File created: 24-Aug-2022 08:54:17 **{DSK}<home>larry>medley>lispusers>PAC-MAN-IDLE.;2**

changes to: (FNS Pac-Man-Eat-Window Pac-Man-Scout-Food)

previous date: 30-Jun-86 18:01:00 {DSK}<home>larry>medley>lispusers>PAC-MAN-IDLE.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1986 by Xerox Corporation.

(RPAQQ PAC-MAN-IDLECOMS
       [(* * The Pac-Man idle function)
        (FNS Pac-Man-Eat-Window Pac-Man-Idle)
        (VARS DefaultPacManEatMask DefaultPacManIcon DefaultPacManMask)
        (INITVARS (Pac-Man-Delay 100)
               (pacManHorizonFactor 0.75)
               (pacManStarvationTime 75)
               (pacManEatMask DefaultPacManEatMask)
               (pacManIcon DefaultPacManIcon)
               (pacManMask DefaultPacManMask))
        (GLOBALVARS Pac-Man-Delay pacManHorizonFactor pacManStarvationTime pacManEatMask pacManIcon pacManMask)
        (FNS Pac-Man-Scout-Food)
        (* * Stuff for counting the bits on in a bitmap)
        (FNS Pac-Man-Amount-Of-Food)
        (MACROS Pac-Man-Convert-Word)
        (VARS (Pac-Man-Convert-Byte-Array (\ALLOCBLOCK 256 T)))
        (GLOBALVARS Pac-Man-Convert-Byte-Array)
        [P (for i from 0 to 255 do (\PUTBASE Pac-Man-Convert-Byte-Array i (bind (j _ i)
                                                                               while
                                                                               (NOT (ZEROP j))
                                                                               count
                                                                               (SETQ j (LOGAND j (SUB1 j]
        (* * Another idle function)
        (FNS Slow-Fade)
        [INITVARS (Slow-Fade-Delay 1000)
               (Default-Slow-Fade-Idle-Function (LISTGET IDLE.PROFILE 'DISPLAYFN]
        (GLOBALVARS Slow-Fade-Delay Default-Slow-Fade-Idle-Function)
        (* * Add them as idle functions)
        (ADDVARS (IDLE.FUNCTIONS ("Pac-man" 'Pac-Man-Idle)
                       ("Slow fade" 'Slow-Fade])


          (* * The Pac-Man idle function)

(DEFINEQ

(Pac-Man-Eat-Window
  [LAMBDA (window)                                          (* smL "30-Jun-86 17:38")

          (* * Comment)

    (RESETLST
        (bind (minX _ (MINUS (QUOTIENT (BITMAPWIDTH pacManIcon)
                                       2)))
              (minY _ (MINUS (QUOTIENT (BITMAPHEIGHT pacManIcon)
                                       2)))
              (maxX _ (DIFFERENCE (WINDOWPROP window 'WIDTH)
                                  (QUOTIENT (BITMAPWIDTH pacManIcon)
                                       2)))
              (maxY _ (DIFFERENCE (WINDOWPROP window 'HEIGHT)
                                  (QUOTIENT (BITMAPHEIGHT pacManIcon)
                                       2)))
              (minimumSpeed _ 0.2)
              (maximumSpeed _ 1.0)
              (icon _ (DEFERREDCONSTANT (BITMAPCOPY pacManIcon)))
              (delayTimer _ (DEFERREDCONSTANT (SETUPTIMER 250)))
              [horizon _ (FIX (MAX 2 (TIMES (MIN (BITMAPWIDTH pacManIcon)
                                                 (BITMAPHEIGHT pacManIcon))
                                            pacManHorizonFactor]
              (delta _ '(0 . 0))
              possibleDeltas x y (xSpeed _ 0)
              (ySpeed _ 0)
              [maxSpeed _ (TIMES 10 (MIN (BITMAPWIDTH pacManIcon)
                                         (BITMAPHEIGHT pacManIcon]
              (maxAcceleration _ (MIN (BITMAPWIDTH pacManIcon)
                                      (BITMAPHEIGHT pacManIcon)))
              (stepsWithoutFood _ 0) first [SETQ possibleDeltas (for pair in '((0 . -1)
                                                                               (-1 . 0)
                                                                               (1 . 0)
                                                                               (0 . 1)
                                                                               (0.707 . 0.707)
```

```
                                                        (-0.707 . 0.707)
                                                        (0.707 . -0.707)
                                                        (-0.707 . -0.707))
                                                  collect (CONS (FIX (TIMES horizon (CAR pair)))
                                                                (FIX (TIMES horizon (CDR pair]
                                                  (* Pick a random starting place)
                                    (SELECTQ (RAND 0 1)
                                         (0 (SETQ x (TIMES (WINDOWPROP window 'WIDTH)
                                                           (RAND 0 1)))
                                            [SETQ y (RAND 0 (WINDOWPROP window 'HEIGHT])
                                         (1 [SETQ x (RAND 0 (WINDOWPROP window 'WIDTH]
                                            (SETQ y (TIMES (WINDOWPROP window 'WIDTH)
                                                           (RAND 0 1))))
                                         NIL)
          while T do

          (* * Try to figure out which direction to go. Pick the one that would get us the most food.
          Make sure to block, and don't move to quickly (hah!))

                    (SETQ delayTimer (SETUPTIMER Pac-Man-Delay delayTimer))
                    [SETQ delta (Pac-Man-Scout-Food window x y pacManEatMask possibleDeltas delta
                                        (DEFERREDCONSTANT (BITMAPCREATE
                                                                (PLUS (TIMES 16 (QUOTIENT (BITMAPWIDTH pacManMask)
                                                                                          16))
                                                                      (if (ZEROP (REMAINDER (BITMAPWIDTH pacManMask)
                                                                                            16))
                                                                          then 0
                                                                        else 16))
                                                                (BITMAPHEIGHT pacManMask]
                    (COND
                       ((NOT (NULL delta))                          (* Found some food)
                        NIL)
                       ((GREATERP stepsWithoutFood pacManStarvationTime)
                                                                    (* Starving, so make a random jump)
                        (change xSpeed (RAND (DIFFERENCE minX x)
                                             (DIFFERENCE maxX x)))
                        (change ySpeed (RAND (DIFFERENCE minY y)
                                             (DIFFERENCE maxY y)))
                        (SETQ stepsWithoutFood 0)
                        (SETQ delta (CONS xSpeed ySpeed)))
                       (T (add stepsWithoutFood 1)
                          (change xSpeed (RAND (MINUS maxAcceleration)
                                               maxAcceleration))
                          (change xSpeed (MAX (DIFFERENCE minX x)
                                              (MIN (DIFFERENCE maxX x)
                                                   DATUM)))
                          (change ySpeed (RAND (MINUS maxAcceleration)
                                               maxAcceleration))
                          (change ySpeed (MAX (DIFFERENCE minY y)
                                              (MIN (DIFFERENCE maxY y)
                                                   DATUM)))
                          (SETQ delta (CONS xSpeed ySpeed)))
                       (T (SETQ stepsWithoutFood 0)
                          (SETQ xSpeed 0)
                          (SETQ ySpeed 0)))
                    (do (BLOCK) repeatuntil (TIMEREXPIRED? delayTimer))

          (* * Eat the food at the current location)

                    (BITBLT pacManEatMask NIL NIL window x y NIL NIL 'INPUT 'ERASE)

          (* * Update my location)

                    [change x (FIX (MAX minX (MIN maxX (PLUS DATUM (TIMES (RAND minimumSpeed maximumSpeed)
                                                                          (CAR delta]
                    [change y (FIX (MAX minY (MIN maxY (PLUS DATUM (TIMES (RAND minimumSpeed maximumSpeed)
                                                                          (CDR delta]
                    (BITBLT window x y icon NIL NIL NIL NIL 'INPUT 'REPLACE)
                    (BITBLT pacManMask NIL NIL icon NIL NIL NIL NIL 'INPUT 'ERASE)
                    (BITBLT pacManIcon NIL NIL icon NIL NIL NIL NIL 'INPUT 'PAINT)
                    (BITBLT icon NIL NIL window x y NIL NIL 'INPUT 'REPLACE))])

(Pac-Man-Idle
  [LAMBDA (window)                                              (* smL "30-Jun-86 17:41")

          (* * A hungry idle function)

    (BITBLT (WINDOWPROP window 'IMAGECOVERED)
           NIL NIL window NIL NIL NIL NIL 'INVERT 'REPLACE)
    (Pac-Man-Eat-Window window])

)

(RPAQQ DefaultPacManEatMask 
)
```

```
(RPAQQ DefaultPacManIcon
```

```
)

(RPAQQ DefaultPacManMask
```

```
)

(RPAQ? Pac-Man-Delay 100)

(RPAQ? pacManHorizonFactor 0.75)

(RPAQ? pacManStarvationTime 75)

(RPAQ? pacManEatMask DefaultPacManEatMask)

(RPAQ? pacManIcon DefaultPacManIcon)

(RPAQ? pacManMask DefaultPacManMask)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS Pac-Man-Delay pacManHorizonFactor pacManStarvationTime pacManEatMask pacManIcon pacManMask)
)

(DEFINEQ

## (Pac-Man-Scout-Food

```
  [LAMBDA (window x y mask possibleDeltas prevDelta tempBitMap) (* smL "29-Apr-86 12:55")

          (* * Return the x-y pair of directions to go to get the most food)

    (for i from 1 to 8 bind direction
       thereis [SETQ direction (for offsetPair in [for x in possibleDeltas collect (CONS (TIMES i (CAR x))
                                                                                          (TIMES i (CDR x]
                                    bind xoffset yoffset amountOfFood (mostFood _ 0)
                                      (mostFoodDirections _ NIL)
                                    do (SETQ xoffset (CAR offsetPair))
                                       (SETQ yoffset (CDR offsetPair))

          (* * Build a bitmap of the food available at the location. -
          This requires computing the number of bits that are black both in the window and in the mask.
          -
          We want black bits in the window because things have been inverted by idle and we are trying to eat white bits, and we
          want black bits in the mask because that is what defines the mask.)
                                                                  (* Copy the screen bits into the temp bitmap.)
                                    (BITBLT NIL NIL NIL tempBitMap NIL NIL NIL NIL 'TEXTURE 'REPLACE WHITESHADE)
                                    (BITBLT window (PLUS xoffset x)
                                           (PLUS yoffset y)
                                           tempBitMap NIL NIL NIL NIL 'INPUT 'REPLACE)
                                                                  (* Or in the white bits of the mask at the appropriate location.)
                                    (BITBLT mask NIL NIL tempBitMap NIL NIL NIL NIL 'INVERT 'ERASE)
                                                                  (* Clear out the image of the current position of the mask.)
                                    (BITBLT NIL NIL NIL tempBitMap NIL NIL NIL NIL 'TEXTURE 'INVERT BLACKSHADE)
                                    (BITBLT mask (MAX 0 xoffset)
                                           (MAX 0 yoffset)
                                           tempBitMap
                                           (MAX 0 (MINUS xoffset))
                                           (MAX 0 (MINUS yoffset))
                                           NIL NIL 'INPUT 'PAINT)
                                    (BITBLT NIL NIL NIL tempBitMap NIL NIL NIL NIL 'TEXTURE 'INVERT BLACKSHADE)
                                                                  (* Compute the amount of food)
                                    (SETQ amountOfFood (Pac-Man-Amount-Of-Food tempBitMap))
                                                                  (* Remember the directions with the most food)
                                    (if (LESSP amountOfFood mostFood)
                                        then   (* This direction loses)
                                             NIL
                                      elseif (EQP amountOfFood mostFood)
                                        then   (* This is a possible direction)
                                             (push mostFoodDirections offsetPair)
                                      else   (* This direction dominates)
                                           (SETQ mostFood amountOfFood)
                                           (SETQ mostFoodDirections (LIST offsetPair)))
                                 finally (RETURN (if (ZEROP mostFood)
                                                     then NIL
                                                   else (CAR (NTH mostFoodDirections (RAND 1 (LENGTH
                                                                                               mostFoodDirections
                                                                                               ]
       finally (RETURN direction])

)

          (* * Stuff for counting the bits on in a bitmap)
```
```

```
(DEFINEQ

(Pac-Man-Amount-Of-Food
  [LAMBDA (bitMap)                                                    (* smL "29-Apr-86 13:23")

          (* * How much food is there in the bitmap?)

    (for j from 0 to (QUOTIENT (TIMES (BITMAPHEIGHT bitMap)
                                      (BITMAPWIDTH bitMap))
                               16)
       bind (bitmapBase _ (fetch (BITMAP BITMAPBASE) of bitMap)) sum (Pac-Man-Convert-Word (\GETBASE bitmapBase j
                                                                                                ])

)

(DECLARE%: EVAL@COMPILE

(PROGN (DEFMACRO Pac-Man-Convert-Word (word)

          (* * Count up the number of bits on in the word)

          `(PLUS (\GETBASE Pac-Man-Convert-Byte-Array (LRSH %, word 8))
                 (\GETBASE Pac-Man-Convert-Byte-Array (LOGAND %, word 255))))
       NIL)
)

(RPAQ Pac-Man-Convert-Byte-Array (\ALLOCBLOCK 256 T))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS Pac-Man-Convert-Byte-Array)
)

[for i from 0 to 255 do (\PUTBASE Pac-Man-Convert-Byte-Array i (bind (j _ i) while (NOT (ZEROP j))
                                                                  count (SETQ j (LOGAND j (SUB1 j]


          (* * Another idle function)

(DEFINEQ

(Slow-Fade
  [LAMBDA (window)                                                    (* smL "30-Jun-86 17:16")

          (* * Slowly fade the idle window to black)

    (BITBLT (WINDOWPROP window 'IMAGECOVERED)
            NIL NIL window NIL NIL NIL NIL 'INVERT 'REPLACE)
    [LET [(fadeTextures (for i from 0 to 15 collect (LLSH 1 i]
         (while fadeTextures bind selectedTexture do (BLOCK Slow-Fade-Delay)
                                                     (SETQ selectedTexture (LLSH 1 (RAND 0 15)))
                                                     (BITBLT NIL NIL NIL window NIL NIL NIL NIL 'TEXTURE
                                                            'ERASE selectedTexture)
                                                     (SETQ fadeTextures (DREMOVE selectedTexture fadeTextures]
    (BLOCK Slow-Fade-Delay)
    (APPLY* (OR Default-Slow-Fade-Idle-Function (FUNCTION IDLE.BOUNCING.BOX))
            window])

)

(RPAQ? Slow-Fade-Delay 1000)

(RPAQ? Default-Slow-Fade-Idle-Function (LISTGET IDLE.PROFILE 'DISPLAYFN))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS Slow-Fade-Delay Default-Slow-Fade-Idle-Function)
)


          (* * Add them as idle functions)

(ADDTOVAR IDLE.FUNCTIONS ("Pac-man" 'Pac-Man-Idle)
                         ("Slow fade" 'Slow-Fade))

(PUTPROPS PAC-MAN-IDLE COPYRIGHT ("Xerox Corporation" 1986))
```

FUNCTION INDEX

VARIABLE INDEX