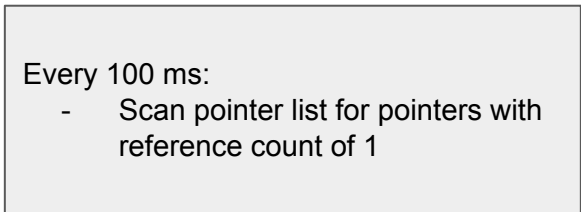


### Collector Thread Pointer List

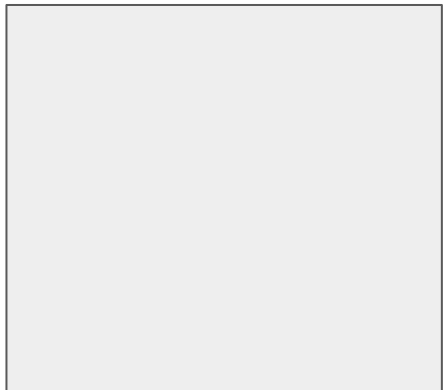


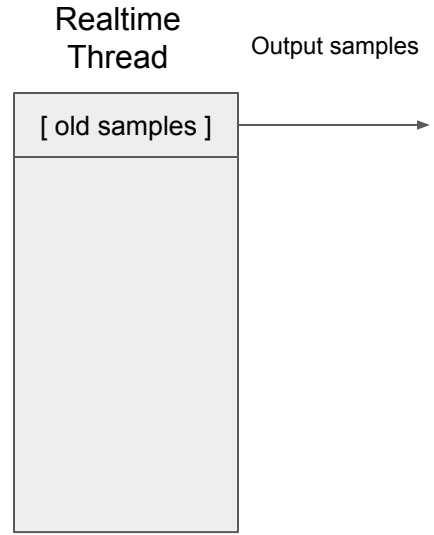
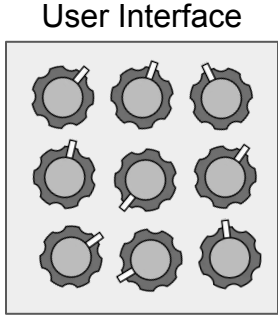
### Collector Thread



First, the UI tells the UI thread that something happened.

### Heap

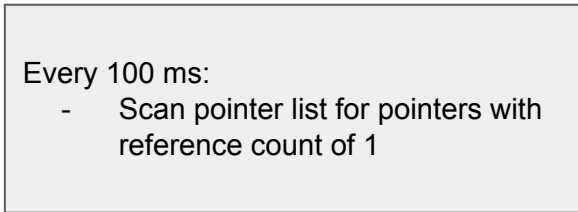




### Collector Thread Pointer List

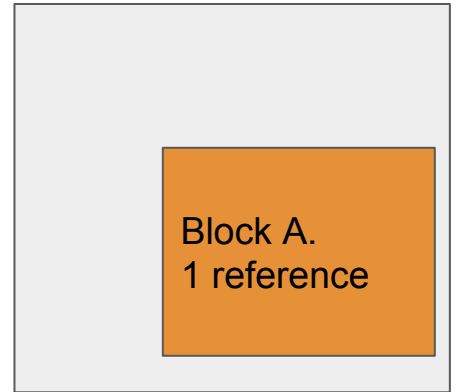


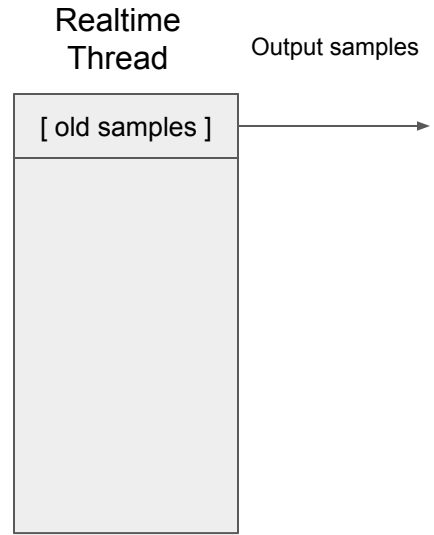
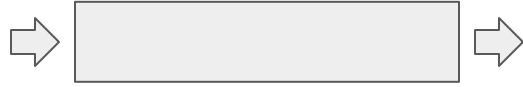
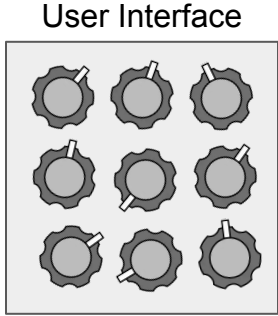
### Collector Thread



Next, the UI thread performs an allocation with a reference counted pointer.

### Heap

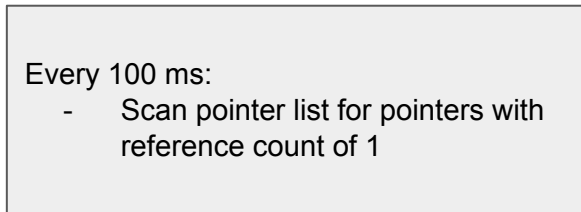




### Collector Thread Pointer List

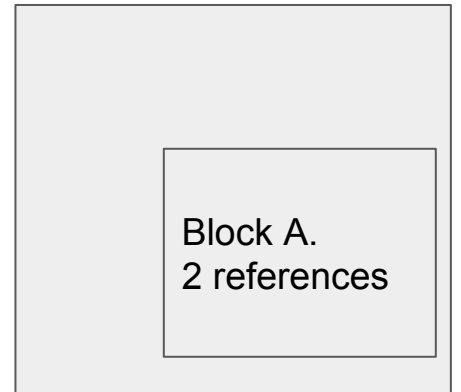


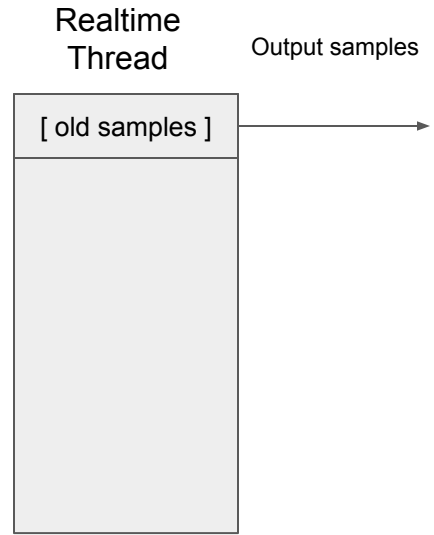
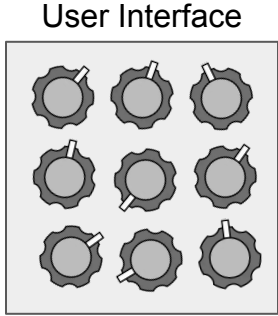
### Collector Thread



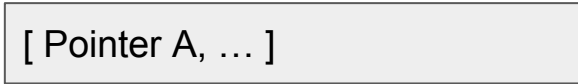
The UI thread then registers the new reference counted pointer with the collector thread.

### Heap

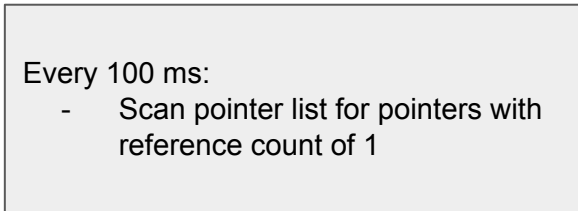




**Collector Thread Pointer List**



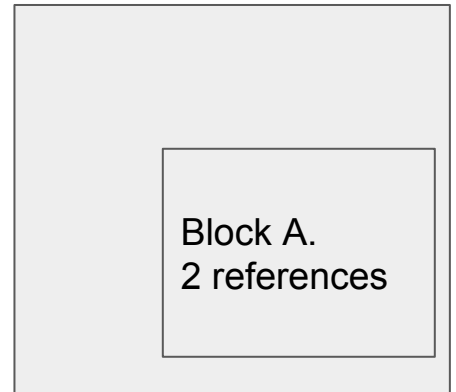
**Collector Thread**

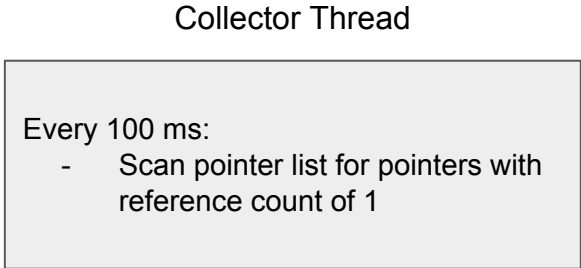
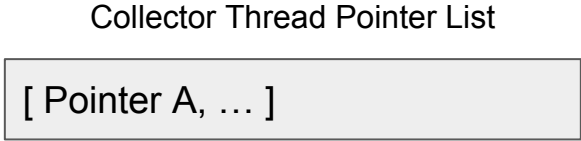
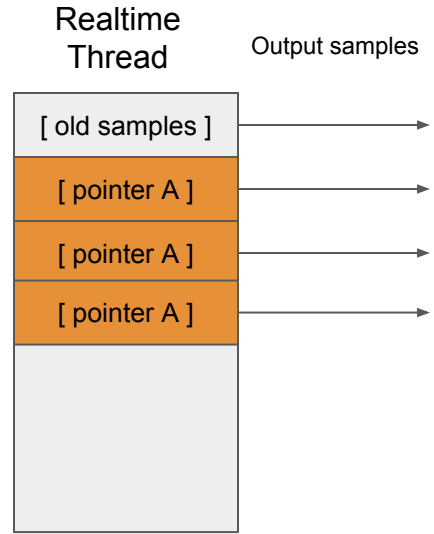
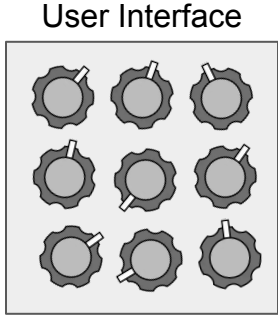


The UI thread sends the messages over the lock free queue.

The UI thread drops its reference, but the queue adds one.

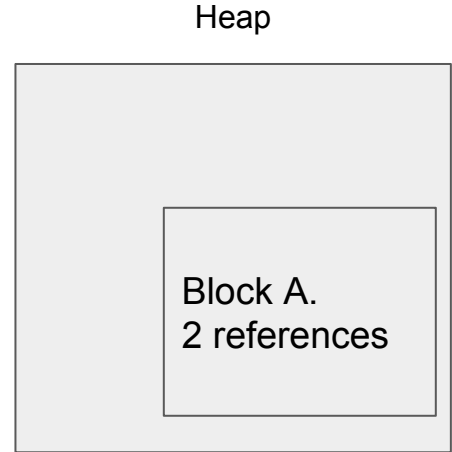
**Heap**

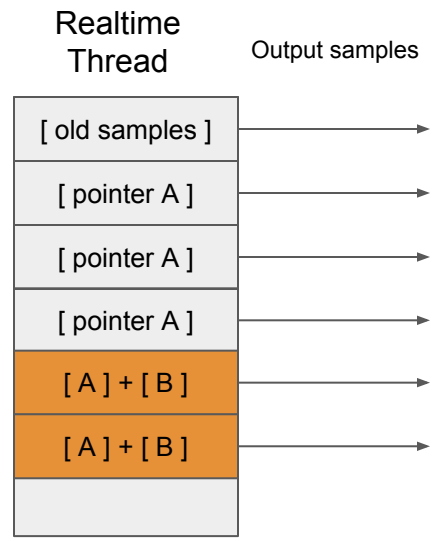
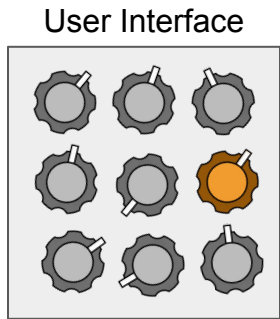




The realtime thread reads the new samples, and uses them for a while.

The queue drops a reference but the realtime thread adds one.

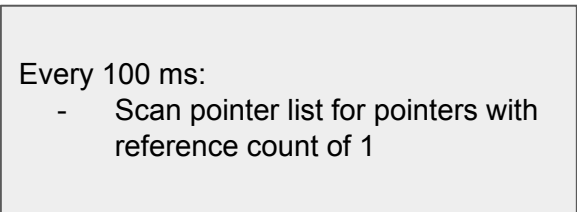




Collector Thread Pointer List



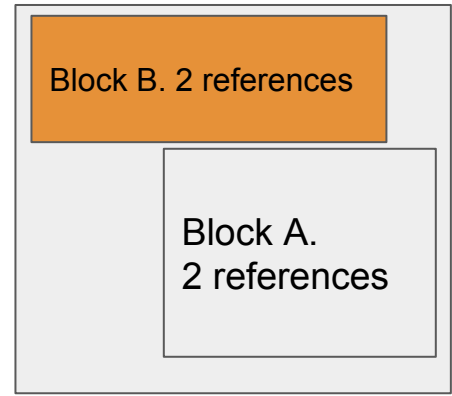
Collector Thread

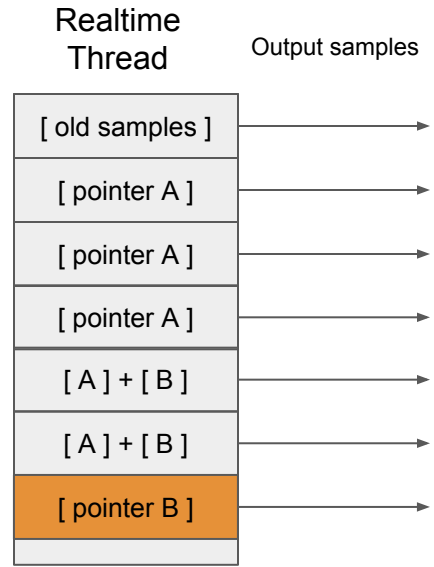
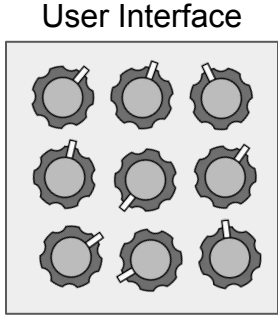


The entire process happens again...

The realtime thread gradually transitions between the "A" samples and the "B" samples.

Heap

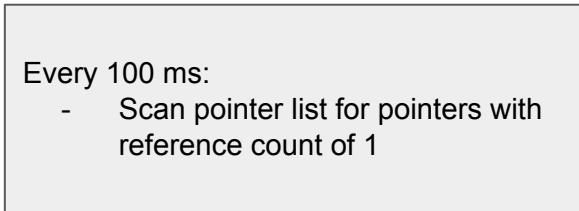




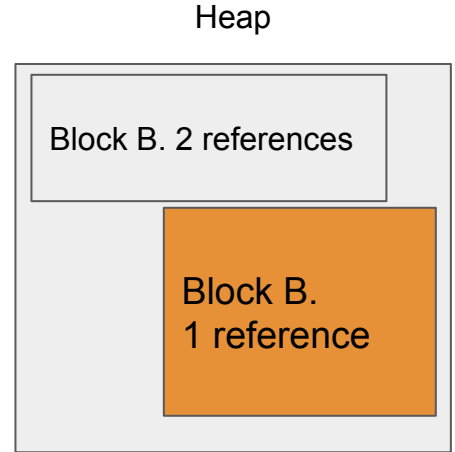
Collector Thread Pointer List

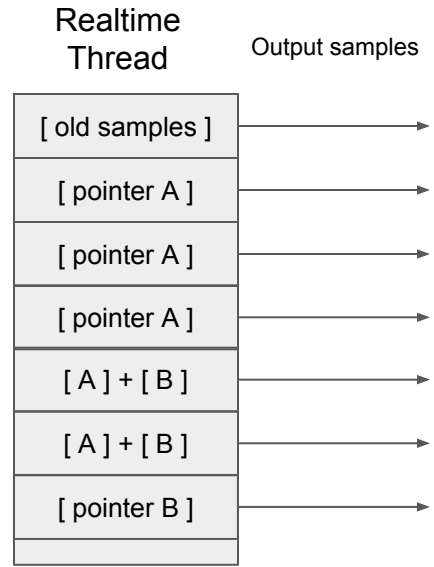
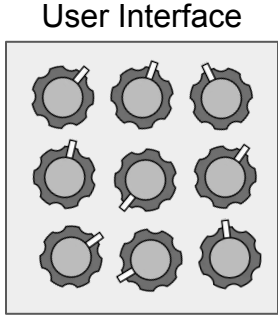


Collector Thread



Eventually, the realtime thread finishes using the A samples and drops its reference to the message.

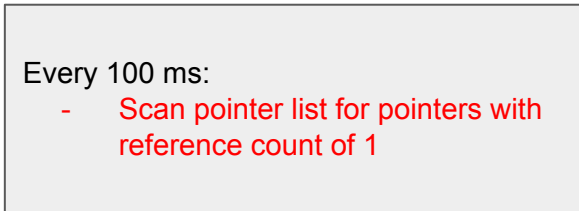




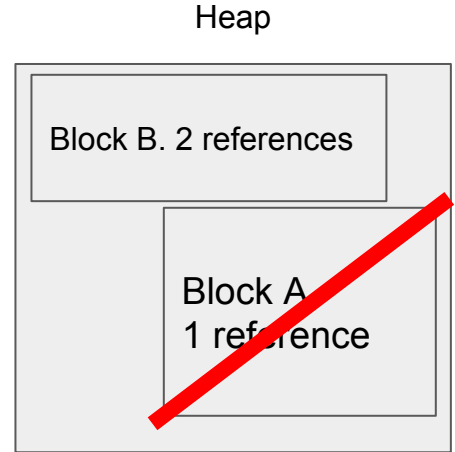
Collector Thread Pointer List



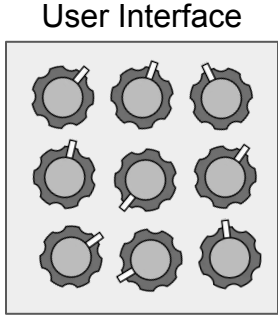
Collector Thread



Sometime later, the collector thread notices that pointer A is only being referred to by the collector thread and frees the memory.



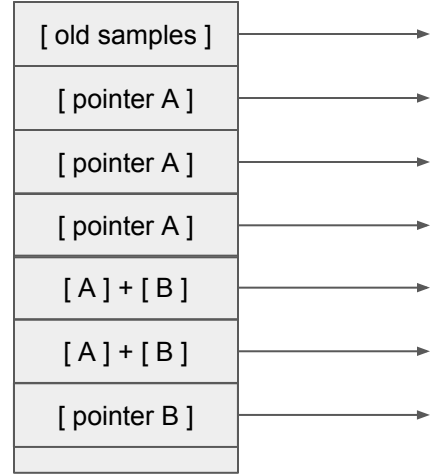




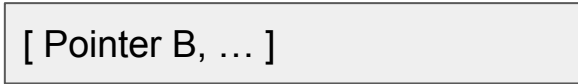
### UI Thread



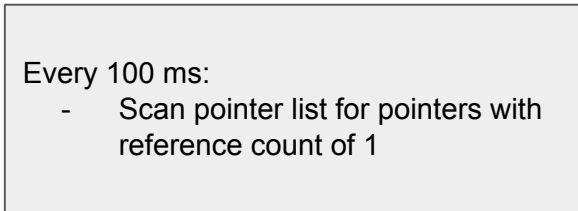
### Realtime Thread



### Collector Thread Pointer List



### Collector Thread



The process starts over!

### Heap

