

Tic Tac Doh PseudoCode

EventCheckers.c

```
boolean checkButton1(void);
If more than debounce interval has passed since last check
    Check state of button1
    If state of button 1 different than previous check
        If button 1 is down
            Post Button1Press Event to list 00 and 02
            Return true
        End if
    Else
        Return false
    End if
Record time of last sample
End(s)
```

```
boolean checkButton2(void);
If more than debounce interval has passed since last check
    Check state of button2
    If state of button 2 different than previous check
        If button 2 is down
            Post Button2Press Event to list 00 and 02
            Return true
        End if
    Else
        Return false
    End if
Record time of last sample
End(s)
```

```
boolean checkAnalog(void);
Post event AnalogUpdate to list 01
Return true
```

Call State Machines in this order: AnalogStuffSM, TicTacDohSM

AnalogStuffSM.c - PostEventList 01

```
CurrentState = QueryTicTacDohSM()
Switch CurrentState
    Case InitPState

        Case WaitingToStart
```

```

    Do Nothing
    Set coinTossStartTime = getTime()
Case CoinToss
    Read both accelerometers
    Keep track of each players shaking amplitudes
    Keep track of time within this state
    If Timeout
        Calculate who won the shake off
        Post event PlayerWinsToss with parameter 1 or 2 to Lists 00 02
    End
Case Player1Aiming
    Keep track of player 1's time remaining
    Read accelerometer of Player1
    Move servos to aim laser
    Read potentiometer of Player 2
    Move Board servo to match pot
    If Player 1 Time decrements past 2 sec interval
        Post event Player1TimeDecrement with to list 02
    If Player 1 Time all runs out
        Post Event Player1Timeout to lists 00 and 02
Case Player2 Aiming
    Keep track of player 2's time remaining
    Read accelerometer of Player2
    Move servos to match motion
    Read potentiometer of Player 1
    Move Board servo to match pot
    If Player 2 Time decrements past 2 sec interval
        Post event Player2TimeDecrement to list 02
    If Player 2 Time all runs out
        Post Event Player2Timeout to lists 00 and 02
Case Victory Dance
    Keep track of victory dance time
    If victory dance time decrements by 1 sec interval
        Post event VictoryDanceTimeDecrement to list 02
    If victory dance time expires
        Post event VictoryDanceTimeout to list 00 and 02
    End if

```

BoardScoreSM.c – PostEventList 02

```

CurrentState = QueryTicTacDohSM()
Switch CurrentState
    Case InitPState

        Case WaitingToStart
            Turn off all lights

```

```

    If event Button1Press or Button2Press
        Turn on Coin toss light
        Turn on all timer lights
    End
Case CoinToss
    If event Player1 WinsToss
        Turn off coin toss light
        Turn on Player 1 turn light
    If event Player2WinsToss
        Turn off coin toss light
        Turn on Player 2 turn light
Case Player1Aiming
    If event Button1Press
        Check phototransistor inputs
        If a target is hit and it is not opposing players color
            Update board state and corresponding lights
            Post event TargetHit to List 00
        End if
        If current board state indicates someone won
            Post event WinEvent to List 00
            Turn on all LEDs in winning players color
        Else if current board state full
            See which player has more squares
            Turn on all LEDs in winning players color
            Post WinEvent to List 00
        End if
    End if
    If event Player1TimeDecrement
        Turn off corresponding timer LED
    Endif
    If event Player1Timeout
        Turn on all LEDs in Player2's color
    End if
Case Player2 Aiming
    If event Button2Press
        Check phototransistor inputs
        If a target is hit and it is not opposing players color
            Update board state and corresponding lights
            Post event TargetHit to List 00
        End if
        If current board state indicates someone won
            Post event WinEvent to List 00
            Turn on all LEDs in winning players color
        Else if current board state full
            See which player has more squares
            Turn on all LEDs in winning players color

```

```

        Post WinEvent to List 00
    End if
End if
If event Player2TimeDecrement
    Turn off corresponding timer LED
Endif
If event Player2Timeout
    Turn on all LEDs in Player1's color
End if

```

```

Case VictoryDance
    If event VictoryDanceTimeDecrement
        Toggle states of all LEDs in winning players color
    Else if event VictoryDanceTimeout
        Turn off all LEDs
    End if
End case

```

TicTacDohSM.c - PostEventList 00

```

Switch CurrentState
Case InitPState
    ADCInit();
    InitializeIOs()
    CurrentState = WaitingToStart
Case WaitingToStart
    If event Button1Press or Button2Press
        CurrentState = CoinToss
    End if
Case CoinToss
    If event Player1WinsToss
        CurrentState = Player1Aiming
    Else if event Player2WinsToss
        CurrentState = Player2Aiming
    End if
Case Player1Aiming
    If event TargetHit
        CurrentState = Player2Aiming
    Else if event Player1Timeout
        CurrentState = VictoryDance
    Else if event WinEvent
        CurrentState = VictoryDance
    End if
Case Player2Aiming
    If event TargetHit
        CurrentState = Player1Aiming
    Else if event Player2Timeout
        CurrentState = VictoryDance
    Else if event WinEvent
        CurrentState = VictoryDance
    End if
Case VictoryDance

```

If event VictoryDanceTimeout
CurrentState = WaitingToStart