**JoustingHSM 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 JoustingHSMS (CurrentEvent)**

 // Handle history or irregular entry here

If not history entry

 CurrentState = Accelerating;

 End

 /\* Call ENTRY function for JoustingHSMS \*/

 Run JoustingHSMS(CurrentEvent)

End Start

**JoustingHSMS\_t RunJoustingHSMS (CurrentEvent)**

 Init MakeTransition, NextState, EntryEventKind, ReturnEvent

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

Switch (CurrentState)

 Case Accelerating:

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

 CurrentEvent = During Accelerating (CurrentEvent)

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

 If ES\_NO\_EVENT does not == CurrentEvent

 Switch (CurrentEvent)

 Case Accel Timeout

 // Any transition action here

 If (MaxSpeed <= GetSpeed())

 NextState = Attacking;

 EntryEventKind = EV\_CONDITIONAL\_ENTRY;

 Else

 NextState = Accelerating;

 EntryEventKind = EV\_ENTRY;

 End if

 MakeTransition = True; ReturnEvent = CurrentEvent;

 Break;

 End switch

 End if

 Break

 Case Attacking:

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

 CurrentEvent = DuringAttacking(CurrentEvent);

 // Final state – no transitions

 Break;

 End switch

 /\* Handle transitions between states \*/

 If (MakeTransition)

 /\* Exit this state \*/

 CurrentEvent.EventType = EV\_EXIT;

 Run JoustingHSMS(CurrentEvent)

/\* Enter next state \*/

CurrentState = NextState; // Change to next state

CurrentEvent.EventType = EventEntryKind;

Run JoustingHSMS(CurrentEvent)

 End if

Return ReturnEvent

End Run

**JoustingHSMS\_t Query JoustingHSMS (CurrentEvent)**

 Return CurrentState;

End Query

**DuringAccelerating(CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // Execute Entry actions here

 Start Accel Timer

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

 Else if (EV\_ExIt == CurrentEvent)

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

 // Execute Exit actions here

 Increase speed

 Else

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

 // Execute any internal event response here

 End if

 return ReturnEvent

End During

**JoustingHSMS\_t DuringAttacking (CurrentEvent)**

 Init ReturnEvent

 If (EV\_ENTRY == CurrentEvent)

 // Execute Entry actions here

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

 StartAttackingHSMS(CurrentEvent);

 Else if (EV\_ExIt == CurrentEvent)

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

 RunAttackingHSMS(CurrentEvent);

 // Execute Exit actions here

 Else

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

 ReturnEvent = RunAttackingHSMS(CurrentEvent);

 // Execute any internal event response here

 End if

 return ReturnEvent // or CurrentEvent to avoid event consumption

End During