GUI

Déplacement d'objets à la souris

Tout mouvement ou clic dans un bouton de la souris est un événement. Pour gérer la souris, il faudra de redéfinir la méthode on-event qui par défaut ne fait rien au sein d'une sous-classe de canvas%. L'evenement lié à la souris intervient en effet dans le canvas.

Exemple : un éditeur de polygone.

Le canvas va présenter un polygone initial constitué de points reliés par des segments. L'utilisateur va pouvoir déplacer les sommets avec la souris. La liste de points est défini dans la variable globale POLY.

Chaque point est matérialisé par un petit cercle de manière à pouvoir l'attraper facilement avec la souris.

GUI

Un point est une structure de type posn. Un point est représenté par une structure à deux champs nommés x et y.

En Racket ; (define-struct posn (x y))

Ce qui veut dire : soit posn un nouveau type de structure à deux champs x et y. Cette structure est fourni par Racket. On peut alors définir des points :

(define A (make-posn 3 10)) ; point A (3;10)

(define B (make-posn -8 1) ; point B (-8;1)

>A.	>(posn-x A)
#(struct:posn 3 10)	3

GUI

Le polygone initial sera construit par un appel à la fonction build-polygon.

```
126 #lang racket/gui
127 (require lang/posn)

128 ; (build-polygon n f g) construit une liste de n sommets (f(i),g(i))
129 (define (build-polygon n f g)
130   (build-list n (λ (i) (make-posn (f i) (g i)))))

131 (define POLY
132   (build-polygon 9 (λ (i) (* 30 (+ i 1))) (λ (i) (random 300))))
```

GUI

Le dessin du polygone P peut programmer à l'extérieur du paint-callback.

Il faut bien récupérer le peintre dc associé au canvas dans le paint-callback pour le passer à la fonction extérieure (draw-polygon P dc)

Le dessin des arrêtés (edges) se fait après celui des sommets (vertices)

Utiliser les méthodes draw-point et draw-line de la classe dc%

Rappel : origine du canvas est en haut à gauche - axe Oy vers le bas)

GUI

```
137 (define PEN-FOR-LINES (make-object pen% "blue" 1 'solid))   ; fin
138 (define PEN-FOR-POINTS (make-object pen% "red" 8 'solid))   ; épais

139 (define (draw-polygon P dc)          ; P ≡ (#(struct:posn x y) ...)
140   (define (draw-vertices L)          ; dessin des sommets
141     (send dc set-pen PEN-FOR-POINTS)
142     (for-each (lambda (pt) ; le point a une épaisseur de 8
143                 (send dc draw-point
144                    (- (posn-x pt) 4)
145                    (- (posn-y pt) 4)))
146               L))
147   (define (draw-edges L)             ; dessin des arêtes
148     (send dc set-pen PEN-FOR-LINES)
149     (do ((L L (cdr L)))
150       ((null? (cdr L)) (void))
151       (send dc draw-line
152          (posn-x (car L)) (posn-y (car L))
153          (posn-x (cadr L)) (posn-y (cadr L)))))
154   (send dc clear)
155   (when (not (null? P))
156     (draw-vertices P) (draw-edges P)))
```

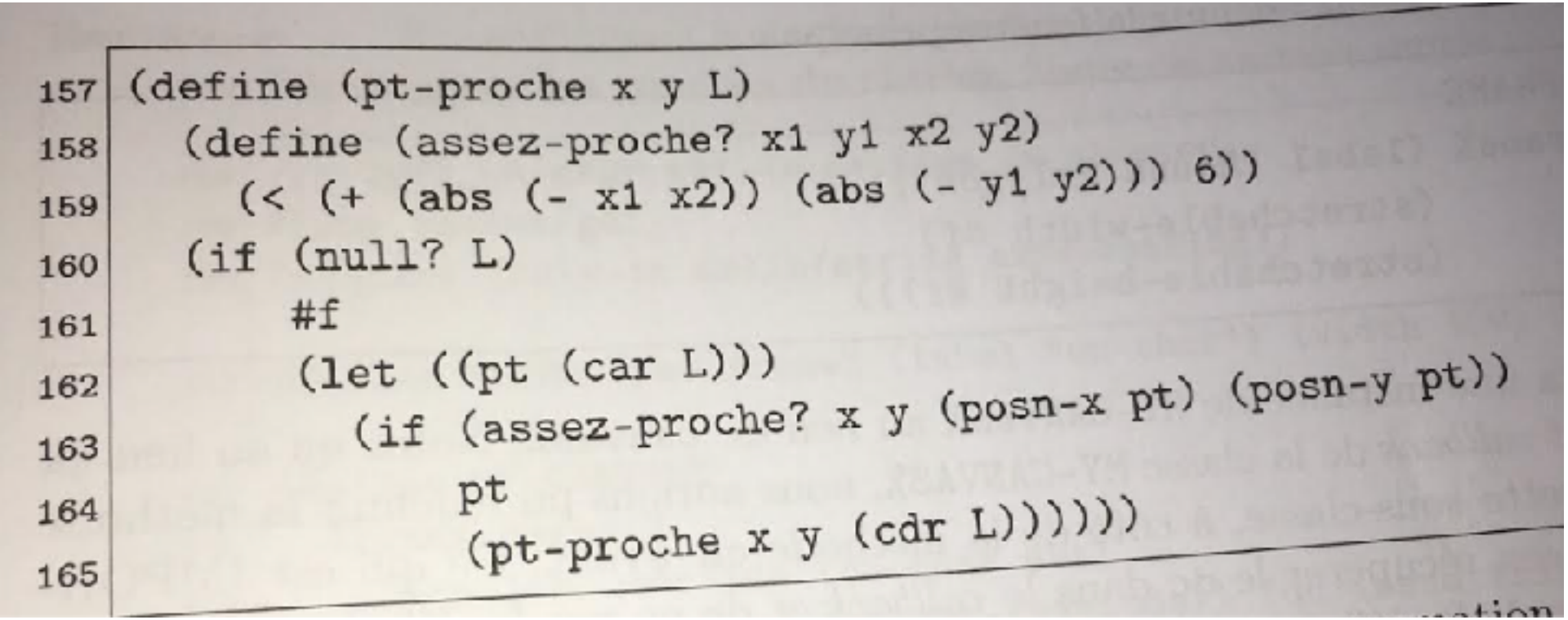
GUI

Lorsqu'on clique au voisinage du polygone, on doit savoir si on est assez prés d'un point. Si oui, on retourne les coordonnées du point et si non on retourne #f.

Définir une fonction (pt-proche x y P) - P polygone

On utilise la fonction abs(x1-x2) + abs(y1 -y2) pour laquelle le voisinage d'un point est un losange autour de ce point et non un disque.

GUI



```
157 (define (pt-proche x y L)
158   (define (assez-proche? x1 y1 x2 y2)
159     (< (+ (abs (- x1 x2)) (abs (- y1 y2))) 6))
160   (if (null? L)
161       #f
162       (let ((pt (car L)))
163         (if (assez-proche? x y (posn-x pt) (posn-y pt))
164             pt
165             (pt-proche x y (cdr L))))))
```

GUI

Point crucial : programmation de la sous-classe MY-CANVAS% de la classe Canvas%

On va redéfinir la méthode on-event pour gérer les événements souris.

£on veut pouvoir cliquer sur un point V dans le canvas avec le bouton gauche de la souris, pour le déplacer en laissant le bouton enfoncé (drag and drop).

L'événement *bouton gauche enfoncé* a pour type 'left-down.

L'événement *bouton gauche relâché* a pour type 'left-up.

GUI

Soit V le point recherché (qui vaut #f tant qu'il n'est pas trouvé).

Si on est en left-down : on note les coordonnées (x,y) de la souris et on cherche si (x,y) est voisin d'un sommet du polygone.Si c'est le cas, V devient ce point.

Si on est en left-up, on libber le point V

Dans tous les autres cas (on doit être en motion), la situation se divise en 2 : (1) si V est un point, on est en train de la déplacer et on doit modifier en temps réel les coordonnées du point déplacé dans la liste POLY et on poste un message on-paint pour redessiner le canvas ; (2) si V vaut #f, c'est que on fait glisser la souris bouton relâché et on fait rien.

GUI

```
166 (define MY-CANVAS%                                  ; pour gérer la souris
167   (class canvas%                                    ; une sous-classe de canvas%
168     (define/override (on-event evt)
169       (case (send evt get-event-type)
170         ((left-down) (let ((x (send evt get-x)) (y (send evt get-y)))
171                        (set! V (pt-proche x y POLY))))
172         ((left-up) (set! V #f))
173         (else (when V (set-posn-x! V (send evt get-x))
174                       (set-posn-y! V (send evt get-y))
175                       (send this on-paint)))))
176     (super-new)))
```

GUI

Définition de la fenêtre principale

```
177 (define FRAME
178   (new frame% (label "Mouse Polygon")
179               (stretchable-width #f)
180               (stretchable-height #f)))
```

Le canvas sera une instance de MY-CANVAS%

Rappel : on peut soit utiliser paint-callback pour dessiner le polygone ou bien redéfinir la méthode on-paint à coté de la méthode on-event dans la sous-classe.

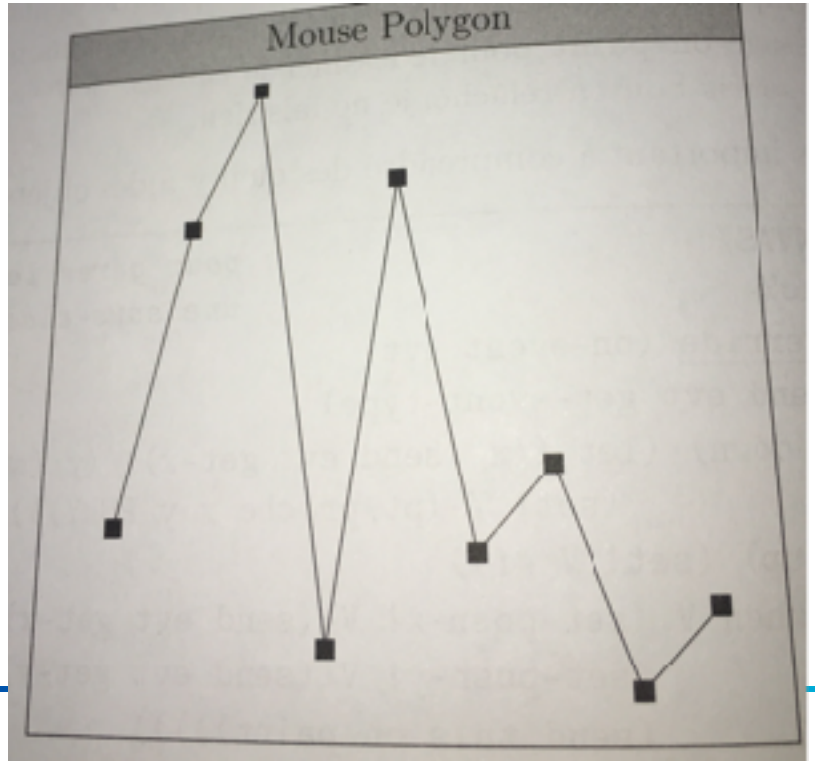
Il faut juste récupérer le dc dans la callback et pas le redéfinir après la definition du canvas

GUI

```
181 (define CANVAS
182   (new MY-CANVAS% (parent FRAME)
183                   (min-width 300)
184                   (min-height 300)
185                   (paint-callback (lambda (obj dc)
186                                     (draw-polygon POLY dc)))))
187 (send FRAME show #t)
```

GUI

On prend la souris et on bouge les points. Si vous faites afficher la variable POLY au top-level pour voir qu'elle a été modifiée.



GUI

Boutons, radio, cases à cocher,etc..

L'API graphique propose d'autres composants graphiques :

. les cases à cocher [check-box%] - plusieurs peuvent être cochées

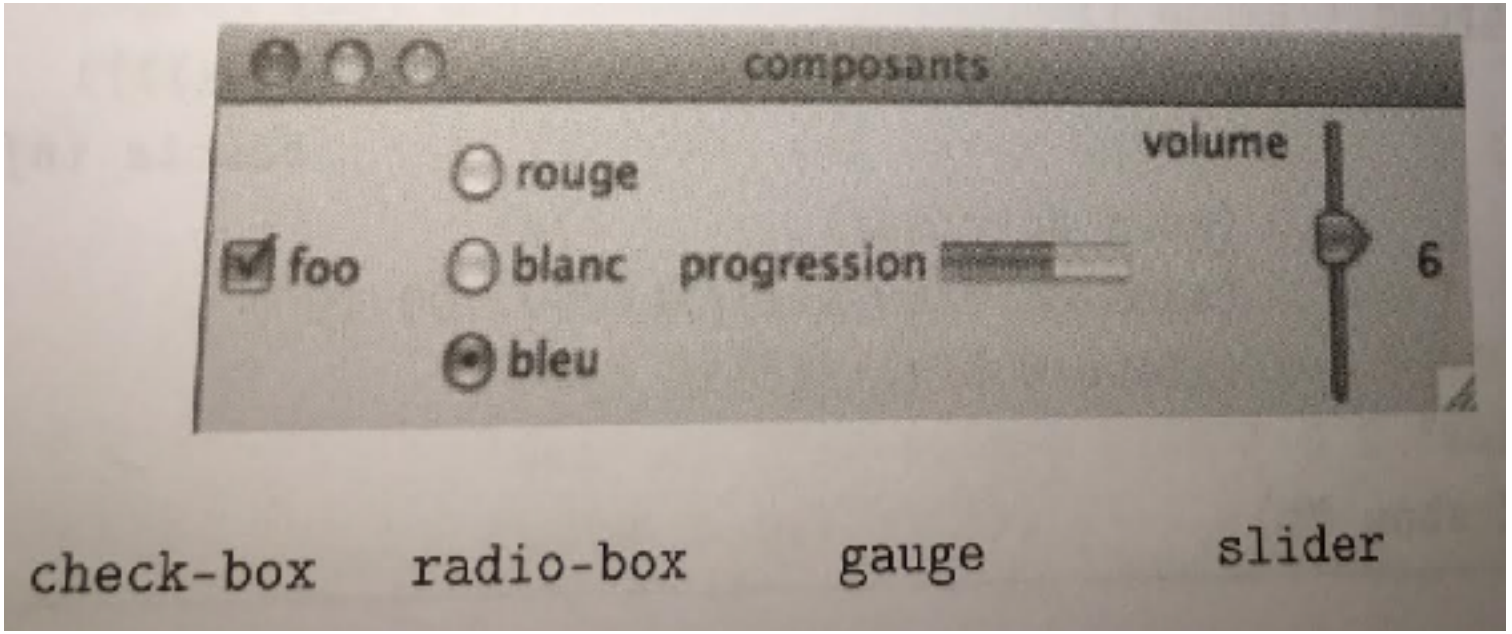
. les ensembles de votons radio [radio-box%] - un seul peut être sélectionné.

. les jauges [gauge%] pour visualiser un pourcentage d'avancement.

. les regrettes [sliders%] pour régler à la souris la valeur d'une variable entre deux valeurs extremes.

. la liste de choix [choice% et list-box%]

GUI



Voir l'aide en ligne : https://docs.racket-lang.org/gui/Widget_Gallery.html?q=check-box

TP N°4