

**CSCU9YM – The ODD Protocol**

Contents

**[1](#_Toc34051800)****[Introduction](#_Toc34051800)** [2](#_Toc34051800)

[**2** **Overview** 2](#_Toc34051801)

[**2.1** **Purpose** 2](#_Toc34051802)

[**2.2** **Entities, State Variables, and Scale** 2](#_Toc34051803)

[**2.3** **Process Overview and Scheduling** 4](#_Toc34051804)

[**3** **Design concepts** 4](#_Toc34051805)

[**3.1** **Basic Principles** 4](#_Toc34051806)

[**3.2** **Emergence** 5](#_Toc34051807)

[**3.3** **Adaptation** 5](#_Toc34051808)

[**3.4** **Objectives** 5](#_Toc34051809)

[**3.5** **Learning** 5](#_Toc34051810)

[**3.6** **Prediction** 6](#_Toc34051811)

[**3.7** **Sensing** 6](#_Toc34051812)

[**3.8** **Interaction** 6](#_Toc34051813)

[**3.9** **Stochasticity** 6](#_Toc34051814)

[**3.10** **Collectives** 6](#_Toc34051815)

[**3.11** **Observation** 6](#_Toc34051816)

[**4** **Details** 7](#_Toc34051817)

[**4.1** **Initialization** 7](#_Toc34051818)

[**4.2** **Input** 7](#_Toc34051819)

# **Introduction**

Traffic jams is an unpleasant experience in people's daily lives. Traffic congestion occurs when the cars stop to make the "Wave Phenomena" happen. A behavior that despite the progression of all cars, traffic jams tend to be reversed. While this is a real event in our daily lives, there are many other factors that may cause traffic jams and that should be taken into account, such as patience for drivers, which means how long they will wait in the queue before losing patience and change the lane that could cause further traffic jams. This model is an extension of the existing model “Traffic 2 Lanes” (Wilensky, U & Payette, N. 1998).

# **Overview**

## **Purpose**

The model shows how traffic jams can form and explores additional factors in order to test whether or not traffic jams occur most frequently. For instance, what happens if the model has additional lanes? What if the car breaks down and blocks the whole lane for quite some time? How will the patience of the driver affect the overall traffic if drivers change the paths constantly when the car is slower in front of them?

## **Entities, State Variables, and Scale**

The model is spatially clear, consisting of a grid of 41 by 17 patches (road) and inhabited by turtles (vehicles). The world is a rectangular grid of cylinders that only wraps horizontally. The model is explicitly temporary and lasts 1,000 time-steps. A traffic jam will occur according to lanes, number of cars, acceleration speed, speed declaration and patience parameters of the driver. Table T1 provides a list of parameters for turtles, patches and global variables.

|  |  |  |
| --- | --- | --- |
| **Table T1** Turtles, patch, and global variables | | |
| **Variable** | **Value** | **Description** |
| **Turtles (cars)** | | |
| X and Y coordinates | X coordinate: random x of a road lane  Y coordinate: random ymap coordinate (lane) | Every turtle has a random location on the grid which is a random x position of a random road lane. |
| Speed | 0 – 0.5 | Current vehicle speed |
| Top Speed | ≤ 0.5 | Maximum vehicle speed (different for each car) |
| Target Lane | Patch y coordinate | The car's lane of choice |
| Patience | 1 – 100 | The current level of patience of the driver. For each driver a random value of patience is applied Each time the car hits the breaks, the patience level drops, and the driver switches the lane as he goes 0. |
| Acceleration | 0.001 – 0.010 | This applies to the car’s speed as a gas when there is no other car in front of it.  “speed (speed + acceleration)” |
| Deceleration | 0.01 – 0.10 | This applies to the car's speed in order to avoid a collision when approaching a slower car. “speed (speed – deceleration)” |
| **Patches** | | |
| X and Y coordinates | X coordinate: -20 – 20  Y coordinate: -8 – 8 | Each patch is explicitly placed on the road. Patches cannot move. |
| Patch type | “environment”, “lanes” | This defines if the patch is an empty environment (grass, etc.) or a lane. |
| **Global** | | |
| Total number of cars | 1 – 328 | This describes the total number of cars in the model. |
| Lanes |  | This describes the y coordinates of different lanes |
| Selected Car |  | This is the location of the chosen car and it's an Observer Agent. It can be used to track or collect data for this specific car as a vehicle representative of the whole. |
| Broken Car |  | This variable is a random car that has been broken down and blocks the lane for a period of time. Another random car breaks down when it's "fixed" and begins to move again. |

|  |  |
| --- | --- |
| **Table T2** Default Parameters of the model | |
| **Parameter** | Value |
| Number of Lanes | 8 |
| Number of Cars | 70 |
| Acceleration | 0.005 |
| Deceleration | 0.02 |
| Max Patience | 30 |

## **Process Overview and Scheduling**

When the "setup" button is pressed, patches begin to generate (grass and road) with a variety of shades and then a temporary invisible turtle draws straight yellow and dashed white road lines using pen-down and pen-up commands. Cars are randomly placed on the accessible lanes, which do not come into conflict with other cars, faced in the right direction. On the "go" button, the cars start going from one patch to another in the right direction. At every step of the process, the car moves forward once the speed is gradually increased and checks whether another car is in front of it. The car will hit the brakes and stop if another vehicle blocks the road while the driver loses patience at the same time. The car will change lane above or beneath when the patience variable reaches zero (only if there are no cars above or beneath). The model's most interesting part is the broken car. When the model starts a randomly chosen car "breaks down" for a period of time, obviously blocking the road for the next cars, making them lose more patience and switch the lane. The following figure shows the annotated screenshot of the model (Fig 1).

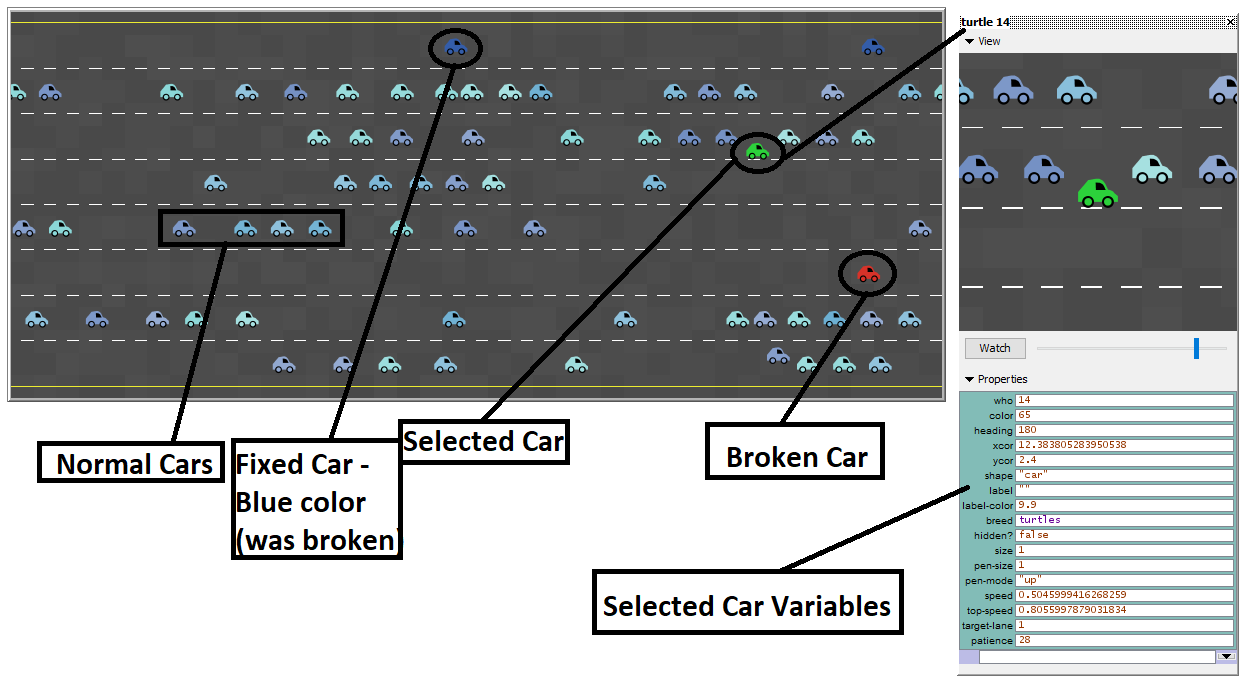


Figure 1: Annotated screenshot of the model after 700 time-steps

# **Design concepts**

## **Basic Principles**

The question and the hypothesis of the model is that the problem of traffic can be resolved by adding further roads? Or even more lanes added what happens if a different variable is added, a broken random car that every time blocks the entire lane.

The principles behind this model can be as follows:

* Cars are always moving to the right at increasing speed until some other car blocks its road (slower or broken car).
* If a traffic occurs, this does not necessarily mean that the car will switch the lane. Only when the level of patience drops to zero.
* If the patience level drops to zero, an alternative lane will be taken either above or below.
* All cars have a randomly pre-determined top speed and maximum level of patience.
* A random car must always break.

These principles have been incorporated into the model as the basis for the rules on the movement and reaction of cars.

## **Emergence**

The result of the model is the average speed of all cars and the patience of the chosen car during the run-time. With lower average speed and higher average patience, huge traffic jams occurred during the model run-time. The population and the number of lanes have a huge role to play, as the rules of movement will eventually force the cars to move at any cost (maybe stuck between two lanes due to high traffic and population), while other factors will play a significant role in shaping their behavior.

## **Adaptation**

For this section, the "Selected Car" will represent the entire car population, let's call it Turtle-1. Turtle-1 will spawn randomly on any x coordinate of a random lane like any other turtle. In this example, the properties of Turtle-1 are:

* Ycor: -5 (7th lane)
* Xcor: -5 (random x position of 7th lane)
* Current-speed: 0.5
* Top-speed: 0.9
* Patience: 19

By pressing "go," Turtle-1 moves to the next patch at 0.5 speed and gradually increases its speed until it reaches the top speed (0.9) and remains there. A broken car blocks the road after a while, causing Turtle-1 to stop and lose 1 patience. But eventually another slower car in front of Turtle-1 causes speed and patience to fall again. Finally, when the patience value drops to 0, Turtle-1 searches for a free space to change the lane. The lane above has another car coming, but it's free below, so it decides to go below and resets the patience value to another random number.

## **Objectives**

There is no explicit goal in the underlying model, as cars tend to move to x-coordinates and change lanes in specific situations.

## **Learning**

There is no learning in this model.

## **Prediction**

There is no explicit prediction in this model.

## **Sensing**

When the cars have to switch a lane, they know that they can go either up or down, but, for example, if there is no other lane below, they know that they can only get to the lane above. In addition, cars do not have details of distant patches or other turtles that cause them to stop immediately instead of slowing down when a car is right in front of them (e.g. a broken car), but this can be balanced by their ability to determine whether they should move to the upper or lower lane without error.

## **Interaction**

As in real life, drivers tend to change their behavior on the road (accelerate, slow down, change the lane) taking into account the behavior of other drivers around them in order to avoid accidents. There is an interaction between the turtles on this model, but they change their behavior only when they almost collide with other turtles. All other factors (bad road, turns, intersections, traffic lights, etc.) are excluded from this model.

## **Stochasticity**

The concept of what occurs when a random car breaks down in the middle of the road is represented with stochasticity. In reality, drivers slow down and attempt to change their lanes to prevent traffic and keep traveling.

## **Collectives**

There are no collectives represented in this model.

## **Observation**

The Ycor of Cars Plot shows a histogram of how many cars are in each lane, as determined by their y-coordinates. In addition, the driver patience chart shows four quantities for the current patience of the drivers: the maximum, the minimum, the average and the current patience of the driver of the selected car. The Car Speeds Plot displays four quantities over time: the maximum speed of any car–BLUE, the minimum speed of any car–RED, the average speed of all cars–VIOLET, the speed of the selected car–GREEN. Plots and file output will be used to obtain data on how traffic can be resolved.

# **Details**

## **Initialization**

The model is initialized by a button press, with initial conditions such as the generation of road and grass; the hidden agent draws the straight lines of the road; 70 cars1 are placed randomly on the x axis with 8 lanes[[1]](#footnote-1); each car has its own patience value and its own maximum speed.

## **Input**

No input other than the parameters already specified is needed or expected, as the model is intended to be self-contained and there is no need for additional variables.

1. Default values [↑](#footnote-ref-1)