File created: 5-Dec-2020 16:37:08   **{DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>LDE>ROOMS>MEDLEY-35>TOUCHY-BUTTONS.;2**

previous date: 17-Aug-90 14:46:54   {DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>LDE>ROOMS>MEDLEY-35>TOUCHY-BUTTONS.;1

  Read Table: XCL

    Package: ROOMS

      Format: XCCS

```
(IL:RPAQQ IL:TOUCHY-BUTTONSCOMS
        (
;; Includer buttons so you can have dynamic mixin rooms.  For example, you can have a "Notecards-Mixin" or "Programming-Mixin" Room
;; and have buttons to include these in "Pockets"  then whenever you need these facilities you can mix them in.

        (IL:FUNCTIONS MAKE-INCLUDER INCLUDER-TEXT INCLUDER-ACTION)

;; Toggle buttons for switching between variable settings.  This should obviously be generalized to something that allows you to select or
;; circulate through value settings.

        (IL:FUNCTIONS MAKE-TOGGLER MAKE-N-VALUER MAKE-EXSET-TOGGLER MAKE-RANGE-TOGGLER N-VALUER-ACTION
                N-VALUER-TEXT CHECK-EVAL)
        (IL:FUNCTIONS MAKE-SWITCH SWITCH-ACTION)

;;

;; (MAKE-ONCE-ONLY (IL:PROMPTPRINT "Hello, World") "Fire..." "Exhausted")

        (IL:FUNCTIONS MAKE-ONCE-ONLY ONCE-ONLY-ACTION ONCE-ONLY-TEXT)
        (IL:DECLARE\: IL:DONTCOPY (IL:PROPS (IL:TOUCHY-BUTTONS IL:MAKEFILE-ENVIRONMENT)
                                        (IL:TOUCHY-BUTTONS IL:FILETYPE)))))
```

;; Includer buttons so you can have dynamic mixin rooms. For example, you can have a "Notecards-Mixin" or "Programming-Mixin" Room and have
;; buttons to include these in "Pockets" then whenever you need these facilities you can mix them in.

```
(DEFUN MAKE-INCLUDER (ROOM-NAME)
    (MAKE-BUTTON-WINDOW (MAKE-BUTTON :TEXT-FORM (LIST 'INCLUDER-TEXT ROOM-NAME)
                                :ACTION
                                'INCLUDER-ACTION :INCLUDER-ROOM-NAME ROOM-NAME)))


(DEFUN INCLUDER-TEXT (INCLUDER-ROOM-NAME)
    (LET ((INCLUSIONS (ROOM-INCLUSIONS *CURRENT-ROOM*)))
        (IF (LISTP INCLUSIONS)
            (IF (MEMBER INCLUDER-ROOM-NAME INCLUSIONS :TEST #'EQUAL)
                (FORMAT NIL "Exclude ~A" INCLUDER-ROOM-NAME)
                (FORMAT NIL "Include ~A" INCLUDER-ROOM-NAME))
            (FORMAT NIL "*-???-* ~A" INCLUDER-ROOM-NAME))))


(DEFUN INCLUDER-ACTION (DSP BUTTON)
    (LET* ((ROOM *CURRENT-ROOM*)
           (INCLUDER-ROOM-NAME (BUTTON-PROP BUTTON :INCLUDER-ROOM-NAME)))
        (UPDATE-PLACEMENTS)
        (IF (MEMBER INCLUDER-ROOM-NAME (ROOM-INCLUSIONS ROOM)
                    :TEST
                    #'EQUAL)
            (SETF (ROOM-INCLUSIONS ROOM)
                  (DELETE INCLUDER-ROOM-NAME (ROOM-INCLUSIONS ROOM)
                          :TEST
                          #'EQUAL))
            (PUSH INCLUDER-ROOM-NAME (ROOM-INCLUSIONS ROOM)))
        (ROOM-CHANGED ROOM :EDITED)))
```

;; Toggle buttons for switching between variable settings. This should obviously be generalized to something that allows you to select or circulate
;; through value settings.

```
(DEFUN MAKE-TOGGLER (VARIABLE-NAME &REST KEYS)
    (APPLY #'MAKE-N-VALUER VARIABLE-NAME '(NIL T)
        KEYS))


(DEFUN MAKE-N-VALUER (VARIABLE-NAME N-VALUES &REST KEYS)
    (MAKE-BUTTON-WINDOW (APPLY #'MAKE-BUTTON :TEXT-FORM `(N-VALUER-TEXT ',VARIABLE-NAME)
                                :ACTION
                                'N-VALUER-ACTION :VARIABLE-NAME VARIABLE-NAME :N-VALUES N-VALUES KEYS)))


(DEFUN MAKE-EXSET-TOGGLER (VARIABLE-NAME EXPLICIT-SET &REST KEYS)
    (APPLY #'MAKE-N-VALUER VARIABLE-NAME EXPLICIT-SET :HELP (FORMAT NIL "Set variable ~S" VARIABLE-NAME)
        KEYS))


(DEFUN MAKE-RANGE-TOGGLER (VARIABLE-NAME RANGE-START RANGE-END &REST KEYS)
```

```
      (APPLY #'MAKE-N-VALUER VARIABLE-NAME (DO ((I RANGE-START (1+ I))
                                               (ACCUMULATOR NIL ACCUMULATOR))
                                              ((> I RANGE-END)
                                               (NREVERSE ACCUMULATOR))
                                            (PUSH I ACCUMULATOR))
             :HELP
             (FORMAT NIL "Set variable ~S" VARIABLE-NAME)
             KEYS))


(DEFUN N-VALUER-ACTION (DSP BUTTON)
   (LET* ((N-VALUES (BUTTON-PROP BUTTON :N-VALUES))
          (VARIABLE-NAME (BUTTON-PROP BUTTON :VARIABLE-NAME)))
      (SETQ N-VALUES (REMOVE (CHECK-EVAL VARIABLE-NAME)
                             N-VALUES))
      (SET VARIABLE-NAME (IF (EQ (LENGTH N-VALUES)
                                 1)
                             (CAR N-VALUES)
                             (IL:MENU (IL:CREATE IL:MENU
                                           IL:ITEMS IL:_ N-VALUES))))))


(DEFUN N-VALUER-TEXT (VARIABLE-NAME)
   (FORMAT NIL "~A is ~a" VARIABLE-NAME (CHECK-EVAL VARIABLE-NAME)))


(DEFUN CHECK-EVAL (VARIABLE-NAME)
   (IF (BOUNDP VARIABLE-NAME)
       (EVAL VARIABLE-NAME)
       "Unbound"))


(DEFUN MAKE-SWITCH (DECISION-FN SET-FN TEXT-FORM &REST KEYS)
   (MAKE-BUTTON-WINDOW (APPLY #'MAKE-BUTTON :TEXT-FORM TEXT-FORM :ACTION 'SWITCH-ACTION :DECISION-FN DECISION-FN
                              :SET-FN SET-FN KEYS)))


(DEFUN SWITCH-ACTION (DSP BUTTON)
   (LET* ((VALUE (FUNCALL (BUTTON-PROP BUTTON :DECISION-FN)
                          BUTTON)))
      (FUNCALL (BUTTON-PROP BUTTON :SET-FN)
               BUTTON VALUE)))
```

;;
;; (MAKE-ONCE-ONLY (IL:PROMPTPRINT "Hello, World") "Fire..." "Exhausted")

```
(DEFUN MAKE-ONCE-ONLY (FORM INITIAL-TEXT FINAL-TEXT)
   (MAKE-BUTTON-WINDOW (MAKE-BUTTON :TEXT INITIAL-TEXT :TEXT-FORM 'ONCE-ONLY-TEXT :ACTION 'ONCE-ONLY-ACTION
                              :ONCE-ONLY-FORM FORM :INITIAL-TEXT INITIAL-TEXT :FINAL-TEXT FINAL-TEXT)))


(DEFUN ONCE-ONLY-ACTION (STREAM BUTTON)
   (UNLESS (IL:STREAMPROP (IL:GETSTREAM STREAM)
                  BUTTON)
```

      ;; store the state of the button on its host, so that the state is reset each time the button is reconstituted.

```
      (EVAL (BUTTON-PROP BUTTON :ONCE-ONLY-FORM))
      (IL:STREAMPROP (IL:GETSTREAM STREAM)
             BUTTON T)))


(DEFUN ONCE-ONLY-TEXT (STREAM BUTTON)
   (WHEN STREAM
      (IF (IL:STREAMPROP (IL:GETSTREAM STREAM)
                  BUTTON)
          (BUTTON-PROP BUTTON :FINAL-TEXT)
          (BUTTON-PROP BUTTON :INITIAL-TEXT))))
```

(IL:DECLARE\: IL:DONTCOPY

(IL:PUTPROPS **IL:TOUCHY-BUTTONS IL:MAKEFILE-ENVIRONMENT** (:PACKAGE "ROOMS" :READTABLE "XCL"))

(IL:PUTPROPS **IL:TOUCHY-BUTTONS IL:FILETYPE** :COMPILE-FILE)
)

FUNCTION INDEX

PROPERTY INDEX