```
(@htdw ListOfBall)

;; ============================================================================
;; Constants:
(define WIDTH  605)
(define HEIGHT 535)

(define BALL-RADIUS 10)

(define TOP (+ 0        BALL-RADIUS));these constants define the "inner box"
(define BOT (- HEIGHT 1 BALL-RADIUS));that constrain the center of the ball
(define LEF (+ 0        BALL-RADIUS));
(define RIG (- WIDTH  1 BALL-RADIUS));

(define BALL (circle BALL-RADIUS "solid" "white"))

(define MTS (rectangle WIDTH HEIGHT "solid" "green"))


;; ============================================================================
;; Data definitions:

(@htdd Ball)
(define-struct ball (x y dx dy))
;; Ball is (make-ball Number Number Number Number)
;; interp. (make-ball x y dx dy) is ball
;;    – position x, y    in screen coordinates
;;    – velocity dx, dy  in pixels/tick
;; CONSTRAINT: x is in [LEF, RIG]; y is in [TOP, BOT]
(define B1 (make-ball (/ WIDTH 2) (/ HEIGHT 2) 4 -3))

(@dd-template-rules compound)

(define (fn-for-ball b)
  (... (ball-x b)
       (ball-y b)
       (ball-dx b)
       (ball-dy b)))

(@htdd ListOfBall)
;; ListOfBall is one of:                A
;;   – empty
;;   – (cons Ball ListOfBall)
;; interp. a list of balls
(define LOB1 empty)
(define LOB2 (cons B1 empty))

(@dd-template-rules one-of
                    atomic-distinct
                    compound
                    ref          B
                    self-ref)

(define (fn-for-lob lob)
  (cond [(empty? lob) (...)]
        [else
         (... (fn-for-ball (first lob))
              (fn-for-lob (rest lob)))]))
```

```
59
60
61  ;; ========================================================================
62  ;; Functions:
63
64  (@htdf render-balls)          C
65  (@signature ListOfBall -> Image)
66  ;; render all balls onto MTS
67  (check-expect (render-balls empty) MTS)
68  (check-expect (render-balls (cons (make-ball 10 20 3 4)
69                                    (cons (make-ball 30 40 1 2)
70                                          empty)))
71                (place-ball (make-ball 10 20 3 4)
72                            (place-ball (make-ball 30 40 1 2)
73                                        MTS)))
74
75  ;(define (render-balls lob) MTS) ;stub
76
77  (@template-origin ListOfBall)
78
79  (@template
80   (define (render-balls lob)
81     (cond [(empty? lob) (...)]
82           [else
83            (... (fn-for-ball (first lob))
84                 (render-balls (rest lob)))])))
85                                          D
86  (define (render-balls lob)
87    (cond [(empty? lob) MTS]
88          [else                    E
89           (place-ball (first lob)
90                       (render-balls (rest lob)))]))      F
```