**RobotControlHSM 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 RobotControlHSMS (CurrentEvent)**

 // Handle history or irregular entry here

If not history entry

 CurrentState = WaitingToStart;

 End

 /\* Call ENTRY function for RobotControlHSMS \*/

 Run RobotControlHSMS(CurrentEvent)

End Start

**RobotControlHSMS\_t Run RobotControlHSMS (CurrentEvent)**

 Init MakeTransition, NextState, EntryEventKind, ReturnEvent

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

Switch (CurrentState)

 Case WaitingToStart

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

 CurrentEvent = DuringWaitingToStart (CurrentEvent);

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

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case EV\_ROUND\_1\_START:

 // Any transition action here

 NextState = Jousting;

 MakeTransition = True;

 EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 End switch

 End if

 Break;

 Case Jousting

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

 CurrentEvent = DuringJousting (CurrentEvent);

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

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case EV\_RECESS\_1\_START:

Case EV\_RECESS\_2\_START:

Case EV\_RECESS\_3\_START:

Case EV\_CRAWL\_SPEED\_REACHED:

 // Any transition action here

 NextState = GetHome;

 MakeTransition = True;

 EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 End switch

 End if

 Break;

 Case GetHome:

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

 CurrentEvent = DuringGetHome (CurrentEvent);

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

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case EC\_AWAY\_WALL\_HIT:

 // Any transition action here

 NextState = FiringAtGoal;

 MakeTransition = True;

 EntryEventKind = EV\_CONDITIONAL\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 Case EC\_HOME\_WALL\_HIT:

 // Any transition action here

 NextState = Reloading;

 MakeTransition = True;

 EntryEventKind = EV\_CONDITIONAL\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 End switch

 End if

 Break;

 Case Reloading:

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

 CurrentEvent = DuringReloading (CurrentEvent);

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

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case EV\_ROUND\_3\_START:

 // Any transition action here

 NextState = Jousting;

 MakeTransition = True;

 EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 Case EV\_RECESS\_3\_START:

 // Any transition action here

 NextState = GetHome;

 MakeTransition = True;

 EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 End switch

 End if

 Break;

 Case FiringAtGoal:

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

 CurrentEvent = DuringFiringAtGoal (CurrentEvent);

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

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case EV\_ROUND\_2\_START:

 If (CheckingForGoal == QueryFiringAtGoal())

 // Any transition action here

 NextState = Jousting;

 MakeTransition = True;

 EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 End if

 break;

 Case EV\_OUT\_OF\_BALLS:

 If (Shooting == QueryFiringAtGoal())

 // Any transition action here

 NextState = Jousting;

 MakeTransition = True;

 EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 End if

 break;

 Case EV\_RECESS\_2\_START:

 // Any transition action here

 NextState = GetHome;

 MakeTransition = True;

 EntryEventKind = EV\_ENTRY;

 ReturnEvent = CurrentEvent;

 break;

 End switch

 End if

 Break;

 End switch

 /\* Handle transitions between states \*/

 If (MakeTransition)

 /\* Exit this state \*/

 CurrentEvent.EventType = EV\_EXIT;

 Run RobotControlHSMS(CurrentEvent)

/\* Enter next state \*/

CurrentState = NextState; // Change to next state

CurrentEvent.EventType = EventEntryKind;

Run RobotControlHSMS(CurrentEvent)

 End if

Return ReturnEvent

End Run

**RobotControlHSMS\_t Query RobotControlHSMS (CurrentEvent)**

 Return CurrentState;

End Query

**RobotControlHSMS\_t During WaitingToStart (CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // No Entry actions

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

 StartWaitingToStartHSMS(CurrentEvent)

 Else if (EV\_ExIt == CurrentEvent)

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

 RunWaitingToStartHSMS(CurrentEvent)

 // No Exit actions

 Else

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

 ReturnEvent = RunWaitingToStartHSMS (CurrentEvent)

 // No internal event response

 End if

 return ReturnEvent // or CurrentEvent to avoid event consumption

End During

**RobotControlHSMS\_t During Jousting (CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // No Entry actions

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

 StartJoustingHSMS(CurrentEvent)

 Else if (EV\_ExIt == CurrentEvent)

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

 RunJoustingHSMS(CurrentEvent)

 // No Exit actions

 Else

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

 ReturnEvent = RunJoustingHSMS (CurrentEvent)

 // No internal event response

 End if

 return ReturnEvent // or CurrentEvent to avoid event consumption

End During

**RobotControlHSMS\_t During FiringAtGoal (CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // No Entry actions

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

 StartFiringAtGoalHSMS(CurrentEvent)

 Else if (EV\_ExIt == CurrentEvent)

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

 RunFiringAtGoalHSMS(CurrentEvent)

 // No Exit actions

 Else

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

 ReturnEvent = RunFiringAtGoalHSMS (CurrentEvent)

 // No internal event response

 End if

 return ReturnEvent // or CurrentEvent to avoid event consumption

End During

**RobotControlHSMS\_t During Reloading (CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // No Entry actions

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

 StartReloadingHSMS(CurrentEvent)

 Else if (EV\_ExIt == CurrentEvent)

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

 RunReloadingHSMS(CurrentEvent)

 // No Exit actions

 Else

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

 ReturnEvent = RunReloadingHSMS (CurrentEvent)

 // No internal event response

 End if

 return ReturnEvent // or CurrentEvent to avoid event consumption

End During

**RobotControlHSMS\_t During GetHome (CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // No Entry actions

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

 StartGetHomeHSMS(CurrentEvent)

 Else if (EV\_ExIt == CurrentEvent)

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

 RunGetHomeHSMS(CurrentEvent)

 // No Exit actions

 Else

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

 ReturnEvent = RunGetHomeHSMS (CurrentEvent)

 // No internal event response

 End if

 return ReturnEvent // or CurrentEvent to avoid event consumption

End During