```
;source code

extensions [ sound ] ;; to produce various sounds during simulation
especially ambulance

breed[pedestrians pedestrian]
breed[cars car]
breed[ambulances ambulance]
breed[water]


globals
[
  no-of-cars-met-with-accident ;;to determine how many cars met with
an accident
  emergency-dead              ;; to determine whether emergency
vehicle moved out of scope or not
  emergency-lane              ;; used as a temporary variable to
store the lane of ambulance
  ecounter                    ;; used in determining when to make
ambulance move out of space
  emergency-north-east        ;; used to store the
  emergency-north-west        ;; state of traffic lights
  emergency-south-east        ;; when emergency vehicle
  emergency-south-west        ;; created

  pedestrian-lane0            ;;
  pedestrian-lane1            ;;
  pedestrian-lane2            ;;  To group Which patches
  pedestrian-lane3            ;;  corresponds to which
  pedestrian-lane4            ;;  pedestrian lane
  pedestrian-lane5            ;;
  pedestrian-lane6            ;;
  pedestrian-lane7            ;;

  glane0                      ;;
  glane1                      ;;
  glane2                      ;;  To group which patches
  glane3                      ;;  corresponds to which
  glane4                      ;;  road lanes
  glane5                      ;;
  glane6                      ;;
  glane7                      ;;

  intersections               ;;;;co-ordinates where all the lanes
meet
  North-west                  ;;;; patch at (-2,2)
  North-east                  ;;;; patch at (2,2)
  South-west                  ;;;; patch at (-2,-2)
  South-east                  ;;;; patch at (2,-2)

  counter
  snow-counter
  tempcounter
  tempcounter1
  Is-ambulance-already-created?

]




patches-own
[
  pedestrian-lane0?             ;;
  pedestrian-lane1?             ;;
  pedestrian-lane2?             ;; To determine which
  pedestrian-lane3?             ;; patch belongs to
  pedestrian-lane4?             ;; which pedestrian lane
  pedestrian-lane5?             ;;
  pedestrian-lane6?             ;;
  pedestrian-lane7?             ;;

  plane0?                       ;;
  plane1?                       ;;
  plane2?                       ;; To determine which
  plane3?                       ;; patch belongs to which
  plane4?                       ;; road lanes
  plane5?                       ;;
  plane6?                       ;;
  plane7?                       ;;

  intersection?

]




cars-own
[
  wait-time                    ;; For how much time a car is waiting
at a red signal
  actual-speed                 ;; speed of the turtle corresponding
to real world
  actual-acceleration          ;; acceleration of the car in the
real world
  actual-deceleration          ;; deceleration of the car in the
real world
  speed                        ;; speed of each car
  max-speed                    ;; maximum speed of the car
  min-speed                    ;; minimum speed of the car
  lane                         ;; randomly deciding on to which lane
car has to be placed
  actual-lane                  ;; depending upon the lane assigned,
randomly assigning x-cor or y-cor
  change-lane                  ;; new lane into which each car has
to be changed
```

```
  front-car                     ;; if there is any car in the front
  acceleration                  ;; tells how much a car should
accelerate
  deceleration                  ;; tells how much a car should
decelerate
  rash-car?                     ;; To determine whether a driver is
reckless or not
  dying?                        ;; To know whether a car is met with
an accident or not
]


ambulances-own
[
  ambulance-actual-speed          ;; speed of the ambulance
corresponding to real world
  ambulance-actual-acceleration  ;; acceleration of the ambulance
corresponding to real world
  ambulance-actual-deceleration  ;; deceleration of the ambulance
correspondhin to real world
  front-vehicle                   ;; determines which vehicle is
infront of the ambulance
  speed                           ;; speed of the ambulance
  max-speed                       ;; maximum speed an ambulance can
reach
  min-speed                       ;; minimum speed an ambulance
should maintain
  decision-speed                  ;; used to move the ambulances from
halt state to movement state
  lane                            ;; lane on which ambulance should
be moved
  change-lane                     ;; determines the lane into which
ambulance should take a turn
  acceleration                    ;; accleration of the ambulance
  deceleration                    ;; deceleration of the ambulance
]




pedestrians-own
[
 direction                        ;; direction in which pedestrians
should move along the lanes
 p-tempspeed                      ;; to temporarily store the speed
of the pedestrian
 p-speed                          ;; speed of the pedestrians
 p-minspeed                       ;; minimum speed a pedestrian
should move
 p-maxspeed                       ;; maximum speed a pedestrian
should move
 p-acceleration                   ;; acceleration of the pedestrian
 p-deceleration                   ;; deceleration of the pedestrian
 pedestrian-lane                  ;; lane on which a pedestrian is
```

```
moving
 change-pedestrian-lane          ;; lane onto which a pedestrian
should change while taking a turn
]


water-own
[
  snow-speed                     ;; speed of the snow turtles
]

;;----------------------Procedure to create snow turtles when snow
option is selected--------------------------------------------------
to create-snow

  ifelse ( snow )
  [

 create-water 100
 [
   set snow-speed 2
   set color white
   set shape "circle"
   set size 0.5
   set heading 180
   set xcor random min-pxcor
   set ycor random max-pycor
 ]

  create-water 100
 [
   set snow-speed 2
   set color white
   set shape "circle"
   set size 0.5
   set heading 180
   set xcor random min-pxcor
   set ycor random min-pycor
 ]
  create-water 100
 [
   set snow-speed 2
   set color white
   set shape "circle"
   set size 0.5
   set heading 180
   set xcor random max-pxcor
   set ycor random min-pycor
 ]
  create-water 100
 [
   set snow-speed 2
   set color white
   set shape "circle"
   set size 0.5
   set heading 180
   set xcor random max-pxcor
   set ycor random max-pycor
 ]
  for-snow
  ]
  [
    user-message ( word "Turn snow on ")  ;; Asking user to select
snow option first before creating snow turtles
  ]
end

;;-----------------------Procedure responsible for snow turtles to
move so as to create snow effect-----------------------------------
to for-snow

  if ( snow )
  [
      ask water
  [
    fd snow-speed
  ]

    if ( ticks = number-of-ticks )
    [
    ask water [ die ]
    stop
    ]
  ]
end

;;--------------------------------Procedure responsible for creation
of lanes along axis, for creating cars,
;;                                  intersections, and also
initializes all the variables used in subsequent
;;
procedures------------------------------------------;;

to setup

  clear-all
  reset-ticks
  create-text-file
  set snow-counter 0
  set Is-ambulance-already-created? false
  set no-of-cars-met-with-accident 0
   ask patches
  [
    set intersection? false
    set pedestrian-lane0? false          ;; At the start of the
simulation and before
    set pedestrian-lane1? false          ;; creating pedestrian
lanes, no patch will
```

```
    set pedestrian-lane2? false          ;; come under pedestrian-
lanes group. Therefore
    set pedestrian-lane3? false          ;; making this field false
will ensure that no
    set pedestrian-lane4? false          ;; pedestrian patch is
there before creation of
    set pedestrian-lane5? false          ;; pedestrian lanes.
    set pedestrian-lane6? false          ;;
    set pedestrian-lane7? false          ;;

    set plane0? false                    ;;
    set plane1? false                    ;; At the start of the
simulation and before creating lanes
    set plane2? false                    ;; in the NetLogo world, no
patch will come under road lanes group.
    set plane3? false                    ;; Therefore making this
field false will ensure that no patch
    set plane4? false                    ;; belongs to road lane
before creation of roads
    set plane5? false                    ;;
    set plane6? false                    ;;
    set plane7? false                    ;;
  ]
  set tempcounter 0
  set tempcounter1 0
  draw-margins
  set-intersections
  setup-pedestrian-lanes
  setup-cars
  setup-pedestrians
  rash-driving-car
  set North-west "green"
  set counter 0
  change-green-light-NW

end

;;------------------------------------Procedure is responsible for
giving the user a provision to create a
;;                                   text file and to store that
text file at his/her desired location.
;;                                   After opening this file,status
of all the turtles are entered---------------------------------;;

to create-text-file
  file-open user-new-file
end

;;-------------------------------------Procedure is responsible for
entering the report
;;                                    generated by all the turtles
after each tick. This procedure
;;                                    enters wait time, speed, who
is the front-vehicle regarding each
```

```
;;                                      turtle after every
tick--------------------------------------------------------;;

to write-to-file
  file-print " "
  file-print ( word " Tick Number - " ticks )
  file-print ( word "
------------------------------------------------------------" )
  ask cars
  [
    file-print ( word " Who - " who "---speed - " actual-speed "---
lane - " lane " ---wait-time - " wait-time " ----front car - "
front-car)
    file-print " "
  ]

  if ( counter >= 1 )
  [
    file-print( word " ***********************")
    file-print ( word " Average wait time of all cars is - " ((mean
[ wait-time] of cars) * 10) "mts" )
    file-print( word " ***********************")
    file-print (word " Average speed of all the cars is - " mean
[ actual-speed] of cars "mph")
    file-print (word "  ************************" )
    file-print (word " Density of cars on lane-0 - " count cars with
[lane = 0] )
    file-print (word " Density of cars on lane-1 - " count cars with
[lane = 1] )
    file-print (word " Density of cars on lane-2 - " count cars with
[lane = 2] )
    file-print (word " Density of cars on lane-3 - " count cars with
[lane = 3] )
    file-print (word " Density of cars on lane-4 - " count cars with
[lane = 4] )
    file-print (word " Density of cars on lane-5 - " count cars with
[lane = 5] )
    file-print (word " Density of cars on lane-6 - " count cars with
[lane = 6] )
    file-print (word " Density of cars on lane-7 - " count cars with
[lane = 7] )
  ]
end



to write-to-file-when-dying
  ask cars
  [
    file-print( word "The car - " who " died at lane - " lane )
  ]
end

;;-------------------------------Procedure is responsible for
```

```
creation of intersection.
;;                              In the sense that, it specifies
which all patches comes
;;                              under the intersections. After
determining the patches, it
;;                              will set the intersection? property
to true.------------------------------------------;;

to set-intersections
  set intersections patches with [ ( pxcor >= -2 and pxcor <= 2 and
pycor = 2 ) or
                                   ( pxcor = -2  and pycor >= -2 and
pycor <= 2 ) or
                                   ( pxcor = 2   and pycor >= -2 and
pycor <= 2 ) or
                                   (pxcor >= -2  and pxcor <= 2 and
pycor = -2 ) or
                                   (pxcor >= -2 and pxcor <= 2 and
pycor = 1 ) or
                                    (pxcor >= -2 and pxcor <= 2 and
pycor = 0 ) or
                                    (pxcor >= -2 and pxcor <= 2 and
pycor = -1 )
  ]

  ask intersections [ set intersection? true ]
end

;;------------------------------Procedure is responsible for
creation of lanes along the
;;                              x-axis and y-axis. In total 8 lanes
are created along both
;;                              the axis and also determines which
patch comes under which lane.
;;                              If one patch come under lane 0 then
plane0? property of that patch
;;                              is made true and all remaining
false-----------------------------------------------------;;

to draw-margins
  let x-cor max-pxcor mod 2
  let y-cor max-pycor mod 2

    ;---------to get white and red margins if the max-pxcor and max-
pycor are even numbers-------------------------;;

    if ( max-pxcor mod 2 = 0) and (max-pycor mod 2 = 0 )
    [
      ask patches with [ (pxcor >= x-cor - 2)  and (pxcor <= x-cor +
2) ]
      [set pcolor white ]

    ;;;;------------------To get the green grass effect alongside
the roads
```

```
      ask patches with [ (pxcor < x-cor - 2 ) or ( pxcor > x-cor +
2 ) ]
      [set pcolor scale-color green ((random 1000) + 8000) 10
15000 ]


   ;;;;------------------To get White margins along the Y-axis
      ask patches with [ (pycor >= y-cor - 2) and (pycor <= y-cor +
2) ]
      [set pcolor white ]

    ]


   ;;---------to get white margins and red margins and if max-pxcor
and max-pycor are odd
numbers--------------------------------------------;;

   if ( max-pxcor mod 2 = 1 ) and ( max-pycor mod 2 = 1 )
   [
       ask patches with [ (pxcor >= x-cor + -3)  and (pxcor <= x-cor
+ 1) ]
       [set pcolor white ]

    ;;;;---------------------To get the green grass effect alongside
the roads
       ask patches with [ (pxcor < x-cor + -3 ) or ( pxcor >= x-cor
+ 2 ) ]
       [set pcolor scale-color green ((random 1000) + 8000) 10
15000 ]


   ;;;;----------------To get White margins along the Y-axis
       ask patches with [ (pycor >= y-cor + -3) and (pycor <= y-cor
+ 1) ]
       [set pcolor white ]

    ]


     ;;;;---------------To get the effect of two lanes on a single
road
      ask patches with [ (pxcor mod 2 = 0) and (pycor = 0) and
(pxcor != 2) and (pxcor != -2) ]
       [ set pcolor black ]
      ask patches with [ (pycor mod 2 = 0) and (pxcor = 0) ]
        [set pcolor black ]

       ;;;;;;------------To delete the red patches and make them
white at the intersection of the two roads
        ask patches with [ (pxcor >= -2 ) and (pxcor <= 2 ) and
( pycor = 0) or
                              (pycor >= -2 ) and (pycor <= 2) and
(pxcor = 0) ]
```

```
         [set pcolor white ]

;;-------------------------------------------------------------------
--------------------------------------------------------------------
----;;
 ;;------------------------------------To assign patches to their
respective
lanes---------------------------------------------------------------
-;;


 set glane0 patches with [ pxcor < -2 and (pycor = 2  or pycor =
1)]   ;;
 set glane1 patches with [ pxcor < -2 and (pycor = -2 or pycor =
-1)]  ;; assigning patches
 set glane2 patches with [ pxcor > 2  and (pycor = 2  or pycor =
1)]   ;; to different lanes
 set glane3 patches with [ pxcor > 2  and (pycor = -2 or pycor =
-1)]  ;; as per their
 set glane4 patches with [ (pxcor = 2  or pxcor = 1)  and pycor >
2]   ;; co-ordinates
 set glane5 patches with [ (pxcor = -2 or pxcor = -1) and pycor >
2]   ;;
 set glane6 patches with [ (pxcor = 2  or pxcor  = 1) and pycor <
-2]  ;;
 set glane7 patches with [ (pxcor = -2 or pxcor = -1) and pycor <
-2]  ;;



ask glane0 [ set plane0?
true]                                    ;;
ask glane1 [ set plane1?
true ]                                   ;; If a patch belongs
to lane A
ask glane2 [ set plane2?
true ]                                   ;; then planeA?
property of that patch
ask glane3 [ set plane3?
true ]                                   ;; is made true, so
that by seeing that
ask glane4 [ set plane4?
true ]                                   ;; particular field one
can know to which
ask glane5 [ set plane5?
true]                                    ;; lane that patch
belongs to
ask glane6 [ set plane6?
true ]                                   ;;
ask glane7 [ set plane7?
true]                                    ;;


ask patches with [ pxcor = 0 and pycor = 0 ] [ set pcolor black ]
```

```
end

;;------------------------------------Procedure is responsible for
creating pedestrian-lanes.
;;                                   Creation of pedestrian-lanes
includes grouping patches depending
;;                                  upon the lane on which they
exist.-------------------------;;

to setup-pedestrian-lanes

  ask patches with [ ( pxcor >= 3 and pycor = 3 ) or
                     ( pxcor >= 3 and pycor = -3 ) or
                     ( pxcor = 3  and pycor >= 3 ) or
                     ( pxcor = 3  and pycor <= -3 ) or
                     ( pxcor = -3 and pycor <= -3 ) or
                     ( pxcor = -3 and pycor >= 3 ) or
                     ( pxcor <= -3 and pycor = 3 ) or
                     ( pxcor <= -3 and pycor = -3 ) ]
   [ set pcolor black ]

 set pedestrian-lane0 patches with [ pxcor <= -4 and pycor = 3 ]
 set pedestrian-lane1 patches with [ pxcor <= -4 and pycor = -3 ]
 set pedestrian-lane2 patches with [ pxcor >= 4 and pycor = 3 ]
 set pedestrian-lane3 patches with [ pxcor >= 4 and pycor = -3 ]
 set pedestrian-lane4 patches with [ pxcor = 3 and pycor >= 4 ]
 set pedestrian-lane5 patches with [ pxcor = -3 and pycor >= 4 ]
 set pedestrian-lane6 patches with [ pxcor = 3 and pycor <= -4 ]
 set pedestrian-lane7 patches with [ pxcor = -3 and pycor <= -4 ]

 ask pedestrian-lane0 [ set pedestrian-lane0? true ]
 ask pedestrian-lane1 [ set pedestrian-lane1? true ]
 ask pedestrian-lane2 [ set pedestrian-lane2? true ]
 ask pedestrian-lane3 [ set pedestrian-lane3? true ]
 ask pedestrian-lane4 [ set pedestrian-lane4? true ]
 ask pedestrian-lane5 [ set pedestrian-lane5? true ]
 ask pedestrian-lane6 [ set pedestrian-lane6? true ]
 ask pedestrian-lane7 [ set pedestrian-lane7? true ]


end

;;-----------------------------------Procedure is responsible for
the creation
;;                                  of pedestrians along the
pedestrian-lanes. This method
;;                                  will assign values to different
parameters of pedestrians
;;                                  and makes sure that they only
move in lanes-------------------------;;


to setup-pedestrians
```

```
let color-list [yellow pink cyan gray orange sky violet ]
    create-pedestrians 10
    [
      set size 1
      set shape "circle"
      set color one-of color-list
      set p-speed random 0.5
      set p-minspeed 0.2
      set p-maxspeed 0.5
      set p-deceleration 0.05
      set p-acceleration 0.0035

      ;set direction random 2
      set pedestrian-lane random 8
      set change-pedestrian-lane 0

      if ( pedestrian-lane = 0 )
      [
        set xcor random min-pxcor
        set ycor 3
        set heading 90

      ]

      if ( pedestrian-lane = 1 )
      [
        set xcor random min-pxcor
        set ycor -3
        set heading -90

      ]

      if ( pedestrian-lane = 2 )
      [
        set xcor random max-pxcor
        set ycor 3
        set heading 90
      ]

      if ( pedestrian-lane = 3 )
      [
        set xcor random max-pxcor
        set ycor -3
       set heading -90
      ]

      if ( pedestrian-lane = 4 )
      [
        set xcor 3
        set ycor random max-pycor
        set heading 180
      ]

      if ( pedestrian-lane = 5 )
```

```
        [
          set xcor -3
          set ycor random max-pycor
          set heading 0
        ]

        if ( pedestrian-lane = 6 )
        [
          set xcor 3
          set ycor random min-pycor
          set heading 180
        ]

        if ( pedestrian-lane = 7 )
        [
          set xcor -3
          set ycor random min-pycor
          set heading 0
        ]

        pedestrians-only-in-lanes
      ]
end

;;------------------------------------Procedure is responsible for
creating pedestrian turtles
;;                                    only in lanes and not outside
of the lanes. It checks the
;;                                    co-ordinates of each and then
decides to do the needed operation--------------------;;

to pedestrians-only-in-lanes
  if ( pedestrian-lane = 0 or pedestrian-lane = 1 )
  [
    if ( xcor >= -3 )
    [
      let temp random min-pxcor
      ifelse ( temp >= -3 )
      [
        pedestrians-only-in-lanes
      ]
      [
        set xcor temp
      ]
    ]
  ]

  if ( pedestrian-lane = 2 or pedestrian-lane = 3 )
  [
     if ( xcor <= 3 )
     [
       let temp random max-pxcor
       ifelse ( temp <= 3 )
       [
          pedestrians-only-in-lanes
        ]
        [
          set xcor temp
        ]
      ]
    ]

  if ( pedestrian-lane = 4 or pedestrian-lane = 5 )
  [
    if ( ycor <= 3 )
    [
      let temp random max-pycor
      ifelse ( temp <= 3 )
      [
        pedestrians-only-in-lanes
      ]
      [
       set ycor temp
      ]
    ]
  ]

  if ( pedestrian-lane = 6 or pedestrian-lane = 7 )
  [
    if ( ycor >= -3 )
    [
      let temp random min-pycor
      ifelse ( temp >= -3 )
      [
        pedestrians-only-in-lanes
      ]
      [
        set ycor temp
      ]
    ]
  ]
end

;;----------------------------------Procedure is responsible for
movement of
;;                                  pedestrians along the lanes. It
also checks
;;                                  the speed of all pedestrian
turtles--------------;;

to forward-pedestrians

  if ( p-speed < p-minspeed ) [ set p-speed p-minspeed ]
  if ( p-speed > p-maxspeed ) [ set p-speed p-maxspeed ]
  fd p-speed
end

;;----------------------------------Procedure is responsible for
```

```
calling other
;;                                methods which are responsible
for changing lanes.
;;                               Depending upon the co-ordinates
on which a pedestrian
;;                              stand, it will decide whether to
change lane or not---------------------------;;


to move-pedestrians
  ask pedestrians
  [
    assign-different-lanes-to-pedestrians

    if ( pedestrian-lane = 0 or pedestrian-lane = 5)
    [
      ifelse ( [pxcor] of patch-here = -3 and [pycor] of patch-here
= 3 )
      [
        ifelse ( change-pedestrian-lane = 0 )
        [
          assign-different-lanes-during-direction-change-to-
pedestrians

        ]
        [
         set p-tempspeed p-speed
         set p-speed 0
        ]
       ]
       [ forward-pedestrians ]
    ]

    if ( pedestrian-lane = 1 or pedestrian-lane = 7 )
    [
      ifelse ( [pxcor] of patch-here = -3 and [pycor] of patch-here
= -3 )
      [
         ifelse ( change-pedestrian-lane = 0 )
        [
          assign-different-lanes-during-direction-change-to-
pedestrians

        ]
        [
         set p-tempspeed p-speed
         set p-speed 0
        ]
      ]

      [ forward-pedestrians ]
    ]

    if ( pedestrian-lane = 3 or pedestrian-lane = 6 )
    [
      ifelse ( [pxcor] of patch-here = 3 and [pycor] of patch-here =
-3 )
      [
        ifelse ( change-pedestrian-lane = 0 )
        [
          assign-different-lanes-during-direction-change-to-
pedestrians

        ]
        [
         set p-tempspeed p-speed
         set p-speed 0
        ]
       ]
      [ forward-pedestrians ]
    ]

    if ( pedestrian-lane = 4 or pedestrian-lane = 2 )
    [
      ifelse ( [pxcor] of patch-here = 3 and [pycor] of patch-here =
3 )
      [
        ifelse ( change-pedestrian-lane = 0 )
        [
          assign-different-lanes-during-direction-change-to-
pedestrians

        ]
        [
         set p-tempspeed p-speed
         set p-speed 0
        ]
      ]
      [ forward-pedestrians ]
    ]

  ]
end

;;-----------------------------------------------Procedure is
responsible for changing of lanes.
;;                                               Depending upon the
random value assigned to change-pedestrian-lane,
;;                                               this method makes
turtle to change its direction and also its lane----------;;

to assign-different-lanes-during-direction-change-to-pedestrians


    if ( change-pedestrian-lane = 0 )
        [
          if ( pedestrian-lane = 0 )
          [
            set xcor -3
            set ycor 4
            set heading 0
            forward-pedestrians
           ]

          if ( pedestrian-lane = 1 )
          [
            set xcor -3
            set ycor -4
            set heading 180
            forward-pedestrians
           ]


          if ( pedestrian-lane = 7 )
          [
            set xcor -4
            set ycor -3
            set heading -90
            forward-pedestrians
           ]


          if ( pedestrian-lane = 4)
          [
            set xcor 4
            set ycor 3
            set heading 90
            forward-pedestrians
           ]

         if ( pedestrian-lane = 3 )
          [
            set xcor 3
            set ycor -4
            set heading 180
            forward-pedestrians
           ]


          if ( pedestrian-lane = 2 )
          [
            set xcor 3
            set ycor 4
            set heading 0
            forward-pedestrians
           ]

          if ( pedestrian-lane = 6 and heading = 0)
          [
            set xcor 4
            set ycor -3
            set heading 90
```

```
            forward-pedestrians
           ]

        ]

end

;;-------------------------------------------Procedure is
responsible for changing lanes of pedestrians.
;;                                          For example, if a
pedestrian is of lane 2 when created, and while
;;                                          moving if it is on Lane
4, the pedestrian-lane property of the turtle
;;                                          is changed from 2 to
4---------------------------------------------------------;;



to assign-different-lanes-to-pedestrians

    if ( [pedestrian-lane2?] of patch-here )
    [
      set pedestrian-lane 2
    ]

    if ( [pedestrian-lane3?] of patch-here )
    [
      set pedestrian-lane 3
    ]

    if ( [pedestrian-lane6?] of patch-here )
    [
      set pedestrian-lane 6
    ]

    if ( [pedestrian-lane7?] of patch-here )
    [
      set pedestrian-lane 7
    ]

    if ( [pedestrian-lane0?] of patch-here )
    [
      set pedestrian-lane 0
    ]

    if ( [pedestrian-lane1?] of patch-here )
    [
      set pedestrian-lane 1
    ]

    if ( [pedestrian-lane4?] of patch-here )
    [
      set pedestrian-lane 4
    ]
    if ( [pedestrian-lane5?] of patch-here )
    [
      set pedestrian-lane 5
    ]

end

;;--------------------------------------------Procedure is
responsible for calling
;;                                           different methods
which will instantiate and
;;                                          initiate different car
properties--------------------------------------------------;;

to setup-cars

 ifelse ( number-of-cars <= floor ( (((max-pxcor - 5 ) * 30)  /
100 ) * 8) )  ;; Checks whether the no of cars creating are less
than the size of all the road lanes together.
 [
      create-cars number-of-cars
      [
        car-parameters
        separate-cars ;;to prevent more than one turtle on the same
patch
        cars-only-in-lanes ;; cars should not be present beyond the
traffic lights when the simulation starts
      ]
 ]
 [
    user-message(word " Reduce the number-of-cars and then try
again")
  ]
end

;;-------------------------------------procedure is responsible for
creating
;;                                    new cars when the number of
cars created
;;                                     are less when compared to
the start of the simulation and
;;                                     call other methods to
initiate car parameters---------------------;;

to count-cars
let no_of_cars count cars

if ( no_of_cars < number-of-cars )
[
  create-cars ( number-of-cars - no_of_cars )
  [
    car-parameters
    separate-cars
    cars-only-in-lanes
    ]
    ]
    end


to car-parameters
set rash-car? false
set dying? false
set size 1
     set color blue
     set speed 0.2 + random-float 1.1
     set max-speed 1.2
     set min-speed 0.3
     set deceleration 0.05
     set acceleration 0.0035
      set lane (random 8)

;;------------------------------Assigning co-ordinates to cars
(turtles) depending upon the lanes they are assigned to----------;;

      if ( lane = 0 )
       [
         set xcor random min-pxcor
         set ycor 2
       ]

      if ( lane = 1)
       [
         set xcor random min-pxcor
         set ycor -2
        ]

       if ( lane = 2 )
       [
         set xcor random max-pxcor
         set ycor 2
       ]

       if ( lane = 3 )
       [
         set xcor random max-pxcor
         set ycor -2
        ]

       if ( lane = 4 )
       [
          set xcor 2
          set ycor random max-pycor
       ]

       if ( lane = 5 )
        [
          set xcor -2
          set ycor random max-pycor
        ]

        if ( lane = 6 )
        [
          set xcor 2
          set ycor random min-pycor
        ]

       if ( lane = 7 )
        [
          set xcor -2
          set ycor random min-pycor
        ]

        ;;---------------------------------if not on correct
lanes-------------------------------------------------------------;;

          if ( lane = 0 ) or ( lane = 2 )
         [
           if ( ycor > 2 )
            [

              die
            ]
         ]

         if ( lane = 1 ) or ( lane = 3 )
         [
           if ( ycor < -2 )
            [

              die
            ]
         ]

         if ( lane = 4 ) or ( lane = 6 )
         [
           if ( xcor > 2 )
            [

              die
            ]
         ]

         if ( lane = 5 ) or ( lane = 7 )
          [
             if ( xcor < -2 )
              [

                die
              ]
          ]
  ;;---------------------------------------------Setting headings to
```

```
the cars
--------------------------------------------------------------------
------------------;;


       if ( lane = 0 ) or ( lane = 2 )
        [ set heading 90 ]

       if ( lane = 3)
       [ set heading 270 ]

       if (lane = 1)
       [ set heading 270 ]

       ;if ( lane = 3 ) or ( lane = 1 )
        ;[ set heading 270 ]

       if ( lane = 5 ) or ( lane = 7 )

        [ set heading 0 ]


       if ( lane = 6 ) or ( lane = 4 )
        [ set heading 180 ]


        assign-actual-speed-to-cars

end

;;----------------------------------------correlates the NetLogo
speed of the car
;;                                        to the actual speed of the
car in the real world-------------------------;;

to assign-actual-speed-to-cars

  if ( speed >= 0.1 and speed <= 0.5 ) [ set actual-speed 20 ]
  if ( speed > 0.5 and speed <= 0.8 )   [ set actual-speed 45 ]
  if ( speed > 0.8 and speed <= 1.3 )   [ set actual-speed 70 ]
  set actual-acceleration 4
  set actual-deceleration 8
end

;;----------------------------------------When turtles are created,
more than one turtle
;;                                       can be placed on one patch.
Following method will check
;;                                       and assign turtles on
different patches------------------------------------;;


to separate-cars
```

```
 if any? other cars-here
 [
   ifelse ( [pcolor] of patch-ahead 1 != red )
    [
     fd 1
     separate-cars
    ]
    [
      set dying? true
      write-to-file-when-dying
      set no-of-cars-met-with-accident no-of-cars-met-with-accident
+ 1
       die
     ]
  ]
end

;;--------------------------------------------------------------------
--------------------------------------------------------------------
---------------------------------------;;

;;-------------------------------Procedure to prevent cars beyond
traffic lights when simulation
starts--------------------------------------------------------------
----------;;

to cars-only-in-lanes

if ( lane = 0 ) or ( lane = 1 )
[
 if ( xcor >= -3 )
 [
  let temp-xcor random min-pxcor
  ifelse ( temp-xcor < -2 )
  [ set xcor temp-xcor ]
  [ cars-only-in-lanes ]
 ]
]

if ( lane = 2 ) or ( lane = 3 )
[
 if ( xcor <= 3 )
 [
  let temp-xcor random max-pxcor
  ifelse ( temp-xcor > 2 )
  [ set xcor temp-xcor ]
  [ cars-only-in-lanes ]
 ]
]

if ( lane = 4 ) or ( lane = 5 )
[
 if ( ycor <= 3 )
 [
    let temp-ycor random max-pycor
    ifelse ( temp-ycor > 2 )
    [ set ycor temp-ycor ]
    [ cars-only-in-lanes ]
  ]
]

if ( lane = 6 ) or ( lane = 7 )
[
  if ( ycor >= -3 )
  [
    let temp-ycor random min-pycor
    ifelse ( temp-ycor < -2 )
    [ set ycor temp-ycor ]
    [ cars-only-in-lanes ]
  ]
]

end

;;------------------------------------------Procedure selects one-of
the cars
;;                                          as rash car and call
other methods which
;;                                          formulates rash
behaviour--------------------------------------------------------;;

to rash-driving-car

  ask one-of cars [set rash-car? true ]
  let rash-car one-of cars with [rash-car? = true ]
  ask cars with [ rash-car? = true ]
  [
    set color red
  ]

end



;;------------------------------ To let cars move along the lanes
--------------------------------------------------------------------
-----------------------------------------;;

to go

;;---------------------------if snow is off, all snow turtles will
die------------------------;;

if (not snow )
[
  if( snow-counter = 1 )
  [
    ask water [ die ]
```

```
 ]
]


if ( ticks = number-of-ticks )
[
 ask turtles [ die ]
 clear-all
 stop
]
cars-new-behaviour
rash-driver-behaviour

;;-----------------------Periodically calculates the number of cars
in the NetLogo world-----------------------------;;

if ( (counter mod 4000) = 0 )
[
 count-cars
]

periodic-change-of-lanes-in-vehicles
move
move-pedestrians
set counter counter + 1

;;-----------------------Responsible for chanmging of lights from
red to orange----------------------------------------;;

if ( counter mod Traffic-Lights-timer? = (Traffic-Lights-timer? -
4 ))
 [
   if ( North-west = "green" )
   [ ask patches with [ pxcor = 2 and pycor = 3 ] [ set pcolor
orange ] ]

   if ( North-east = "green" )
   [ ask patches with [ pxcor = 3 and pycor = -2 ] [ set pcolor
orange ] ]

   if ( South-east = "green" )
   [ ask patches with [ pxcor = -2 and pycor = -3 ] [ set pcolor
orange ] ]

   if ( South-west = "green" )
   [ ask patches with [ pxcor = -3 and pycor = 2 ] [ set pcolor
orange ] ]
 ]

;;------------------------Responsible for calling traffic lights
change method----------------------;;

if counter mod Traffic-Lights-timer? = 0
[
 change-globals-red
]

tick
write-to-file
end

;;-------------------------------------------------------------------
--------------------------------------------------------------------
---------;;

to accelerate-the-car
   set speed speed + acceleration
end

to decelerate-the-car
  set speed [ speed ] of front-car - deceleration
end

;;--------------------------------This procedure call various
methods
;;                                which are responsible for the
movement of cars
;;                                depending upon the traffic light
ahead of them------------------------------------;;

to move

    if ( North-west = "green" )
    [

      ask cars
      [
        if [intersection?] of patch-here or [plane5?] of patch-here
        [
          if ( lane = 7 )
          [
            let tempspeed speed
            set lane  5
            set speed 1.2
            forward-cars-as-per-speed
          ]
        ]

      ]
    ]

if ( North-west = "green"  )
[
 ask cars
 [
  if ( lane = 0 or lane = 2 or lane = 5 or lane = 6 or lane = 1)
  [
  set wait-time 0
```

```
   move-cars
  ]
 if ( lane = 4 or lane = 7 or lane = 3 )
  [
   set wait-time wait-time + 1
   separate-cars2
  ]
 ]
]

;;-------------------------------------------------------------------
---;;

    if (North-east = "green" )
    [
       ask cars
      [
        if [intersection?] of patch-here or [plane2?] of patch-here
        [
          if ( lane = 0 )
          [
            let tempspeed speed
            set lane  2
            set speed 1.2
           forward-cars-as-per-speed
          ]
        ]
      ]
    ]


if ( North-east = "green" and North-east != orange )
[
 ask cars
 [
  if ( lane = 4 or lane = 6 or lane = 2 or lane = 5 or lane = 1)
  [
   set wait-time 0
   move-cars
  ]
  if ( lane = 0 or lane = 7 or lane = 3 )
  [
    set wait-time wait-time + 1
    separate-cars2
  ]
 ]
]

;;-------------------------------------------------------------------
-------;;

    if ( South-east = "green")
    [
       ask cars
      [
        if [intersection?] of patch-here or [plane6?] of patch-here
        [
          if ( lane = 4 )
          [
            let tempspeed speed
            set lane  6
            set speed max-speed
            forward-cars-as-per-speed

          ]
        ]
      ]
    ]

if ( South-east = "green" and South-east != orange)
[
 ask cars
 [
  if ( lane = 3 or lane = 1 or lane = 2 or lane = 5 or lane = 6)
  [
   set wait-time 0
   move-cars
  ]
  if ( lane = 0 or lane = 4 or lane = 7 )
  [
  set wait-time wait-time + 1
   separate-cars2
  ]
 ]
]

;;--------------------------------------------------------------------
--------;;

     if ( South-west = "green")
     [

      ask cars
      [
        if [intersection?] of patch-here or [plane1?] of patch-here
        [
          if ( lane = 3 )
          [
            let tempspeed speed
            set lane  1
            set speed max-speed
            forward-cars-as-per-speed
          ]
        ]
      ]
     ]

if ( South-west = "green" )
```

```
[
 ask cars
 [
  if ( lane = 7 or lane = 5 or lane = 2 or lane = 6 or lane = 1)
  [
  set wait-time 0
   move-cars
  ]
  if ( lane = 0 or lane = 3 or lane = 4 )
  [
   set wait-time wait-time + 1
   separate-cars2
  ]
 ]
]


end

;;-------------------------------------------Procedure responsible
for changing lane
;;                                           property of cars when
they move from one lane to another
;;                                           and also call other
procedures responsible for movement of
cars-------------------------;;

to move-cars

  if ( lane = 2 )
  [
    if ([plane0?] of patch-here )
    [
      set lane 0
    ]
  ]
  if ( lane = 0 )
  [
    if ([plane2?] of patch-here )
    [
      set lane 2
    ]
  ]

  if ( lane = 4 )
  [
    if ( [plane6?] of patch-here )
    [
      set lane 6
    ]
  ]
  if ( lane = 6 )
  [
    if ([plane4?] of patch-here)
```

```
        [
          set lane 4
        ]
      ]

      if ( lane = 7 )
      [
        if ([plane5?] of patch-here)
        [
          set lane 5
        ]
      ]
      if ( lane = 5 )
      [
        if ([plane7?] of patch-here)
        [
          set lane 7
        ]
      ]

      if ( lane = 3 )
      [
        if ([plane1?] of patch-here)
        [
          set lane 1
        ]
      ]
      if ( lane = 1 )
      [
        if ( [plane3?] of patch-here)
        [
          set lane 3
        ]
      ]


        forward-cars-as-per-speed
      end

    ;;---------------------------Procedure responsible for forwarding cars
    ;;                          as per their speeds, checks speeds and
    call other procedures
    ;;                          responsible for changing of
    lanes--------------------------------------------------------;;

    to forward-cars-as-per-speed

               set front-car one-of cars-on patch-ahead 1
               ifelse ( front-car = nobody )
               [ accelerate-the-car ]
               [ decelerate-the-car ]


              if  ( speed < min-speed ) [ set speed min-speed + 1 ]
              if  ( speed > max-speed ) [ set speed max-speed  ]

             fd speed
             changing-lanes

    end

    ;;------------------------Changing of traffic lights as time
    progresses----------------------------------------------;;

    to change-globals-red

                if ( North-west = "green" )
                 [
                   set North-west "red"
                   set North-east "green"
                   change-green-light-NE
                   stop
                 ]
                  if ( North-east = "green" )
                 [
                   set South-east "green"
                   set North-east "red"
                   change-green-light-SE
                   stop
                 ]

                if ( South-east = "green" )
                 [
                   set South-west "green"
                   set South-east "red"
                   change-green-light-SW
                   stop
                 ]
                  if ( South-west = "green" )
                 [
                   set North-west "green"
                   set South-west "red"
                   change-green-light-NW
                   stop
                 ]
    end

    ;;----------------------------------Changing of traffic lights a
    time
    progresses----------------------------------------------------------
    ---------;;

    to change-green-light-NW

          if ( North-west = "green" )
           [
             ask patch -3 2   [ set pcolor green ]
             ask patch  2 3   [ set pcolor red ]
```

```
          ask patch  3 -2  [ set pcolor red ]
          ask patch -2 -3  [set pcolor red ]
        ]

end

to change-green-light-NE

        if ( North-east = "green" )
        [
          ask patch -2 2   [ set pcolor white ]
          ask patch  3 2   [ set pcolor white ]
          ask patch 2 3    [ set pcolor green ]
          ask patch  2 -3  [ set pcolor white ]
          ask patch -2 -3  [set pcolor red ]
          ask patch -3 -2   [ set pcolor white ]
          ask patch 3 -2    [ set pcolor red ]
          ask patch -3 2    [set pcolor red ]
        ]

end

to change-green-light-SE
        if ( South-east = "green" )
        [
          ask patch -3 2   [ set pcolor red ]
          ask patch -3 -2   [ set pcolor white ]
          ask patch 3 2    [ set pcolor white ]
          ask patch  2 3   [ set pcolor red ]
          ask patch  3 -2  [ set pcolor green ]
          ask patch -2 -3  [ set pcolor red ]
          ask patch 2 -3   [ set pcolor white ]

        ]

end

to change-green-light-SW
        if ( South-west = "green" )
        [
          ask patch -3 2   [ set pcolor red ]
          ask patch  2 3   [ set pcolor red ]
          ;ask patch  2 -2  [ set pcolor red ]
          ask patch -2 -3  [set pcolor  green ]
          ask patch 3 2    [ set pcolor white ]
          ask patch -3 2   [ set pcolor red ]
          ask patch 3 -2   [ set pcolor red ]
        ]

end

;;------------------------------------------Procedure responsible
for modifying various parameters
;;                                        of cars depending upon the
```

```
lane from which they are taking a turn
;;                                          into which lane ( new
lane )---------------------------;;

to changing-lanes

  if ( lane = 0 )
  [
    if (change-lane = 1 )
    [
        if ( pxcor = 1 and pycor = 1 )
        [
          set heading 180
          set lane 6
          assign-correct-coordinates
        ]
    ]
   if ( change-lane = 2 )
    [
      if ( pxcor = -2 and pycor = 2 )
      [
        set heading 0
        set lane 5
        assign-correct-coordinates
      ]
    ]
   ]

  ;;-----------------------------------------------------------------
------------;;

  if ( lane = 3 )
  [
    if ( change-lane = 2 )
    [
     if ( pxcor = 2 and pycor = -2 )
      [
       set heading 180
        set lane 6
        assign-correct-coordinates
      ]
    ]
      if ( change-lane = 1 )
      [
       if ( pxcor = -1 and pycor = -1 )
        [
          set heading 0
          set lane 5
          assign-correct-coordinates
        ]
      ]
    ]

  ;;-----------------------------------------------------------------
```

```
------------;;

     if ( lane = 4 )
    [
    if ( change-lane = 1 )
    [
     if ( pxcor = 1 and pycor = -1 )
    [
       set heading -90
       set lane 1
       assign-correct-coordinates
      ]
    ]

      if ( change-lane = 2 )
      [
        if ( pxcor = 2 and pycor = 2 )
        [
          set heading 90
          set lane 2
          assign-correct-coordinates
        ]
      ]
    ]

;;-----------------------------------------------------------------
----------;;

   if ( lane = 7 )
   [
    if ( change-lane = 2 )
    [
     if ( pxcor = -2 and pycor = -2 )
      [
       set heading -90
       set lane 1
       assign-correct-coordinates
      ]
    ]
      if ( change-lane = 1 )
      [
        if ( pxcor = -1 and pycor = 1 )
        [
          set heading 90
          set lane 2
          assign-correct-coordinates
        ]
      ]
    ]

end

to assign-correct-coordinates
```

```
if ( lane = 1)
[
  if ( change-lane != 1 )
  [
    if ( ycor < -2 or (ycor < 0 and ycor > -2 ) )
    [
      set ycor -2
    ]
  ]
  if ( change-lane = 1)
  [
    if ( ycor != -1 )
      [
       set ycor -1
      ]
  ]
]
 if (lane = 2)
 [
   if ( change-lane != 1)
   [
       if ( ycor > 2 or ( ycor > 0 and ycor < 2 ))
       [
          set ycor 2
        ]
   ]
   if ( change-lane = 1 )
   [
     if ( ycor != 1)
     [
       set ycor 1
     ]
   ]
 ]

if (lane = 6)
[
  if ( change-lane != 1)
  [
      if ( xcor > 2 or (xcor > 0 and xcor < 2))
      [
        set xcor 2
      ]
  ]
  if ( change-lane = 1 )
  [
   if ( xcor != 1)
   [
     set xcor 1
   ]
  ]
]

if (lane = 5)
```

```
  [
    if ( change-lane != 1)
    [
        if ( xcor < -2  or ( xcor < 0 and xcor > -2))
        [
          set xcor -2
        ]
    ]
    if ( change-lane = 1 )
    [
      if ( xcor != -1 )
      [
        set xcor -1
      ]
    ]
  ]

end

;;------------------------------------------------Procedure 
responsible for stopping cars
;;                                            cars infront of red 
light. When a car is stopped
;;                                         at a red signal, all the 
other cars stops right behind the stopped vehicle---------------;;

to separate-cars2


  if ( lane = 0)
  [
    if ( xcor >= min-pxcor and ( ycor > 0 and ycor <= 2 ))
    [
      separate-cars
      ifelse ( change-lane = 1 ) [ temp-procedure ]
      [stop-at-the-red-light]
    ]
  ]

  if ( lane = 4 )
  [
    if ( ycor <= max-pycor and (xcor > 0 and xcor <= 2) )
    [
      separate-cars
       ifelse ( change-lane = 1 ) [ temp-procedure ]
      [stop-at-the-red-light]
    ]
  ]

  if ( lane = 3 )
  [
    if ( xcor <= max-pxcor and (ycor >= -2 and ycor < 0))
    [
      separate-cars
```

```
       ifelse ( change-lane = 1 ) [ temp-procedure ]
      [stop-at-the-red-light]
    ]
  ]

  if ( lane = 7 )
  [
    if( (xcor < 0 and xcor >= -2) and ycor >= min-pycor)
    [
      separate-cars
       ifelse ( change-lane = 1 ) [ temp-procedure ]
      [stop-at-the-red-light]
    ]
  ]

end

;;-------------------------Procedure responsible for stopping
vehicles in front of red light----------------------------;;

to stop-at-the-red-light
  let t-ahead one-of cars-on patch-ahead 1
  if ( t-ahead = nobody and [pcolor] of patch-ahead 1 != red and
[pcolor] of patch-ahead 1 != orange )
  [
    fd 1
    slow-cars-at-the-red-light
    if ( (tempcounter mod 5) = 0 )
    [
      stop-at-the-red-light
    ]
  ]

end


to temp-procedure

  let t-ahead one-of cars-on patch-ahead 1
  if ( t-ahead = nobody )
  [
    if (lane = 0 )
    [
      if ( ([pcolor] of patches with [pxcor = -3 and pycor = 2] !=
[15] and
        [pcolor] of patches with [pxcor = -3 and pycor = 2] != [25])
or
        [pxcor] of patch-ahead 1 != -3  )
      [
       fd 1
       slow-cars-at-the-red-light1
       if ( (tempcounter1 mod 5) = 0 )
       [
         temp-procedure
        ]
      ]
    ]

     if (lane = 4 )
    [
      if ( ([pcolor] of patches with [ pxcor = 2 and pycor = 3] !=
[15] and
        [pcolor] of patches with [ pxcor = 2 and pycor = 3] !=
[25] )or
        [pycor] of patch-ahead 1 != 3 )
      [
       fd 1
       slow-cars-at-the-red-light1
       if ( (tempcounter1 mod 5) = 0 )
       [
         temp-procedure
       ]
      ]
    ]

     if (lane = 3 )
    [
      if ( ([pcolor] of patches with [pxcor = 3 and pycor = -2] !=
[15] and
        [pcolor] of patches with [pxcor = 3 and pycor = -2] != [25])
or
        [pxcor] of patch-ahead 1 != 3 )
      [
       fd 1
       slow-cars-at-the-red-light1
       if ( (tempcounter1 mod 5) = 0 )
       [
         temp-procedure
       ]
      ]
    ]

     if (lane = 7 )
    [
      if ( ([pcolor] of patches with [pxcor = -2 and pycor = -3] !=
[15] and
        [pcolor] of patches with [pxcor = -2 and pycor = -3] !=
[25]) or
        [pycor] of patch-ahead 1 != -3 )
      [
       fd 1
       slow-cars-at-the-red-light1
       if ( (tempcounter1 mod 5) = 0 )
       [
         temp-procedure
       ]
      ]
    ]
```

```
  ]
end



to slow-cars-at-the-red-light
  set tempcounter tempcounter + 1
end

to slow-cars-at-the-red-light1
   set tempcounter1 tempcounter1 + 1
end

;;--------------------------------------Periodically changes the
change-lane property of vehicles into lanes
;;                                    which is responsible for their
turning into a particular lane-----------------------------;;

to periodic-change-of-lanes-in-vehicles
   if ( ( ticks mod 300) = 0 )
   [
     ask cars
     [
       set change-lane random 3
     ]
   ]
end

;;------------------------------------Procedure responsible for
simulation of emergency vehicles and also
;;                                                  initializes
their parameters-----------------------------;;
to Emergency-Vehicles

 ifelse (not Is-ambulance-already-created? )
 [
  set ecounter 0
  set emergency-dead 0
  set Is-ambulance-already-created? true
   create-ambulances 1
   [
     set size 2
     set decision-speed 1.2
     ;set lane random 8
     if ( North-west = "green" ) [ set lane 0 ]
     if ( North-east = "green")  [ set lane 4 ]
     if ( South-west = "green" ) [ set lane 7 ]
     if ( South-east = "green" ) [ set lane 3 ]
     set shape "arrow"
     set change-lane random 3
     set speed 1.2
     set color blue
     if ( lane = 0 )
       [
         setxy min-pxcor 1
         set heading 90
       ]

       if ( lane = 3 )
       [
         setxy max-pxcor -1
         set heading -90
        ]

       if ( lane = 4 )
       [
          setxy 1 max-pycor
          set heading 180
       ]

       if ( lane = 7 )
        [
          setxy -1 min-pycor
          set heading 0
        ]


        assign-actual-speed-to-ambulances
        set emergency-lane lane
        stop-vehicles-during-emergency

    ]
  ]
  [
    user-message ( word " ambulance already created")
  ]

end

;;---------------------Repeatedly calls other functions which are
responsible for movement of emergency vehicles----------;;

to go-emergency

 if ( ticks = number-of-ticks)
 [
  stop
]

ask ambulances
[
 set hidden? false
 ]
  Emergency-vehicles-changing-lanes
forward-the-emergency-vehicle
```

```
end

;;-----------------------Procedure responsible for changing lanes by
emergency vehicles----------------;;

to Emergency-vehicles-changing-lanes

    ask ambulances
    [
    if ( lane = 0 )
    [
   if ( change-lane = 0 or change-lane = 3 )
   [
     if ( pxcor = -3 and pycor = 1 )
     [
     set ecounter 1
     set lane 2

     ]
   ]
    if (change-lane = 1 )
    [
        if ( pxcor = 1 and pycor = 1 )
        [

          set heading 180
          set lane 6
          set ecounter 1
        ]
    ]
    if ( change-lane = 2 )
    [
      if ( pxcor = -1 and pycor = 1 )
      [
        set heading 0
        set lane 5
        set ecounter 1
      ]
    ]
   ]
  ;;----------------------------------------------------------------
------------;;
  if ( lane = 3 )
  [
     if ( change-lane = 0 or change-lane = 3 )
   [
     if ( pxcor = 3 and pycor = -1 )
     [
     set ecounter 1
     set lane 1

     ]
   ]
```

```
    if ( change-lane = 2 )
    [
     if ( pxcor = 1 and pycor = -1 )
      [
        set heading 180
        set lane 6
        set ecounter 1
      ]
    ]
      if ( change-lane = 1 )
      [
       if ( pxcor = -2 and pycor = -1 )
        [
          set heading 0
          set lane 5
          set ecounter 1
        ]
      ]
    ]

  ;;-----------------------------------------------------------------
------------;;

     if ( lane = 4 )
  [
    if ( change-lane = 0 or change-lane = 3 )
   [
     if ( pxcor = 1 and pycor = -2 )
     [
     set ecounter 1
     set lane 6

     ]
   ]

    if ( change-lane = 1 )
    [
     if ( pxcor = 1 and pycor = -2 )
      [
        set heading -90
        set lane 1
        set ecounter 1
      ]
    ]
      if ( change-lane = 2 )
      [
        if ( pxcor = 1 and pycor = 1 )
        [
          set heading 90
          set lane 2
          set ecounter 1
        ]
      ]
    ]
```

```
;;-------------------------------------------------------------------
----------;;
   if ( lane = 7 )
  [
   if ( change-lane = 0 or change-lane = 3 )
   [
     if ( pxcor = -1 and pycor = -3 )
     [
     set ecounter 1
     set lane 5

     ]
   ]
    if ( change-lane = 2 )
    [
     if ( pxcor = -1 and pycor = -1 )
      [
        set heading -90
        set lane 1
        set ecounter 1

      ]
    ]
      if ( change-lane = 1 )
      [
        if ( pxcor = -1 and pycor = 2 )
        [
          set heading 90
          set lane 2
          set ecounter 1
        ]
      ]
    ]
    ]

end

;;---------------------------------------Procedure correlates the
parameters of emergency vehicles
;;                                       in the NetLogo world to the
values of the real world-------------------------;;

to assign-actual-speed-to-ambulances

   if ( speed >= 0.1 and speed <= 0.5 ) [ set ambulance-actual-speed
20 ]
  if ( speed > 0.5 and speed <= 0.8 )   [ set ambulance-actual-speed
45 ]
  if ( speed > 0.8 and speed <= 1.3 )   [ set ambulance-actual-speed
70 ]
  set ambulance-actual-acceleration 4
  set ambulance-actual-deceleration 8
end
```

```
;;---------------------------Accelerates and Decelerates the
emergency vehicles------------------------------;;

to accelerate-the-ambulance
  set speed speed + acceleration
  end


to decelerate-the-ambulance
  set speed [speed] of front-vehicle - deceleration
 end

;;--------------------------------------------Procedure responsible
for the forward movement of cars
;;                                        and also call other
procedures responsible for behaviour of emergency vehicles--------;;

 to forward-the-emergency-vehicle

   ask ambulances
   [
     if ( ecounter = 1 )
     [
       if ( xcor >= max-pxcor or xcor <= min-pxcor or ycor >= max-
pycor or ycor <= min-pycor )
       [
         set emergency-dead 1
         set Is-ambulance-already-created? false
         die
       ]
     ]
     if (( xcor < max-pxcor or xcor > min-pxcor or ycor < max-pycor
or ycor > min-pycor ))
       [
          set front-vehicle one-of cars-on patch-ahead 1
           ifelse ( front-vehicle = nobody )
           [ accelerate-the-ambulance ]
           [ decelerate-the-ambulance ]
           fd speed

           ambulance-decision-during-red-light
           test-sound
       ]
     ]
   if ( emergency-dead = 1 )
   [
    resume-all-the-vehicles-after-emergency-vehicle-moved-out
   ]
 end

;;-------------------------Procedure responsible for sound evolving
from emergency vehicles-------------------------;;
```

```
to  test-sound
  ;sound:play-sound "/Users/jay/Downloads/Ambulance.wav"
end

;;----------------------------------- Changes lane property of
emergency vehicles
;;                                  when it moved from one lane to
another different lane altogether-----------------;;

to change-lanes-for-emergency-vehicles

   if ( [intersection?] of patch-here )
  [
  if ( lane = 0  )
  [
    if ( change-lane = 2 ) [ set lane 5]
    if ( change-lane = 0 or change-lane = 3 ) [ set lane 2 ]
    if ( change-lane = 1 ) [ set lane 6 ]
  ]

  if ( lane = 4  )
  [
   if ( change-lane = 0 or change-lane = 3 ) [ set lane 6 ]
   if ( change-lane = 2 ) [ set lane 2 ]
   if ( change-lane = 1 ) [ set lane 1 ]
  ]

  if ( lane = 3 )
  [
   if ( change-lane = 0 or change-lane = 3 ) [ set lane 1 ]
   if ( change-lane = 2 ) [ set lane 6 ]
   if ( change-lane = 1 ) [ set lane 5 ]
  ]

  if ( lane = 7 )
  [
   if ( change-lane = 0 or change-lane = 3 ) [ set lane 5 ]
   if ( change-lane = 2 ) [ set lane 1 ]
   if ( change-lane = 1 ) [ set lane 2 ]
  ]
  ]

end

;;-----------------------------------Depending upon the lane into
which an emergency vehicle should turn, its behaviour at the red
light will differ. For example,
;;                if emergency vehicle is on lane 0 and it is red
light and if the vehicle should have to turn into lane 5, it will
turn even when
;;                it is red light. But if it has to turn into lane 2
or lane 6 it will stop at red light, as there will be other vehicles
moving along
;;                those
```

```
lanes-------------------------------------------------------------------
------------;;

to ambulance-decision-during-red-light

  if ( lane = 0 )
  [
     if ( [pcolor] of patches with [ pxcor = -3 and pycor = 2 ] =
[15] and ( change-lane != 2 )
                and [pxcor] of patch-ahead 1 = -3)
       [
         set speed 0
       ]
       if ( [pcolor] of patches with [ pxcor = -3 and pycor = 2 ] =
[55]
                and ([pxcor] of patch-here = -4 or [pxcor] of patch-
here > -3 )and speed = 0)
       [
         set  speed decision-speed
       ]
   ]

  if ( lane = 4 )
  [
    if ( [pcolor] of patches with [ pxcor = 2 and pycor = 3] =
[ 15 ] and ( change-lane != 2 )
         and [pycor] of patch-ahead 1 = 3 )
      [
        set speed 0
      ]
      if ( [pcolor] of patches with [ pxcor = 2 and pycor = 3] =
[ 55 ]
         and ([pycor] of patch-here = 4 or [pycor] of patch-here <
3 ) and speed = 0 )
      [
        set  speed decision-speed
      ]
  ]
  if ( lane = 3 )
  [
    if ( [pcolor] of patches with [ pxcor = 3 and pycor = -2 ] =
[15] and ( change-lane != 2 )
         and [pxcor] of patch-ahead 1 = 3)
      [
        set speed 0
      ]

      if ( [pcolor] of patches with [ pxcor = 3 and pycor = -2 ] =
[55]
        and ([pxcor] of patch-here = 4 or [pxcor] of patch-here <
3 )and speed = 0)
      [
        set  speed decision-speed
      ]
  ]

  if ( lane = 7 )
  [
    if ( [pcolor] of patches with [ pxcor = -2 and pycor = -3 ] =
[15] and (change-lane != 2 )
       and [pycor] of patch-ahead 1 = -3)
    [
      set speed 0
    ]

    if ( [pcolor] of patches with [ pxcor = -2 and pycor = -3 ] =
[55]
       and ([pycor] of patch-here = -4 or [pycor] of patch-here >
-3 ) and speed = 0)
    [
      set  speed decision-speed
    ]

  ]

end

;;----------------------Responsible for making car move onto a lane
from intersection during emergency vehicle------------;;

to move-on-to-lane-during-emergency-vehicle

  ask cars
    [
      if ( [intersection?] of patch-here )
      [
        fd 1
        move-on-to-lane-during-emergency-vehicle
      ]
    ]
end

;;-------------------Stops the vehicles when there is an emergency
vehicle in the world--------------------------;;


to stop-vehicles-during-emergency

    set emergency-north-east North-east
    set emergency-north-west North-West
    set emergency-south-east South-east
    set emergency-south-west South-west

  if ( emergency-lane = 0 )
  [
    ask patches with [ pxcor = 3 and pycor = -2] [ set pcolor red]
    ask patches with [ pxcor = 2 and pycor = 3]  [ set pcolor red]
    ask patches with [ pxcor = -2 and pycor = -3][ set pcolor red]
    set North-east "red"
    set South-east "red"
    set South-west "red"
  ]
  if ( emergency-lane = 4 )
  [
    ask patches with [ pxcor = -3 and pycor = 2] [ set pcolor red]
    ask patches with [ pxcor = 3 and pycor = -2] [ set pcolor red]
    ask patches with [ pxcor = -2 and pycor = -3][ set pcolor red]
    set North-west "red"
    set South-east "red"
    set South-west "red"
  ]
  if ( emergency-lane = 3 )
  [
    ask patches with [ pxcor = -3 and pycor = 2] [ set pcolor red]
    ask patches with [ pxcor = 2 and pycor = 3]  [ set pcolor red]
    ask patches with [ pxcor = -2 and pycor = -3][ set pcolor red]
    set North-east "red"
    set North-west "red"
    set South-west "red"
  ]
  if ( emergency-lane = 7 )
  [
    ask patches with [ pxcor = -3 and pycor = 2] [ set pcolor red]
    ask patches with [ pxcor = 3 and pycor = -2] [ set pcolor red]
    ask patches with [ pxcor = 2 and pycor = 3]  [ set pcolor red]
    set North-east "red"
    set North-west "red"
    set South-east "red"
  ]

  move-on-to-lane-during-emergency-vehicle

  ask cars with [ lane = 5 or lane = 1 or lane = 6 ]
  [ forward-cars-as-per-speed ]
end

;;---------------------Resumes all other cars when emergency vehicle
moved out (In case if all cars are stopped)---------;;

to resume-all-the-vehicles-after-emergency-vehicle-moved-out

  if ( (North-west = "red" or North-west = 0) and
       (North-east = "red"  or North-east = 0) and
       (South-east = "red"  or South-east = 0) and
       (South-west = "red"  or South-west = 0))
  [
    set North-east emergency-north-east
    set North-West emergency-north-west
    set South-east emergency-south-east
    set South-west emergency-south-west

    if ( North-west = "green" )
```

```
      [
       ask patch -3 2 [ set pcolor green]
      ]

      if ( North-east = "green" )
      [
        ask patch 2 3  [ set pcolor green]
      ]

      if ( South-east = "green")
      [
         ask patch 3 -2  [ set pcolor green]
      ]

      if ( South-west = "green" )
      [
       ask patch -2 -3  [ set pcolor green]
      ]
   ]
end

;;-----------------------Procedure responsible for assigning rash
behaviour to the rash car--------------------;;

to rash-driver-behaviour

  ask cars with [ rash-car? = true ]
  [
     if ( front-car != nobody and front-car != 0)
    [
     if ( lane = 0 or lane = 2)
     [
      set xcor xcor + 1
      set ycor  1
    ]

    if ( lane = 1 or lane = 3)
    [
      set xcor xcor - 1
      set ycor  -1
    ]

    if ( lane = 4 or lane = 6)
    [
      set xcor 1
      set ycor ycor - 1
    ]

    if ( lane = 5 or lane = 7)
    [
      set xcor -1
      set ycor ycor + 1
    ]
  ]
  if ( front-car = nobody )
  [
  if ( (lane = 0 or lane = 2 ) and ycor = 1)
  [
    if ( not any? cars-at (xcor + 2) 2 )
    [
      set xcor xcor + 2
      set ycor 2
    ]
  ]


  if ( (lane = 3 or lane = 1 ) and ycor = -1 )
  [
    if ( not any? cars-at ( xcor - 2) -2 )
     [
        set xcor xcor - 2
        set ycor -2
     ]
  ]

  if ( (lane = 4 or lane = 6 ) and xcor = 1 )
  [
    if ( not any? cars-at 2 (ycor - 2) )
      [
        set xcor 2
        set ycor ycor - 2
      ]
    ]

  if ( (lane = 5 or lane = 7 ) and xcor = -1 )
  [
    if ( not any? cars-at -2 (ycor + 2) )
      [
        set xcor -2
        set ycor ycor + 2
      ]
  ]
  ]

  if ( lane = 4 and ([pycor] of patch-here <= 3  or [intersection?]
of patch-here ) and (North-east = "red" or North-east = 0 ))
      [
         die
      ]
      if ( lane = 0 and ([pxcor] of patch-here >= -3 or
[intersection?] of patch-here ) and (North-west = "red" or North-
west = 0))
      [
         die
      ]

      if ( lane = 3 and ([pxcor] of patch-here <= 3 or
[intersection?] of patch-here ) and (South-east = "red" or South-
east = 0) )
      [
        die
      ]

      if ( lane = 7 and ([pycor] of patch-here >= -3 or
[intersection?] of patch-here ) and (South-west = "red" or South-
west = 0 ) )
      [
         die
      ]
]

end


to cars-new-behaviour

  ask cars
  [
     if ( front-car != nobody and front-car != 0)
    [
      if ( change-lane = 1 )
      [
       if ( lane = 0 or lane = 2)
       [
         set xcor xcor + 1
         set ycor  1
       ]

      if ( lane = 1 or lane = 3)
      [
       set xcor xcor - 1
       set ycor  -1
      ]

      if ( lane = 4 or lane = 6)
      [
       set xcor 1
       set ycor ycor - 1
      ]

      if ( lane = 5 or lane = 7)
      [
       set xcor -1
       set ycor ycor + 1
     ]
    ]
   ]
  ]

end
```