

File created: 5-Dec-2020 16:24:55 {DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>LDE>ROOM
S>MEDLEY-35>ROOMS-D.;2

previous date: 17-Aug-90 12:43:06 {DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>LDE>ROOMS>MEDLEY-35>ROOMS-D
.;1

Read Table: XCL

Package: ROOMS

Format: XCCS

; Copyright (c) 1987, 1988, 1990, 2020 by Venue & Xerox Corporation. All rights reserved.

```
(IL:RPAQQ IL:ROOMS-DCOMS
;; Rooms' interface to Interlisp-D window system
(FILE-ENVIRONMENTS IL:ROOMS-D)
(IL:P (EXPORT '(*WHO-LINE-ENTRY*)))
(IL:TYPES BITMAP FONT TEXTURE)

;; windows
(IL:FUNCTIONS MOVE-WINDOW SHAPE-WINDOW OPEN-WINDOW CLOSE-WINDOW)
(IL:FUNCTIONS WINDOW-REGION MAIN-WINDOW WINDOW-TITLE WINDOW-VISIBLE-P)
;; regions
(IL:STRUCTURES REGION)
(IL:FUNCTIONS
  MAKE-POSITION POSITION-X POSITION-Y GET-POSITION)
;; positions
(IL:FUNCTIONS
  SHRINK-WINDOW EXPAND-WINDOW ICON? SHRUNKEN? ICON-POSITION WINDOW-POSITION WINDOW-ICON
  DELETE-WINDOW-ICON)
;; icons
;; user interface
(IL:FUNCTIONS MENU PROMPT-USER CONFIRM NOTIFY-USER GET-MESSAGE-STREAM SELECT-WINDOW SELECT-BAGGAGE
  EXTERNALIZE-FONT)
(IL:FUNCTIONS
  ;; keyboard interpretation
  ;; these have gotten out of control. it might be worth converting these to one function which returns a keyword naming the
  ;; selected operation. EDITCALLERS, anyone?
  COPY-KEY-DOWN-P HELP-KEY-DOWN-P DELETE-KEY-DOWN-P EDIT-KEY-DOWN-P MOVE-KEY-DOWN-P
  EXPAND-KEY-DOWN-P)
(IL:COMS
  ;; add a lafite form for bug reports
  (IL:FUNCTIONS MAKE-ROOMS-SUPPORT-FORM)
  (IL:VARIABLES IL:ROOMSSUPPORT)
  (IL:ADDVARS (IL:LAFITESPECIALFORMS ("Rooms Report" (IL:FUNCTION MAKE-ROOMS-SUPPORT-FORM)
    "A form to report a Rooms bug or suggestion")))
  (IL:P (SETQ IL:LAFITEFORMSMENU NIL))
  ;; provide a who line entry
  (IL:VARIABLES *WHO-LINE-ENTRY*)
  (IL:P (WHEN (BOUNDP 'IL:*WHO-LINE-ENTRY-REGISTRY*)
    (PUSHNEW *WHO-LINE-ENTRY* IL:*WHO-LINE-ENTRY-REGISTRY* :TEST 'EQUAL :KEY
      'CAR))))
  (IL:GLOBALVARS IL:PROMPTWINDOW IL:CROSSHAIRS IL:WINDOWBACKGROUNDSHADE IL:WHOLESCREEN
    IL:LAFITESPECIALFORMS IL:LAFITEFORMSMENU IL:DEFAULTICONFN IL:*WHO-LINE-ENTRY-REGISTRY*))
  )
)
```

;; Rooms' interface to Interlisp-D window system

```
(DEFINE-FILE-ENVIRONMENT IL:ROOMS-D :COMPILER :COMPILE-FILE
:PACKAGE "ROOMS"
:READTABLE "XCL")
```

```
(EXPORT '(*WHO-LINE-ENTRY*))
```

```
(DEFTYPE BITMAP ()
  `(SATISFIES IL:BITMAP))
```

```
(DEFTYPE FONT ()
  `(SATISFIES IL:FONT))
```

```
(DEFTYPE TEXTURE ()
  `(SATISFIES IL:TEXTURE))
```

;; windows

```
(DEFUN MOVE-WINDOW (WINDOW POS &OPTIONAL (CURRENT-REGION (WINDOW-REGION WINDOW)))
  (UNLESS (IL:EQMEMB 'IL:DON'T (IL:WINDOWPROP WINDOW 'IL:MOVEFN))
    (LET ((CURRENT-MAIN-WINDOW-REGION (IL:WINDOWPROP WINDOW 'IL:REGION)))
      ;; have to compensate for (possible) windows attached at left or bottom. IL:SHAPEW does this for us, but not IL:MOVEW...
      (IL:MOVEW WINDOW (+ (POSITION-X POS)
        (- (REGION-LEFT CURRENT-MAIN-WINDOW-REGION))
      )
    )
  )
)
```

```

                (REGION-LEFT CURRENT-REGION)))
    (+ (POSITION-Y POS)
      (- (REGION-BOTTOM CURRENT-MAIN-WINDOW-REGION)
         (REGION-BOTTOM CURRENT-REGION))))))

(defun SHAPE-WINDOW (WINDOW DESIRED-REGION &KEY (CURRENT-REGION (WINDOW-REGION WINDOW))
                    NO-SHAPE)

  ;; open up IL:SHAPEW a bit
  (MULTIPLE-VALUE-BIND (VALUE CONDITION)
    (IGNORE-ERRORS (IF (OR
      ;; if we don't really need to reshape
      (AND (= (REGION-WIDTH DESIRED-REGION)
              (REGION-WIDTH CURRENT-REGION))
            (= (REGION-HEIGHT DESIRED-REGION)
              (REGION-HEIGHT CURRENT-REGION)))
      ;; or we're not supposed to reshape
      (IL:\\USERFNISSDON\\'T (IL:WINDOWPROP WINDOW 'IL:RESHAPEFN))
      NO-SHAPE)
      ;; then just move
      (MOVE-WINDOW WINDOW (MAKE-POSITION (REGION-LEFT DESIRED-REGION)
                                           (REGION-BOTTOM DESIRED-REGION))
                    CURRENT-REGION)
      ;; otherwise do the reshape
      (FUNCALL (OR (IL:WINDOWPROP WINDOW 'IL:DOSHAPEFN)
                   'IL:SHAPEW1)
               WINDOW
               (COPY-REGION DESIRED-REGION))))
    (WHEN CONDITION (NOTIFY-USER "Error reshaping ~A: ~A" WINDOW CONDITION)
      VALUE))

(defmacro OPEN-WINDOW (WINDOW)
  `(IL:OPENW ,WINDOW))

(defmacro CLOSE-WINDOW (WINDOW)
  `(IL:CLOSEW ,WINDOW))

(defmacro WINDOW-REGION (WINDOW)
  `(IL:WINDOWREGION ,WINDOW))

(defmacro MAIN-WINDOW (WINDOW)
  `(LET ((WINDOW ,WINDOW))
    (OR (IL:WINDOWPROP WINDOW 'IL:ICONFOR)
        (IL:MAINWINDOW WINDOW T))))

(defmacro WINDOW-TITLE (WINDOW)
  `(IL:WINDOWPROP ,WINDOW 'IL:TITLE))

(defun WINDOW-VISIBLE-P (WINDOW)
  (AND (IL:OPENWP WINDOW)
        (IL:REGIONSINTERSECTP (WINDOW-REGION WINDOW)
                                IL:WHOLESCREEN)))

;; regions

(defstruct (REGION (:TYPE LIST))

  ;; overlay onto an Interlisp-D region, so we don't have to use il:fetch cruft.

  LEFT
  BOTTOM
  WIDTH
  HEIGHT)

(defmacro MAKE-POSITION (X Y)
  `(CONS ,X ,Y))

(defmacro POSITION-X (POS)
  `(CAR ,POS))

(defmacro POSITION-Y (POS)
  `(CDR ,POS))

```

```
(DEFUN GET-POSITION (MESSAGE &REST MESSAGE-ARGS)
  (APPLY #'NOTIFY-USER MESSAGE MESSAGE-ARGS)
  (IL:GETPOSITION))
```

```
(DEFMACRO SHRINK-WINDOW (WINDOW POS)
  `(IL:SHRINKW ,WINDOW NIL ,POS))
```

```
(DEFUN EXPAND-WINDOW (WINDOW)
  `(IL:EXPANDW (WINDOW-ICON ,WINDOW)))
```

```
(DEFMACRO ICON? (WINDOW)
  `(IL:WINDOWPROP ,WINDOW 'IL:ICONFOR))
```

```
(DEFUN SHRUNKEN? (WINDOW)
  (IL:EQMEMB (IL:FUNCTION IL:CLOSEICONWINDOW)
    (IL:WINDOWPROP WINDOW 'IL:OPENFN)))
```

```
(DEFUN ICON-POSITION (WINDOW)
  (LET ((ICON-WINDOW (WINDOW-ICON WINDOW)))
    (WHEN ICON-WINDOW (WINDOW-POSITION ICON-WINDOW))))
```

```
(DEFUN WINDOW-POSITION (WINDOW)
  (LET ((REGION (WINDOW-REGION WINDOW)))
    (MAKE-POSITION (REGION-LEFT REGION)
      (REGION-BOTTOM REGION))))
```

```
(DEFMACRO WINDOW-ICON (WINDOW)
  `(IL:WINDOWPROP ,WINDOW 'IL:ICONWINDOW))
```

```
(DEFUN DELETE-WINDOW-ICON (WINDOW)
```

```
;;; delete the icon for WINDOW, if any. We know WINDOW is expanded.
```

```
  (IL:WINDOWPROP WINDOW 'IL:ICONWINDOW NIL)
  (IL:WINDOWPROP WINDOW 'IL:ICONPOSITION NIL))
```

```
;; user interface
```

```
(DEFUN MENU (ITEMS &OPTIONAL TITLE MESSAGE &REST MESSAGE-ARGS)
  (WHEN MESSAGE
    (APPLY #'NOTIFY-USER MESSAGE MESSAGE-ARGS))
  (IL:MENU (IL:CREATE IL:MENU
    IL:ITEMS IL:_ ITEMS
    IL:TITLE IL:_ TITLE
    IL:CENTERFLG IL:_ T)))
```

```
(DEFUN PROMPT-USER (PROMPT &OPTIONAL MESSAGE &REST MESSAGE-ARGS)
```

```
;;; prompt the user for a string. input should end when CR is typed.
```

```
  (WHEN MESSAGE
    (APPLY #'NOTIFY-USER MESSAGE MESSAGE-ARGS))
  (IL:RESETFORM (IL:TTYDISPLAYSTREAM (GET-MESSAGE-STREAM))
    (IL:PROMPTFORWARD PROMPT NIL NIL NIL NIL 'IL:TTY (IL:CHARCODE (IL:EOL)))))
```

```
(DEFUN CONFIRM (&OPTIONAL MESSAGE &REST MESSAGE-ARGS)
```

```
;; make sure prompt-window is un-hidden
```

```
(LET ((STREAM (GET-MESSAGE-STREAM)))
```

```
;; use IL:MOUSECONFIRM
```

```
  (PROG2 (TERPRI STREAM)
    (IL:MOUSECONFIRM (WHEN MESSAGE
      (APPLY #'FORMAT NIL MESSAGE MESSAGE-ARGS))
      NIL STREAM T)
    (TERPRI STREAM)))
```

```
(DEFUN NOTIFY-USER (FORMAT-STRING &REST ARGS)
```

```
  (LET ((STREAM (GET-MESSAGE-STREAM)))
    (TERPRI STREAM)
    (APPLY #'FORMAT STREAM FORMAT-STRING ARGS)
    (TERPRI STREAM)))
```

```
(DEFUN GET-MESSAGE-STREAM ())
```

```
;;; return an output stream for user messages
```

```
(WHEN (%WINDOW-HIDDEN? IL:PROMPTWINDOW)
  (UN-HIDE-WINDOW IL:PROMPTWINDOW))
(IL:GETSTREAM IL:PROMPTWINDOW))
```

```
(DEFUN SELECT-WINDOW (&OPTIONAL MESSAGE &REST MESSAGE-ARGS))
```

```
;;; get the user to select a window on the screen
```

```
(WHEN MESSAGE
  (APPLY #'NOTIFY-USER MESSAGE MESSAGE-ARGS))
(IL:RESETFORM
  (IL:CURSOR IL:CROSSHAIRS)
  (LET
    (WINDOW)
    (LOOP
      (WHEN (IL:MOUSESTATE (OR IL:LEFT IL:MIDDLE))
        (RETURN (LET ((WINDOW (IL:WHICHW)))
          (WHEN WINDOW
            (UNWIND-PROTECT
              (PROGN (IL:INVERTW WINDOW)
                (LOOP (WHEN (NOT (IL:MOUSESTATE (OR IL:LEFT IL:MIDDLE)))
                  (RETURN (LET ((NEW-WINDOW (IL:WHICHW)))
                    (WHEN (AND NEW-WINDOW (EQ (MAIN-WINDOW NEW-WINDOW)
                      (MAIN-WINDOW WINDOW)))
                      (MAIN-WINDOW WINDOW))))))))
              (IL:INVERTW WINDOW))))))))))
```

```
(DEFUN SELECT-BAGGAGE ())
```

```
;;; returns a list of selected placements.
```

```
;;; we presume UPDATE-PLACEMENTS has just been called & won't be called again by GO-TO-ROOM.
```

```
(LET (WINDOW PLACEMENT ROOM BAGGAGE)
  (LOOP (LET ((OP (COND
    ((MOVE-KEY-DOWN-P)
     :MOVE)
    ((COPY-KEY-DOWN-P)
     :COPY)
    (T (RETURN))))))
    (SETQ WINDOW (SELECT-WINDOW "Select placement to ~A" OP))
    (UNLESS WINDOW (RETURN))
    (MULTIPLE-VALUE-SETQ (PLACEMENT ROOM)
      (FIND-PLACEMENT WINDOW))
    (WHEN PLACEMENT
      (CASE OP
        (:MOVE (DELETE-PLACEMENT PLACEMENT ROOM))
        (:COPY (SETQ PLACEMENT (COPY-PLACEMENT-INTERNAL PLACEMENT))))
    (PUSHNEW PLACEMENT BAGGAGE :KEY #'PLACEMENT-WINDOW :TEST 'EQ))))
  BAGGAGE))
```

```
(DEFUN EXTERNALIZE-FONT (FONT)
  (LIST (IL:FONTPROP FONT 'IL:FAMILY)
    (IL:FONTPROP FONT 'IL:SIZE)
    (IL:FONTPROP FONT 'IL:FACE)))
```

```
(DEFMACRO COPY-KEY-DOWN-P ()
  `(OR (IL:KEYDOWNP 'IL:COPY)
    (AND (IL:SHIFTDOWNP 'IL:SHIFT)
      (NOT (OR (IL:SHIFTDOWNP 'IL:CTRL)
        (IL:SHIFTDOWNP 'IL:META))))))
```

```
(DEFMACRO HELP-KEY-DOWN-P ()
  `(OR (IL:KEYDOWNP 'HELP)
    (IL:KEYDOWNP 'IL:DBK-HELP)))
```

```
(DEFMACRO DELETE-KEY-DOWN-P ()
  `(OR (IL:KEYDOWNP 'IL:DELETE)
    (AND (IL:SHIFTDOWNP 'IL:CTRL)
      (IL:SHIFTDOWNP 'IL:META)
      (NOT (IL:SHIFTDOWNP 'IL:SHIFT))))))
```

```
(DEFMACRO EDIT-KEY-DOWN-P ()
  `(AND (IL:SHIFTDOWNP 'IL:CTRL)
    (NOT (OR (IL:SHIFTDOWNP 'IL:SHIFT)
      (IL:SHIFTDOWNP 'IL:META))))))
```

```
(DEFMACRO MOVE-KEY-DOWN-P ()
  `(OR (IL:KEYDOWNP 'IL:MOVE)
        (AND (IL:SHIFTDOWNP 'IL:CTRL)
              (IL:SHIFTDOWNP 'IL:SHIFT)
              (NOT (IL:SHIFTDOWNP 'IL:META)))))
```

```
(DEFMACRO EXPAND-KEY-DOWN-P ()
  `(OR (IL:KEYDOWNP 'IL:EXPAND)
        (IL:KEYDOWNP 'IL:ESCAPE)))
```

:: add a lafite form for bug reports

```
(DEFUN MAKE-ROOMS-SUPPORT-FORM ()
  (IL:MAKEXXSUPPORTFORM "Rooms" IL:ROOMSSUPPORT *ROOMS-SYSTEM-DATE*))
```

```
(DEFGLOBALVAR IL:ROOMSSUPPORT "RoomsSupport^.PA")
```

```
(IL:ADDOVAR IL:LAFITESPECIALFORMS ("Rooms Report" (IL:FUNCTION MAKE-ROOMS-SUPPORT-FORM)
                                         "A form to report a Rooms bug or suggestion"))
```

```
(SETQ IL:LAFITEFORMSMENU NIL)
```

:: provide a who line entry

```
(DEFPARAMETER *WHO-LINE-ENTRY*
  `("Room:" (AND *CURRENT-ROOM* (ROOM-NAME *CURRENT-ROOM*))
    10
    ,#'(LAMBDA NIL (INTERACTIVE-GO-TO-ROOM :ALLOW-NEW? T))))
```

```
(WHEN (BOUNDP 'IL:*WHO-LINE-ENTRY-REGISTRY*)
  (PUSHNEW *WHO-LINE-ENTRY* IL:*WHO-LINE-ENTRY-REGISTRY* :TEST 'EQUAL :KEY 'CAR))
```

```
(IL:DECLARE\ : IL:DOEVAL@COMPILE IL:DONTCOPY
```

```
(IL:GLOBALVARS IL:PROMPTWINDOW IL:CROSSHAIRS IL:WINDOWBACKGROUNDSHADE IL:WHOLESCREEN IL:LAFITESPECIALFORMS
  IL:LAFITEFORMSMENU IL:DEFAULTICONFN IL:*WHO-LINE-ENTRY-REGISTRY*)
)
```

```
(IL:PUTPROPS IL:ROOMS-D IL:COPYRIGHT ("Venue & Xerox Corporation" 1987 1988 1990 2020))
```

FUNCTION INDEX

CONFIRM	3	GET-POSITION	3	NOTIFY-USER	3	SHRUNKEN?	3
DELETE-WINDOW-ICON	3	ICON-POSITION	3	PROMPT-USER	3	WINDOW-POSITION	3
EXPAND-WINDOW	3	MAKE-ROOMS-SUPPORT-FORM .	5	SELECT-BAGGAGE	4	WINDOW-VISIBLE-P	2
EXTERNALIZE-FONT	4	MENU	3	SELECT-WINDOW	4		
GET-MESSAGE-STREAM	4	MOVE-WINDOW	1	SHAPE-WINDOW	2		

MACRO INDEX

CLOSE-WINDOW	2	EXPAND-KEY-DOWN-P .	5	MAKE-POSITION	2	POSITION-Y	2	WINDOW-TITLE	2
COPY-KEY-DOWN-P ...	4	HELP-KEY-DOWN-P ...	4	MOVE-KEY-DOWN-P ...	5	SHRINK-WINDOW	3		
DELETE-KEY-DOWN-P .	4	ICON?	3	OPEN-WINDOW	2	WINDOW-ICON	3		
EDIT-KEY-DOWN-P ...	4	MAIN-WINDOW	2	POSITION-X	2	WINDOW-REGION	2		

VARIABLE INDEX

WHO-LINE-ENTRY	5	IL:LAFITESPECIALFORMS	5	IL:ROOMSSUPPORT	5
------------------------	---	-----------------------------	---	-----------------------	---

TYPE INDEX

BITMAP	1	FONT	1	TEXTURE	1
--------------	---	------------	---	---------------	---

STRUCTURE INDEX

REGION	2
--------------	---

FILE-ENVIRONMENT INDEX

IL:ROOMS-D	1
------------------	---
