```
*****************************************************************
```

Pseudo-code for Constructing SM (second level from top)

```
Module variables: CurrentState, MyTeam, CSG, CSR, ActiveArea,
GreenScore, RedScore, GameStatus,ResponseReadyByte, Acknowledge,
NextLocation
States: Drive to Checkin, Checkin, Drive to Shooting, Shooting, Drive
to Loading, Loading
Events Posted: ES_QUERY, ES_DONE_DRIVING, ES_DRIVE_CHECKIN,
ES_DRIVE_SHOOT, ES_SHOOT, ES_LOAD, ES_DRIVE_LOAD
```

RunConstructingSM

```
Parameters: ES_Event: the event to process
Returns: ES_Event: an event to return

Set MakeTransition variable to false, because we are not making a
transition currently
Set state type variable NextState to CurrentState
Set event type EntryEventKind to ES_ENTRY (default to normal entry to
new state)
Set event type ReturnEvent to CurrentEvent, assuming we are not
consuming event

  switch ( CurrentState )
      case DRIVE_TO_CHECKIN :
        Execute DuringDriveToCheckin. Pass CurrentEvent.
           If an event is active (not ES_NO_EVENT)
switch (CurrentEvent)            {
         case ES_DONE_DRIVING :
                NextState = CHECKIN
                mark that we are taking a transition
                EntryEventKind.EventType = ES_ENTRY
                consume or re-map this event for the upper
               level state machine
                ReturnEvent = ES_NO_EVENT
        else (Current Event is now ES_NO_EVENT. Correction
2/20/17). Probably means that CurrentEvent was consumed by       lower
level. in that case update ReturnEvent to    CurrentEvent.
case CHECKIN
        execute DuringCheckin(CurrentEvent)
        //process any events
        If an event is active (not ES_NO_EVENT)
           switch (CurrentEvent)
              case ES_DRIVE_CHECKIN
```

```
                 NextState = DRIVE_TO_CHECKIN
       mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
                 ReturnEvent = ES_NO_EVENT
            case ES_DRIVE_SHOOT
      NextState = SHOOTING
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
                 ReturnEvent = ES_NO_EVENT
        else (Current Event is now ES_NO_EVENT. Correction
2/20/17). Probably means that CurrentEvent was consumed by       lower
level. in that case update ReturnEvent to    CurrentEvent.
case DRIVE_TO_SHOOTING
        Execute DuringDriveToShooting(CurrentEvent)
        //process any events
        If an event is active (not ES_NO_EVENT)
           switch (CurrentEvent)
              case ES_DONE_DRIVING
                 NextState = SHOOTING
      mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
                 ReturnEvent = ES_NO_EVENT
         case ES_TIMEOUT
           if ( CurrentEvent.EventParam is
                DRIVING_DEBUG_TIMER )
                     start DRIVING_DEBUG_TIMER
                     NextState = SHOOTING
       mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
                 ReturnEvent = ES_NO_EVENT
 else (Current Event is now ES_NO_EVENT. Correction
2/20/17). Probably means that CurrentEvent was consumed by       lower
level. in that case update ReturnEvent to    CurrentEvent.
case SHOOTING
      Execute DuringShooting(CurrentEvent)
```

```
        //process any events
        If an event is active (not ES_NO_EVENT)
           switch (CurrentEvent)
              case ES_DRIVE_CHECKIN
                 NextState = DRIVE_TO_CHECKIN
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
              case ES_SHOOT
                 NextState = SHOOTING
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
             case ES_DRIVE_LOAD
                 NextState = DRIVE_TO_LOADING
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
             case ES_TIMEOUT
      if (CurrentEvent.EventParam = ShootingTimer)
                 NextState = SHOOTING
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
       else (Current Event is now ES_NO_EVENT. Correction
2/20/17). Probably means that CurrentEvent was consumed by       lower
level. in that case update ReturnEvent to    CurrentEvent.

      case DRIVE_TO_LOADING
        Execute DuringDriveToLoading(CurrentEvent)
        //process any events
        If an event is active (not ES_NO_EVENT)
        switch (CurrentEvent)
              case ES_DONE_DRIVING
                 NextState = LOADING
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
      else (Current Event is now ES_NO_EVENT. Correction
     2/20/17). Probably means that CurrentEvent was consumed by
           lower   level. in that case update ReturnEvent to
           CurrentEvent.
     case LOADING
        Execute DuringLoading(CurrentEvent)
        //process any events
        If an event is active (not ES_NO_EVENT)
        switch (CurrentEvent)
              case ES_DRIVE_CHECKIN
                 NextState = DRIVE_TO_CHECKIN
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
             case ES_DRIVE_SHOOT
                 NextState = DRIVE_TO_SHOOTING
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
           case ES_LOAD
                 NextState = LOADING
                 mark that we are taking a transition
                 EntryEventKind.EventType = ES_ENTRY
                 consume or re-map this event for the upper
                level state machine
      else (Current Event is now ES_NO_EVENT. Correction
     2/20/17). Probably means that CurrentEvent was consumed by
           lower   level. in that case update ReturnEvent to
           CurrentEvent.
   If we are making a state transition
      Execute exit function for current state
 RunConstructingSM(ExitEvent)
      Modify state variable to next state
     Execute entry function for new state. this defaults to ES_ENTRY
      RunConstructingSM(EntryEventKind)
    return(ReturnEvent)
```

StartConstructingSM

Takes nothing, returns nothing.

```
  MyTeam = GetMyTeam()
      if (CurrentEvent is not ES_ENTRY_HISTORY )
```

```
       CurrentState = ENTRY_STATE
  RunConstructingSM(CurrentEvent)
```

QueryConstructingSM

```
Takes nothing, returns the current state of the constructing state
machine.
  return(CurrentState)
```

DuringDriveToCheckin

```
Takes in an event, returns an event.
   assume no re-mapping or consumption: set return event as passed in
     event.
     process ES_ENTRY, ES_ENTRY_HISTORY & ES_EXIT events
   if Event is ES_ENTRY or Event is ES_ENTRY_HISTORY
     DETERMINE THE CHECKIN LOCATION by posting ES_QUERY event to
Comm. EventParam = GameStatusCMD.
     after that start lower level machine: Driving SM
   else if Event is ES_EXIT
       on exit, give the lower levels a chance to clean up first
       RunDrivingSM(Event);
   else pass the event down
     if Event is ES_RESPONSE_READY
          run lower level state machine
          ReturnEvent = RunDrivingSM(Event)
   return ReturnEvent to allow the lower level machine to remap the
current event
```

DuringCheckin

```
Takes in an event, returns an event.
   assume no re-mapping or consumption: set return event as passed in
     event.
     process ES_ENTRY, ES_ENTRY_HISTORY & ES_EXIT events
   if Event is ES_ENTRY or Event is ES_ENTRY_HISTORY
after that start lower level machine: Checkin SM
   else if Event is ES_EXIT
       on exit, give the lower levels a chance to clean up first
       RunCheckinSM(Event)
   else pass the event down
       ReturnEvent = RunCheckinSM(Event)
   return ReturnEvent to allow the lower level machine to remap the
current event
```

DuringDriveToShooting

```
Takes in an event, returns an event.
   assume no re-mapping or consumption: set return event as passed in
     event.
     process ES_ENTRY, ES_ENTRY_HISTORY & ES_EXIT events
   if Event is ES_ENTRY or Event is ES_ENTRY_HISTORY
     Determine the shooting location by posting ES_QUERY event to
Comm. EventParam = GameStatusCMD.
     after that start lower level machine: DrivingSM
   else if Event is ES_EXIT
       on exit, give the lower levels a chance to clean up first
       post ES_QUERY event to comm master sm. Event param is
        GameStatusCMD.
       RunDrivingSM(Event)
   else pass the event down
       ReturnEvent = RunDrivingSM(Event)
   return ReturnEvent to allow the lower level machine to remap the
current event
```

DuringShooting

```
Takes in an event, returns an event.
   assume no re-mapping or consumption: set return event as passed in
     event.
   if Event is ES_ENTRY or Event is ES_ENTRY_HISTORY
        start ShootingTimer
       after that start lower level machine: ShootingSM
   else if Event is ES_EXIT
       on exit, give the lower levels a chance to clean up first
       RunShootingSM(Event)
        KillFlyWheel()
   else pass the event down
       ReturnEvent = RunShootingSM(Event)
        if shooting timer timed out then we need to go somewhere else
             and check in again
                if event is the shooting timeout
                     if we are out of COWs
                          new event is ES_DRIVE_LOAD
                     else
                          new event is ES_DRIVE_CHECKIN
                     post the new event to master sm

   return ReturnEvent to allow the lower level machine to remap the
current event
```

DuringDriveToLoading

```
Takes in an event, returns an event.
   assume no re-mapping or consumption: set return event as passed in
     event.
   if Event is ES_ENTRY or Event is ES_ENTRY_HISTORY
     determine the loading location by posting ES_QUERY event to
Comm. EventParam = GameStatusCMD.
     after that start lower level machine: DrivingSM
   else if Event is ES_EXIT
       on exit, give the lower levels a chance to clean up first
       RunDrivingSM(Event)
   else pass the event down
       ReturnEvent = RunDrivingSM(Event)
return ReturnEvent to allow the lower level machine to remap the
current event
```

DuringLoading

```
Takes in an event, returns an event.
   assume no re-mapping or consumption: set return event as passed in
     event.
   if Event is ES_ENTRY or Event is ES_ENTRY_HISTORY
     after that start lower level machine: LoadingSM
   else if Event is ES_EXIT
on exit, give the lower levels a chance to clean up first
      RunLoadingSM(Event)
   else pass the event down
ReturnEvent = RunLoadingSM(Event)
return ReturnEvent to allow the lower level machine to remap the
current event
```