## <<Interface>>

## StatefulWorld<%>

on-tick: -> Void

on-mouse: Integer Integer MouseEvent -> Void

on-key: KeyEvent -> Void

on-draw: ->Scene target-x: -> Integer target-y: -> Integer

target-selected? -> Boolean get-toys: -> ListOfStatefulToy<%> for-test:get-x-click -> Integer for-test:get-y-click -> integer for-test:get-speed -> PosInt

## World%

x,y:PosInt

X-click, y-click: Integer selected?: Boolean square-speed: PosInt toys: ListOfStatefulToy<%>>

on-tick: -> Void

on-mouse: Integer Integer MouseEvent -> Void world-after-button-down: Integer Integer -> Void

click-in-target? : Integer Integer -> Boolean world-after-button-up : Integer Integer -> Void world-after-drag : Integer Integer -> Void

world-after-mouse-event:

Integer Integer Integer Boolean -> Void

on-key: KeyEvent -> Void

on-draw : -> Scene target-x: -> Integer target-y: -> Integer

target-selected?: -> Boolean get-toys: -> ListOfStatefulToy<%> for-test:get-x-click: -> Integer for-test:get-y-click: -> Integer for-test:get-speed: -> PosInt <<interface>>
StatefulToy<%>

on-tick: -> Void

add-to-scene: Scene -> Scene

toy-x : -> PosInt toy-y : -> PosInt

toy-color: -> ColorString

for-test:get-fields: ->ListOfFields

SquareToy%

x,y,square-speed: PosInt

dir: Direction

on-tick: -> Void

right-sq-toy-on-tick: -> Void left-sq-toy-on-tick: -> Void new-r-dir: PosInt -> Direction new-l-dir: PosInt -> Direction add-to-scene: Scene -> Scene

toy-x : -> PosInt toy-y : -> PosInt

toy-color: -> ColorString for-test:toy-dir: -> Direction for-test:toy-speed: -> PosInt for-test:get-fields: ->ListOfFields CircleToy%

x,y: PosInt

tick-counter: PosNum color: ColorString

on-tick: -> Void

switch-color-and-reset: CircleToyColor -> Void
change-color : CircleToyColor -> CircleToyColor

add-to-scene : Scene -> Scene

toy-x : -> PosInt toy-y : -> PosInt

toy-color : -> CircleToyColor for-test:tick-ctr : -> PosInt

for-test:get-fields : -> ListOfFields