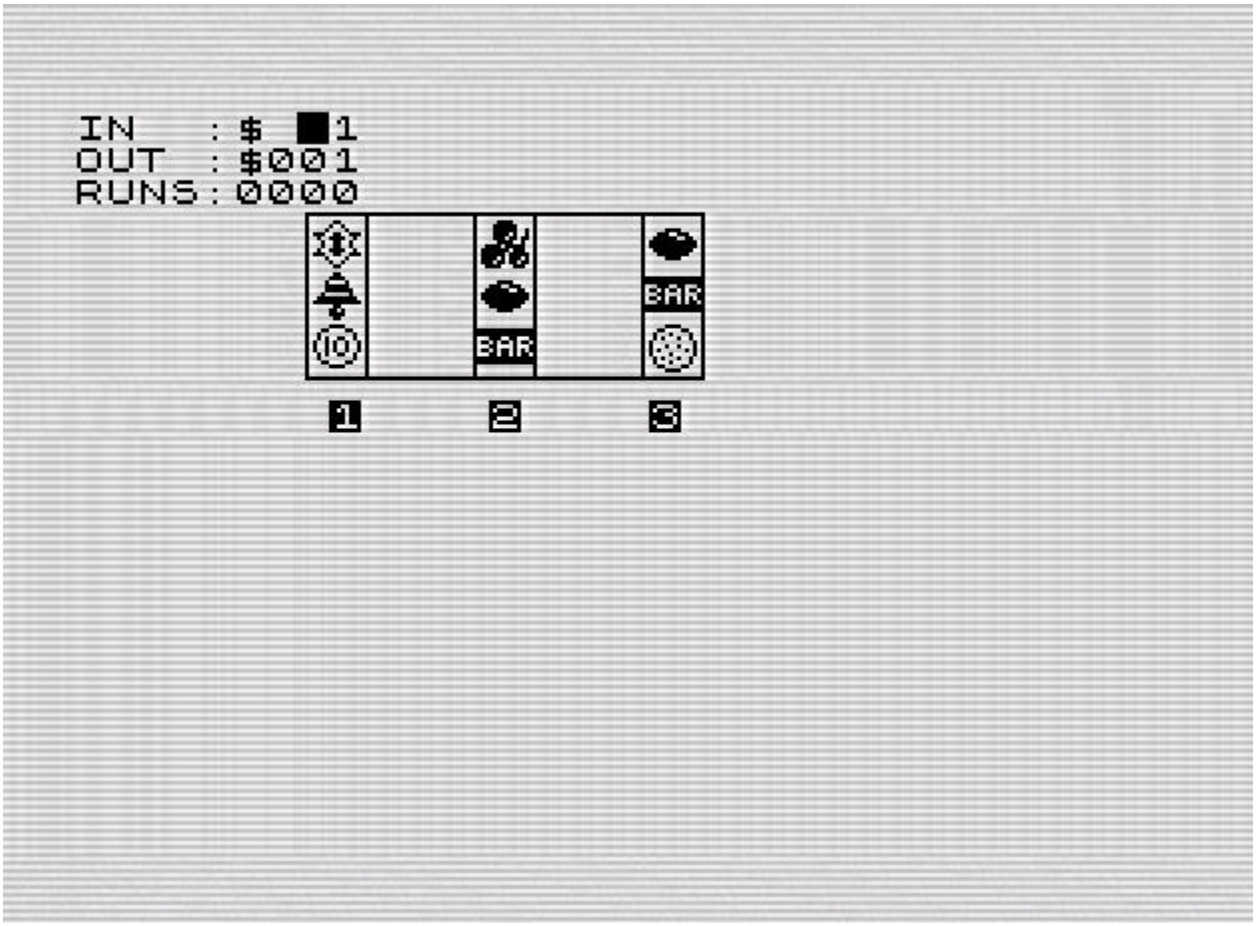
Fruitmachine simulator



This game hardly has a screenmemory. The only screenlines shown are the top and bottomline of the box. All other lines pick up some UDG's and show these on screen. The disadvantage of this: I couldn't double use the screen for initialization and the code got tight.
I finally had just enough room to fix the 48K bug and end this 31th 1K hires game.

```
; Fruit Machine Simulator
; Based on the idea from Shaun Bebbington

; Scoretable
;  25: 3x lemon
;  50: 3x cherry
;  75: 3x horseshoe
; 100: 3x citrus
; 125: 3x grapes
; 150: 3x dime
; 175: 3x bell
; 200: 3x star
; 225: 3x orange
; 250: 3x bar


? * TORNADO *

            ORG  #4009              ;#4009
            DUMP #C009
```

```
basiccall  JP   init              ; rest of initialization

d_file     DEFW dfile
dfcc       DEFW dfile+1
var        DEFW vars
dest       DEFW 0
eline      DEFW last              ; start needed
chadd      DEFW last-1            ; start needed
xptr       DEFW 0                 ; start needed
stkbot     DEFW 0
           DEFW 0                 ; memory above reused for data

berg       DEFB 0

mem        DEFW 0                 ; x

           DEFB 0
           DEFB 2
           DEFW 0

lastk      DEFB 255,255,255       ; used by ZX81
margin     DEFB 55                ; used by ZX81

nxtlin     DEFW basic

           DEFB 0
           DEFB 0

flagx      DEFB 0                 ; x
strlen     DEFW 0

taddr      DEFW 3213

seed       DEFW 0
frames     DEFW 65535             ; used by ZX81
coords     DEFB 0,0
prcc       DEFB 188
sposn      DEFB 33,24
cdflag     DEFB 64                ; fixed value

delay      LD   HL,frames         ; only called once
           LD   A,(HL)            ; replacing could save
           SUB  8                 ; some bytes when moving in
wfr        CP   (HL)              ; other routine
           JR   NZ,wfr
           RET

hr         LD   B,14              ; start lowres on screen
hr00       DJNZ hr00

           LD   HL,lowres+#8000
           LD   A,#1E
           LD   I,A
           LD   A,#F5
           LD   BC,#608           ; 6*8+8=56 lines
           CALL #2B5              ; lowres displaylines

           LD   B,13              ; line out hiresdisplay
hr01       DJNZ hr01

roll3      LD   BC,#230           ; roll 3 index
roll2      LD   DE,#2B2           ; roll 2 index
```

```
            DEFB #DD
            LD   H,40              ; 3x13+1=40 hires lines

            LD   HL,line
            CALL showline+#8000    ; show topline, 57 lines

            CALL showline+3        ; delay routine
            RET  Z                 ; 5 tstate more delay for sync

roll1       LD   HL,#200           ; roll 1 index, 2x compressed

            LD   A,#41
            LD   I,A               ; highbyte of hiresscreen

hr1loop     JP   lbuf+#8000

lbuf        LD   A,L               ; fetch roll1 index
            LD   R,A
            DEFW 0                 ; show udg
            LD   A,E               ; fetch roll2 index
            LD   R,A
            DEFW 0                 ; show udg
            LD   A,C               ; fetch roll3 index
            LD   R,A
            DEFW 0                 ; show udg
            JP   highback

highback    DEFB #DD
            DEC  H                 ; dec IXh
            JP   Z,hrout           ; end of rolls? tot 97 lines

;compresroutine roll3 for 3x line0  32 32 needs both same timing
            LD   A,C               ;  4  4 fetch index
            CP   B                 ;  4  4 compare with repeat
            JP   NC,normalc        ; 10 10 beyond repeat?
            DEC  B                 ;  4    decrease repeater
            JP   etest             ; 10    same line again
normalc     INC  BC                ;     6 point to next line
            INC  C                 ;     4 of udg in 10 tstates
            NOP                    ;     4 sync both routines


etest       LD   A,E               ; compresroutine roll 2
            CP   D                 ; as roll3, but with DE
            JP   NC,normale
            DEC  D
            JP   ltest
normale     INC  DE
            INC  E
            NOP

ltest       LD   A,L               ; compresroutine roll 1
            CP   H                 ; and finally with HL
            JP   NC,normalh
            DEC  H
            JP   looptest
normalh     INC  HL
            INC  L
            NOP

looptest    JP   hr1loop           ; do next hires line

hrout       EX   (SP),HL           ; sync display
```

```
            EX   (SP),HL           ; timing delay
            EX   (SP),HL
            EX   (SP),HL

            LD   HL,line           ; 97+1=98 lines
            CALL showline+#8000    ; show bottomline

            CALL showline          ; sync timing for hold-line
            PUSH HL
            LD   A,(HL)
            POP  HL

            LD   HL,holdline+#8000 ; 2*8+5=21 lines
            LD   A,#1E
            LD   I,A
            LD   A,#F5
            LD   BC,#205           ; 98+21=119 lines total
            CALL #2B5              ; display hold-line

            LD   C,5               ; 5x16=80 lines fill
app16lines  DJNZ app16lines        ; app. 16 screenlines delay
            DEC  C
            JR   NZ,app16lines     ; total 199 screenlines

            CALL #292              ; and back to mainprogram
            CALL #220
            LD   IX,hr
            JP   #2A4

showline    LD   A,H               ; line in other 256bytes range
            LD   I,A               ; so other I-reg needed
            LD   A,L
            LD   R,A               ; full line indexed
            DEFW 0,0,0,0           ; show line
            DEFB 0
rollm2      LD   H,2               ; part of showlinebuf as code
            INC  L                 ; saves some bytes elsewhere
            INC  L
            RET

space4100   EQU  #4100-$           ; filler to UDG-bank at #4100
            DEFS space4100

; here comes 256 bytes of UDG
; first compressed, 1x then 9x full size 2x(9x13+11)=256 bytes

lemon       DEFB 128,1
            DEFB 131,193
            DEFB 142,113
            DEFB 159,153
            DEFB 191,253
            DEFB 191,253
            DEFB 159,249
            DEFB 143,241
            DEFB 131,193
            DEFB 128,1
            DEFB 128,1

bar         DEFB 128,1
            DEFB 128,1
            DEFB 255,255
            DEFB 255,255
            DEFB 142,99
            DEFB 181,173
```

```
            DEFB 140,35
            DEFB 181,171
            DEFB 141,173
            DEFB 255,255
            DEFB 255,255
            DEFB 128,1
            DEFB 128,1

orange      DEFB 128,1
            DEFB 131,193
            DEFB 140,49
            DEFB 145,9
            DEFB 148,73
            DEFB 160,5
            DEFB 162,69
            DEFB 168,21
            DEFB 161,5
            DEFB 148,73
            DEFB 145,9
            DEFB 140,49
            DEFB 131,193

star        DEFB 128,1
            DEFB 129,129
            DEFB 130,65
            DEFB 132,33
            DEFB 185,157
            DEFB 147,201
            DEFB 137,145
            DEFB 137,145
            DEFB 147,201
            DEFB 185,157
            DEFB 132,33
            DEFB 130,65
            DEFB 129,129

bell        DEFB 128,1
            DEFB 129,129
            DEFB 131,193
            DEFB 132,33
            DEFB 135,225
            DEFB 135,241
            DEFB 136,17
            DEFB 159,249
            DEFB 191,253
            DEFB 129,129
            DEFB 131,65
            DEFB 131,193
            DEFB 129,129

dime        DEFB 128,1
            DEFB 131,193
            DEFB 140,49
            DEFB 144,9
            DEFB 148,201
            DEFB 165,37
            DEFB 165,37
            DEFB 165,37
            DEFB 148,201
            DEFB 144,9
            DEFB 140,49
            DEFB 131,193
            DEFB 128,1
```

```
grapes     DEFB 128,1
           DEFB 128,241
           DEFB 129,129
           DEFB 135,241
           DEFB 137,145
           DEFB 153,153
           DEFB 166,101
           DEFB 166,101
           DEFB 153,153
           DEFB 137,145
           DEFB 134,97
           DEFB 130,65
           DEFB 129,129

citrus     DEFB 128,1
           DEFB 129,249
           DEFB 135,233
           DEFB 143,249
           DEFB 159,169
           DEFB 191,217
           DEFB 187,241
           DEFB 191,113
           DEFB 183,225
           DEFB 189,193
           DEFB 191,1
           DEFB 128,1
           DEFB 128,1

horseshoe  DEFB 128,1
           DEFB 190,125
           DEFB 170,85
           DEFB 190,125
           DEFB 138,81
           DEFB 158,121
           DEFB 150,105
           DEFB 188,61
           DEFB 172,53
           DEFB 190,125
           DEFB 151,233
           DEFB 157,185
           DEFB 143,241

cherry     DEFB 128,1
           DEFB 131,193
           DEFB 135,225
           DEFB 135,229
           DEFB 135,165
           DEFB 135,109
           DEFB 131,201
           DEFB 158,249
           DEFB 191,185
           DEFB 191,125
           DEFB 189,117
           DEFB 187,109
           DEFB 158,57

; end of UDG-bank, now at #4200

n          EQU  27

lowres     DEFB 118,118,118
           DEFB "I"-n,"N"-n,0,0,14,13,0
paid       DEFB 28,28,118
           DEFB "O"-n,"U"-n,"T"-n,0,14,13
```

```
pout       DEFB 28,28,28,118
           DEFB "R"-n,"U"-n,"N"-n,"S"-n,14
credits    DEFB 28,28,28,28
holdline   DEFB 118
           DEFB 0,0,0,0,0,0,0,0
           DEFB 29+#80,0,0,0,0    ; inverted 1
           DEFB 30+#80,0,0,0,0    ; inverted 2
           DEFB 31+#80            ; inverted 3
dfile      DEFB 118

line       DEFB 255,255,255,255,255,255,255,255,255
           DEFB 255,255,255,240   ; top and bottom line of box

init       LD   IX,hr             ; only set after loading
           LD   B,2               ; set bugrange of 512 bytes
           LD   H,#3F             ; HL range #3f00 to #40ff
           LD   D,#BF             ; HL+48K
           LD   E,L
           LDIR                   ; repair 48K bug

start      LD   C,#FE             ; payout routine
amount     INC  C                 ; 1x add dollar on
           LD   B,4               ; 4x decrease runcounter
payout     CALL subcredit
           DJNZ payout
           JR   NZ,amount         ; repeat until empty

           INC  C                 ; need 1 more, also test >0
           LD   HL,pout+3         ; point to payout text
           CALL NZ,ac1            ; set payout amount on screen

waitnl     XOR  A                 ; after payout a restart delay
           IN   A,(254)           ; to show
           RRA                    ; money-in vs money-out
           JR   C,waitnl

; part of basiccode hidden in machinecode to save bytes
basic      LD   B,2               ; line number
           JR   skipbas           ; line length

           DEFB 249,212,28        ; the only basic command
           DEFB 126
           DEFB 143,0,18          ; #4009

skipbas    LD   L,paid*256/256    ; point to paid-in text
           XOR  A                 ; set startmoney on 0
rloop      LD   C,A               ; save result in C
           ADD  A,A               ; x2
           ADD  A,A               ; x4
           ADD  A,C               ; x5
           ADD  A,A               ; x10
           LD   C,A               ; result*10 in C
           PUSH BC                ; temporary save
reread     LD   (HL),128          ; set cursor on inputfield
           LD   BC,(lastk)        ; fetch lastkey pressed
           LD   A,C               ; inport to A
ktest      CP   0                 ; check with previous key
           JR   Z,reread          ; still key down, wait
           LD   (ktest+1),A       ; set new key pressed
           INC  A                 ; is it a no-key?
           JR   Z,reread          ; if so wait for key down
           PUSH HL                ; save textpointer
           CALL #7BD              ; translate to ascii
           LD   A,(HL)            ; fetch asciicode
```

```
            POP  HL                ; get back textpointer
            LD   (HL),A            ; write key to screen
            SUB  28                ; need "0"-"9"
            CP   10                ; test on "0"-"9"
            JR   NC,reread         ; if not read again
            POP  BC                ; fetch result and loop
            ADD  A,C               ; add units to tens
            INC  HL                ; point to next text
            DJNZ rloop             ; read next number
            JR   Z,basic           ; "00" entered, enter again

            LD   C,A               ; dollars to C

            LD   A,28              ; erase payout
            LD   L,pout*256/256
            LD   (HL),A
            INC  HL
            LD   (HL),A
            INC  HL
            LD   (HL),A

            LD   B,4               ; dollars to runs
pay2        PUSH BC                ; 1 dollar = 4 runs
            CALL addcredit
            POP  BC
            DJNZ pay2

nextpay     CALL subcredit         ; test remaining runs
            JR   Z,waitnl          ; out of runs

            LD   C,1               ; undo decrement from test
            CALL addcredit

next        LD   C,7               ; default all keys allow hold
holdreset   LD   A,(hold+1)        ; indicator HOLD
            CP   7                 ; select loop
            JR   NZ,nlread         ; previous was hold/win

holdselect  LD   B,2               ; double loop for flash
flash       LD   HL,dfile-1
            LD   E,C               ; copy from current hold
fullline    LD   A,(HL)
            OR   A
            JR   Z,holdspace       ; skip spaces
            BIT  0,E               ; already hold current roll?
            JR   Z,nexthold
            XOR  128               ; if not do flash
            LD   (HL),A
nexthold    SRL  E                 ; point to next roll
holdspace   DEC  HL
            LD   A,(HL)
            CP   118
            JR   NZ,fullline       ; do a full line check
            CALL delay             ; flashdelay
            DJNZ flash             ; flash off and flash on

; read 1-3
            LD   A,%11110111       ; inport 1-5
            IN   A,(254)
            RRA                    ; shift key 1
            JR   C,test2
            RES  2,C               ; set hold roll1
test2       RRA                    ; shift key 2
            JR   C,test3
```

```
            RES  1,C               ; set hold roll2
test3       RRA                    ; shift key 3
            JR   C,read123
            RES  0,C               ; set hold roll3
read123     LD   A,C
            OR   A                 ; all rolls selected?
            JR   Z,next            ; not allowed, reset select
; read Cash-out or start
nlread      LD   A,%10111110       ; Sh-V and H-NL together
            IN   A,(254)           ; read
            BIT  3,A               ; "C"ash out
            JP   Z,start

            RRA                    ; shift in NL (or Shift)
            JR   C,holdreset       ; continue if not Newline

endselect   LD   A,C               ; get HOLD-buttons
            LD   (hold+1),A        ; set HOLD-buttons for roll
            CALL subcredit         ; pay for turn

; the roll loop
            LD   D,%00000111       ; all rolls roll normally

holdrnd     LD   BC,0              ; fetch a random number
            LD   HL,(frames)
            ADD  HL,BC
            INC  HL
            LD   A,H
            AND  #1F
            LD   H,A
            LD   (holdrnd+1),HL
            LD   A,(HL)
frnd        LD   C,A
            SUB  9
            JR   NC,frnd           ; 0-8

            LD   A,C
            ADD  A,A               ; x2
            ADD  A,C               ; x3
            ADD  A,A               ; x6
            ADD  A,A               ; x12
            ADD  A,C               ; x13
            ADD  A,130             ; +1 full round
            LD   C,A               ; rnd nr of udg +1 round

doroll      LD   B,67              ; delay
wait        CALL showline
            DJNZ wait

hold        LD   B,%00000111       ; which rolls on hold?

            LD   HL,(roll1+1)      ; fetch index roll1
            LD   A,B               ; is it on hold?
            AND  D                 ; or is roll already over?
            BIT  2,A               ; result in this bit
            CALL NZ,rollmove       ; still on roll
            LD   (roll1+1),HL      ; save roll1-index

            LD   HL,(roll2+1)      ; same roll2
            LD   A,B
            AND  D
            BIT  1,A
            CALL NZ,rollmove
            LD   (roll2+1),HL
```

```
            LD   HL,(roll3+1)      ; same roll3
            LD   A,B
            AND  D
            BIT  0,A
            CALL NZ,rollmove
            LD   (roll3+1),HL

            DEC  C                 ; roll the number of moves
            JR   NZ,doroll

            SRL  D                 ; stop a roll
            JR   NZ,holdrnd        ; let next rolls move on

; test on win
            LD   E,L               ; save roll3
            LD   HL,(roll2+1)      ; fetch roll2
            LD   A,(roll1+1)       ; fetch roll1
            CP   L                 ; check with 2
            JR   NZ,nowin          ; not same, no win
            CP   E                 ; check with 3
            JR   NZ,nowin          ; not same, nowin

fiscore     PUSH AF
            LD   A,C
            ADD  A,25              ; calculate winpoints
            LD   C,A
            POP  AF
            ADD  A,26
            JR   NC,fiscore

            CALL addcredit         ; add points
            LD   (hold+1),A        ; never 7, no hold next turn

nowin       JP   nextpay           ; do next payment for play


subcredit   LD   HL,credits        ; pay when possible
            LD   D,4
testcred    LD   A,(HL)
            SUB  28
            JR   NZ,credok         ; not just "0" on counter
            INC  HL
            DEC  D
            JR   NZ,testcred
            RET                    ; zero reached, out of runs

credok      LD   L,credits*256/256+4
            DEFB #3A               ; hide command in LD A,(NN)
decten      LD   (HL),37           ; set "9"
            DEC  HL
            DEC  (HL)              ; down 1 count
            LD   A,(HL)
            CP   27                ; less then "0"
            JR   Z,decten          ; if so dec position before
            RET

addcredit   LD   HL,credits+4      ; like decrease but increase
ac1         PUSH HL                ; also used for OUT, save HL
            DEFB #3A
tens        LD   (HL),28
            DEC  HL
            INC  (HL)              ; add 1
            LD   A,(HL)
```

```
            CP   38                ; > "9"
            JR   Z,tens            ; increase next position
            POP  HL                ; fetch start again
            DEC  C
            JR   NZ,ac1            ; next increase from start
            RET

rollmove    LD   A,L               ; same routine as in hires but
            CP   H                 ; no constant timing needed
            DEC  H                 ; also reset of compresslines
            RET  C
            JP   rollm2            ; storing in LBUF (-2 bytes)

vars        DEFB 128
?
last        EQU  $
```