1) recenter pics for drawing
2) crashes w/space ~5500
seems to happen around time of "garbage coll"

```
to init i
   (dansinit.
    kaosinit.
    "NEXTSYSPIC ← 5.
    "NEXTAPIC ← "NPICS ← 10.
    cel init.
    movie init.
    draw init.
    toneselect init.
    brushselect init.
      ("USERTEMP ← GET USER "DO.
       PUT USER "DO "
         ((mouse 9)
          < - 128?(ourev).
          off?()
          "XCTR ← "YCTR ← 0.
          "XMIN ← "YMIN ← - 128.
          "XMAX ← "YMAX ← 128.
          "RFLAG ← true.
          "UPFLAG ← false.
          display holds run)).
      (interpret append 1.
       interpret movie 2.
       interpret menu 2.
       interpret nilpic 6.
       interpret cel 6.
       interpret freshcel 6).
      ("crosshairs ← apic 5.
       easel load 5.
       for i ← - 20 to 20
         (easel 0 i ← 3.
          easel i 0 ← 3).
       "freshsign ← apic NPICS.
       easel load NPICS.
       easel 1 1 ← 3.
       easel(- 1)
       1 ← 3.
       easel(- 1)(- 1)
       ← 3.
       easel 1(- 1)
       ← 3.
      ).
    "WSIZE ← 60.
    "CROSSFLAG ← false.
    reset.
    display run.
    disp clear.
    cr.
    disp ← '@ SHAZAM at your service, o fearless animator @'.
    cr
      (defs delete "readp.
       defs delete "dansinit.
       defs delete "kaosinit.
       defs delete "buttoncode.
       defs delete "init.
       "readp ← "dansinit ← "kaosinit ← "buttoncode ← "init ← 0.
      ))

to reset i
   ("MOVIES ← obset 10.
```

or swap. Press button 2 then 10 menu times

"defs ← obset 50.

```
    "CELS ← vector 3.
    CELS[1] ← freshcel.
    CELS[2] ← nilpic.
    for i ←(NPICS + 1)
    to NEXTAPIC(erasecel i).
    "NEXTCEL ← 2.
    "NEXTAPIC ← NPICS.
    "FRAMES ← "FRAME ← "FINC ← "F1 ← "F2 ← 1.
    display ← append nilpic nilpic.
   )

to over : under on
   (isnew?
      (:under :on)
    %run?(under run)
    %print?()
    %findpix?
      (under findpix.
       on findpix))

to append bflag t : aunder aon
   (isnew?
      (:aunder :aon)
    %run?
      ("bflag ← true.
       aon run.
       RFLAG?
         ("bflag ← false.
          aunder run.
          UPFLAG?
            ("t ← aon.
             "aon ← aunder.
             "aunder ← t)))
    %add?
      ("bflag ← true.
       aon add.
       aflag?()
       "bflag ← false.
       aunder add.
       aflag?
         ("t ← aon.
          "aon ← aunder.
          "aunder ← t))
    %print?()
    %findpix?
      (aunder findpix.
       aon findpix))

to at xsave ysave : xc yc atpic
   (isnew?
      (:xc :yc :atpic)
    %run?
      ("xsave ← XCTR.
       "ysave ← YCTR.
       centreupdate.
       atpic run.
       "XCTR ← xsave.
       "YCTR ← ysave)
    %print?()
    %findpix?(atpic findpix))
```

```
to nilpic : pic
   (%add?
      ("aflag ← true.
       bflag?
         ("aon ← npic)
       "aunder ← npic)
     %print?("nilpic print)
     isnew?
       ("pic ← 0)
     %findpix?())

to movie num nam : xseq yseq pseq xvec yvec pvec frame finc frames f1 f2 : Menu
   (%init?
      ("Menu ← menu 3 12 3 12 'moviemenu.' 1 4 3 5 6 14 7 8 9)
     isnew?
      (("f1 ← "frame ← 1.
        %of?
          ("f2 ← :frames.
           :finc.
           "xvec ← :xseq vec.
           "yvec ← :yseq vec.
           "pvec ← :pseq vec)
        "finc ← "f2 ← "frames ← 1.
        :num.
          (null num?()
           "f2 ← "frames ← num)
        "xvec ← vector frames.
        "yvec ← vector frames.
        "pvec ← vector frames.
        do frames
          (xvec[N] ← yvec[N] ← 0.
           pvec[N] ← freshcel).
        "xseq ← seq 0 xvec 1.
        "yseq ← seq 0 yvec 1.
        "pseq ← seq 0 pvec 1).
       cr.
       disp ← 'Type MOVIE name: '.
       "nam ←(read)
       [1].
       MOVIES ← nam.
       nam ← SELF.
       SELF wakeup)
     %wakeup?
       (addpic at(- 64) 0 wind(- WSIZE) WSIZE(- WSIZE) WSIZE over SELF outln)
     %run?(mousein?(Menu select))
     %set?
       (xseq set.
        yseq set.
        pseq set)
     %advance?
       ("frame ← frame + 1.
        frame > f2?
          ("frame ← f1))
     %print?()
     %findpix?(pseq findpix)
     %evals?(!(:")
       eval)
```

```
   %knows?(ev))

to update
   (display stop.
      (frame > f2?
         ("frame ← f1)).
    "xseq ← seq frame - 1 xvec[f1 to f2] finc.
    "yseq ← seq frame - 1 yvec[f1 to f2] finc.
    "pseq ← seq frame - 1 pvec[f1 to f2] finc.
    display run)

to cel x y : celpic : Menu
   (%init?
      ("Menu ← menu 3 12 3 12 'paintmenu.' 1 10 3 12 2 13 7 11 9)
    isnew?
      (%no?
         (0 =
            ("x ← oldnos[1 to nnewp] find :y)
         ?
            ("celpic ← "NEXTAPIC ← NEXTAPIC + 1.
             newpix["nnewp ← nnewp + 1] ← SELF.
             "CELS ← vecmod CELS CELS length 0 SELF.
             "NEXTCEL ← NEXTCEL + 1.
             oldnos[nnewp] ← y)
          !newpix[x])
       "celpic ← "NEXTAPIC ← NEXTAPIC + 1.
       "CELS ← vecmod CELS CELS length 0 SELF.
       "NEXTCEL ← NEXTCEL + 1.
       SELF wakeup)
    %wakeup?
      (addpic at 64 0 wind(- WSIZE)
       WSIZE(- WSIZE)
       WSIZE over SELF outln)
    %run?(mousein?(Menu select))
    %print?
      (disp ← 'cel no '.(celpic - NPICS)
       print)
    %celpic?(!celpic)
    %findpix?(newpix ← celpic))

to freshcel : celpic
   (isnew?
      ("celpic ← NPICS)
    %wakeup?(!cel)
    %print?("freshcel print)
    %findpix?())

to addpic npic aflag
   (:npic.
    "aflag ← false.
    display holds add.
    aflag?()
    display ← append display holds npic)

to add ()

to centreupdate
   ("XCTR ← xsave + xc.
    "YCTR ← ysave + yc)

to wind xisave xasave yisave yasave : xmin xmax ymin ymax wpic
```

```
    (isnew?
       (:xmin :xmax :ymin :ymax :wpic)
     %run?
       (wsave.
        wupdate.(mousein?
              ("RFLAG ← false.
               wpic run)).
        wrestore)
     %print?()
     %findpix?(wpic findpix)
     %knows?(ev))

to wsave
   ("xisave ← XMIN.
    "xasave ← XMAX.
    "yisave ← YMIN.
    "yasave ← YMAX)

to wupdate
   ("XMIN ← maxim xisave XCTR + xmin.
    "XMAX ← minim xasave XCTR + xmax.
    "YMIN ← maxim yisave YCTR + ymin.
    "YMAX ← minim yasave YCTR + ymax)

to wrestore
   ("XMIN ← xisave.
    "XMAX ← xasave.
    "YMIN ← yisave.
    "YMAX ← yasave)

to movepic
   (cr.
    disp ← 'MOVE WINDOW'.
    "xc ← mx.
    "yc ← my.
    repeat
      (down?()
       "xc ← xm.
       "yc ← ym.
       done).
    centreupdate.
    wupdate)

to changewindow
   (cr.
    disp ← 'CHANGE SIZE OF WINDOW'.
    "xmax ← mx.
    "ymin ← my.
    repeat
      (down?()
       "ymin ← ymrel.
       "xmax ← xmrel.
       done).
    wupdate)

to movewindow xtemp ytemp ptemp
   (cr.
    disp ← 'MOVE BORDER'.
    "xtemp ← xc.
    "ytemp ← yc.
    "ptemp ← wpic.
```

```
    "wpic ← at neg mxabs neg myabs at xc yc ptemp.
    "xc ← mx.
    "yc ← my.
    repeat
      (down?()
       done)
    "xmax ← xmrel + xmax.
    "xmin ← xmrel + xmin.
    "ymax ← ymrel + ymax.
    "ymin ← ymrel + ymin.
    "xc ← xtemp.
    "yc ← ytemp.
    "wpic ← ptemp.
    wupdate)

to run ()

to mousein
   (XMIN < xm < XMAX?
      (!YMIN < ym < YMAX)
    !false)

to maxim a b
   (
      (0 < :a - :b)
   ?(!a)
    !b)

to minim a b
   (
      (0 < :a - :b)
   ?(!b)
    !a)

to xmrel
   (!(mouse 8)
    - XCTR)

to ymrel
   (!(mouse 9)
    - YCTR)

to ourev
   (kbck?
      (disp ← 8.
       cr.
       read eval print.
       disp ← 20)
    disp ← 8.
    do 10()
    disp ← 20)

to seq temp : n v dn
   (isnew?
      (:n :v :dn)
    %vec?(!v)
    %load?
      (!v[(:)
      ])
    %store?
      (v[(:)
```

```
    ] ← :)
  %set?
    ("n ← frame - 1.
     "dn ← finc.
    )
  %print?
    (disp ← 'seq '.
     n print.
     disp ← ' {'.
     do v length
       (v[N] print.
        sp.
       )
     disp ← '} '.
     dn print.
    )
  %findpix?
    (for temp to v length
       (v[temp] findpix)))

to down
   (!4 = mouse 7)

to up
   (!0 = mouse 7)

to off
   (!2 = mouse 7)

to move xtemp ytemp ptemp xt yt pt i j k len newlen lenadded
   (menuoff.
    cr.
    "MOVEMENT print.
    finc = 1?
      ("len ←(f2 - f1)
       + 1.
       disp ← ' Currently, '.
       f1 print.
       disp ← ' to '.
       f2 print.
       disp ← ' active.' "newlen ← len + 60.
        ("xtemp ← vector newlen.
         "ytemp ← vector newlen.
         "ptemp ← vector newlen.
         xtemp[1 to len] ← xvec[f1 to f2].
         ytemp[1 to len] ← yvec[f1 to f2].
         ptemp[1 to len] ← pvec[f1 to f2].
         for i ← len + 1 to newlen do
           (xtemp[i] ← xvec[f2].
            ytemp[i] ← yvec[f2].
            ptemp[i] ← pvec[f2].
           )
        ("i ← 0.
         "xseq ← mx.
         "yseq ← my.
         "pseq ← ptemp[1].
         repeat
           (down?
              ("i ← i + 1.
               "pseq ← ptemp[i].
               xtemp[i] ← xmrel.
```

conflict w/ turtle not used

finc = 1 always enter and make 'if'...

~~xtemp[len+1 to newlen] ← fill xvec[f2]~~

```
            ytemp[i] ← ymrel.
           )
         off?(done)
         i = newlen?(done)))
      (disp ← ' MOVEMENT has '.
       i print.
       disp ← ' frames. How many do you want~ '.
       "j ←(read)
      [1].
     )
    j = 0?
     (update.
      disp ← ' MOVEMENT ignored.'.
      menuon)
     (
        (j > newlen?
           ("j ← newlen)).
       "lenadded ← j - len.
        (j > i?
           (for k ← i + 1 to j do
              (xtemp[k] ← xtemp[i].
               ytemp[k] ← ytemp[i].
               ptemp[k] ← ptemp[i].
              ))))
      ("xt ← vector frames + lenadded.
       "yt ← vector frames + lenadded.
       "pt ← vector frames + lenadded.
       xt[1 to f1 - 1] ← xvec[1 to f1 - 1].
       yt[1 to f1 - 1] ← yvec[1 to f1 - 1].
       pt[1 to f1 - 1] ← pvec[1 to f1 - 1].
       xt[f1 to f2 + lenadded] ← xtemp[1 to j].
       yt[f1 to f2 + lenadded] ← ytemp[1 to j].
       pt[f1 to f2 + lenadded] ← ptemp[1 to j].
       xt[f2 + 1 + lenadded to frames + lenadded] ← xvec[f2 + 1 to frames].
       yt[f2 + 1 + lenadded to frames + lenadded] ← yvec[f2 + 1 to frames].
       pt[f2 + 1 + lenadded to frames + lenadded] ← pvec[f2 + 1 to frames].
       "xvec ← xt.
       "yvec ← yt.
       "pvec ← pt.
      )
     "f2 ← f2 + lenadded.
     "frames ← frames + lenadded.
     sp.
     f1 print.
     disp ← ' to '.
     f2 print.
     disp ← ' active.'.
     update.
     menuon.
    )
  finc = 0?
   ("xseq ← mx.
    "yseq ← my.
    repeat(off?(done)).
    xvec[frame] ← xmrel.
    yvec[frame] ← ymrel.
    update.
    menuon))

to draw pict : : gt goto
  (%init?
```

```
      (to gt(CODE 36).
       to goto
         (gt 256 +(:)
          256 - :))
    setcursor XCTR YCTR.
    cr.
    "DRAW print.
      (CROSSFLAG?
         ("on ← over crosshairs outln)
       "on ← outln).
    easel load celpic.
    repeat
      (down?()
       done).
    repeat
      (down?(paint run)
       paint stop.
       kbck?(read eval print)
       off?(done)).
    setcursor 0 0.
    easel clear.
    menuon)

to copy pic
   (menuoff.
    cr.
    disp ← 'SHOW PAINT WINDOW'.
    pvec[frame] ← pvec[frame] wakeup.
    pseq store frame pvec[frame].
    repeat
      (down?()
       done).
    menuon)

to select choice
   (menuoff.
    cr.
    disp ← 'SELECT A PICTURE'.
    "choice ← 0.
    repeat
      (off?(done)
       down?
         ("choice ← choice + 1.
          pvec[frame] ← CELS[choice].
          pseq store frame pvec[frame].
          sp.
          CELS[choice] print.
          choice = NEXTCEL?
            ("choice ← 0))).
    menuon)

to singlestep
   (display stop.
    MOVIES map "
      (
         (vec[i] eval)
       evals
         ("finc ← 0).
         (vec[i] eval)
       advance.
         (vec[i] eval)
```

```
        set).
     display run)

to playback
   (cr.
    disp ← 'PLAYBACK MOVIE'.
    display stop.
    MOVIES map "
      (
         (vec[i] eval)
       evals
         ("frame ← 1.
          "finc ← 1).
         (vec[i] eval)
       set).
     display run)

to crossvis
   (
      (CROSSFLAG?
         ("CROSSFLAG ← false.
          cr.
          disp ← ' cross off.').
       "CROSSFLAG ← true.
       cr.
       disp ← ' cross on.').
     repeat
       (down?()
        done))

to brushselect : : Menu
   (%init?
      ("Menu ← menu 2 12 2 12 'brushmenu.' 22 23 24 25.
      )
    Menu select once.
   )

to toneselect : : Menu
   (%init?
      ("Menu ← menu 3 12 3 12 'tonemenu.' 15 16 17 16 18 16 19 20 21)
    Menu select once.
   )

to singstep
   (display stop.
    "finc ← 0.
    under advance.
    under set.
    cr.
    disp ← 'STEP TO FRAME '.
    frame print.
    display run)

to play
   (display stop.
    "finc ← 1.
    under set.
    display run)

to erasecel x
   ("x ← :.
```

```
      CODE 65)

to kaosinit
   ((interpret over 1)(interpret at 2)(interpret mx 3)(interpret my 4)(interpret  **
number 5)(interpret apic 6)(interpret outln 7)(interpret wind 9)(interpret clear  **
10)(interpret seq 11)(interpret neg 13)(interpret mxabs 14)(interpret myabs 15)
      (display configure.
       display ← outln)
      ("black ← "((- 1)(- 1)).
       "white ← "(0 0).
       "grey ← "((- 23131)(- 23131)).
       "light ← "(1285 1285).
       "dark ← "((- 1286)(- 1286)).
       "vlight ← "(1025 1025).
       "trans ← "((- 1)
          0).
       "pin ← 1.
       "dot ← 2.
       "drop ← 3.
       "block ← 4.
      )
      (paint tone ← grey.
       paint brush ← dot))

to display arg0 : : curpic ntodo
   (%←?
      ("arg0 ← :curpic CODE 61)
    %holds?(!curpic)
    %running?
      (0 = mem ntodo?(!false)
       !mem ntodo)
    %run?
      (mem ntodo ←
         (%for?(:)
          - 1).
       active 1024)
    %stop?
      (mem ntodo ← 0.
       inactive 1024)
    %configure?
      ("ntodo ← 8 + mem 67.
       CODE 57)
    !curpic)

to paint arg0 arg1 tone : : brush tone1 tone2 going
   (%running?
      (%←?
         (:going?(active 256)
          inactive 256)
       !going)
    %brush?
      (%←?
         ("brush ← :arg0.
          CODE 62)
       !brush)
    %tone?
      (%←?
         (:tone.
          "arg0 ← "tone1 ← tone[1] eval.
          "arg1 ← "tone2 ← tone[2] eval.
          CODE 63)
```

```
        arg0 ← vector 2.
        arg0[1] ← tone1.
        arg0[2] ← tone2.
        !arg0)
     %run?
       (paint running ← true)
     %stop?
       (paint running ← false))

to easel x y v : : picno
   (%load?
      ("x ← :picno.
       CODE 60.
       !picno)
     %clear?
       (easel load 0.
        sp.
        space print)
     %holds?(!picno)
     :x :y %←?
       (:v.
        CODE 59.
        !v)
     "v ← - 1.
     !CODE 59)

to mx
   (isnew?()
    %print?(disp ← 'mx'))

to my
   (isnew?()
    %print?(disp ← 'my'))

to mxabs
   (%print?(disp ← 'mxabs')
    isnew?())

to myabs
   (%print?(disp ← 'myabs')
    isnew?())

to xm (!mouse 8)

to ym (!mouse 9)

to apic : num
   (isnew?(:num)
    %print?
      ("#print.
       num print)
    %picnum?(!num)
    %findpix?(newpix ← num))

to outln
   (isnew?()
    %print?(disp ← 'outln'))

to active
   (mem 299 ←
      ((:)
```

```
      &+ mem 299))

to inactive
   (mem 299 ←
      ((- 1)
       &-(:))
    &* mem 299)

to not
   ((:)
   ?(!false)
    !true)

to interpret clas n
   (:#clas.
    :n.
    "clas ← point clas.
    CODE 56)

to neg : n
   (isnew?(:n)
    %print?
      (disp ← 'neg' sp n print))

to point obj
   (:#obj.
    CODE 58)

to space q
   ("q ← mem 67.
    !
      (mem mem 7 + q)
    - mem mem 6 + q)

to setcursor q
   ("q ← mem 67.
    mem q + 9 ← :.
    mem q + 10 ← :)

to buttoncode (!"
      ((movepic)(draw)(down?
            (cr.
             disp ← 'SLEEP'.
             bflag?
               ("aon ← nilpic)
             "aunder ← nilpic))(move)(singstep)(select)(movewindow)(copy)(changew **
indow)(crossvis)(erasecel celpic)
         (brushselect.
          Menu on.
         )
         (toneselect.
          Menu on.
         )(play)
         (paint tone ← black)
         (paint tone ← white)
         (paint tone ← dark)
         (paint tone ← trans)
         (paint tone ← grey)
         (paint tone ← vlight)
         (paint tone ← light)
         (paint brush ← pin)
```

```
        (paint brush ← dot)
        (paint brush ← drop)
        (paint brush ← block)))

to menu t k emx emy : menux menuy mpic buttons rows cols rwidth cwidth
.  (isnew?
      ("rows ← :.
       "rwidth ← :.
       "cols ← :.
       "cwidth ← :.
       "buttons ← vector(rows * cols)
       + 1.
       display ← "mpic ← apic "NEXTSYSPIC ← NEXTSYSPIC + 1.
       picin(:)
       apic NEXTSYSPIC.
       buttons[1] ← "().
       for k ← 2 to(rows * cols)
       + 1 do
         (null :t?(done)
          buttons[k] ← buttoncode[t]))
    %on?(menuon)
    %off?(menuoff)
    %select?
      (SELF on.
         (%once?(repeat(mousein?(down?(selectit.
                        done))))
          repeat
            (mousein?
               (kbck?(read eval print)
               down?(selectit))
             done)).
       SELF off.
      )
    %print?())

to menuon
   ("menux ←((XMIN + XMAX)
       / 2)
    - XCTR.
    "menuy ←((YMIN + YMAX)
       / 2)
    - YCTR.
    "on ← over outln Menu)

to menuoff
   ("on ← outln)

to inmenu
   (
      (-(cols * cwidth)
       / 2)
    < emx <((cols * cwidth)
       / 2)
   ?
      (!
         (-(rows * rwidth)
          / 2)
       < emy <(rows * rwidth)
       / 2)
    !false)
```

```
to { vec i len
   ("vec ← vector "len ← 4.
    "i ← 0.
    repeat
      (%)?
         (!vec[1 to i])
         (i = len?
            ("vec ← vec[i to "len ← 2 * len]))
       vec["i ← i + 1] ← :))

to , (:)

to moviein newpix nnewp oldnos M f i
   ("f ← file :i old?
      (display stop.
       "newpix ← vector 20.
       "oldnos ← vector 20.
       "nnewp ← 0.
       "M ← usereader f eval.
       for i to nnewp
         (readpic f newpix[i] celpic).
       display run)
    disp ← 'no such file'.
   )

to movieout newpix M f i
   (display stop.
    "f ← file :.
    :M.
    usedisp f
      (M print.
       disp ← '').
    "newpix ← obset 20.
    M findpix.
    newpix map "
      (writepic f vec[i]).
    f close.
    display run.
   )

to dansinit
   (
      (GET number "DO)
   [10][13] ← ', -'.
    filin evals
      (addto fseq "
         (%evals?(!(:")
             eval))))

to usedisp disp
   (:disp.(:")
    eval)

to usereader fi i
   (:fi.
    !filin evals
      ("f ← fi.
       "reader ← fseq fi evals sadr.
       reader evals
         ("ptr ← fi evals bytec).
       "i ← read.
```

```
      reader evals
        ((bridge?
              (0 > "ptr ← ptr - stop?
                 ("ptr ← ptr + 512.
                  fi evals
                    ("pagen ← pagen - 1))))
           fi evals
             ("bytec ← ptr)).
        !i))

to readpic adr afree bmin f mmax picnum picsiz
   ("f ← :.
    "picnum ← :.
    "afree ← mem 6 + mem 67.
    "bmin ← mem mem 7 + mem 67.
    "mmax ← mem mem 11 + mem 67.
    f eof?
      (disp ← 'file eof'.
       cr)
    0 = "picsiz ← f next word?
      (disp ← 'zero pic size'.
       cr)
    0 > mem mmax - picnum?
      (disp ← 'picture in use'.
       cr).
    bmin > picsiz + "adr ← mem afree?
      (mem afree ← picsiz + adr.
       mem mmax - picnum ← adr - mmax - picnum.
       mem adr ← picsiz.
       mem adr + 1 ← picnum.
       disp ← 'filed picture '.
       f next word print.
       disp ← ' stored as picture '.
       picnum print.
       cr.
       f readseq adr + 2 picsiz - 2)
    disp ← 'storage full'.
    cr)

to writepic f adr mmaxp
   ("f ← :.
    "mmaxp ←
      (mem mem 11 + mem 67)
    - :.
    "adr ← mmaxp + mem mmaxp.
    f writeseq adr mem adr)

to inrow i j
   (for i to rows do
      ("j ←((rows * rwidth)
           / 2)
       -(rwidth *(i - 1)).
       j > emy >(j - rwidth)
      ?(!i))
    !0)

to incol i j
   (for i to cols do
      ("j ←
         (-(cols * cwidth)
           / 2)
```

```
        +(cwidth *(i - 1)).
        j < emx <(j + cwidth)
       ?(!i))
     !0)

to picin f celpic
   ("f ← file(:)
    old?
      ("celpic ←
         (%.
         ?(cel.
             NEXTAPIC)
          %apic?(:)
          NPICS + :)
        erasecel celpic.
        readp f celpic.
        f close)
     disp ← 'no such file'.
     cr)

to readp adr afree bmin f mmax picnum picsiz
   ("f ← :.
    "picnum ← :.
    "afree ← mem 6 + mem 67.
    "bmin ← mem mem 7 + mem 67.
    "mmax ← mem mem 11 + mem 67.
    f eof?()
    0 = "picsiz ← f next word?()
    0 > mem mmax - picnum?()
    bmin > picsiz + "adr ← mem afree?
      (mem afree ← picsiz + adr.
       mem mmax - picnum ← adr - mmax - picnum.
       mem adr ← picsiz.
       mem adr + 1 ← picnum.
       f next word.
       f readseq adr + 2 picsiz - 2)
     disp ← 'storage full' cr)

to selectit x
   ("emx ← xmrel - menux.
    "emy ← ymrel - menuy.
    inmenu?
      ("x ← incol +(cols *(inrow - 1))
       + 1.
       x > 0?
         (buttons[x] eval))
    "UPFLAG ← true.
   )

to tablet
   ("down ← #down1.
    "up ← #up1.
    "off ← #off1.
    "button1 ← #down1)

to up1
   (!0 =(- 8192)
    &* mem - 2)

to down1
   (!(16384 + 16384)
```

loop

up not used!

```
    =(- 8192)
    &* mem - 2)

to off1
   (!(- 8192)
    =(- 8192)
    &* mem - 2)
```