----- Paragraph @Chartable@ -----

chartable
!byte 7,7,7,6,7,7,6,7,7,5,6,7,5,8,7,7,7,7,7,7,7,7,8,7,7,7,5,7,5,7,7
!byte 4,4,7,8,7,7,8,5,5,5,9,7,4,7,4,8,7,7,7,7,8,7,7,7,7,7,5,4,5,6,5,7
!byte 1,7,7,7,7,7,7,7,7,3,7,7,7,8,7,7,7,7,7,7,7,7,7,7,8,7,7,7

;----- ende -----
```

```
;-----  
;  
;----- Paragraph @Irq1: Koala fadein@ -----  
;  
;-----  
  
!zone  
irq1          inc $d019  
  
          lda #$3b  
          sta $d011  
          lda #$19  
          sta $d018  
          lda #$d8  
          sta $d016  
  
.wait        lda $d012  
.zeile       cmp #$31  
          bcc .wait  
  
          lda $d011  
          ora #$40  
          ldy $d012  
          cpy $d012  
          beq *-3  
          sta $d011  
  
          inc .zeile+1  
          lda .zeile+1  
          cmp #$7c  
          bne .ende  
          lda #<irq2  
          sta $0314  
          lda #>irq2  
          sta $0315  
          lda #$2e  
          sta $d012  
  
.ende        jsr soundplay  
          jmp $ea7e  
  
;----- ende -----  
  
;-----  
;  
;----- Paragraph @Irq2: Koala Display@ -----  
;  
;-----
```

```
!zone
irq2      inc $d019

          lda #$3b
          sta $d011
          lda #$19
          sta $d018
          lda #$d8
          sta $d016

          jsr soundplay

IRQC1     lda #<irq3
          sta $0314
IRQC2     lda #>irq3
          sta $0315
          lda #$80
          sta $d012

          jmp $ea7e

;----- ende -----

;-----
;
;----- Paragraph @Irq3: Fadein Gray@ -----
;
;-----

!zone
irq3      inc $d019

.farbe    lda #0
          ldy $d012
          cpy $d012
          beq *-3
          sta $d020
          sta $d021

          lda #$1b
          sta $d011
          lda #$c8
          sta $d016
          lda #$35
          sta $d018

.wait     lda $d012
```



```
        cmp #$fe
        bcc .wait

        ldx #7
.warten ldy $d012
        cpy $d012
        beq *-3
        nop:nop:nop
        dex
        bne .warten

        lda #0
        ldy $d012
        cpy $d012
        beq *-3
        sta $d020
        sta $d021

.pause  dec .pause+1
        lda #4
        bne .weiter
        lda #2
        sta .pause+1
.cnt    ldx #0
        lda .fcol,x
        sta .farbe+1
        inc .cnt+1
        lda .cnt+1
        cmp #8
        bne .weiter
        lda #<irq4
        sta IRQC1+1
        lda #>irq4
        sta IRQC2+1

        ldx #79
        lda #$ff
.loop   sta $0568,x
        dex
        bpl .loop

        lda #AN
        sta WARTEN+1

.weiter lda #<irq2
        sta $0314
        lda #>irq2
        sta $0315
        lda #$2e
        sta $d012
```

```
        jmp $ea7e

.fcol
!byte $00,$0b,$09,$02,$08,$0a,$0f,$0f

;----- ende -----

;-----
;
;----- Paragraph @Irq4: Fadein Gray finished@ -----
;
;-----

!zone
irq4      inc $d019

        lda #15
        ldy $d012
        cpy $d012
        beq *-3
        sta $d020
        sta $d021

        lda #$3b
        sta $d011
        lda #$c8
        sta $d016
        lda #$19
        sta $d018

.wait2    lda $d012
        cmp #$fe
        bcc .wait2

        ldx #7
.warten  ldy $d012
        cpy $d012
        beq *-3
        nop:nop:nop
        dex
        bne .warten

        lda #0
        ldy $d012
        cpy $d012
        beq *-3
        sta $d020
```

```

        sta $d021

        dec .wait+1
.wait   lda #4
        bne .ende
        lda #4
        sta .wait+1

PAUSE   lda #1
        beq .ende

        lda $dc01
        cmp #$ef
        bne .ende

        lda #AN
        sta TASTE+1

.ende   lda #<irq2
        sta $0314
        lda #>irq2
        sta $0315
        lda #30
        sta $d012

        jmp $ea7e

;----- ende -----

;-----
;
;----- Paragraph @Sub-Route: Plasma-Effect@ -----
;
;-----

!zone
plasma lda .pcounter
        sta .sincnt+1

.zeile  ldy #0           ;row-counter
.sincnt ldx #0         ;sinus-counter

        jsr .effekt     ;call plasma sub routine

        inc .sincnt+1
        inc .zeile+1
        lda .zeile+1
        cmp #PZEILEN   ;all 14 rows done?
        bne .zeile

```

```
    lda .pcounter
    clc:adc #3
    sta .pcounter

    inc .pcounter+1

    lda #0
    sta .zeile+1
    rts

.effekt lda .pcounter+1
        sta .weiter+1
        lda #<pscreen
        sta z1
        lda #>pscreen
        sta z2

        ;set row of the color ram

.loop1   cpy #0                ;in y we stored the current line.
        beq .weiter           ;if equal zero, start plasma
        lda z1                ;else add 40 chars to row-counter...
        clc:adc #40
        sta z1
        bcc .loop2
        inc z2
.loop2   dey                  ;...repeat this until counter for color ram
        bne .loop1           ;is set to current row.

        ;here we go!

.weiter ldy #0
        lda plasmasinus,x
        clc
        adc plasmasinus,y
        adc .pcounter
        adc .pcounter+1
        and #$7f
        lsr
        tay
        lda plasmacolors,y

.charcnt ldy #0                ;counter for columns (i.e. width of plasma in
chars)
        sta (z1),y
        inc .weiter+1
        inc .charcnt+1
        lda .charcnt+1
        cmp #PSPALTEN        ;number of columns
```

```

        bne .weiter

        lda #0
        sta .charcnt+1
        rts

.pcounter
!byte 0,0

;----- ende -----

;-----
;
;----- Paragraph @Plasma-Colors@ -----
;
;-----

*= plasmacolors

!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf
!byte $bf,$9f,$2f,$cf,$af,$ff,$7f,$1f
!byte $7f,$ff,$af,$cf,$2f,$9f,$bf,$bf
!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf
!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf
!byte $bf,$6f,$4f,$ef,$5f,$3f,$df,$1f
!byte $df,$3f,$5f,$ef,$4f,$6f,$bf,$bf
!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf

!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf
!byte $bf,$9f,$2f,$cf,$af,$7f,$1f,$7f
!byte $af,$cf,$2f,$9f,$bf,$bf,$bf,$bf
!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf
!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf
!byte $bf,$6f,$4f,$ef,$5f,$3f,$df,$1f
!byte $df,$3f,$5f,$ef,$4f,$6f,$bf,$bf
!byte $bf,$bf,$bf,$bf,$bf,$bf,$bf,$bf

;-----
;
;----- Paragraph @Plasma-Sinus@ -----
;
;-----

*= plasmasinus

!byte $1f,$1f,$1f,$1f,$1f,$20,$20,$21
!byte $21,$22,$23,$25,$26,$27,$29,$2a

```

```
!byte $2b,$2d,$2e,$2f,$30,$30,$30,$30
!byte $30,$2f,$2e,$2c,$2a,$28,$25,$22
!byte $1f,$1c,$18,$15,$12,$0e,$0b,$09
!byte $06,$04,$02,$01,$00,$00,$00,$00
!byte $01,$02,$04,$06,$08,$0b,$0e,$10
!byte $13,$15,$18,$1a,$1c,$1d,$1e,$1f
!byte $1f,$1f,$1e,$1d,$1c,$1a,$18,$15
!byte $13,$10,$0e,$0b,$08,$06,$04,$02
!byte $01,$00,$00,$00,$00,$01,$02,$04
!byte $06,$09,$0b,$0e,$12,$15,$18,$1c
!byte $1f,$22,$25,$28,$2a,$2c,$2e,$2f
!byte $30,$30,$30,$30,$30,$2f,$2e,$2d
!byte $2b,$2a,$29,$27,$26,$25,$23,$22
!byte $21,$21,$20,$20,$1f,$1f,$1f,$1f
!byte $1f,$1f,$1f,$1f,$1f,$1e,$1e,$1d
!byte $1d,$1c,$1b,$19,$18,$17,$15,$14
!byte $13,$11,$10,$0f,$0e,$0e,$0e,$0e
!byte $0e,$0f,$10,$12,$14,$16,$19,$1c
!byte $1f,$22,$26,$29,$2c,$30,$33,$35
!byte $38,$3a,$3c,$3d,$3e,$3e,$3e,$3e
!byte $3d,$3c,$3a,$38,$36,$33,$30,$2e
!byte $2b,$29,$26,$24,$22,$21,$20,$1f
!byte $1f,$1f,$20,$21,$22,$24,$26,$29
!byte $2b,$2e,$30,$33,$36,$38,$3a,$3c
!byte $3d,$3e,$3e,$3e,$3e,$3d,$3c,$3a
!byte $38,$35,$33,$30,$2c,$29,$26,$22
!byte $1f,$1c,$19,$16,$14,$12,$10,$0f
!byte $0e,$0e,$0e,$0e,$0e,$0f,$10,$11
!byte $13,$14,$15,$17,$18,$19,$1b,$1c
!byte $1d,$1d,$1e,$1e,$1f,$1f,$1f,$1f
```

```
;-----
;
; scrolltext
;
```

```
;----- Paragraph @Text@ -----
```

```
*= stext
;ä = ^
;ö = \h
;ü = _
```

```
;255 = new line
;253 = new page
;254 = end
```

```
!ct scr

!tx "Welcome, ladies and gentlemen, to the short note!"
!by 255, 255
!tx "First of all the credits for this note:"
!by 255
!tx "-----"
!by 255,255
!tx "Coding by Testicle/Payday"
!by 255
!tx "Logo by Zeitgeist/Civitas"
!by 255
!tx "Music by Dalezy/Creators"
!by 255,255
!tx "This charset is a slight modified version of the"
!by 255
!tx "original C64-charset. Modifications done by me,"
!by 255
!tx "Testicle."
!by 253

!tx "I think, the credits for the demo Oedipus Complex"
!by 255
!tx "are not necessary to mention, because you can (or"
!by 255
!tx "must, hehe), read them in the scrollers of each"
!by 255
!tx "part."
!by 255,255
!tx "All I want to say, is that all coding was done by"
!by 255
!tx "Testicle/Payday and all graphics where made by"
!by 255
!tx "Fabu/Payday. I think, that's important! ;-)"
!by 255,255
!tx "Now some words to the demo..."
!by 253

!tx "Well, the effects in this demo are quite old. Most"
!by 255
!tx "of them were done short before the Payday-break"
!by 255
!tx "in 1995. But it was a lot of work to me and that's"
!by 255
!tx "why I wanted to release these parts as a demo."
!by 255,255
!tx "The small competition at the forum-64.de-forum"
!by 255
!tx "gave me the motivation to complete my goal! :-)"
!by 255,255
!tx "I don't think, we in Payday will release another"
```

```
!by 255
!tx "demo, because the main-reason why we started"
!by 255
!tx "our 'comeback' was our beloved discmag..."
!by 255,255
!tx "
                                POPELGANDA!"
!by 255
!tx "(Applause)"
!by 253

!tx "I hope, we will manage to release the next issue"
!by 255
!tx "within this year, but i'm am quite confident, that"
!by 255
!tx "issue #6 of Popelganda is released in the near"
!by 255
!tx "future!"
!by 255,255
!tx "Now to some more future-plans of Payday. Beside"
!by 255
!tx "Popelganda we are also developing a crossdevelop-"
!by 255
!tx "ment-tool for windows."
!by 255,255
!tx "It's a very useful text-editor for ml-coding and"
!by 255
!tx "can be used with many common assemblers like"
!by 255
!tx "ACME, DAsm, C64Asm or Ca65."
!by 253

!tx "Now you might ask yourself, where to get all this"
!by 255
!tx "funky stuff? simply take a look at our website!"
!by 255,255
!tx "
                                http://www.popelganda.de"
!by 255,255
!tx "Our homepage is currently being redesigned and the"
!by 255
!tx "editor is not ready for download yet. I'll announce"
!by 255
!tx "new releases at the well-known scene-news-sites,"
!by 255
!tx "ofcourse."
!by 253

!tx "In case you want to contact Payday:"
!by 255
!tx "-----"
```



```
!by 255,255
!tx "Testicle/Payday"
!by 255
!tx "tte@popelganda.de"
!by 255,255
!tx "Fabu/Payday"
!by 255
!tx "fabu@popelganda.de"
!by 255,255,255,255
!tx "Thank you for your attention!"
!by 255,255
!tx "
!by 254
```

Testicle/Payday."

From:
<https://codebase64.org/> - Codebase 64 wiki

Permanent link:
https://codebase64.org/doku.php?id=base:proportional-charset-noter_with_plasma-effect

Last update: **2015-04-17 04:33**

