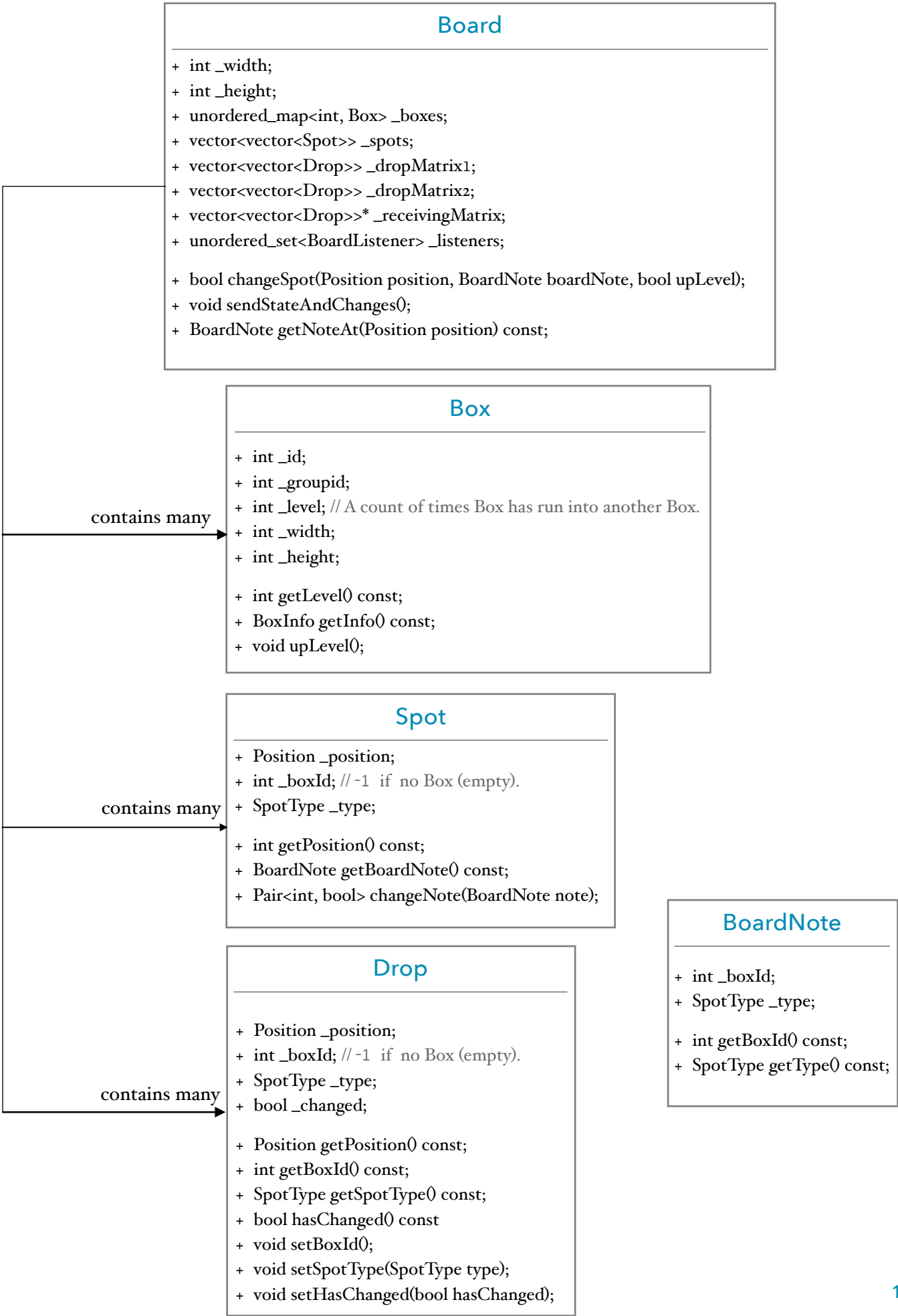


# BOARD CLASS

## Boxes | Spots | Drops

Conceptually a rectangular plane with x-y coordinates. Boxes move across the Board and sit at a coordinate.

Board holds the master copy of Boxes and Spots. Spots hold the state for each coordinate. Box holds the state for each Box. Board contains BoardListeners, that it notifies with updates to the Boxes and Spots.



# BOARDLISTENER

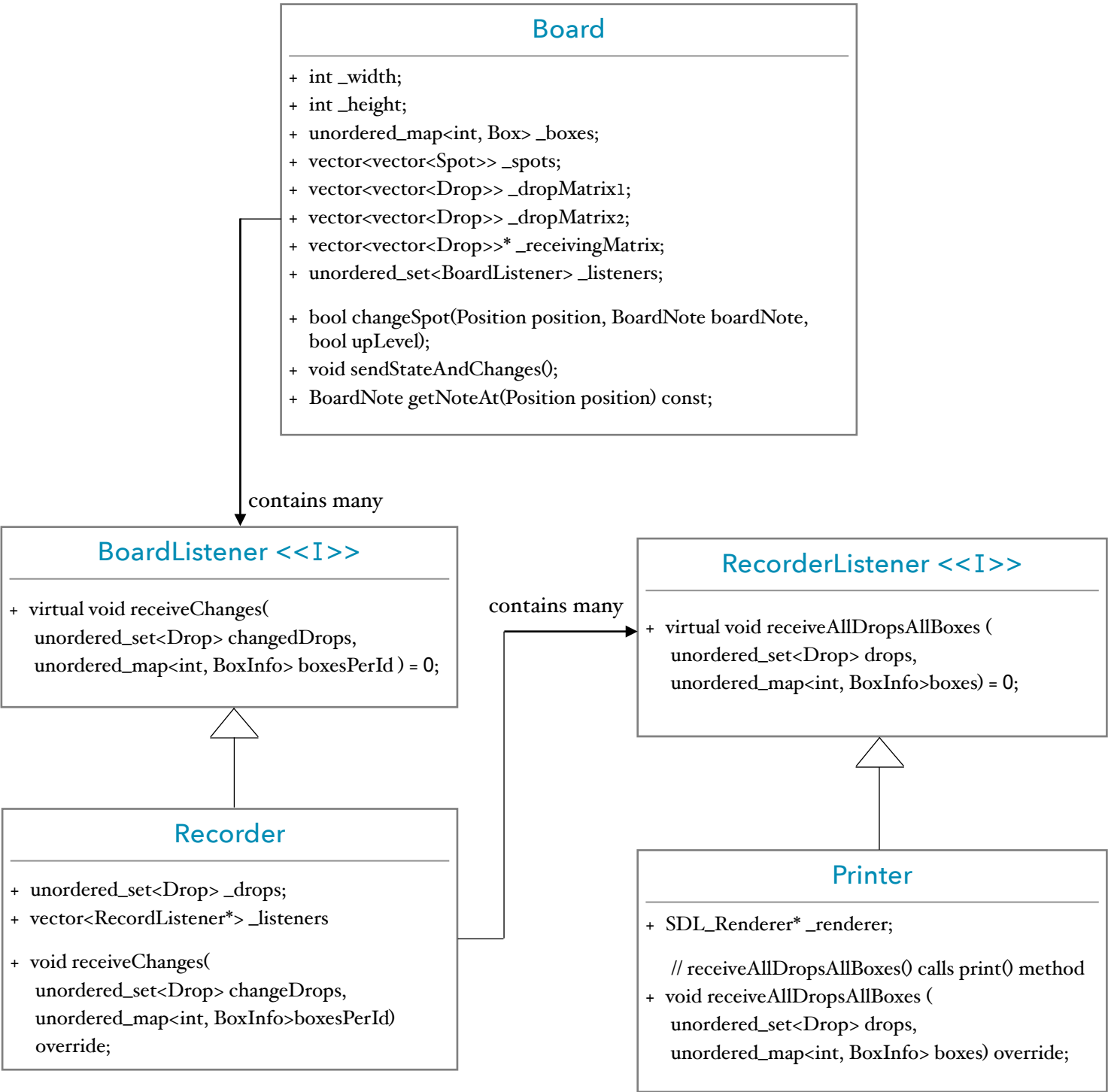
## Recorder | RecorderListener | Printer

When Board’s `sendStateAndChanges()` method is called, it sends the Board’s changes to its BoardListeners.

It sends the current complete state of all of its Boxes via a map of BoxInfos. It however, does not send the complete state of all the Spots. It only sends the changes to the Spots since the last time `sendStateAndChanges()` was called. These changes are sent as an `unordered_set` of Drops.

Recorder, which is a BoardListener, keeps a tally of all the changes, so it can send a complete picture to its RecorderListeners.

Printer is a RecordListener and renders the Board.

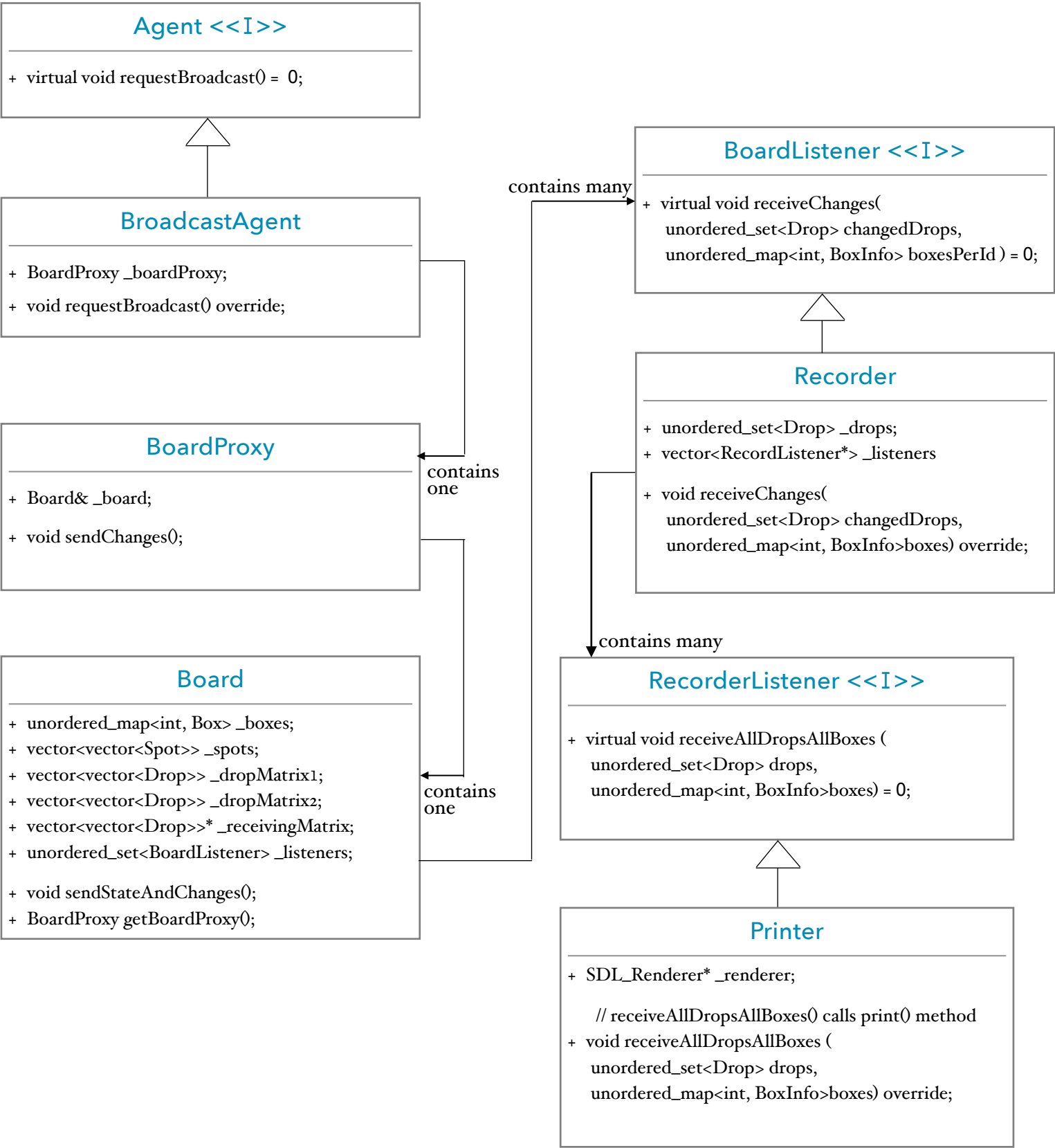


# MAIN'S RENDER LOOP

## BroadcastAgent | BoardProxy | Printer

Main's rendering loop is a while loop that breaks when the user closes the window.

At each iteration the BroadcastAgent requests that the BoardProxy broadcast the Board's state. The BoardProxy which contains a Board, calls the Board's sendStateAndChanges() method. The broadcast is picked up by the Recorder (which is a BoardListener). The Recorder aggregates the data and then broadcasts it out to its RecorderListeners. There's only one RecorderListener, the Printer which takes the data and renders the Board.



# THREAD FUNCTION

## PositionManager | Decider | Mover

Main creates and deletes a thread for each Box. The thread function, funcMoveBox(), contains a while loop that continually asks the Board to move the Box closer to its final destination. First a vector of new positions are gotten from a Position Manager. Then one of the Positions is chosen by a Decider. Then a Mover asks the Board to move the Box to the new position by calling the Board's changeSpot() method.

