12/23/74


```
shazamfns←( init reset append at wind over movie nilpic freshcel cel update addpic add movepic changewindow m **
ovewindow run mousein xmrel ymrel ourev seq down off move movieupdate draw copy select singlestop playback cros **
svis brushselect toneselect singstep play erasecel moviesetup finishup getpoints ).
to init i
  (addto dispframe
    (⊲is⊳
      (⊲dispframe⊳(⋂true)
       ⊲?⊳(⋂dispframe)
       !.
       ⋂false)).
   addto obset
    (⊲contents⊳
      (⋂vec[1 to end])
     ⊲end⊳(⋂end))
   dansinit.
   kaosinit.
   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⊳()
       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.
   reset.
   display run.
   disp clear.
   cr.
   disp ← '(A) SHAZAM at your service, o fearless animator (A)'.
   cr
    (dots ← obset 50.
     dansinit ← kaosinit ← buttoncode ← init ← 0.
    ))

to reset i
  (MOVIES ← obset 10.
   CELS ← vector 3.
   CELS[1] ← freshcel.
   CELS[2] ← nilpic.
```

```
    for i ← (NPICS + 1)
    to NEXTAPIC(erasecel i).
    ☞NEXTCEL ← 2.
    ☞NEXTAPIC ← NPICS.
    display ← append nilpic nilpic.
    )

to append bflag RFLAG t : aunder aon
  (isnew⇒
    (:aunder :aon)
   ⇩run⇒
    (☞bflag ← true.
     aon run.
     RFLAG⇒
       (☞bflag ← false.
        aunder run.
       ))
   ⇩add⇒
    (☞bflag ← true.
     aon add.
     aflag⇒()
     ☞bflag ← false.
     aunder add.
     aflag⇒
       (☞t ← aon.
        ☞aon ← aunder.
        ☞aunder ← t))
   ⇩print⇒
    (disp is dispframe⇒()
     disp ← 'append' sp aunder print sp aon print)
   ⇩findpix⇒
    (aunder findpix.
     aon findpix))

to at : xc yc atpic
  (isnew⇒
    (:xc :yc :atpic)
   ⇩run⇒(atpic run.
    )
   ⇩print⇒
    (disp is dispframe⇒()
     disp ← 'at' sp xc print sp yc print sp atpic print)
   ⇩findpix⇒(atpic findpix))

to wind : xmin xmax ymin ymax wpic
  (isnew⇒
    (:xmin :xmax :ymin :ymax :wpic)
   ⇩run⇒
    (mousein⇒
      (☞RFLAG ← false.
       wpic run)
     ☞RFLAG ← true.
    )
   ⇩print⇒
    (disp is dispframe⇒()
     disp ← 'wind' sp xmin print sp xmax print sp ymin print sp ymax print sp wpic print)
   ⇩findpix⇒(wpic findpix)
   ⇩knows⇒(ev))

to over : under on
  (isnew⇒
    (:under :on)
   ⇩run⇒(under run)
   ⇩print⇒
    (disp is dispframe⇒()
     disp ← 'over' sp under print sp on print)
```

```
  ◊findpix⇒
    (under findpix.
     on findpix))

to movie nam : xseq yseq pseq xvec yvec pvec frame finc frames f1 f2 xpos ypos minx maxx miny maxy : Menu
  (◊wakeup⇒
    (addpic at xpos ypos wind minx maxx miny maxy over SELF outln)
   ◊reset⇒
    (moviesetup.
     SELF wakeup.
    )
   ◊run⇒(mouseIn⇒
       (Menu select.
        movieupdate.
        ))
   ◊set⇒
    (xseq set.
     yseq set.
     pseq set)
   ◊advance⇒
    (☞frame ← frame + 1.
     frame > f2⇒
       (☞frame ← f1))
   ◊print⇒
    (disp is dispframes⇒()
     disp ← 'movie of ' frames print sp finc print sp xseq print sp yseq print sp pseq print.
     sp.
     xpos print.
     sp.
     ypos print.
     sp.
     minx print.
     sp.
     maxx print.
     sp.
     miny print.
     sp.
     maxy print.
    )
   ◊findpix⇒(pseq findpix)
   ◊evals⇒(∏(:☞)
     eval)
   ◊init⇒
    (☞Menu ← menu 3 12 3 12 7 '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.
          ☞xpos ← :.
          ☞ypos ← :.
          ☞minx ← :.
          ☞maxx ← :.
          ☞miny ← :.
          ☞maxy ← :.
          )
        moviesetup.
        ☞finc ← 1.
        ☞frames ← :.
        (null frames⇒
          (☞f2 ← ☞frames ← 1)
         ☞f2 ← frames).
```

```
      (@xvec ← vector frames.
      (@yvec ← vector frames.
      (@pvec ← vector frames.
      do frames
        (xvec[N] ← yvec[N] ← 0.
         pvec[N] ← freshcol).
      (@xseq ← seq 0 xvec 1.
      (@yseq ← seq 0 yvec 1.
      (@pseq ← seq 0 pvec 1).
     repeat
      (cr.
      disp ← 'Type MOVIE name: '.
      (@nam ← (read)
      [1].
      null nam⇒()
      done.
      ).
     MOVIES ← nam.
     nam ← SELF.
     SELF wakeup))

to nilpic : pic
  (◊add⇒
    ((@aflag ← true.
    bflag⇒
      ((@aon ← npic)
    (@aunder ← npic)
   ◊print⇒(@nilpic print)
   isnew⇒
    ((@pic ← 0)
   ◊findpix⇒())

to freshcol : colpic
  (isnew⇒
    ((@colpic ← NPICS)
   ◊wakeup⇒(⋂col)
   ◊print⇒(@freshcol print)
   ◊findpix⇒())

to col x y : colpic CROSSFLAG : Menu
  (◊wakeup⇒
    (addpic at 64 0 wind(- WSIZE)
     WSIZE(- WSIZE)
     WSIZE over SELF outln)
   ◊run⇒(mousein⇒(Menu select))
   ◊print⇒
    (disp ← 'col no '.(colpic - NPICS)
    print)
   ◊colpic⇒(⋂colpic)
   ◊findpix⇒(newpix ← colpic)
   ◊init⇒
    ((@Menu ← menu 3 12 3 12 6 'paintmenu.' 1 10 3 12 2 13 7 11 0)
   isnew⇒
    (◊no⇒
      (0 =
        (@x ← oldnos[1 to nnewp] find y)
       ⇒
        ((@colpic ← @NEXTAPIC ← NEXTAPIC + 1.
        newpix[@nnewp ← nnewp + 1] ← SELF.
        @CELS ← vecmod CELS CELS length 0 SELF.
        @NEXTCEL ← NEXTCEL + 1.
        oldnos[nnewp] ← y)
       ⋂newpix[x])
     (@colpic ← @NEXTAPIC ← NEXTAPIC + 1.
     @CELS ← vecmod CELS CELS length 0 SELF.
     @NEXTCEL ← NEXTCEL + 1.
```

```
    @CROSSFLAG ← false.
    SELF wakeup))

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 addpic npic aflag
  (:npic.
  @aflag ← false.
  display holds add.
  aflag⇒()
  display ← append display holds npic)

to add ()

to movepic
  (cr.
  disp ← 'MOVE WINDOW'.
  @xc ← mx.
  @yc ← my.
  repeat
    (down⇒()
    @xc ← xm.
    @yc ← ym.
    done).
  )

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

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 ←(xm - xtemp)
  + xmax.
  @xmin ←(xm - xtemp)
  + xmin.
  @ymax ←(ym - ytemp)
  + ymax.
  @ymin ←(ym - ytemp)
  + ymin.
  @xc ← xtemp.
```

```
  ☞yc ← ytemp.
  ☞wplc ← ptemp.
 )

to run ()

to mousein
 ((xmin + xc)
  < xm <((xmax + xc)
 ⇒
   (⋂((ymin + yc)
    < ym <((ymax + yc))
  ⋂false)

to xmrel
 (⋂(mouse 8)
  - xc)

to ymrel
 (⋂(mouse 9)
  - yc)

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 is dispframe⇒()
    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 off
```

```
  (⋂2 = mouse 7)

to move xt yt pt xx yy j i
  (menuoff.
   cr.
   (☞MOVEMENT print.
   finc = 0⇒
     (☞xseq + mx.
      ☞yseq + my.
      repeat(off⇒(done))
       xvec[frame] + xmrel.
       yvec[frame] + ymrel.
       update.
       menuon)
   not(finc = 1)
   ⇒(disp + 'MOVEMENT not available if frame increment not 1 or 0')
   cr.
   disp + 'Currently ' f1 print.
   disp + ' to '.
   f2 print.
   disp + ' are active frames.' getpoints.
   cr.
   disp + 'MOVEMENT has '(xx end)
   print.
   disp + ' new frames.' 0 = xx end⇒
     (update.
      disp + ' MOVEMENT ignored.' menuon)
   disp + ' How many do you want ?' 0 = (☞j + (read)
   [1]⇒
     (update.
      disp + 'MOVEMENT ignored.' menuon).
      (j >(xx end)
      ⇒
        (☞j + xx end))
   finishup.
   update.
   menuon.
  )

to movieupdate
  (☞xpos + xc.
   ☞ypos + yc.
   ☞minx + xmin.
   ☞miny + ymin.
   ☞maxx + xmax.
   ☞maxy + ymax.
  )

to draw pict : : gt goto
  (◇init⇒
    (to gt(CODE 36).
     to goto
       (gt 256 +(:)
        256 - :))
   setcursor xc yc.
   cr.
   ☞DRAW print.
     (CROSSFLAG⇒
       (☞on + over crosshairs outln)
     ☞on + outln).
   easel load colpic.
   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
      (☞inc ← 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.
       ☞inc ← 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 9 'brushmenu.' 22 23 24 25.
    )
   Menu select once.
  )

to toneselect : : Menu
  (◇init⇒
    (@Menu + menu 3 12 3 12 8 '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 moviesetup
  (@xpos + - 64.
   @ypos + 0.
   @miny + @minx + - WSIZE.
   @maxx + @maxy + WSIZE.
  )

to finishup
  (@f2 + f2 +(j - 1).
    (@xt + vector frames + j - 1.
     @yt + vector frames + j - 1.
     @pt + vector frames + j - 1).
    (xt[1 to frame - 1] + xvec[1 to frame - 1].
     yt[1 to frame - 1] + yvec[1 to frame - 1].
     pt[1 to frame - 1] + pvec[1 to frame - 1]).
    (@i + xx contents[1 to j].
     xt[frame to
       (frame + j - 1)
     ] + i.
     @i + yy contents[1 to j].
     yt[frame to
       (frame + j - 1)
     ] + i.
     for i + frame to
```

```
      (frame + j - 1)
      (pt[i] ← pvec[frame])).
    (xt[frame + j to frames + j - 1] ← xvec[frame + 1 to frames].
     yt[frame + j to frames + j - 1] ← yvec[frame + 1 to frames].
     pt[frame + j to frames + j - 1] ← pvec[frame + 1 to frames]).
   (☞frames ← frames + j - 1.
   (☞xvec ← xt.
   (☞yvec ← yt.
   (☞pvec ← pt.
   update.
   menuon.
  )

to getpoints
  ((☞xx ← obset 60.
   (☞yy ← obset 60.
   (☞xseq ← mx.
   (☞yseq ← my.
   (☞pseq ← pvec[frame].
   repeat
    (down⇒
      (xx ← xmrel.
       yy ← ymrel.
      )
     off⇒(done)
     60 = xx end⇒(done)))

(☞menufns←(☞( menu buttoncode menuon clipg clipl menuoff inmenu ] , incol inrow selectit ).
to menu t k emx emy : menux menuy mpic buttons rows cols rwidth cwidth
  (isnew⇒
    ((☞rows ← 1.
    (☞rwidth ← 1.
    (☞cols ← 1.
    (☞cwidth ← 1.
    (☞buttons ← vector(rows * cols)
    + 1.
    display ← (☞mpic ← apic (☞t ← 1.

    picin(1)
    apic t.
    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 buttoncode (∩(☞
    ((movepic)(draw)(down⇒
        (cr.
         disp ← 'SLEEP'.
         bflag⇒
           (☞aon ← nilpic)
```

```
      (☞aunder ← aon.
      (☞aon ← nilpic))(move)(singstep)(select)(movewindow)(copy)(changewindow)(crossvis)(erasecel celpic)
    (brushselect.
     Menu on.
    )
    (tonoselect.
     Menu on.
    )(play)
    (paint tone ← ☞((- 1)(- 1)))
    (paint tone ← ☞(0 0))
    (paint tone ← ☞((- 1286)(- 1286)))
    (paint tone ← ☞((- 1)
      0))
    (paint tone ← ☞((- 23131)(- 23131)))
    (paint tone ← ☞(1025 1025))
    (paint tone ← ☞(1285 1285))
    (paint brush ← 1)
    (paint brush ← 2)
    (paint brush ← 3)
    (paint brush ← 4)))

to menuon
  (☞menux ←
    (
      (clipl xc + xmin)
     +
      (clipg xc + xmax)
     - 2 * xc)
   / 2.
   ☞menuy ←
    (
      (clipl yc + ymin)
     +
      (clipg yc + ymax)
     - 2 * yc)
   / 2.
   ☞on ← over outln Menu.
  )

to clipg a
  (
    (☞a ← :.
    )
   > 128⇒(⇑128)
   ⇑a)

to clipl a
  (
    (☞a ← :.
    )
   < - 128⇒(⇑- 128)
   ⇑a)

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[1 to ⍚len ← 2 * len]))
      vec[⍚i ← i + 1] ← :))

to , (:)

to incol i j
  (for i to cols do
    (⍚j ←
      (-(cols * cwidth)
       / 2)
     +(cwidth *(i - 1)).
     j < emx <(j + cwidth)
    ⇒(⋂i))
   ⋂0)

to inrow i j
  (for i to rows do
    (⍚j ← ((rows * rwidth)
       / 2)
     -(rwidth *(i - 1)).
     j > emy >(j - rwidth)
    ⇒(⋂i))
   ⋂0)

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

⍚kaosfns←⍚( kaosinit display paint easel mx my mxabs myabs xm ym apic outln active inactive not interpret neg poi **
nt space setcursor ).
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 myab**
s 15)
    (display configure.
     display ← outln)
    (paint tone ← ⍚((- 23131)(- 23131)).
     paint brush ← 2))

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 ← :)

☞filefns←☞( dansinit usedisp usereader readpic writepic picin moviein movieout ).
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 flg adr afree bmin f mmax picnum picsiz
  (⍐flg ← ⍂noprint.
   ⍐f ← :.
   ⍐picnum ← :.
   ⍐afree ← mem 6 + mem 67.
   ⍐bmin ← mem mem 7 + mem 67.
   ⍐mmax ← mem mem 11 + mem 67.
   f eof⇒
     (flg⇒()
      disp ← 'file eof'.
      cr)
   0 = ⍐picsiz ← f next word⇒
     (flg⇒()
      disp ← 'zero pic size'.
      cr)
   0 > mem mmax - picnum⇒
     (flg⇒()
      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.
        (flg⇒(f next word)
         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 picin f colpic
  (☞f ← file(:)
   old⇒
    (☞colpic ←
      (⊰.
       ⇒(col.
           NEXTAPIC)
        ⊰apic⇒(:)
        NPICS + :)
      erasecol colpic.
      readpic noprint f colpic.
      f close)
    disp ← 'no such file'.
    cr)

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] colpic).
     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.
   )

☞tabletfns←☞( tablet down1 off1 ).
to tablet
  (☞down ← #down1.
   ☞off ← #off1.
   ☞button1 ← #down1)

to down1
  (⋂(16384 + 16384)
   =(- 8192)
   ^* mem - 2)

to off1
  (⋂(- 8192)
   =(- 8192)
   ^* mem - 2)
```

Shazam

core 2737 before init
1714 after init

Dec 9
~~2309~~ 2303 w/o inlet
2302 2561 w

filin 'filepic' (ETHERMICA <owner> FILEPIC)


```
to picin f celpic
  (f ← file(:)
  old⇒
   (
     (apic⇒
       (celpic ← :)
     null
       (celpic ← 1)
     ⇒
     (cel.
       celpic ← NEXTAPIC)
     celpic ← celpic + 10.
     eraseeel
     readpic f celpic.
     f close)
   disp ← 'no such file'.
   cr)

to picout f picnum
  (f ← file :.
   picnum ←
    (apic⇒(:)
     10 + :).
   writepic f picnum.
   f close)

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)n
```

interim picture saver - version 1

Suggested usage for saving

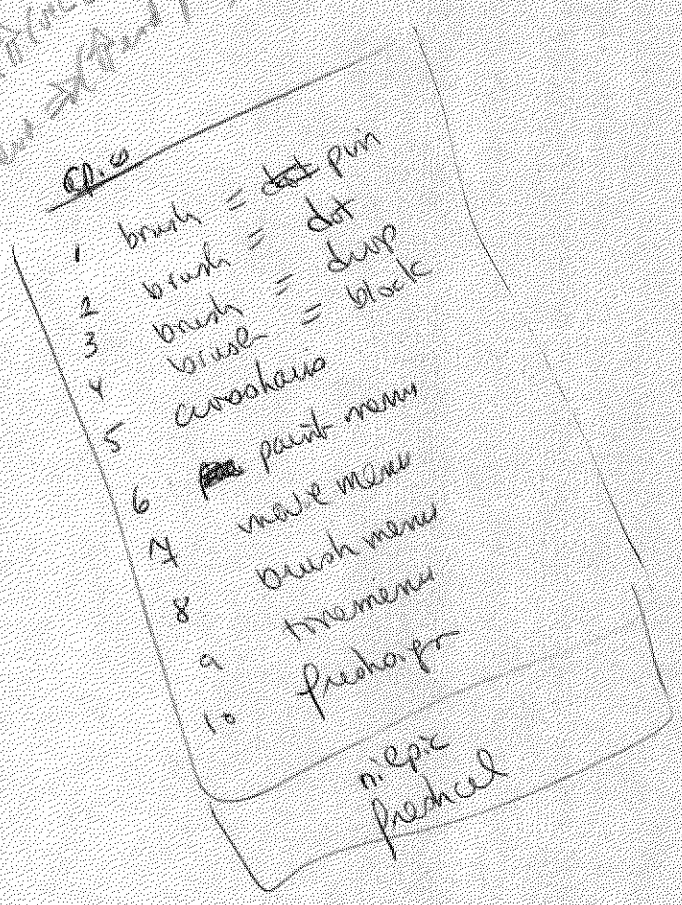
display stop.
picout <filename> <cell number>
OR picout <filename> apic <apic number>
:
display run.

... for loading

display stop.
picin <filename>. [uses next available cell number]
OR picin <filename> <cell number>
[erases cell + replaces it with file contents]
OR picin <filename> apic <apic number>
[erase cell + replace it with file contents]
:
display run.

<filename> is a Smalltalk string e.g. 'zap.'
<cell number> is an integer between 1 and 245
<apic number> is an integer between 1 and 255

```
☞shazamfns←☞( init reset append at wind over movie nilpic freshcel cel update addpic add movepic changewindow m **
ovewindow run mousein xmrel ymrel ourev seq down off move movieupdate draw copy select singlestep playback cros **
svis brushselect toneselect singstep play erasecel moviesetup ).
to init i
  (addto dispframe ☞
    (⍁is⇒
      (⍁dispframe⇒(⋂true)
      ⍁?⇒(⋂☞dispframe)
      !☞.
      ⋂false)).
  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⇒()
      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 ← obset 50.
    ☞readp ← ☞dansinit ← ☞kaosinit ← ☞buttoncode ← ☞init ← 0.
   ))
to reset i
  (☞MOVIES ← obset 10.
  ☞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 append bflag RFLAG t : aunder aon
  (isnew⇒
    (:aunder :aon)
  ◇run⇒
    (☞bflag ← true.
     aon run.
     RFLAG⇒
       (☞bflag ← false.
        aunder run.
       ))
  ◇add⇒
    (☞bflag ← true.
     aon add.
     aflag⇒()
     ☞bflag ← false.
     aunder add.
     aflag⇒
       (☞t ← aon.
        ☞aon ← aunder.
        ☞aunder ← t))
  ◇print⇒
    (disp is dispframe⇒()
     disp ← 'append' sp aunder print sp aon print)
  ◇findpix⇒
    (aunder findpix.
     aon findpix))

to at : xc yc atpic
  (isnew⇒
    (:xc :yc :atpic)
  ◇run⇒(atpic run.
    )
  ◇print⇒
    (disp is dispframe⇒()
     disp ← 'at' sp xc print sp yc print sp atpic print)
  ◇findpix⇒(atpic findpix))

to wind : xmin xmax ymin ymax wpic
  (isnew⇒
    (:xmin :xmax :ymin :ymax :wpic)
  ◇run⇒
    (mousein⇒
       (☞RFLAG ← false.
        wpic run)
     ☞RFLAG ← true.
    )
  ◇print⇒
    (disp is dispframe⇒()
     disp ← 'wind' sp xmin print sp xmax print sp ymin print sp ymax print sp wpic print)
  ◇findpix⇒(wpic findpix)
  ◇knows⇒(ev))

to over : under on
  (isnew⇒
    (:under :on)
  ◇run⇒(under run)
  ◇print⇒
    (disp is dispframe⇒()
     disp ← 'over' sp under print sp on print)
  ◇findpix⇒
```

```
    (under findpix.
     on findpix))

to movie nam : xseq yseq pseq xvec yvec pvec frame finc frames f1 f2 xpos ypos minx maxx miny maxy : Menu
  (◁init⇒
    (☞Menu ← menu 3 12 3 12 'moviemenu.' 1 4 3 5 6 14 7 8 9)
   isnew⇒
    (moviesetup.
      (☞f1 ← ☞frame ← 1.
       ◁of⇒
        (☞f2 ← :frames.
         :finc.
          ☞xvec ← :xseq vec.
          ☞yvec ← :yseq vec.
          ☞pvec ← :pseq vec)
        ☞finc ← 1.
        ☞frames ← :.
          (null frames⇒
            (☞f2 ← ☞frames ← 1)
          ☞f2 ← frames).
        ☞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).
      repeat
       (cr.
        disp ← 'Type MOVIE name: '.
        ☞nam ←(read)
        [1].
        null nam⇒()
        done.
       ).
      MOVIES ← nam.
      nam ← SELF.
      SELF wakeup)
   ◁wakeup⇒
    (addpic at xpos ypos wind minx maxx miny maxy over SELF outln)
   ◁reset⇒
    (moviesetup.
     SELF wakeup.
    )
   ◁run⇒(mousein⇒
       (Menu select.
        movieupdate.
       ))
   ◁set⇒
    (xseq set.
     yseq set.
     pseq set)
   ◁advance⇒
    (☞frame ← frame + 1.
     frame > f2⇒
      (☞frame ← f1))
   ◁print⇒
    (disp is dispframe⇒()
     disp ← 'movie of ' frames print sp finc print sp xseq print sp yseq print sp pseq print)
   ◁findpix⇒(pseq findpix)
   ◁evals⇒(⇑(:☞)
     eval))

to nilpic : pic
```

```
  (◁add⇒
    (☞aflag ← true.
     bflag⇒
       (☞aon ← npic)
     ☞aunder ← npic)
   ◁print⇒(☞nilpic print)
   isnew⇒
    (☞pic ← 0)
   ◁findpix⇒())

to freshcel : celpic
  (isnew⇒
    (☞celpic ← NPICS)
   ◁wakeup⇒(⇑cel)
   ◁print⇒(☞freshcel print)
   ◁findpix⇒())

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 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 addpic npic aflag
  (:npic.
   ☞aflag ← false.
   display holds add.
   aflag⇒()
   display ← append display holds npic)

to add ()

to movepic
  (cr.
```

```
   disp ← 'MOVE WINDOW'.
   ☞xc ← mx.
   ☞yc ← my.
   repeat
    (down⇒()
     ☞xc ← xm.
     ☞yc ← ym.
     done).
  )

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

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 ←(xm - xtemp)
   + xmax.
   ☞xmin ←(xm - xtemp)
   + xmin.
   ☞ymax ←(ym - ytemp)
   + ymax.
   ☞ymin ←(ym - ytemp)
   + ymin.
   ☞xc ← xtemp.
   ☞yc ← ytemp.
   ☞wpic ← ptemp.
  )

to run ()

to mousein
  ((xmin + xc)
   < xm <(xmax + xc)
  ⇒
    (⇑(ymin + yc)
     < ym <(ymax + yc))
    ⇑false)

to xmrel
  (⇑(mouse 8)
   - xc)

to ymrel
  (⇑(mouse 9)
   - yc)

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 is dispframe⇒()
     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 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].
```

```
to move xt yt pt xx yy j i
  (menuoff.
   cr.
   ☞MOVEMENT print.
   finc = 0⇒
    (☞xseq ← mx.
     ☞yseq ← my.
     repeat(off⇒(done))
     xvec[frame] ← xnrel.
     yvec[frame] ← ynrel.
     update.
     menuon)
   not(finc = 1)
   ⇒(disp ← 'MOVEMENT not available if frame increment not 1 or 0')
   cr.
   disp ← 'Currently ' f1 print.
   disp ← ' to '.
   f2 print.
   disp ← ' are active frames.' getpoints.
   cr.
   disp ← 'MOVEMENT has '(xx end)
   print.
   disp ← ' new frames.' 0 = xx end⇒
    (update.
     disp ← ' MOVEMENT ignored.' menuon)
   disp ← ' How many do you want ?' 0 = ☞j ←(read)
   [1]⇒
    (update.
     disp ← 'MOVEMENT ignored.' menuon).
     (j >(xx end)
     ⇒
       (☞j ← xx end))
   finishup.
   update.
   menuon.
   )

to finishup
   (☞f2 ← f2 +(j - 1).
     (☞xt ← vector frames + j - 1.
      ☞yt ← vector frames + j - 1.
      ☞pt ← vector frames + j - 1).
     (xt[1 to frame - 1] ← xvec[1 to frame - 1].
      yt[1 to frame - 1] ← yvec[1 to frame - 1].
      pt[1 to frame - 1] ← pvec[1 to frame - 1]).
     (xt[frame to
       (frame + j - 1)
     ] ← xx contents[1 to j].
      yt[frame to
       (frame + j - 1)
     ] ← yy contents[1 to j].
      for i ← frame to
       (frame + j - 1)
       (pt[i] ← pvec[frame])).
     (xt[frame + j to frames + j - 1] ← xvec[frame + 1 to frames].
      yt[frame + j to frames + j - 1] ← yvec[frame + 1 to frames].
      pt[frame + j to frames + j - 1] ← pvec[frame + 1 to frames]).
     ☞frames ← frames + j - 1.
     ☞xvec ← xt.
     ☞yvec ← yt.
     ☞pvec ← pt.
     update.
     menuon.
     )
```

```
to getpoints
  (☞xx ← obset 60.
   ☞yy ← obset 60.
   ☞xseq ← mx.
   ☞yseq ← my.
   ☞pseq ← pvec[frame].
   repeat
     (down⇒
       (xx ← xmrel.
        yy ← ymrel.
       )
      off⇒(done)
      60 = xx end⇒(done))) ←
```

```
        ptemp[i] ← pvec[f2].
       ))
     (☞i ← 0.
     ☞xseq ← mx.
     ☞yseq ← my.
     ☞pseq ← ptemp[1].
     repeat
       (down⇒
         (☞i ← i + 1.
          ☞pseq ← ptemp[i].
          xtemp[i] ← xmrel.
          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 movieupdate
  (☞xpos ← xc.
   ☞ypos ← yc.
   ☞minx ← xmin.
   ☞miny ← ymin.
   ☞maxx ← xmax.
   ☞maxy ← ymax.
  )

to draw pict : : gt goto
  (◇init⇒
    (to gt(CODE 36).
     to goto
      (gt 256 +(:)
       256 - :))
   setcursor xc yc.
   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 moviesetup
   (☞xpos ← - 64.
   ☞ypos ← 0.
   ☞miny ← ☞minx ← - WSIZE.
   ☞maxx ← ☞maxy ← WSIZE.
   )
```

```
☞menufns←☞( menu buttoncode menuon clipg clipl menuoff inmenu ⍞ , incol inrow selectit ).
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 buttoncode (⇑☞
   ((movepic)(draw)(down⇒
      (cr.
       disp ← 'SLEEP'.
       bflag⇒
        (☞aon ← nilpic)
       ☞aunder ← aon.
       ☞aon ← nilpic))(move)(singstep)(select)(movewindow)(copy)(changewindow)(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 menuon
  (☞menux ←
   (
    (clipl xc + xmin)
   +
    (clipg xc + xmax)
   - 2 * xc)
```

```
  / 2.
  ☞menuy ←
   (
     (clipl yc + ymin)
    +
     (clipg yc + ymax)
    - 2 * yc)
  / 2.
  ev.
  ☞on ← over outln Menu.
  )

to clipg a
  (
    (☞a ← i.
    )
    > 128⇒(⇑128)
    ⇑a)

to clipl a
  (
    (☞a ← i.
    )
    < - 128⇒(⇑- 128)
    ⇑a)

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
     (∜i⇒
       (⇑vec[1 to i])
       (i = len⇒
         (☞vec ← vec[1 to ☞len ← 2 * len]))
      vec[☞i ← i + 1] ← i))

to , (i)

to incol i j
  (for i to cols do
    (☞j ←
      (-(cols * cwidth)
       / 2)
     +(cwidth *(i - 1)).
     j < emx <(j + cwidth)
    ⇒(⇑i))
   ⇑0)

to inrow i j
```

```
  (for i to rows do
    (☞j ←((rows * rwidth)
        / 2)
      -(rwidth *(i - 1)).
      j > emy >(j - rwidth)
     ⇒(⋔i))
    ⋔0)

to selectit x
  (☞emx ← xmrel - menux.
   ☞emy ← ymrel - menuy.
   inmenu⇒
    (☞x ← incol +(cols *(inrow - 1))
     + 1.
     x > 0⇒
       (buttons[x] eval)))
```

```
☞kaosfns←☞( kaosinit display paint easel mx my mxabs myabs xm ym apic outln active inactive not interpret neg poi **
nt space setcursor ).
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 myab**
s 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 outin
  (isnew▷()
  ◁print▷(disp ← 'outin'))

to active
  (mem 299 ←
    ((:)
     ^+ mem 299))

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

to not
```

```
((:)
⇒(⇑false)
 ⇑true)

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

to neg : n
  (isnew⇒(:in)
   ⇦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 ← t.
   mem q + 10 ← :)
```

```
☞filefns←☞( dansinit usedisp usereader readpic readp writepic picin moviein movieout ).
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 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 writepic f adr mmaxp
  (☞f ← :.
  ☞mmaxp ←
   (mem mem 11 + mem 67)
  - :.
  ☞adr ← mmaxp + mem mmaxp.
  f writeseq adr mem adr)

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 moviein newpix nnewp oldnos M f i
  (☞f ← file :! 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.
  )
```

```
☞tabletfns←☞( tablet down1 off1 ).
to tablet
  (☞down ← #down1.
   ☞off ← #off1.
   ☞button1 ← #down1)

to down1
  (⋔(16384 + 16384)
   =(- 8192)
   ^* mem - 2)

to off1
  (⋔(- 8192)
   =(- 8192)
   ^* mem - 2)
```