**ReloadingHSM State Pseudo Code**

**Key:**

Capital Case used for variables and function names (Ed’s Convention)

/\* comments \*/

// place holder for code

Only code needed for lowest level state (During func)

Write Durings for everything, but comment out the function call if not needed

**Start ReloadingHSMS (CurrentEvent)**

 // Handle history or irregular entry here

CurrentState = ReloadingCompleted;

 If (EV\_CONDITIONAL\_ENTRY == CurrentEvent)

 Save ReloadAllowed = QueryJSRReloadStatus()

 If (ReloadAllowed)

 CurrentState = RequestingBall;

 End if

 End if

 /\* Call ENTRY function for ReloadingHSMS \*/

 Run ReloadingHSMS(CurrentEvent)

End Start

**ReloadingHSMS\_t Run ReloadingHSMS (CurrentEvent)**

 Init MakeTransition, NextState, EntryEventKind, ReturnEvent

 /\* React differently based on the state you’re in \*/

Switch (CurrentState)

 Case RequestingBall:

 /\* Execute During to respond to event \*/

 CurrentEvent = During RequestingBall (CurrentEvent)

 /\* Handle all events relevant to RequestingBall \*/

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case EV\_REQUEST\_SENT:

 // Any transition action here

 NextState = ReceivingBall;

 MakeTransition = True; EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 End switch

 End if

 break;

 Case ReceivingBall:

 /\* Execute During to respond to event \*/

 CurrentEvent = During ReceivingBall (CurrentEvent)

 /\* Handle all events relevant to ReceivingBall \*/

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case EV\_BALL\_RECEIVED:

 // Any transition action here

 If (NumBallsRemaining < MaxBalls)

 NextState = RequestingBall;

 Else

 NextState = ReloadingCompleted;

 End if

 MakeTransition = True; EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 End switch

 End if

 break;

 Case ReloadingCompleted:

 /\* Execute During to respond to event \*/

 CurrentEvent = DuringReloadingCompleted (CurrentEvent);

 // Final state – no transitions

 Break;

 End switch

 /\* Handle transitions between states \*/

 If (MakeTransition)

 /\* Exit this state \*/

 CurrentEvent.EventType = EV\_EXIT;

 Run ReloadingHSMS(CurrentEvent)

/\* Enter next state \*/

CurrentState = NextState; // Change to next state

CurrentEvent.EventType = EventEntryKind;

Run ReloadingHSMS(CurrentEvent)

 End if

Return ReturnEvent

End Run

**ReloadingHSMS\_t Query ReloadingHSMS (CurrentEvent)**

 Return CurrentState;

End Query

**DuringRequestingBall (CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // Execute Entry actions here

 Send Ball Request

 /\* No lower level SM(s) \*/

 Else if (EV\_ExIt == CurrentEvent)

 /\* No lower level SM(s) \*/

 // Execute Exit actions here

 Rotate Hopper

 Else

 /\* No lower level SM(s) \*/

 // Execute any internal event response here

 End if

 return ReturnEvent

End During

**DuringReceivingBall(CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // Execute Entry actions here

 /\* No lower level SM(s) \*/

 Else if (EV\_ExIt == CurrentEvent)

 /\* No lower level SM(s) \*/

 Increment Balls Remaining counter

 // Execute Exit actions here

 Else

 /\* No lower level SM(s) \*/

 // Execute any internal event response here

 End if

 return ReturnEvent

End During

**DuringReloadingCompleted(CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // Execute Entry actions here

 /\* No lower level SM(s) \*/

 Else if (EV\_ExIt == CurrentEvent)

 /\* No lower level SM(s) \*/

 // Execute Exit actions here

 Else

 /\* No lower level SM(s) \*/

 // Execute any internal event response here

 End if

 return ReturnEvent

End During