

MODULE <i>HKFM</i>	
EXTENDS <i>Integers, Sequences</i>	
CONSTANTS <i>Client, Song</i>	
VARIABLES <i>inbox, state</i>	
Definitions	
<i>vars</i>	$\triangleq \langle inbox, state \rangle$
<i>Server</i>	$\triangleq \text{CHOOSE } x : x \notin Client$
<i>Node</i>	$\triangleq Client \cup \{Server\}$
<i>Idx</i>	$\triangleq Nat \cup \{-1\}$
<i>Playlist</i>	$\triangleq Seq(Song)$
<i>Playhead</i>	$\triangleq [i : Idx, t : Nat]$
<i>Stopped</i>	$\triangleq [i \mapsto -1, t \mapsto 0]$
<i>State</i>	$\triangleq [playlist : Playlist, playhead : Playhead]$
<i>InitState</i>	$\triangleq [playlist \mapsto \langle \rangle, playhead \mapsto Stopped]$
<i>Message</i>	$\triangleq [action : \{\text{"sync"}\}, data : State] \cup$ $[action : \{\text{"add"}\}, data : Song, sender : Client] \cup$ $[action : \{\text{"seek"}, \text{"skip"}\}, data : Playhead, sender : Client]$
<i>TypeOK</i>	$\triangleq \wedge inbox \in [Node \rightarrow Seq(Message)]$ $\wedge state \in [Node \rightarrow State]$
Message Constructors	
<i>SyncMsg</i>	$\triangleq$ $[action \mapsto \text{"sync"}, data \mapsto state'[Server]]$
<i>AddMsg(client, song)</i>	$\triangleq$ $[action \mapsto \text{"add"}, data \mapsto song, sender \mapsto client]$
<i>SeekMsg(client, playhead)</i>	$\triangleq$ $[action \mapsto \text{"seek"}, data \mapsto playhead, sender \mapsto client]$
<i>SkipMsg(client, playhead)</i>	$\triangleq$ $[action \mapsto \text{"skip"}, data \mapsto playhead, sender \mapsto client]$
Client Actions	
<i>SendAdd(self, song)</i>	$\triangleq$ LET $msg \triangleq AddMsg(self, song)$ IN

$$\begin{aligned}
& \wedge \text{inbox}' = [\text{inbox} \text{ EXCEPT } ![Server] = \text{Append}(\text{inbox}[Server], \text{msg})] \\
& \wedge \text{UNCHANGED } state \\
\\
\text{RecvSync}(self) & \triangleq \\
& \wedge \text{inbox}[self] \neq \langle \rangle \\
& \wedge \text{Head}(\text{inbox}[self]).\text{action} = \text{"sync"} \\
& \wedge \text{LET} \\
& \quad \text{newState} \triangleq \text{Head}(\text{inbox}[self]).\text{data} \\
& \text{IN} \\
& \quad \wedge \text{inbox}' = [\text{inbox} \text{ EXCEPT } ![self] = \text{Tail}(\text{inbox}[self])] \\
& \quad \wedge \text{state}' = [\text{state} \text{ EXCEPT } ![self] = \text{newState}] \\
\\
\text{SendSeek}(self) & \triangleq \\
& \text{LET} \\
& \quad \text{playhead} \triangleq \text{state}[self].\text{playhead} \\
& \quad \text{msg} \triangleq \text{SeekMsg}(self, [\text{playhead} \text{ EXCEPT } !.t = \text{playhead}.t + 1]) \\
& \text{IN} \\
& \quad \wedge \text{playhead} \neq \text{Stopped} \\
& \quad \wedge \text{inbox}' = [\text{inbox} \text{ EXCEPT } ![Server] = \text{Append}(\text{inbox}[Server], \text{msg})] \\
& \quad \wedge \text{UNCHANGED } state \\
\\
\text{SendSkip}(self) & \triangleq \\
& \text{LET} \\
& \quad \text{playhead} \triangleq \text{state}[self].\text{playhead} \\
& \quad \text{msg} \triangleq \text{SkipMsg}(self, \text{playhead}) \\
& \text{IN} \\
& \quad \wedge \text{playhead} \neq \text{Stopped} \\
& \quad \wedge \text{inbox}' = [\text{inbox} \text{ EXCEPT } ![Server] = \text{Append}(\text{inbox}[Server], \text{msg})] \\
& \quad \wedge \text{UNCHANGED } state
\end{aligned}$$


---

Server Actions

$$\begin{aligned}
\text{BroadcastSync} & \triangleq \\
& \wedge \text{inbox}' = [n \in \text{Node} \mapsto \text{IF } n = \text{Server} \\
& \quad \text{THEN } \text{Tail}(\text{inbox}[n]) \\
& \quad \text{ELSE } \text{Append}(\text{inbox}[n], \text{SyncMsg})] \\
\\
\text{RecvAdd} & \triangleq \\
& \wedge \text{inbox}[Server] \neq \langle \rangle \\
& \wedge \text{LET} \\
& \quad \text{server} \triangleq \text{state}[Server] \\
& \quad \text{msg} \triangleq \text{Head}(\text{inbox}[Server]) \\
& \text{IN} \\
& \quad \wedge \text{msg}.\text{action} = \text{"add"} \\
& \quad \wedge \text{LET} \\
& \quad \quad \text{newPlaylist} \triangleq \text{Append}(\text{server}.\text{playlist}, \text{msg}.\text{data})
\end{aligned}$$

$$\begin{aligned}
& \text{newPlayhead} \triangleq \text{IF } \text{server.playhead} = \text{Stopped} \\
& \quad \text{THEN } [i \mapsto \text{Len}(\text{server.playlist}), t \mapsto 0] \\
& \quad \text{ELSE } \text{server.playhead} \\
& \text{IN} \\
& \quad \wedge \text{state}' = [\text{state} \text{ EXCEPT } ![\text{Server}] = [\text{playlist} \mapsto \text{newPlaylist}, \\
& \quad \quad \quad \text{playhead} \mapsto \text{newPlayhead}]] \\
& \quad \wedge \text{BroadcastSync} \\
\text{RecvSeek} & \triangleq \\
& \quad \wedge \text{inbox}[\text{Server}] \neq \langle \rangle \\
& \quad \wedge \text{LET} \\
& \quad \quad \text{server} \triangleq \text{state}[\text{Server}] \\
& \quad \quad \text{msg} \triangleq \text{Head}(\text{inbox}[\text{Server}]) \\
& \quad \text{IN} \\
& \quad \quad \wedge \text{msg.action} = \text{"seek"} \\
& \quad \quad \wedge \text{msg.data.i} = \text{server.playhead.i} \\
& \quad \quad \wedge \text{msg.data.t} > \text{server.playhead.t} \\
& \quad \quad \wedge \text{state}' = [\text{state} \text{ EXCEPT } ![\text{Server}].\text{playhead.t} = \text{msg.data.t}] \\
& \quad \quad \wedge \text{BroadcastSync} \\
\text{RecvSkip} & \triangleq \\
& \quad \wedge \text{inbox}[\text{Server}] \neq \langle \rangle \\
& \quad \wedge \text{LET} \\
& \quad \quad \text{server} \triangleq \text{state}[\text{Server}] \\
& \quad \quad \text{msg} \triangleq \text{Head}(\text{inbox}[\text{Server}]) \\
& \quad \text{IN} \\
& \quad \quad \wedge \text{msg.action} = \text{"skip"} \\
& \quad \quad \wedge \text{msg.data.i} = \text{server.playhead.i} \\
& \quad \quad \wedge \text{LET} \\
& \quad \quad \quad \text{newIndex} \triangleq \text{server.playhead.i} + 1 \\
& \quad \quad \quad \text{newPlayhead} \triangleq \text{IF } \text{newIndex} < \text{Len}(\text{server.playlist}) \\
& \quad \quad \quad \quad \text{THEN } [i \mapsto \text{newIndex}, t \mapsto 0] \\
& \quad \quad \quad \quad \text{ELSE } \text{Stopped} \\
& \quad \text{IN} \\
& \quad \quad \wedge \text{state}' = [\text{state} \text{ EXCEPT } ![\text{Server}].\text{playhead} = \text{newPlayhead}] \\
& \quad \quad \wedge \text{BroadcastSync}
\end{aligned}$$


---

Randomly lose a message from an *inbox*

$$\begin{aligned}
\text{Remove}(i, \text{seq}) & \triangleq \\
& \quad [j \in 1 \dots (\text{Len}(\text{seq}) - 1) \mapsto \text{IF } j < i \text{ THEN } \text{seq}[j] \text{ ELSE } \text{seq}[j + 1]] \\
\text{LoseMsg} & \triangleq \\
& \quad \exists n \in \text{DOMAIN } \text{inbox} : \\
& \quad \quad \exists i \in \text{DOMAIN } \text{inbox}[n] :
\end{aligned}$$

$$\begin{aligned} &\wedge \text{inbox}' = [\text{inbox} \text{ EXCEPT } ![n] = \text{Remove}(i, \text{inbox}[n])] \\ &\wedge \text{UNCHANGED } \text{state} \end{aligned}$$

---

Spec

$$\begin{aligned} \text{Init} &\triangleq \\ &\wedge \text{inbox} = [n \in \text{Node} \mapsto \langle \rangle] \\ &\wedge \text{state} = [n \in \text{Node} \mapsto \text{InitState}] \\ \text{Next} &\triangleq \\ &\vee \exists \text{self} \in \text{Client}, \text{song} \in \text{Song} : \text{SendAdd}(\text{self}, \text{song}) \\ &\vee \exists \text{self} \in \text{Client} : \text{RecvSync}(\text{self}) \\ &\vee \exists \text{self} \in \text{Client} : \text{SendSeek}(\text{self}) \\ &\vee \exists \text{self} \in \text{Client} : \text{SendSkip}(\text{self}) \\ &\vee \text{RecvAdd} \\ &\vee \text{RecvSeek} \\ &\vee \text{RecvSkip} \\ &\vee \text{LoseMsg} \\ \text{Spec} &\triangleq \\ &\text{Init} \wedge \square[\text{Next}]_{\text{vars}} \end{aligned}$$

---

Invariants

$$\begin{aligned} \text{PlayheadOK} &\triangleq \\ &\text{LET} \\ &\quad \text{server} \triangleq \text{state}[\text{Server}].\text{playhead} \\ &\text{IN} \\ &\quad \vee \text{server} = \text{Stopped} \\ &\quad \vee \forall c \in \text{Client} : \\ &\quad \quad \text{LET} \\ &\quad \quad \quad \text{client} \triangleq \text{state}[c].\text{playhead} \\ &\quad \quad \text{IN} \\ &\quad \quad \quad \vee \text{client}.i < \text{server}.i \\ &\quad \quad \quad \vee \wedge \text{client}.i = \text{server}.i \\ &\quad \quad \quad \wedge \text{client}.t \leq \text{server}.t \\ \text{SeekAdvances} &\triangleq \\ &\quad \vee \text{state}[\text{Server}].\text{playhead}.i \neq \text{state}'[\text{Server}].\text{playhead}.i \\ &\quad \vee \text{state}[\text{Server}].\text{playhead}.t \leq \text{state}'[\text{Server}].\text{playhead}.t \\ \text{Synced} &\triangleq \\ &\quad \forall c \in \text{Client} : \text{state}[c] = \text{state}[\text{Server}] \end{aligned}$$


---

THEOREM  $Spec \Rightarrow \Box TypeOK$

THEOREM  $Spec \Rightarrow \Box PlayheadOK$

THEOREM  $Spec \Rightarrow \Box \Diamond Synced$

THEOREM  $Spec \Rightarrow \Box [SeekAdvances]_{vars}$

---