

```

// 8 July 2024
//WORKING COPY OF 'goLeft' ROUTINE BEFORE AMENDMENTS MADE 8 JULY 2024
MAINLY FOR 'otherTrack'

//MARK: - goLeft decides whether to change lanes or not
func goLeft(teeVeh: inout [NodeData]) async {

    //teeVeh[index].mySetGap ~3 secs = min gap to vehicle in front
    //These values used to test distance between this vehicle & the
    one behind in the other lane.
    let oRearDecel: CGFloat = 3.5 //m/s2 used to calc spd diff
    (2.4-5.8secs x 3.5m/s2) = 8.4-20.3 kph
    var oRearMaxGap: CGFloat //((2.4-5.8) secs
    var oRearMinGap: CGFloat //min gap for overtaking XXX
    var oRearGapRange: CGFloat
    var maxORearSpdDiff: CGFloat //((2.4-5.8secs x 3.5m/s2) =
    8.4-20.3 kph

    var oRearSub: CGFloat //Used to calc speed
    difference
    var oRearOKGap: CGFloat //Used to store min allowable
    oRearGap

    var oFrontSub: CGFloat //Used to calc speed
    difference
    var oFrontOKGap: CGFloat //Used to store min
    allowable oRearGap

    var bigGap: CGFloat = 0 //bigGap = 110% of 3 sec gap.
    Calc'd for each vehicle during loop.

    /// Setup multiplier to work out minimum speed for change into left lane
    ///- Result = (100-numVehicles)[result capped @ 80] * 11/80 + 5
    ///- min spd = 16kph if <= 20 Vehicles
    ///- min spd = 5kph if 100 Vehicles
    var minChangeLaneSpdL: CGFloat //Minimum speed where lane change
    permitted!
    /// Setup multiplier to work out minimum speed for change into right lane
    ///- Result = (100-numVehicles)[result capped @ 80] * 23/80 + 5
    ///- min spd = 28kph if <= 20 Vehicles
    ///- min spd = 5kph if 100 Vehicles
    var minChangeLaneSpdR: CGFloat //Minimum speed where lane change
    permitted!
    //NOTE: May have to reduce the above when 'numVehicles' becomes
    large!!!
    var tmp = CGFloat(numVehicles)
    if tmp < 20 { tmp = 20 }
    tmp = 100 - tmp
    var tmp2: CGFloat = 23 / 80 // = 0.2875 = kph diff
    //23 = 28kph - 5kph
    minChangeLaneSpdR = tmp * tmp2 + 5.0

```

```

tmp2 = 11 / 80 // = 0.1375 = kph diff
//11 = 16kph - 5kph
minChangeLaneSpdL = tmp * tmp2 + 5.0

//Tests indicated vehicle speeds for various numVehicles limited
as follows (due to proximity to vehicle in front):
// 20 Vehicles 32kph ;Formulas above prevent lane changes
@ low speeds
// 30 Vehicles 24kph ;subject to no. of vehicles.
// 40 Vehicles 22kph ;
// 50 Vehicles 19kph ; -> Left Lane 5kph 100 vehicles
// 60 Vehicles 15kph ; 28kph 20 vehicles
// 70 Vehicles 11kph ;
// 80 Vehicles 8kph ; -> Right Lane 5kph 100 vehicles
// 110 Vehicles 6kph ; 16kph 20 vehicles

//Start loop
for indx in teeVeh.indices {
    // for (indx, vehc) in teeVeh.enumerated() {
    if indx == 0 { continue } //Skip loop for element[0] =
    All Vehicles

    //vvvvvvvvvv Force Lane to 0 or 1 vvvvvvvvvv
    if teeVeh[indx].indicator != .off { continue } //Lane change
    already in progress
    //NOTE: If inst's below omitted then each veh will O/T or
    return ONLY once!
    // (value of indicator changed by 0.002! - laneChange not
    EXACTLY 1.0 lanes)
    if teeVeh[indx].lane > 0.5 { //Ensure lane only = 1 or 0
    when here!
        teeVeh[indx].lane = 1
    } else {
        teeVeh[indx].lane = 0
    }
    //^^^^^^^^^^ Force Lane to 0 or 1 ^^^^^^^^^^

    oRearMaxGap = teeVeh[indx].mySetGap //~ 3 secs (2.4-5.8
    secs)
    // oRearMinGap = (oRearMaxGap / 6) //(~3 secs / 6) =
    ~0.5 secs (0.4-0.933s) = min gap for overtaking
    oRearMinGap = (teeVeh[indx].myMinGap * 2) //(0.4-1.2secs *
    2) = (0.8-2.4secs) = min gap for overtaking
    oRearGapRange = (oRearMaxGap - oRearMinGap)
    maxORearSpdDiff = (oRearMaxGap * oRearDecel) //(2.4-5.8secs @
    3.5m/s2) = 8.4-20.3kph (was~15 kph)

    //***** Test for permissible oRearGap \
    *****

```

```

oRearSub = (teeVeh[indx].oRearSpd -
teeVeh[indx].currentSpeed) //Used to calc speed difference

if oRearSub < 0 {
    oRearSub = 0 //This vehicle faster
    than oRearSpd - SAFE to allow min gap
    //NOTE: the minimum gap = 'myMinGap' (0.4-1.2secs) or
    'minGap' (3.5m), whichever is greater.
    // 'myMinGap' calc'd on oRearSpd for rear or
    currentSpeed for front!
} else {
    if oRearSub > maxORearSpdDiff {
        oRearSub = maxORearSpdDiff //This vehicle slower
        than (oRearSpd - ~15kph) - require max gap
    }
}

oRearOKGap = oRearMinGap + ((oRearGapRange / maxORearSpdDiff)
* oRearSub) //returns min gap allowed = 0.5 - 3 secs
oRearOKGap = (teeVeh[indx].oRearSpd * oRearOKGap) / 3.6 // =
permissible gap in metres

if oRearOKGap < minGap { oRearOKGap = minGap } //Limit
minimum gap to 3.5m at low speeds

if teeVeh[indx].oRearGap <= oRearOKGap { continue }
//oRearGap insufficient to change lanes. End.
//oRearGap OK - Test other factors.

//***** Test for permissible oRearGap /\
//*****
//***** Test for permissible oFrontGap /\
//*****
//For now same constants used for front as for back. Name not
changed as may later be changed.
oFrontSub = (teeVeh[indx].currentSpeed -
teeVeh[indx].oFrontSpd) //Used to calc speed difference

if oFrontSub < 0 {
    oFrontSub = 0 //This vehicle slower
    than oFrontSpd - SAFE to allow min gap
    //NOTE: the minimum gap = 'myMinGap' (0.4-1.2secs) or
    'minGap' (3.5m), whichever is greater.
    // 'myMinGap' calc'd on oRearSpd for rear or
    currentSpeed for front!
} else {
    if oFrontSub > maxORearSpdDiff {
        oFrontSub = maxORearSpdDiff //This vehicle faster
        than (oFrontSpd + ~15kph) - require max gap
    }
}

```

```

oFrontOKGap = oRearMinGap + (oRearGapRange / maxORearSpdDiff
    * oFrontSub) //returns min gap allowed = ~0.5 - ~3 secs
oFrontOKGap = (teeVeh[indx].currentSpeed * oFrontOKGap) / 3.6
// = permissible gap in metres

if oFrontOKGap <= minGap { oFrontOKGap = minGap }
//Limit minimum gap to 3.5m at low speeds

if teeVeh[indx].otherGap <= oFrontOKGap { continue }
//oFrontGap insufficient to change lanes. End.
//oRearGap OK - Test other factors.

//***** Test for permissible oFrontGap /\
//*****
//***** Test for minimum speed to permit lane change /\
//*****
//
//      if teeVeh[indx].currentSpeed < minChangeLaneSpdR {
continue } //Don't permit lane change when vehicle speed < 5-28
kph.

//***** Test for minimum speed to permit lane change /\
//*****

bigGap = ((1.05 * teeVeh[indx].mySetGap) *
    teeVeh[indx].currentSpeed) / 3.6 //bigGap = 105% of 2.4-5.8
    sec gap = 2.52-6.09sec gap
//bigGap = metres travelled during (1.05 * 'mySetGap')
    seconds @ currentSpeed.
//Bigger so lane change starts b4 vehicle slows down.
//Can't be too big ????? TBC ?????
if teeVeh[indx].lane == 0 { //Preferred lane
    (Left)
    //***** Test for permissible gap/otherGap
        from lane 0 /\ *****
    if teeVeh[indx].currentSpeed < minChangeLaneSpdR {
        continue } //Don't permit lane change when vehicle
        speed < 5-28 kph (subject to numVehicles).
    if teeVeh[indx].frontSpd >= teeVeh[indx].currentSpeed {
        continue //Stay in left lane
    } else { //Going faster than vehicle in front
        if teeVeh[indx].gap > bigGap { continue } //gap >
            105% 3 sec gap. Stay in left lane.
        if teeVeh[indx].gap >= teeVeh[indx].otherGap {
            continue } //LHS gap > otherGap. Stay in this lane.
    }
// if teeVeh[indx].otherGap <= minGap { continue } //Limit minimum
    gap to 3.5m at low speeds. ALREADY DONE!
// print("to 1\t\t(indx)\t\t(teeVeh[indx].lane)")
// teeVeh[indx].lane = 1 //Overtake (done elsewhere in SKAction)
    teeVeh[indx].indicator = .overtake
    //Move to right (overtaking) lane

```

```

// sKLAllVehicles[indx].indicator = .overtake      //Move to right
(overtaking) lane
        teeVeh[indx].startIndicator = true //Flag
        used to start lane change
// print("to 1\t\t(indx)\t\t(teeVeh[indx].lane)")
        continue
    }
//May later compare frontSpd to oFrontSpd too!

    //***** Test for permissible gap/otherGap
    from lane 0 /\ *****
}        //End in lane 0 checks

if teeVeh[indx].lane == 1 {        //Overtaking lane
(Right)
    //***** Test for permissible gap/otherGap
    from lane 1 /\ *****
    if teeVeh[indx].currentSpeed < minChangeLaneSpdL {
        continue } //Don't permit lane change when vehicle
        speed < 5-16 kph (subject to numVehicles).
//        if teeVeh[indx].otherGap > teeVeh[indx].mySetGap {
    if teeVeh[indx].otherGap > bigGap {
        if teeVeh[indx].otherTrack == true { continue }
        //Stay in lane if otherTrack
        //        teeVeh[indx].lane = 0
        //Return to left lane
        teeVeh[indx].indicator = .endOvertake
        //Return to left lane
        //        sKLAllVehicles[indx].indicator
        = .endOvertake //Return to left lane
        teeVeh[indx].startIndicator = true //Flag used
        to start lane change
        //
        print("ta\t\t(indx)\t\t(teeVeh[indx].lane)\tenablSpd:
        \(enableMinSpeed)")
        continue //End this vehicle
    } else { //otherGap <= mySetGap
        if teeVeh[indx].otherGap >= teeVeh[indx].gap {
//            if true == true { //Temporarily REPLACES
//                INSTRUCTION ABOVE – FOR TESTING ONLY!!!
//if teeVeh[indx].otherGap <= minGap { continue } //Limit
//                minimum gap to 3.5m at low speeds ALREADY DONE!
//                teeVeh[indx].lane = 0 //Return to
//                left lane
//                teeVeh[indx].indicator = .endOvertake
//                //Return to left lane
//                sKLAllVehicles[indx].indicator =
//                .endOvertake //Return to left lane
//                teeVeh[indx].startIndicator = true //Flag
//                used to start lane change
//                print("tb 0\t\t(indx)\t\t(teeVeh[indx].lane)")

```

```
//           let tst =
Vehicle.startOvertake(sKLAllVehicles[indx])
                continue
        }
    }
//***** Test for permissible gap/otherGap from lane 1 /\
*****
    }           //End in lane 1 checks
}           //End 'for' loop
//           return teeVeh
}           //end of goLeft function
```