ReadMe

Emmanuel Obafemi Buraimo

April 22, 2020

ReadMe

The read me file is very similar to the appendix of the report, it details all that needs to be done to run and view the simulation. This read me file can be ignored if the appendix is read. The main difference between the appendi and the read me file is the exclusion of reused code

User Guide

The steps taken to install and use the simulation are detailed below.

- Download and Setup: The first step is to download the NetLogo application. Various versions of the application are found here https://ccl.northwestern.edu/netlogo/download.shtml Since the model was developed in version 6.0.4, this should be the version preferably downloaded, although the newer version available would also be able to run the program but will be prompted that this model was made on an older model. If the user is using a Linux computer like that found in the labs at Bush House then this step can be skipped.

- Opening Netlogo: After successfully downloading the model, the user should open the program, depending on the operating system this will involve you opening the Netlogo download folder and then opening the application. This is how it would display on a Mac OS

- Opening the Model: After NetLogo has been successfully downloaded and installed and opened the next step is to open the model, after submitted the user should download the .nlogo file from the Keats page where it was uploaded, navigate to file from NetLogo, click open and then search and open the file.

- Running the simulation: After successfully opening the NetLogo platform and the simulation, the program should be opened on a full screen to make space for the graphs. After that the number of cars the user wants to run should be selected first with the slider then the autonomy level, after this is done the "setup" button should be clicked to display all the agents and after the "go" button should be clicked first, this starts the simulation and continues it until one of the 3 end statements is reached as to run infinitely or for too long an amount of time. The speed of the simulation could also be adjusted but making the simulation too fast will not let the user observe anything happening and only see the results.

- Collecting data: While the simulation runs it continues till one of the three end cases are reached, this is based on the amount of cars and the autonomy level as a simulation with a high amount of cars will end much faster than a simulation with a medium amount of cars, and a simulation with a low amount of cars could also end quickly if the law amount of law breaking cars is too low to extract any information from.

Source Code

The source code will be divided into two sections, the source code written by me and the source code either reused or influenced by users on different platforms.

Code Written by the Author

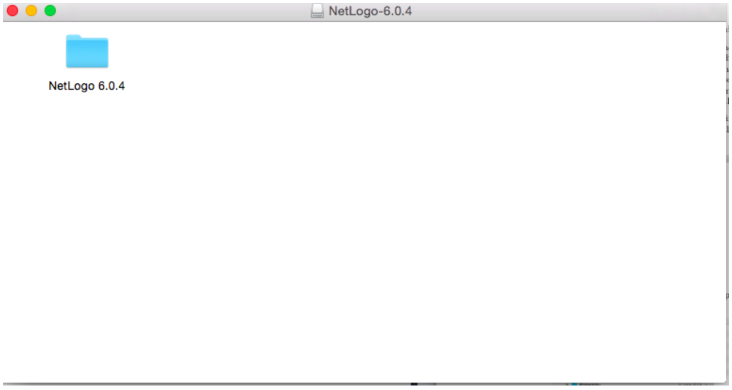
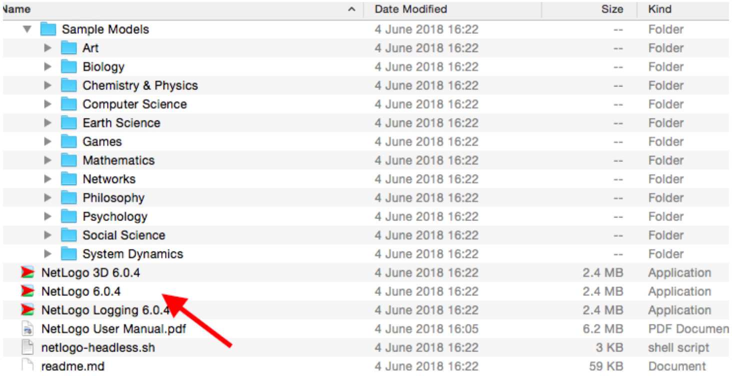


Figure 1: Setup Folder



Name	Date Modified	Size	Kind
▼ Sample Models	4 June 2018 16:22	--	Folder
▶ Art	4 June 2018 16:22	--	Folder
▶ Biology	4 June 2018 16:22	--	Folder
▶ Chemistry & Physics	4 June 2018 16:22	--	Folder
▶ Computer Science	4 June 2018 16:22	--	Folder
▶ Earth Science	4 June 2018 16:22	--	Folder
▶ Games	4 June 2018 16:22	--	Folder
▶ Mathematics	4 June 2018 16:22	--	Folder
▶ Networks	4 June 2018 16:22	--	Folder
▶ Philosophy	4 June 2018 16:22	--	Folder
▶ Psychology	4 June 2018 16:22	--	Folder
▶ Social Science	4 June 2018 16:22	--	Folder
▶ System Dynamics	4 June 2018 16:22	--	Folder
NetLogo 3D 6.0.4	4 June 2018 16:22	2.4 MB	Application
NetLogo 6.0.4	4 June 2018 16:22	2.4 MB	Application
NetLogo Logging 6.0.4	4 June 2018 16:22	2.4 MB	Application
NetLogo User Manual.pdf	4 June 2018 16:05	6.2 MB	PDF Documen
netlogo-headless.sh	4 June 2018 16:22	3 KB	shell script
readme.md	4 June 2018 16:22	59 KB	Document

Figure 2: Application

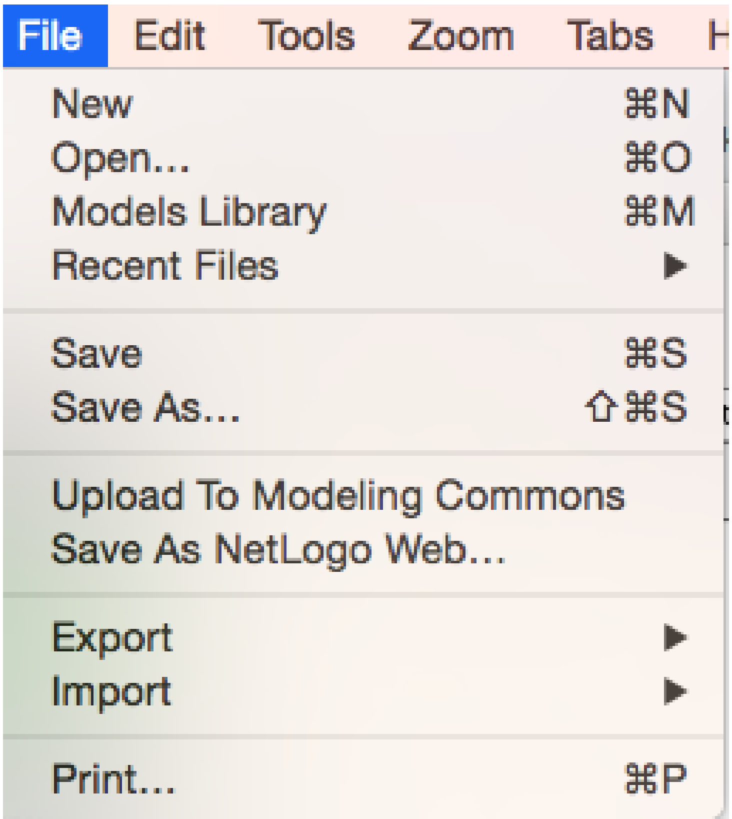


Figure 3: Navigation

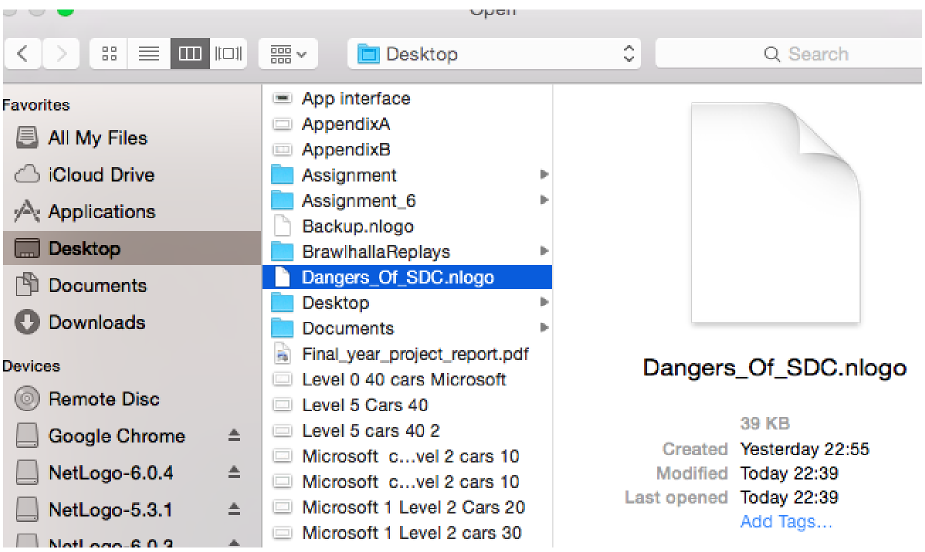
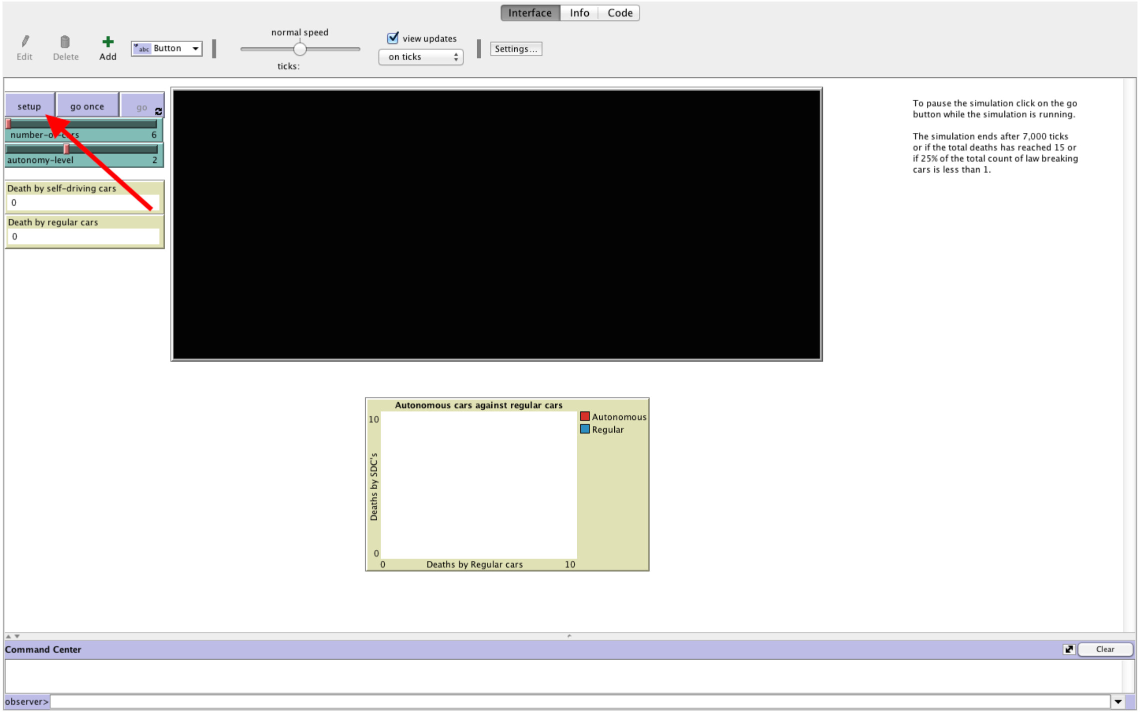


Figure 4: Opening the file



number-of-cars 6

autonomy-level 2

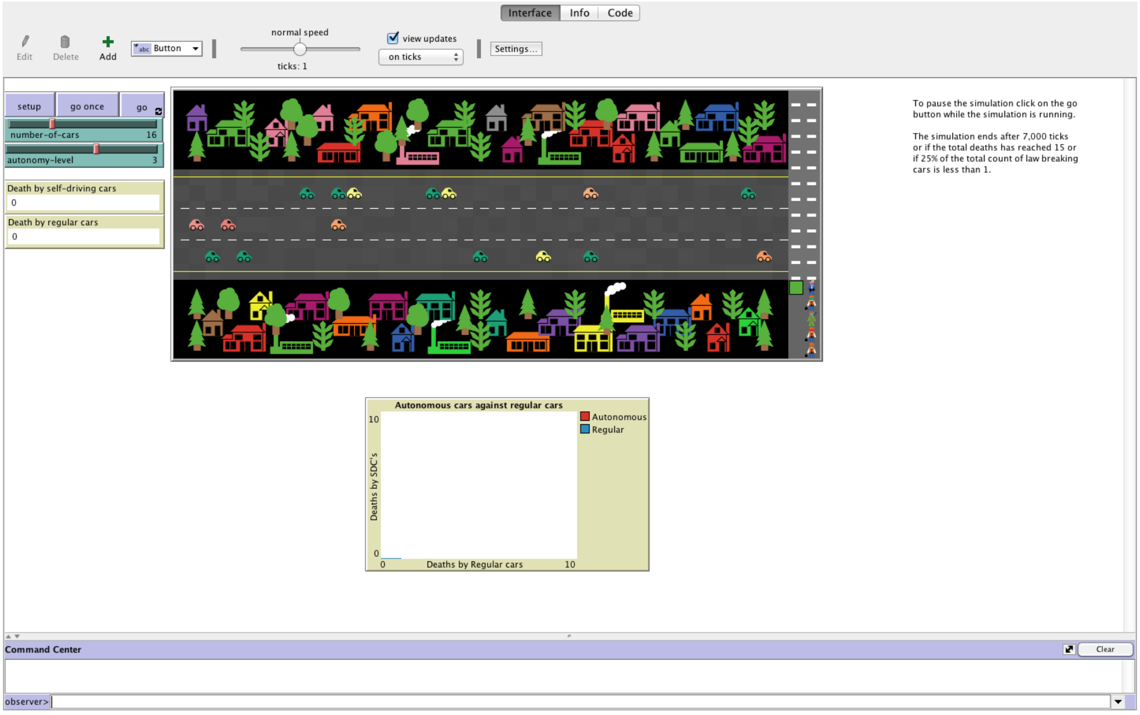
Death by self-driving cars 0

Death by regular cars 0

To pause the simulation click on the go button while the simulation is running.

The simulation ends after 7,000 ticks or if the total deaths has reached 15 or if 25% of the total count of law breaking cars is less than 1.

Figure 5: Setup



number-of-cars 16

autonomy-level 3

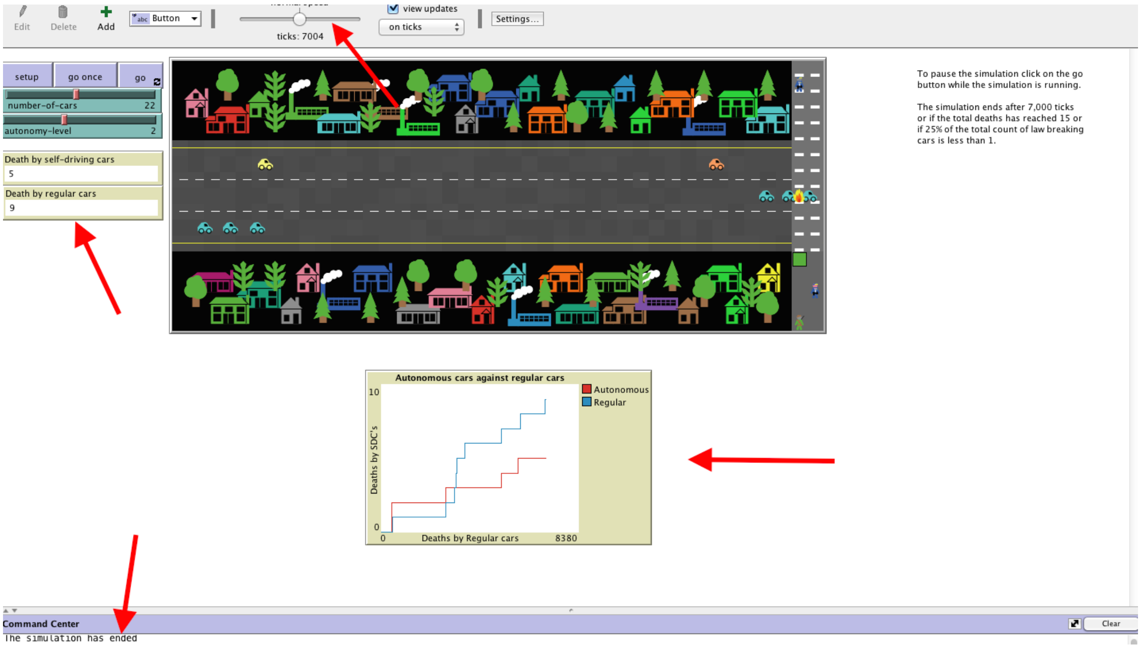
Death by self-driving cars 0

Death by regular cars 0

To pause the simulation click on the go button while the simulation is running.

The simulation ends after 7,000 ticks or if the total deaths has reached 15 or if 25% of the total count of law breaking cars is less than 1.

Figure 6: Go



ticks: 7004

number-of-cars 22

autonomy-level 2

Death by self-driving cars
5

Death by regular cars
9

To pause the simulation click on the go button while the simulation is running.

The simulation ends after 7,000 ticks or if the total deaths has reached 15 or if 25% of the total count of law breaking cars is less than 1.

```
The simulation has ended
```

Figure 7: Simulation

```
globals [
lanes ;Reports the amount of lanes
y-values ;These are the y cooridantes of the people agents
autonomous-death-toll ;The total amount of deaths recorded by
    autonomous cars.
regular-death-toll ;The total amount of deaths recorded by regular
    cars.
totalDeaths ;The sum of the autonomous death toll and the regular
    death toll
]

;These respective breed commands creates an set of all the breeds used
    in the simulation, an agentset can be seen as a collection of
    similar items and each item is known as the breed name.
;For example every item in the trees breed agentset is a tree.
breed [people person]
breed [cars car]
breed[houses house] ;
breed[trees tree]
breed[lights light]
breed[fires fire]
cars-own [
speed ; the current speed of the car
top-speed ; the maximum speed of the car
target-lane ; the desired lane of the car
patience ; the cars current level of patience
current-autonomy-level ; tracks the cars current level of autonomy
autonomous? ; Boolean variable to check if the car is autonomous or not
Law-Abiding? ; Checks if the car stops at a red light
max-patience ;The max patience for the cars, this is 100.
acceleration ;The rate at which cars accelerate
deceleration ;The rate at which cars decelerate
]


;These are variables specific to the lights breeds are placed here.
lights-own

[
time-passed ;Checks the amount of time elapsed as lights change
    periodically.
colors-list ;The list of colors the light agent can be, red, yellow
    and green.
]

;Variables specific to the fire agent are placed here.
fires-own

[
ticksPast ;This keeps track of the amount of ticks past, the fire
    agent dies after a certain amount of ticks.
]


;The setup procedure is called when the setup button is clicked.
;The setup procedure set the shapes of all the agents, the shape of
    the cars lights and fires are fixed but the shape of the people
    trees and houses can be any one of the options listed.
to setup
clear-all
set-default-shape cars "car"
set-default-shape people one-of ["person␣police" "person␣police"
    "person␣soldier" "person␣construction"]
set-default-shape houses one-of [ "house␣bungalow" "house␣ranch"
    "house␣colonial" "house␣efficiency" "house␣two␣story" "factory"]
set-default-shape trees one-of ["tree" "tree␣pine" "plant"]
set-default-shape lights "lights"
set-default-shape fires "fire"
add-trees-and-houses ;This procedure adds trees and houses to the left
    and right areas of the simulation
addLights ; This procedure places the light agent on the road
linesAcross ;This procedure draws lines across the road
set-autonomy-level ;This procedure sets the current autonomy level of
    the car

;The ask command can be used for agentsets or agents, in this case it
    asks or rather sets the max patience, acceleration and
    deceleration of the cars.
;It also sets cars to be intially all be law abiding, and then asks 2
    cars from both autonomous and non-autonomous cars to not be law
    abiding.
;After these 4 cars die, 25% of cars available will be randomly picked
    to not be law abiding
ask cars
[
set max-patience 100
set acceleration 0.0020
set deceleration 0.02]
```

```
ask cars [ set Law-Abiding? true ]
ask n-of 2 cars with [autonomous? = true][set Law-Abiding? 0 ]
ask n-of 2 cars with [autonomous? = 0][set Law-Abiding? 0 ]

reset-ticks ;resets the ticks and start.
tick
end

;The go proceudre keeps running until an end condition is reached. It
    also controls the flow of color change of the traffic lights and
    the movements of the cars and people.
to go
control-traffic-lights
set-patience
ask fires[ if ticks - ticksPast > 60 [die]]

obeyLights ;This procedure tells cars to stop at the traffic light
    when its red
speedOnRed ;This sets the speed of the randomly selected non law
    abiding cars.
walk-forward ;This procedure tells the people to move forward in an
    upward direction
continue ;This procedure tells the people to move forward in a
    downward direction
driveSafe ;This procedure regulates the speed of the cars in normal
    motion
record-accidents ;This proceudre keeps tracks of the accidents that
    have occured
update-cars ;This sets 25% of the cars available to not be law-abiding.


  ;The end condition, if the total death toll is equal to 15 or ticks
      goes past 7000 or there's only 1 car available after the inital
      4 to be non law abiding the simulation stops.

  set totalDeaths (autonomous-death-toll  + regular-death-toll)

 if (totalDeaths = 15) or (ticks > 7000) or (count cars * 0.25 < 1)

 [print "The␣simulation␣has␣ended" stop ]

tick
end

to set-patience

  ;This procedure sets the patience of the autonomous cars.

 ask cars with [autonomous? = true]
 [
   if autonomy-level = 0 [set current-autonomy-level 0 set patience
       (30 + random 15)]
   if autonomy-level = 1 [set current-autonomy-level 1 set patience
       (40 + random 15)]
   if autonomy-level = 2 [set current-autonomy-level 2 set patience
       (40 + random 15)]
   if autonomy-level = 3 [set current-autonomy-level 3 set patience
       (45 + random 15)]
    if autonomy-level = 4 [set current-autonomy-level 4 set patience
        (65 + random 15)]
    if autonomy-level = 5 [set current-autonomy-level 5 set patience
        (85 + random 15)]


 ]
end

to add-trees-and-houses

ask patches with [pcolor != 19] [
    if count neighbors with [pcolor = black] = 8 and not any? turtles
        in-radius 2 [
      sprout-houses 1 [
        set shape one-of [ "house␣bungalow" "house␣ranch" "house␣
            colonial" "house␣efficiency" "house␣two␣story" "factory"]
        set size 3
        stamp
      ]
    ]
 ]
ask patches with [pcolor != 19] [
    if count neighbors with [pcolor = black] = 8 and not any? turtles
        in-radius 1[
        sprout-trees 1 [
          set shape one-of ["tree" "tree␣pine" "plant"]
          set size 2
          set color green
          stamp
        ]

    ]
 ]
end

;This procedure draws a footpath for the people
to draw-footpath
ask patches
[ if pxcor >= 19 ;; patches on the far right side
[ set pcolor grey - 1.5 ] ]

end
;This adds the traffic light to the simualtion.
to addLights
 ask patches with [(pycor = -4) and pxcor = 19] [
 sprout-lights 1 [
 set color green
 set shape "lights"
 set colors-list [red yellow green]
 set time-passed 0
 ]
 ]
end
to walk-forward
 ask one-of people
 [ if not any? (lights with [color = green or color = yellow]) and
     (pxcor = 20) or (pxcor = 20 and pycor >= 3)
```

```
   [ set heading 0
     forward 1

     rotate-left ]
    ]
end


to rotate-left

  ask one-of people
 [if (pxcor = 20 and pycor = 8)
   [lt 90 forward 1]
   ]
end

to continue
ask one-of people
[if not any? (lights with [color = green or color = yellow]) and
   (pxcor = 19)
   [set heading 0
     forward -1
     rotate-right
      ]
   ]
end
to rotate-right

 ask one-of people

  [if (pxcor = 19 and pycor = -8 )

   [ rt 90 forward 1]


 ]
end






;This procedure instructs cars to move when the light is green or
   yellow but stop when the light is red.
to obeyLights

 ask cars

 [ifelse not any? (lights) with [color = red] [move-forward] [set
    speed 0]

 ]


end
```

```
;This procedure gives the respective speeds of the non law abiding
   cars, this reduces as the autonomy level increases.
to speedOnRed



   ask cars with [Law-Abiding? = 0]

 [



   if (autonomous? =  0   and any?(lights) with [color = red]) [fd 1.2]
   if (autonomy-level = 0 and (autonomous? = True) and any?(lights)
      with [color = red]) [fd 1.2]
   if (autonomy-level = 1 and any?(lights) with [color = red]) [fd 0.8]
   if (autonomy-level = 2 and any?(lights) with [color = red]) [fd
      0.35]
   if (autonomy-level = 3 and any?(lights) with [color = red]) [fd 0.3]
   if (autonomy-level = 4 and any?(lights) with [color = red]) [fd 0.2]
   if (autonomy-level = 5 and any?(lights) with [color = red]) [fd 0.1]



   ]
end

  set autonomous-death-toll autonomous-death-toll + count people with
     [any? cars-here with [autonomous? = true]]
     set regular-death-toll regular-death-toll + count people with
        [any? cars-here with [autonomous? = 0]]


     sprout-fires 1[
       set shape "fire"
       set size 1
       set ticksPast ticks
     ]

     ask people-here
     [die]

     ask cars-here
     [die]


 ]

   if count people = 0
   [create-pedestrians]


 ]


end
```

```
;This procedure updates randomly picks cars whenever there is only 1 law abiding car available to 25% of the total count of cars.

to update-cars

 ask cars
 [
  if count cars with [Law-Abiding? = 0] = 1

  [ ask n-of(count cars * 0.25) cars

   [set Law-Abiding? 0]

 ]]


end

;This procedure controls the speed of the cars as they drive and are in contact with people. This is improved as the autonomy level increases.
 to driveSafe

 ask cars
[

    if(any? people in-cone 1 45) and any? (lights) with [color != red] and autonomous? = 0[


 set speed 0.008

   ]


   if(any? people in-cone 1 45) and any? (lights) with [color != red] and autonomy-level = 1 and autonomous? = true  [


 set speed 0.0009

   ]


    if(any? people in-cone 2 45) and any? (lights) with [color != red] and autonomy-level = 2 and autonomous? = true[


      set speed 0.00007

   ]

  if(any? people in-cone 2 45)or (any? people in-cone 2 45) and any? (lights) with [color != red] and autonomy-level = 3 and autonomous? = true[

   set speed 0.00008

   ]

   if(any? people in-cone 2 45) or (any? people in-cone 2 45) and any?
      (lights) with [color != red] and autonomy-level = 4 and
      autonomous? = true[


   set speed 0.000007

   ]

    if(any? people in-cone 2 45) or (any? people in-cone 2 45) and
       any? (lights) with [color != red] and autonomy-level = 5 and
       autonomous? = true[


   set speed 0.000007

   ]

 ]
end

to control-traffic-lights

ask lights
  [
  set time-passed time-passed + 1


   let temp 0
   if item 0 colors-list = green [set temp 105]
   if item 0 colors-list = yellow [set temp 15]
   if item 0 colors-list = red [set temp 75]

   if time-passed = temp
  [
     set time-passed 0
     set colors-list lput first colors-list colors-list
     set colors-list remove-item 0 colors-list
     set color first colors-list
   ]
 ]
 tick


end
```

Code written by Author

Just some notes, not visible in pdf.