Normal design recipe

;; A MouseEvent is one of:

;; - “button-down”

;; - “button-up”

;; - “drag”

;; - “move”

;; - “enter” ; into the canvas area

;; - “leave” ; from the canvas area

(universe u0

[on-new new-expr]

[on-msg msg-expr]

[on-tick tick-expr]

[on-disconnect dis-expr]

[to-string render-expr]

[port port-expr])

; A UoB is (Universe or Bundle)

; new-expr : Universe Iworld → UoB

; msg-expr : Universe Iworld S-Expression → UoB

; tick-expr : Universe → UoB

; dis-expr : Universe Iworld → UoB

; render-expr : Universe → String

; port-expr : NatNum

; make-bundle : Universe [ListOf Mail]

[ListOf Iworld] → Bundle

; make-mail : Iworld S-Expression → Mail

1. Data definitions
2. Sig, PS, Header
3. Functional examples
4. Template
5. Function definition
6. Testing

Generative recursion:

Write why it terminates

Accumulators:

Say what the accumulator represents/is

(animate draww)

; draww : NatNum → Scene/image

Big-bang

(big-bang w0

[to-draw render-expr]

[on-tick tick-expr]

[on-key key-expr]

[on-release release-expr]

[on-mouse mouse-expr]

[stop-when stop-expr]

[name naym]

[register ip-expr]

[port port-expr]

[on-receive recv-expr])

; render-expr : World → Image/Scene

; tick-expr : World → World

; key-expr : World KeyEvent → World

; release-expr : World KeyEvent → World

; on-mouse : World Number Number MouseEvent → World

; stop-expr : World → Boolean

; naym : String or Symbol

; ip-expr : String (or literally LOCALHOST)

; port-expr : NatNum

; recv-expr : World S-Expression → (Package or World; if Package, sends it to the server)

; make-package : World S-Expression → Package

; text : String Int String (color)