```
/*************************************************************************************
Module
TotGameTimer.c
Description
 This implements a state machine for the tot insertion and game timing subsystem.
*************************************************************************************/
InitTotGameTimer
        Enable pin B1 as a digital input with internal pullup resistor
        Initialize the PWM for all servos of interest
        Set the Tot Servo to its CCW limit
        Set CurrentState to InitPState
        Post ES_INIT event to TotGameTimer
End InitTotGameTimer


PostTotGameTimer
        Post event to this state machine's queue
End PostTotGameTimer


CheckTotBeamBreak
        Get the Current Tot Beam Status by reading the input line B1
        If the Current Tot Beam Status is not equal to the Last Tot Beam Status
                If the Current Tot Beam Status is low
                        Post BeamBroken event to TotGameTimer
                Else if the Current Tot Beam Status is high
                        Post BeamActivated event to TotGameTimer
        Set the Last Tot Beam Status equal to the Current Tot Beam Status
End CheckTotBeamBreak

RunTotGameTimer
```

```
Use GetServoPosition() to save the position of the tot servo
Current State is InitPState
    If Event is ES_INIT
        Set Tot servo to its CCW limit
        Set Tot Blocking Servo to its open state (to allow tot to fall through)
        Set NextState to Waiting2Start
` End InitPState block
Current State is Waiting2Start
    If Event is BeamBroken
        Init Tot Servo Timer with Tot Servo Delay
        Post GameStart event to the post list
        Set Tot Blocking Servo to its closed state (to prevent tot from falling through)
        Set NextState to GameInProgress
End Waiting2Start block
Current State is GameInProgress
    If Event is a Tot Servo Timer timeout
        Use GetServoPosition() to save the position of the tot servo
        Set tot servo to its current position plus the Tot Servo Increment
        Init Tot Servo Timer with Tot Servo Delay
        If the tot servo position is at its CW limit
            Post GameWon event to the post list
            Init Shake timer
            Set Next State to Shaking
            Set Shake Count to 0
    If Event is GameLost
        Init Tot Reset Timer
        Set Next State to LostMovingToEnd
End GameInProgress block
Current State is LostMovingToEnd
```

```
    If Event is a Tot Rest Timer timeout
        Set tot servo position to its current position plus the Servo Dump increment
        If the tot servo position is at its CW limit
            Init Shake timer
            Set Next State to Shaking
            Set Shake Count to 0
End LostMovingToEnd block
Current State is Shaking
    If Event is a Tot Shake Timer timeout
        If the value of Shake Count is even
            Set tot servo position to a shake increment CCW of the CW limit
            Init Shake timer
            Increase Shake Count by 1
        Else If the value of Shake Count is odd
            Set tot servo position to its CW limit
            Init Shake Timer
            Increase Shake Count by 1
`       Else if Shake Count is equal to desired number of shakes
            Init Tot Reset Timer
            Set Next State to Resetting
            Set Shake Count = 0
End Shaking block
Current State is Resetting
    If Event is a Tot Reset Timer Timeout
        Set tot servo position to its current position + Tot Reset increment
        Init Tot Reset Timer
    If the Tot servo position is at its CCW limit
        Set tot blocking servo to its open state
        Set Next State to Waiting2Start
    End Resetting block
`   Set Current State equal to Next State
End RunTotGameTimer
```