

Ocean/Imagine Transfer - v1

```

;-----
;Ocean/Imagine Transfer.
;For loader type 1 only.
;
;Fungus/Nostalgia 2005
;-----

        *= $0810

mainmsg = $0b40      ;version msg
msg1    = $0bb8      ;skipping files
msg2    = $0bc8      ;loading file
msg3    = $0bd8      ;rewind to start
msg4    = $0c00      ;saving file
msg5    = $0c10      ;transfer complete
msg6    = $0c38      ;rewind to continue

;-----
;setup, get name, and skip loader

        lda $ba
        sta dvn
        lda #$00      ;init file count
        sta filecnt
        sta blkcnt1  ;init block count
        sta blkcnt2
        sta blkcnt1+1
        sta blkcnt2+1
        lda #$2f      ;normal i/o
        sta $00
        lda #$37
        sta $01
        ldx #$00      ;black
        stx $d020
        stx $d021
        stx $d01a    ;rasters off
        stx $9d       ;msgs
        lda #$80      ;disable shift+c=
        sta $0291
        lda #$17      ;lower/upper
        sta $d018
clrscn  lda #$20      ;clr screen
        sta $0400,x
        sta $0500,x
        sta $0600,x
        sta $06e8,x
        lda #$05      ;clr color mem

```

```

        sta $d800,x
        sta $d900,x
        sta $da00,x
        sta $dae8,x
        inx
        bne clrscn
pmain   lda mainmsg,x
        sta $0400,x ;print version
        inx
        cpx #$78
        bne pmain
        ldx #$27      ;rewind to start
pmsg3   lda msg3,x
        sta $07c0,x
        dex
        bpl pmsg3
start   lda #$ef      ;wait space
        cmp $dc01
        bne *-3
        jsr clrbot
main    jsr ton      ;motor on
        jsr lod      ;load header
        jsr tof      ;motor off
        jsr namer    ;get the name
        ldx #$0f     ;print name
putname lda name,x
        sta $0460,x
        dex
        bpl putname
        jsr delay
        jsr led      ;relocate load
        jsr tof      ;motor off
        jmp load     ;transfer file

ton     lda $01      ;motor on
        and #$df
        sta $01
        rts
tof     lda $01      ;motor off
        ora #$20
        sta $01
        rts
lod     lda #$01     ;load header
        tax
        tay
        jsr $ffbba
        lda #$00
        jsr $ffbd
        lda #$00
        sta $c0
        sta $93

```

```
    jsr $f7d7
    jmp $f84f
led   lda #$00      ;relocate loader
      sta $c3      ;to $1000
      lda #$10
      sta $c4
      lda $033f    ;calc length
      sbc $033d
      tax
      lda $0340
      sbc $033e
      tay
      clc
      txa
      adc $c3      ;calc new end addy
      sta $ae
      tya
      adc $c4
      sta $af
      lda $c3      ;set new load addy
      sta $c1      ;after calcs
      sta $ac
      lda $c4
      sta $c2
      sta $ad
      jsr ton      ;motor on
      jmp $f84f    ;load file

namer ldy #$00      ;get the filename
      ldx #$00
dname lda $0341,y
      cmp #$20
      beq ok1
      cmp #$2f
      bcc der
      cmp #$5a
      bcs der
      cmp #$40
      bcs ok1
      cmp #$39
      bcs der
ok1   sta name,x
      inx
der   iny
      cpy #$10
      bne dname
      lda filecnt
      jsr getvals
      sta name+$0e
      stx name+$0f
      rts
```

```

delay      sei
           ldx #$80
           ldy #$02
yaled      bit $d011
           bpl *-3
           bit $d011
           bmi *-3
           dex
           bne yaled
           dey
           bne yaled
           cli
           rts
getvals    pha
           and #$0f
           jsr conv1
           tax
           pla
           lsr a
           lsr a
           lsr a
           lsr a
conv1      ora #$30
           cmp #$3a
           bcc conv2
           sbc #$39
           ora #$40
conv2      rts

name       .text "
dvn        .byte $00

;-----
;load file

buffer     = $0d00

load
           lda #<buffer;reset load buffer
           sta $08
           lda #>buffer
           sta $09
           lda #$00      ;store 1st block
           sta mod1+1    ;load addy
           sta mod2+1
           sei
           ldy #$00
           sty $d020
           sty $d021
           sty $04      ;low byte of fetch
           lda #$e0      ;$01e0 threshold
    
```

```

        sta $dd04
        lda #$01
        sta $dd05
        lda #$19
        sta $dd0e
        lda #$07
        sta $01
reset   lda #$ff      ;wait for stream
        sta $03      ;of 00 pulses
sync1  jsr getbit
        bcs reset
        dec $03
        bne sync1
sync2  jsr getbit    ;wait out pilot
        bcc sync2   ;(#$00)
        lda blkcnt2+1
        bne skipem1
        lda blkcnt2 ;check if skipping
        beq readfile;needed
skipem1 jsr skipblks
readfile
        ldx #$0f    ;print loading
pmsg2  lda msg2,x
        sta $0450,x
        dex
        bpl pmsg2
        ldy #$00
header jsr getbyte   ;skips control byte
        jsr getbyte ;load addy hi byte
        sta $05
mod1   ldx #$00
        bne skip1
        sta $07    ;1st in chain
        inc mod1+1 ;only
skip1  cmp #$00
        beq eof    ;last file
mod2   ldx #$00    ;skip check on
        beq nochk  ;1st block
        sec
        sbc $06    ;detect eof
        cmp #$01   ;in chain
        bne eof
nochk  inc mod2+1    ;enable eof check
lblock jsr getbyte
        inc $01
        sta ($08),y ;store byte
        dec $01
        inc $08    ;buffer low addy
        inc $04    ;block count low
        bne lblock
        inc blkcnt1 ;inc block count

```

```

        bne hblock1
        inc blkcnt1+1
hblock1 lda $05      ;block addy
        sta $06      ;temp save
        inc $09      ;buffer addy hi
        bne header
eof      lda blkcnt1
        sta blkcnt2 ;save # blocks to
        ;skip
        lda blkcnt1+1
        sta blkcnt2+1
        lda #$00
        sta blkcnt1 ;reset block count
        sta blkcnt1+1
        jsr tof
        jsr save      ;save file
        jsr clrstat
        jsr ton
        lda $05
        beq done
pmsg6   ldx #$27      ;print rewind
        lda msg6,x
        sta $07c0,x
        dex
        bpl pmsg6
        inc filecnt
        lda filecnt
        cmp #$02
        bne goon
        lda #$13
        sta $0bc5
goon    cli
        jmp start
done
        ldx #$27      ;xfer complete
pmsg5   lda msg5,x
        sta $07c0,x
        dex
        bpl pmsg5
        jmp *
skipblks
pmsg1   ldx #$0f      ;print skipping
        lda msg1,x
        sta $0450,x
        dex
        bpl pmsg1

        jsr getbyte ;skip control
        jsr getbyte ;skip load addy
        ldy #$00
skipem2 jsr getbyte ;skip blocks

```

```

        iny
        bne skipem2
        inc blkcnt1
        bne hblock2
        inc blkcnt1+1
hblock2
        lda blkcnt1+1
        cmp blkcnt2+1
        bne skipblks
        lda blkcnt1
        cmp blkcnt2
        bne skipblks
        jmp readfile;load next file
getbyte
        lda #$80      ;get a byte
        sta $02
byteget
        jsr getbit
        ror $02
        bcc byteget
        lda $02
        rts
getbit
        lda #$10      ;get a bit
        bit $dc0d
        beq getbit
        ldx $dd05
        lda $dd0d
        lsr a
        lda #$19
        sta $dd0e
        stx $d020     ;load effect
        rts

clrbot
        ldx #$27
        lda #$20
botclr
        sta $07c0,x
        dex
        bpl botclr
        rts

clrstat
        ldx #$0f
        lda #$20
statclr
        sta $0450,x
        dex
        bpl statclr
        rts

;-----
;save routine

```

```

save                                ;saving
                                ldx #$0f
pmsg4                                lda msg4,x
                                sta $0450,x
                                dex
                                bpl pmsg4

                                lda #$00    ;save file
                                sta $d020    ;border = black
                                lda #$1b
                                sta $d011    ;screen on
                                lda #<buffer
                                sta $fb      ;init save addy
                                lda #>buffer
                                sta $fc
                                lda #$01    ;open file for
                                ldx dvn      ;write
                                ldy #$01
                                jsr $ffba    ;setlfs
                                lda #$10
                                ldx #<name
                                ldy #>name
                                jsr $ffbd    ;setname
                                jsr $ffc0    ;open
                                ldx #$01
                                jsr $ffc9    ;chkout
                                lda #$00    ;start addy low
                                jsr $ffa8    ;send
                                lda $07     ;start addy high
                                jsr $ffa8    ;send

saveb                                ldy #$00    ;save the file
                                sei
                                inc $01
                                lda ($fb),y
                                dec $01
                                jsr $ffa8
                                inc $d020
                                dec $d020
                                inc $fb
                                bne asave
                                inc $fc

asave                                lda $fc
                                cmp $09
                                bne saveb
                                lda $fb
                                cmp $08
                                bne saveb
                                lda #$01
                                jmp $ffc3    ;close the file

```



```
blkcnt1 .byte $00,$00 ;cur block count
blkcnt2 .byte $00,$00 ;eof block count
filecnt .byte $00      ;file count
```

From:

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

Permanent link:

https://codebase64.org/doku.php?id=base:ocean_imagine_transfer_-_v1

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

