File created: 14-Jun-2024 15:48:55 **{WMEDLEY}<lispusers>SOLITAIRE.;4**

edit by: rmk

changes to: (RECORDS CARD)
(FNS GETCARD MOVECARD UPCARD NXTCARD)

previous date: 24-Aug-2022 08:54:17 {WMEDLEY}<lispusers>SOLITAIRE.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ SOLITAIRECOMS
       [(FNS SOLO SOLITAIRE)
        (FNS CARDIMAGE COUNTCARDS CREATEHAND CREATESTACK DEALDECK FLIPSTACK GETCARD GOODMOVE? HTOS? MOVECARD
             DOMOVE MOVEHS MOVES MOVES1 UPCARD MOVESSS NXTCARD PUSHCARD POSTVALUE SEARCHSTACKS SHOWCARDSTACK
             SHUFFLEDECK STACKLOC STOS? TOPSUITSTACK)
        (FNS HIST ARRAYMAX)
        (DECLARE%: DONTCOPY (FNS SHOWCONFIG PRINTCARDSTACK CARDNAME))
        (DECLARE%: EVAL@COMPILE DONTCOPY (MACROS BOTTOM KINGP STACK TOP)
               (RECORDS CARD CARDSTACK)
               (CONSTANTS (Spades 0)
                      (Clubs 1)
                      (Diamonds 2)
                      (Hearts 3)
                      (CostOfDeck 50)
                      (PayForCard 5)
                      (NStacks 7)
                      (NSuits 4)
                      (CardsPerSuit 13)
                      (TotalCards 52))
               (CONSTANTS (BACKSHADE 52275)
                      (BetweenStacks 2)
                      (Overlap 0.667)
                      (CardWidth 30)
                      (CardHeight 45))
               (GLOBALVARS HAND STACKS SUITSTACKS DECK SOLORESULTS)
               (GLOBALVARS MaxCardMove WaitBetweenMoves))
        (INITRECORDS CARD CARDSTACK)
        (BITMAPS SpadesBits ClubsBits DiamondsBits HeartsBits 10Bits)
        (INITVARS (MaxCardMove 8)
               (WaitBetweenMoves 100)
               (DECK)
               (SOLORESULTS))
        (ADDVARS (IDLE.FUNCTIONS ("Solitaire" ’SOLO])

(DEFINEQ

(SOLO
  [LAMBDA (W)                                                      (* bas%: "15-Jan-86 23:31")
    (if (AND (BOUNDP ’SOLORESULTS)
            (ARRAYP SOLORESULTS))
      else (SETQ SOLORESULTS (ARRAY (ADD1 TotalCards)
                                   ’FIXP 0 0)))
    (if (WINDOWP W)
      else (SETQ W (CREATEW [GETREGION (CONSTANT (ITIMES (IPLUS BetweenStacks (ITIMES NStacks (ADD1 BetweenStacks
                                                                                                    )))
                                                        CardWidth))
                                   (CONSTANT (ITIMES CardHeight (IPLUS 5 (FTIMES CardsPerSuit Overlap]
                         "Just waiting Patiently..."))
          (DSPTEXTURE 1088 W)
          (DSPFONT (FONTCREATE ’HELVETICA 18)
                 W))
    (bind X do (SETQ X (SOLITAIRE W))
               (DISMISS 1500)
               (SETA SOLORESULTS X (ADD1 (ELT SOLORESULTS X)))
               (HIST SOLORESULTS W "Number of cards up")
               (DISMISS 1500])

(SOLITAIRE
  [LAMBDA (SOLOW REPLAY)                                           (* bas%: "15-Dec-85 21:33")
    (DECLARE (SPECVARS SOLOW))
    (CLEARW SOLOW)
    (DEALDECK REPLAY)
    (PROG ((GAMEVALUE (IMINUS CostOfDeck)))
          (DECLARE (SPECVARS GAMEVALUE))
          (POSTVALUE GAMEVALUE)
          (while (OR [for I from NStacks to 1 by -1 thereis (MOVES (STACK I)
                                                                   (STOS? (STACK I]
                    (MOVESSS HAND)
                    (MOVEHS HAND (HTOS? HAND))
                    (for I to NStacks thereis (MOVESSS (STACK I)))
                    (FLIPSTACK HAND))
             do (DISMISS WaitBetweenMoves)))
    (COUNTCARDS])

)

(DEFINEQ

(CARDIMAGE
  [LAMBDA (C)                                                     (* bas%: "15-Jan-86 21:37")
    (PROG [(BM (BITMAPCREATE CardWidth CardHeight))
           (SUITBM (SELECTQ (fetch SUIT of C)
                       (0 SpadesBits)
                       (1 ClubsBits)
                       (2 DiamondsBits)
                       (3 HeartsBits)
                       (SHOULDNT)))
           (RANKBM (if (EQ 10 (fetch (CARD RANK) of C))
                      then 10Bits
                    else (GETCHARBITMAP (SELECTQ (fetch (CARD RANK) of C)
                                             (13 (CHARCODE K))
                                             (12 (CHARCODE Q))
                                             (11 (CHARCODE J))
                                             (IPLUS (fetch (CARD RANK) of C)
                                                    (CHARCODE 0)))
                                (FONTCREATE 'HELVETICA 18]
          (BLTSHADE BLACKSHADE BM 0 0 CardWidth CardHeight 'REPLACE)
          (BLTSHADE WHITESHADE BM 1 1 (IDIFFERENCE CardWidth 2)
                 (IDIFFERENCE CardHeight 2)
                 'REPLACE)
          (BITBLT SUITBM 0 0 BM 2 32 NIL NIL 'INPUT 'REPLACE)
          (BITBLT RANKBM 0 0 BM (IQUOTIENT (IDIFFERENCE CardWidth (fetch BITMAPWIDTH of RANKBM))
                                       2)
                 (IQUOTIENT (IDIFFERENCE CardHeight (fetch BITMAPHEIGHT of RANKBM))
                        2)
                 NIL NIL 'INPUT 'REPLACE)
          (BITBLT SUITBM 0 0 BM 17 3 NIL NIL 'INPUT 'REPLACE)
          (RETURN BM])

(COUNTCARDS
  [LAMBDA NIL                                                     (* bas%: "15-Jan-86 21:37")
    (for S from Spades to Hearts sum (fetch (CARD RANK) of (TOPSUITSTACK S])

(CREATEHAND
  [LAMBDA (F)                                                     (* bas%: "15-Jan-86 23:25")
    (FLIPSTACK (SHOWCARDSTACK (create CARDSTACK
                                      FACEUP _ NIL
                                      FACEDOWN _ (for I from F to TotalCards collect (GETCARD I))
                                      CSX _ (fetch CSX of (STACK 1))
                                      CSY _ (FIX (FTIMES CardHeight 0.6))
                                      XO _ (FIX (FTIMES CardWidth Overlap))
                                      YO _ 0])

(CREATESTACK
  [LAMBDA (N)                                                     (* bas%: "15-Jan-86 23:00")
    (FLIPSTACK (SHOWCARDSTACK (create CARDSTACK
                                      FACEUP _ NIL
                                      FACEDOWN _ (for I from (ADD1 (IQUOTIENT (ITIMES N (SUB1 N))
                                                                          2))
                                                    as J to N collect (GETCARD I))
                                      CSX _ (STACKLOC N NStacks)
                                      CSY _ (ITIMES CardHeight (IPLUS 2 (FTIMES CardsPerSuit Overlap)))
                                      XO _ 0
                                      YO _ (FIX (FTIMES CardHeight (FMINUS Overlap])

(DEALDECK
  [LAMBDA (REDEAL)                                                (* bas%: "11-Jan-86 20:14")
    (OR (AND REDEAL (ARRAYP DECK))
        (SHUFFLEDECK))
    (SETQ SUITSTACKS (ARRAY NSuits 'POINTER (create CARD
                                                  RANK _ 0)
                            0))
    (SETQ STACKS (ARRAY NStacks 'POINTER))
    (for I to NStacks do (SETA STACKS I (CREATESTACK I)))
    (SETQ HAND (CREATEHAND (CONSTANT (ADD1 (IQUOTIENT (ITIMES NStacks (ADD1 NStacks))
                                                   2])

(FLIPSTACK
  [LAMBDA (H)                                                     (* bas%: "29-JUL-82 15:07")
    (if (fetch FACEDOWN of H)
        then (PUSHCARD H (NXTCARD H))
             H
      else NIL])
```

```
(GETCARD
  [LAMBDA (I)                                                    ; Edited 14-Jun-2024 15:48 by rmk
                                                                 (* bas%: "30-JUL-82 19:04")
    (PROG ((C (ELT DECK I)))
          (if (fetch FACE of C)
            else (replace FACE of C with (CARDIMAGE C))
                 (replace (CARD CDSAV) of C with (BITMAPCREATE CardWidth CardHeight)))
          (replace CX of C with (replace CY of C with NIL))
          (RETURN C])

(GOODMOVE?
  [LAMBDA (TOP BOT)                                              (* bas%: "15-Jan-86 21:38")
    (if TOP
        then (AND (EQ (fetch (CARD RANK) of TOP)
                      (ADD1 (fetch (CARD RANK) of BOT)))
                  (NEQ (fetch (CARD COLOR) of TOP)
                       (fetch (CARD COLOR) of BOT)))
      else (KINGP BOT])

(HTOS?
  [LAMBDA (H)                                                    (* bas%: "30-JUL-82 19:30")
    (if (TOP H)
        then (SEARCHSTACKS (TOP H])

(MOVECARD
  [LAMBDA (C X Y)                                                ; Edited 14-Jun-2024 15:46 by rmk
                                                                 (* lmm " 6-Aug-85 00:04")
    (if (fetch CX of C)
        then (DOMOVE (fetch FACE of C)
                     (fetch CX of C)
                     (fetch CY of C)
                     X Y (fetch (CARD CDSAV) of C))
      else (BITBLT SOLOW X Y (fetch (CARD CDSAV) of C)
                  NIL NIL NIL NIL ’INPUT ’REPLACE)
           (BITBLT (fetch FACE of C)
                  NIL NIL SOLOW X Y NIL NIL ’INPUT ’REPLACE))
    (replace CX of C with X)
    (replace CY of C with Y)
    C])

(DOMOVE
  [LAMBDA (IMAGE LEFT BOTTOM NX NY SAVE)                         (* lmm " 6-Aug-85 00:04")
    (PROG (N YWP YFP XWP XFP)
          (SETQ N (IQUOTIENT (IPLUS (IMAX (ABS (IDIFFERENCE NX LEFT))
                                          (ABS (IDIFFERENCE NY BOTTOM)))
                                    (SUB1 MaxCardMove))
                         MaxCardMove))                           (* Number of steps)
          (SETQ XWP (IQUOTIENT (IDIFFERENCE NX LEFT)
                           N))
          (SETQ XFP (IREMAINDER (IDIFFERENCE NX LEFT)
                           N))
          (SETQ YWP (IQUOTIENT (IDIFFERENCE NY BOTTOM)
                           N))
          (SETQ YFP (IREMAINDER (IDIFFERENCE NY BOTTOM)
                           N))
          (bind OLDLEFT OLDLOW (XFC _ 0)
                (YFC _ 0) until (AND (EQ LEFT NX)
                                     (EQ BOTTOM NY))
             do (SETQ OLDLEFT LEFT)
                (SETQ OLDLOW BOTTOM)
                [add LEFT XWP (PROG1 (IQUOTIENT (add XFC XFP)
                                                N)
                                     (SETQ XFC (IREMAINDER XFC N)))]
                [add BOTTOM YWP (PROG1 (IQUOTIENT (add YFC YFP)
                                                  N)
                                       (SETQ YFC (IREMAINDER YFC N)))]
                (BITBLT SAVE 0 0 SOLOW OLDLEFT OLDLOW CardWidth CardHeight ’INPUT ’REPLACE)
                (BITBLT SOLOW LEFT BOTTOM SAVE 0 0 CardWidth CardHeight ’INPUT ’REPLACE)
                (BITBLT IMAGE 0 0 SOLOW LEFT BOTTOM CardWidth CardHeight ’INPUT ’REPLACE])

(MOVEHS
  [LAMBDA (H SN)                                                 (* bas%: "30-JUL-82 19:30")
    (if SN
        then (PUSHCARD SN (pop (fetch FACEUP of H)))
             (OR (TOP H)
                 (FLIPSTACK H))
             SN])

(MOVES
  [LAMBDA (S1 S2)                                                (* bas%: "30-JUL-82 12:47")
```

```
    (if S2
        then (MOVES1 (fetch FACEUP of S1)
                    NIL S2)
             (replace FACEUP of S1 with NIL)
             (FLIPSTACK S1)
             S2])

(MOVES1
  [LAMBDA (L P S2)                                               (* bas%: "30-JUL-82 19:12")
    (if L
        then (MOVES1 (CDR L)
                    (CAR L)
                    S2)
             (UPCARD (CAR L)
                    P)
             (PUSHCARD S2 (CAR L])

(UPCARD
  [LAMBDA (X Y)                                                  ; Edited 14-Jun-2024 15:46 by rmk
                                                                 (* lmm " 6-Aug-85 00:04")

          (* Brings up X image which is assumed to be overlapped by Y image.
          Assumes YOFFSET only)

    (if Y
        then (PROG [(DY (IDIFFERENCE (fetch CY of X)
                                  (fetch CY of Y]
                   (BITBLT (fetch (CARD CDSAV) of X)
                          0 0 (fetch (CARD CDSAV) of Y)
                          0 DY CardWidth (IDIFFERENCE CardHeight DY)
                          'INPUT
                          'REPLACE)
                   (BITBLT SOLOW (fetch CX of X)
                          (fetch CY of X)
                          (fetch (CARD CDSAV) of X)
                          0 0 CardWidth (IDIFFERENCE CardHeight DY)
                          'INPUT
                          'REPLACE)
                   (BITBLT (fetch FACE of X)
                          0 0 SOLOW (fetch CX of X)
                          (fetch CY of X)
                          CardWidth
                          (IDIFFERENCE CardHeight DY)
                          'INPUT
                          'REPLACE])

(MOVESSS
  [LAMBDA (S)                                                    (* bas%: "15-Jan-86 23:23")
    (PROG (STS (TS (TOP S)))
          (DECLARE (USEDFREE GAMEVALUE))
          (AND TS [EQ (fetch (CARD RANK) of TS)
                      (ADD1 (fetch (CARD RANK) of (TOPSUITSTACK (SETQ STS (fetch SUIT of TS]
               (PROGN [SETA SUITSTACKS STS (MOVECARD (pop (fetch FACEUP of S))
                                                  (STACKLOC (ADD1 STS)
                                                         NSuits)
                                                  (IPLUS (fetch CSY of (STACK NStacks))
                                                         (FTIMES CardHeight 1.5]
                      (OR (TOP S)
                          (FLIPSTACK S))
                      (POSTVALUE (add GAMEVALUE PayForCard))
                      (RETURN T])

(NXTCARD
  [LAMBDA (S)                                                    ; Edited 14-Jun-2024 15:46 by rmk
                                                                 (* bas%: "15-Jan-86 21:44")

    (PROG1 (pop (fetch FACEDOWN of S))
        [if (fetch FACEDOWN of S)
          else

          (* Last card up, replace surface of card table and adjust saved image for possibly overlapping exposed card in the pile)

              (BLTSHADE (DSPTEXTURE NIL SOLOW)
                     SOLOW
                     (fetch CSX of S)
                     [IPLUS (fetch CSY of S)
                            (if (OR (NULL (fetch FACEUP of S))
                                   (ZEROP (fetch YO of S)))
                                then 0
                              else (IPLUS CardHeight (fetch YO of S]
                     (if (OR (NULL (fetch FACEUP of S))
                            (ZEROP (fetch XO of S)))
                         then CardWidth
                       else (fetch XO of S))
                      (if (OR (NULL (fetch FACEUP of S))
                              (ZEROP (fetch YO of S)))
                         then CardHeight
                       else (IMINUS (fetch YO of S)))
                     'REPLACE)
               (if (fetch FACEUP of S)
                  then (BLTSHADE (DSPTEXTURE NIL SOLOW)
                              (fetch (CARD CDSAV) of (BOTTOM S))
                              0
                              (IMINUS (fetch YO of S))
                              (IDIFFERENCE CardWidth (fetch XO of S))
                              CardHeight
                              'REPLACE])])
```

```
(PUSHCARD
  [LAMBDA (S C)                                                  (* bas%: "30-JUL-82 14:37")
    [MOVECARD C (IPLUS (fetch XO of S)
                         (if (fetch FACEUP of S)
                             then (fetch CX of (CAR (fetch FACEUP of S)))
                           else (fetch CSX of S)))
           (IPLUS (fetch YO of S)
                  (if (fetch FACEUP of S)
                      then (fetch CY of (CAR (fetch FACEUP of S)))
                    else (fetch CSY of S]
    (push (fetch FACEUP of S)
          C])
```

```
(POSTVALUE
  [LAMBDA (V)                                                    (* bas%: "15-Jan-86 23:25")
    (MOVETO (CONSTANT (FIX (FTIMES 25 Overlap CardWidth)))
           CardHeight SOLOW)
    (DSPFONT (FONTCREATE 'HELVETICA 18)
           SOLOW)
    (BLTSHADE (DSPTEXTURE NIL SOLOW)
           SOLOW
           (DSPXPOSITION NIL SOLOW)
           (IDIFFERENCE (DSPYPOSITION NIL SOLOW)
                  (FONTPROP (DSPFONT NIL SOLOW)
                         'DESCENT))
           1000
           (FONTPROP (DSPFONT NIL SOLOW)
                  'HEIGHT)
           'REPLACE)
    (DSPOPERATION (PROG1 (DSPOPERATION 'PAINT SOLOW)
                      (if (ILESSP V 0)
                          then (printout SOLOW "Down by $" (IMINUS V)
                                      "   ")
                        elseif (ZEROP V)
                          then (printout SOLOW "Dead even!   ")
                        else (printout SOLOW "Ahead by $" V "   ")))
           SOLOW])
```

```
(SEARCHSTACKS
  [LAMBDA (K)                                                    (* bas%: "30-JUL-82 19:19")
    (for I to NStacks when (GOODMOVE? (TOP (STACK I))
                                      K)
       do (RETURN (STACK I])
```

```
(SHOWCARDSTACK
  [LAMBDA (S)                                                    (* lmm " 6-Aug-85 00:04")
    (if (fetch FACEDOWN of S)
        then (BITBLT NIL NIL NIL SOLOW (fetch CSX of S)
                    (fetch CSY of S)
                    CardWidth CardHeight 'TEXTURE 'REPLACE BLACKSHADE)
             (BITBLT NIL NIL NIL SOLOW (ADD1 (fetch CSX of S))
                    (ADD1 (fetch CSY of S))
                    (IDIFFERENCE CardWidth 2)
                    (IDIFFERENCE CardHeight 2)
                    'TEXTURE
                    'REPLACE BACKSHADE))
    S])
```

```
(SHUFFLEDECK
  [LAMBDA NIL                                                    (* bas%: "30-JUL-82 14:08")
    [if (AND (BOUNDP 'DECK)
            (ARRAYP DECK))
      else (SETQ DECK (ARRAY TotalCards 'POINTER))
           (bind (I _ 0) for S from Spades to Hearts
              do (for R to CardsPerSuit do (SETA DECK (add I 1)
                                                (create CARD
                                                       SUIT _ S
                                                       RANK _ R]
    (bind Y for I to TotalCards do    (SETQ Y (RAND 1 TotalCards))
                                      (SETA DECK I (PROG1 (ELT DECK Y)
                                                       (SETA DECK Y (ELT DECK I)))])


(STACKLOC
  [LAMBDA (I N)                                                  (* bas%: "15-Jan-86 22:21")
    (IPLUS [ITIMES I (FIXR (FQUOTIENT (IDIFFERENCE (WINDOWPROP SOLOW 'WIDTH)
                                              (ITIMES N CardWidth))
                                   (ADD1 N]
           (ITIMES CardWidth (SUB1 I])


(STOS?
  [LAMBDA (SN)                                                   (* bas%: " 7-JAN-81 22:01")
    (AND (fetch FACEUP of SN)
         [OR (fetch FACEDOWN of SN)
           (NOT (KINGP (BOTTOM SN]
         (SEARCHSTACKS (BOTTOM SN])


(TOPSUITSTACK
  [LAMBDA (I)                                                    (* bas%: " 4-JAN-81 01:39")
    (ELT SUITSTACKS I])

)

(DEFINEQ

(HIST
  [LAMBDA (A W L)                                                (* bas%: "15-Dec-85 20:22")
    (PROG ((WH (WINDOWPROP W 'HEIGHT))
           (WW (WINDOWPROP W 'WIDTH))
           (HM NIL)
           (VM (IPLUS (FONTPROP (DSPFONT NIL W)
                                 'HEIGHT)
                      4)))
          (SETQ HM VM)                                           (* Margins could be different eg if Y labels were used)
          (BLTSHADE WHITESHADE W 0 0 WW WH 'REPLACE)
          [PROG [(HS (IQUOTIENT (IDIFFERENCE WW (ITIMES HM 2))
                                (ARRAYSIZE A)))
                 (VS (FQUOTIENT (IDIFFERENCE WH (ITIMES VM 2))
                                (ARRAYMAX A]
                (for I from (ARRAYORIG A) to (IPLUS (ARRAYSIZE A)
                                                   (ARRAYORIG A)
                                                   -1)
                   do (BLTSHADE GRAYSHADE W (IPLUS HM (ITIMES I HS))
                                 VM HS (FIX (FTIMES VS (ELT A I)))
                                 'REPLACE]
          (DRAWLINE HM VM (IDIFFERENCE WW HM)
                 VM 2 'REPLACE W)
          (DRAWLINE HM VM HM (IDIFFERENCE WH VM)
                 2
                 'REPLACE W)
          (MOVETO (IDIFFERENCE (IDIFFERENCE WW HM)
                         (STRINGWIDTH L (DSPFONT NIL W)))
                 (IPLUS (FONTPROP (DSPFONT NIL W)
                               'DESCENT)
                        2)
                 W)
          (PRIN1 L W])


(ARRAYMAX
  [LAMBDA (A)                                                    (* bas%: " 5-AUG-82 14:59")
    (bind (M _ 0) for I from (ARRAYORIG A) to (IPLUS (ARRAYSIZE A)
                                                    (ARRAYORIG A)
                                                    -1)
       when (LESSP M (ELT A I)) do (SETQ M (ELT A I)) finally (RETURN M])

)

(DECLARE%: DONTCOPY

(DEFINEQ

(SHOWCONFIG
  [LAMBDA NIL                                                    (* bas%: "30-JUL-82 19:20")
    (printout NIL "Suits: " 10)
    (for I from Spades to Hearts do (printout NIL (CARDNAME (TOPSUITSTACK I))
                                          %,))
    (TERPRI)
    (for I to NStacks do (PRINTCARDSTACK (STACK I)
                               (CONCAT "Stack " I)))
    (PRINTCARDSTACK HAND "Hand"])
```

```
(PRINTCARDSTACK
  [LAMBDA (X S)                                                   (* bas%: " 6-JAN-81 16:47")
    (printout NIL S ":" 10 .I2 (LENGTH (fetch FACEDOWN X))
           " down. Up: ")
    (for J in (fetch FACEUP of X) do (printout NIL (CARDNAME J)
                                                %,))
    (TERPRI])

(CARDNAME
  [LAMBDA (C)                                                     (* bas%: "15-Jan-86 21:40")
    (if (ZEROP (fetch (CARD RANK) of C))
        then "None"
      else (PACK (LIST (SELECTQ (fetch (CARD SUIT) of C)
                            (0 'S)
                            (1 'C)
                            (2 'D)
                            (3 'H)
                            (SHOULDNT))
                       (SELECTQ (fetch (CARD RANK) of C)
                            (1 'A)
                            (11 'J)
                            (12 'Q)
                            (13 'K)
                            (fetch (CARD RANK) of C])
)
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS BOTTOM MACRO [(S)
                        (CAR (LAST (fetch FACEUP of S])

(PUTPROPS KINGP MACRO ((C)
                       (EQ CardsPerSuit (fetch (CARD RANK) of C))))

(PUTPROPS STACK MACRO ((N)
                       (ELT STACKS N)))

(PUTPROPS TOP MACRO ((S)
                     (CAR (fetch FACEUP of S))))
)

(DECLARE%: EVAL@COMPILE

[DATATYPE CARD (SUIT RANK FACE CDSAV CX CY)
       (ACCESSFNS CARD (COLOR (ILESSP (fetch SUIT of DATUM)
                                     Diamonds]

(RECORD CARDSTACK (FACEUP FACEDOWN CSX CSY XO YO))
)

(/DECLAREDATATYPE 'CARD '(POINTER POINTER POINTER POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '12)

(DECLARE%: EVAL@COMPILE

(RPAQQ Spades 0)

(RPAQQ Clubs 1)

(RPAQQ Diamonds 2)

(RPAQQ Hearts 3)

(RPAQQ CostOfDeck 50)

(RPAQQ PayForCard 5)

(RPAQQ NStacks 7)

(RPAQQ NSuits 4)

(RPAQQ CardsPerSuit 13)

(RPAQQ TotalCards 52)

(CONSTANTS (Spades 0)
       (Clubs 1)
       (Diamonds 2)
       (Hearts 3)
       (CostOfDeck 50)
       (PayForCard 5)
       (NStacks 7)
       (NSuits 4)
       (CardsPerSuit 13)
       (TotalCards 52))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ BACKSHADE 52275)

(RPAQQ BetweenStacks 2)

(RPAQQ Overlap 0.667)

(RPAQQ CardWidth 30)

(RPAQQ CardHeight 45)

(CONSTANTS (BACKSHADE 52275)
       (BetweenStacks 2)
       (Overlap 0.667)
       (CardWidth 30)
       (CardHeight 45))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS HAND STACKS SUITSTACKS DECK SOLORESULTS)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MaxCardMove WaitBetweenMoves)
)
)

(/DECLAREDATATYPE 'CARD '(POINTER POINTER POINTER POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '12)

(RPAQQ SpadesBits ♠)

(RPAQQ ClubsBits ♣)

(RPAQQ DiamondsBits ◇)

(RPAQQ HeartsBits ♡)

(RPAQQ 10Bits 10)

(RPAQ? MaxCardMove 8)

(RPAQ? WaitBetweenMoves 100)

(RPAQ? DECK )

(RPAQ? SOLORESULTS )

(ADDTOVAR IDLE.FUNCTIONS ("Solitaire" 'SOLO))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

MACRO INDEX

RECORD INDEX