[illegible]

```
☞SWITCH ← false

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 celpic
   (☞f ← file(:)
    old⇒
      (☞celpic ←
        (☜.
        ⇒(cel.
            NEXTAPIC)
         ☜apic⇒(:)
         NPICS + :)
       erasecel celpic.
       readpic noprint f celpic.
       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] celpic).
       f close.
      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 tablet
  (☞down ← #down1.
   ☞off ← #off1.
   ☞button1 ← #down1)

to down1
  (⇑(16384 + 16384)
   =(- 8192)
   ⧆mem - 2)

to off1
  (⇑(- 8192)
   =(- 8192)
   ⧆mem - 2)

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 :.
     buttons[1] ← ☞().
     for k ← 2 to(rows * cols)
     + 1 do
      (null :t⇒(done)
       buttons[k] ← buttoncode[t]))
   👁on⇒(menuon SELF)
   👁off⇒(menuoff)
   👁pick ⇒ (mousein ⇒ (1 = mouse 7 ⇒ (selectit)))
   👁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 ← ☞((- 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)
          (paint brush ← 11)))

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

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)))

to round it
  (☞it ← :.
   it > 0⇒
    (⇑4 *(it / 4))
   it < 0⇒
    (⇑(4 *(it / 4))
     - 4)
   ⇑it)

to kaosinit
  ((interpret over 1)(interpret at 2)(interpret mx 3)(interpret my 4)(interpret num
**ber 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)
    (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 ⇑v)
```

```
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 init i ff
  (addto dispframe ☞
    (☜is⇒
      (☜dispframe⇒(⇑SWITCH)
       ☜?⇒(⇑☞dispframe)
       :☞.
       ⇑false)).
   addto obset ☞
    (☜append ⇒ ((size = ☞end ← end + 1 ⇒
                      (☞vec ← vec[1 to ☞size ← size + 10]))
               vec[end]←:.)
     ☜contents⇒
      (⇑vec[1 to end])
     ☜end⇒(⇑end))
   dansinit.
   kaosinit.
   ☞NEXTAPIC ← ☞NPICS ← 12.

  (☞ff←file 'shazammenus.' old ⇒
✓  (for i ← 6 to ~~9~~ do —— 10
     (erasecel i. readpic noprint ff i.))
  disp ← 'you do not have the window menus on your disk
obtain shazammenus' cr)
   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).
     easel load 11.
     easel (-1) 1 ← 3.
     easel 1 1 ← 3.
     easel 0 (-1) ← 3.
     ☞freshsign ← apic NPICS.
     easel load NPICS.
     easel 1 1 ← 3.
     easel(- 1)
     1 ← 3.
     easel(- 1)(- 1)
     ← 3.
     easel 1(- 1)
     ← 3.
    ).
   ~~makemenu.~~
   ☞WSIZE ← 60.
   reset.
   display run.
   disp clear.
   cr.
   disp ← '☺ SHAZAM at your service ☺'.
   cr cr
    (☞defs ← obset 50.
     ☞dansinit ← ☞kaosinit ← ☞buttoncode ← ☞init ← ☞~~makemenu~~←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. ⇑ SELF)
   👁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)
    👁replace ⇒ (☞aon←:.))

to at : xc yc atpic
  (isnew⇒
    (:xc :yc :atpic. ⇑ SELF)
   👁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 it : xseq yseq pseq xvec yvec pvec frame finc frames f1 f2 xpos ypos mi
**nx maxx miny maxy : Menu
  (👁wakeup⇒
     (addpic at xpos ypos wind minx maxx miny maxy over SELF
       (👁noframe⇒(nilpic)
        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.
      sp.
      xpos print.
      sp.
      ypos print.
      sp.
      minx print.
      sp.
      maxx print.
      sp.
      miny print.
      sp.
      maxy print.
    )
   👁findpix⇒(pseq findpix)
   👁erase⇒
    (for it to pvec length
      (erasecel pvec[it] celpic))
   👁evals⇒(⇑(:☞)
     eval)
   👁init⇒
    (☞Menu ← menu 3 12 3 12 7 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] ← freshcel).
       ☞xseq ← seq 0 xvec 1.
       ☞yseq ← seq 0 yvec 1.
       ☞pseq ← seq 0 pvec 1).
     repeat
      (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 freshcel : celpic
  (isnew⇒
    (☞celpic ← NPICS)
   👁wakeup⇒(⇑cel)
   👁print⇒(☞freshcel print)
   👁findpix⇒())

to cel x y : celpic CROSSFLAG : Menu
  (👁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)
   👁init⇒
    (☞Menu ← menu 3 12 3 12 6 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.
     ☞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.
    ☞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 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.
   Menu on.
  )

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

to draw pict bb: : gt goto drawmenu
  (◀init⇒
    (to gt(CODE 36).
     to goto
       (gt 256 +(:)
        256 - :)
     (☞drawmenu ←
           menu 12 10 12 10 10 22 23 24 25 26 15 16 17 18 19 20 21))
   setcursor xc yc.
   cr.
   ☞DRAW print.
    (CROSSFLAG⇒
      (☞on ← over crosshairs outln)
     ☞on ← outln).
   drawmenu on.
   easel load celpic.
   repeat
    (down⇒()
     done).
   repeat
    (drawmenu pick.
     down⇒(paint run)
     paint stop.
     kbck⇒(read eval print)
     off⇒(done))
   setcursor 0 0.
   drawmenu off.
   easel clear.
   Menu on)

to copy pic
  (menuoff.
   cr.
   disp ← 'SHOW PAINT WINDOW'.
   pvec[frame] ← pvec[frame] wakeup.
   pseq store frame pvec[frame].
```

```
    repeat
     (down⇒()
      done).
    Menu on)

to select choice
   (menuoff.
    cr.
    disp ← 'SELECT A PICTURE'.
    ☞choice ← 0.
    repeat
     (off⇒(done)
      button 1⇒
        (display stop.
         copypic.
         pvec[frame] ← CELS[NEXTCEL].
         pseq store frame pvec[frame].
         display run.
         done.
        )
      down⇒
        (☞choice ← choice + 1.
         pvec[frame] ← CELS[choice].
         pseq store frame pvec[frame].
         sp.
         CELS[choice] print.
         choice = NEXTCEL⇒
           (☞choice ← 0))).
    Menu on)

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 9 22 23 24 25.
      )
    Menu select once.
    )

to toneselect : : Menu
    (👁init⇒
      (☞Menu ← menu 3 12 3 12 8 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.
   Menu on.
  )

to getpoints i
  (☞xx ← obset 60.
   ☞yy ← obset 60.
   ☞xseq ← mx.
   ☞yseq ← my.
   ☞pseq ← pvec[frame].
   cr.
   ☞i ← 0.
   repeat
    (down⇒
      (xx append xmrel.
       yy append ymrel.
       (☞i ← i + 1) print.
       sp.
      )
     off⇒(done)
    60 = xx end⇒(done)))

to copypic ff tt
  (☞ff ← file 'TEMP.'.
   ☞tt ←
    (choice = 0⇒(NEXTCEL)
     choice).
   writepic ff CELS[tt] celpic.
   ff rewind.
   cel.
   readpic noprint ff NEXTAPIC.
   ff close.
  )
```