**ShootingHSM States Pseudo Code**

**Key:**

/\* comments \*/

// place holder for code

Code that can be ignored for lowest level states (no internal SM)

**Start ShootingHSMS (currentEvent)**

// Handle history or irregular entry here

If not history entry

Current state = CheckingForBall

End

/\* Call ENTRY function for ShootingHSMS \*/

Run ShootingHSMS(currentEvent)

End Start

**Run ShootingHSMS (currentEvent)**

Init MakeTransition, NextState, EntryEventKind, ReturnEvent

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

Switch (currentState)

Case CheckingForBall

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

currentEvent = DuringCheckingForBallHSMS(currentEvent)

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

If ES\_NO\_EVENT does not = currentEvent

Switch (currentEvent)

Case ES\_TIMEOUT and the parameter is Reload timer

// Any transition action here

if BallsRemaining is 0

NextState = ShootingCompleted

ReturnEvent = EV\_OUT\_OF\_BALLS

else if BallsRemaining > 0

NextState = CheckingForBall

MakeTransition = False

EntryEventKind = ES\_ENTRY

ReturnEvent = CurrentEvent

Break

End switch

End if

Break

case ShootingCompleted

//nothing in here

Break

End switch

/\* Handle transitions between states \*/

If (makeTransition)

/\* Exit this state \*/

currentEvent.EventType = ES\_EXIT

Run ShootingHSMS(currentEvent)

/\* Enter new state \*/

currentEvent.EventType = EventEntryKind

currentState = nextState

Run ShootingHSMS(currentEvent)

End if

Return returnEvent

End Run

**During CheckingForBallHSMS (currentEvent)**

Init returnEvent

If (EV\_ENTRY = currentEvent)

// start the reload timer

/\* Start lower level SM(s) \*/

// StartLowerLevelSM(currentEvent)

Else if (EV\_EXIT = currentEvent)

/\* Exit lower level SM(s) \*/

// RunLowerLevelSM(currentEvent)

// launch ball (rotate magazine)

// decrement balls remaining counter

Else

/\* Pass event to lower level SM(s) \*/

// ReturnEvent = RunLowerLevelSM(currentEvent)

End if

return ReturnEvent // or currentEvent to avoid event consumption

End During

**During ShootingCompletedHSMS (currentEvent)**

Init returnEvent

If (EV\_ENTRY = currentEvent)

// do nothing

/\* Start lower level SM(s) \*/

// StartLowerLevelSM(currentEvent)

Else if (EV\_EXIT = currentEvent)

/\* Exit lower level SM(s) \*/

// RunLowerLevelSM(currentEvent)

// do nothing

Else

/\* Pass event to lower level SM(s) \*/

// ReturnEvent = RunLowerLevelSM(currentEvent)

// Execute any internal event response here

End if

return ReturnEvent // or currentEvent to avoid event consumption

End During