**TemplateHSM 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 AttackingHSMS (CurrentEvent)**

// Handle history or irregular entry here

CurrentState = Lancing;

If (EV\_CONDITIONAL\_ENTRY == CurrentEvent)

Save GameState = QueryJSRGameplay()

Switch(GameState)

Case Round1:

Case Round2:

CurrentState = Lancing;

Break;

Case Round3:

Case SuddenDeath:

CurrentState = LancingAndShooting;

Break;

End switch

End if

/\* Call ENTRY function for AttackingHSMS \*/

Run AttackingHSMS(CurrentEvent)

End Start

**AttackingHSMS\_t Run AttackingHSMS (CurrentEvent)**

Init MakeTransition, NextState, EntryEventKind, ReturnEvent

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

Switch (CurrentState)

Case Lancing:

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

CurrentEvent = DuringLancing(CurrentEvent);

// Final state – no transitions

Break;

Case LancingAndShooting:

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

CurrentEvent = DuringLancingAndShooting (CurrentEvent);

// Final state – no transitions

Break;

End switch

/\* Handle transitions between states \*/

If (MakeTransition)

/\* Exit this state \*/

CurrentEvent.EventType = EV\_EXIT;

Run AttackingHSMS(CurrentEvent)

/\* Enter next state \*/

CurrentState = NextState; // Change to next state

CurrentEvent.EventType = EventEntryKind;

Run AttackingHSMS(CurrentEvent)

End if

Return ReturnEvent

End Run

**AttackingHSMS\_t Query AttackingHSMS (CurrentEvent)**

Return CurrentState;

End Query

**AttackingHSMS\_t DuringLancing(CurrentEvent)**

Init ReturnEvent

If (EV\_ENTRY == CurrentEvent)

// Execute Entry actions here

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

StartLancingHSMS(CurrentEvent)

Else if (EV\_ExIt == CurrentEvent)

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

RunLancingHSMS(CurrentEvent)

// Execute Exit actions here

Else

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

ReturnEvent = RunLanchingHSMS (CurrentEvent)

// Execute any internal event response here

End if

return ReturnEvent // or CurrentEvent to avoid event consumption

End During

**AttackingHSMS\_t DuringLancingAndShooting(CurrentEvent)**

Init ReturnEvent

If (EV\_ENTRY == CurrentEvent)

// Execute Entry actions here

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

StartLancingHSMS(CurrentEvent);

StartShootingHSMS(CurrentEvent);

Else if (EV\_ExIt == CurrentEvent)

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

RunLanchingHSMS(CurrentEvent);

RunShootingHSMS(CurrentEvent);

// Execute Exit actions here

Else

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

ReturnEvent = RunShootingHSMS(CurrentEvent);

ReturnEvent = RunLanchingHSMS (CurrentEvent)

// Execute any internal event response here

End if

return ReturnEvent // or CurrentEvent to avoid event consumption

End During