|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | Barrier |  |  |  |  |  |  |  |  |
|  | vehicle spawn | |  |  |  |  |  |  |  |
|  | Vehicle |  |  |  |  |  |  |  |  |

Each cell in the grid has attributes:

Type/populated

Speed

Direction

Speed target

Neighbors[]

Percent\_through\_cell

There is a global speed\_goal and each cell has its own speed\_goal that is +-20 from the global

The state if each cell can be any of the types in table/diagram above

Barrier signals that the vehicle cannot go past that direction

Vehicle is a cell that is populated with a simulated vehicle

Spawn is a cell that will cause a cell in front of it to be a vehicle as soon as its an un-populated cell (traffic coming in from one side of road, or could be an on-ramp onto road

Each cycle of the grid will determine where each vehicle is (which cells are populated) will be done by taking cell\_speed\*time\_step\_duration

So if a cell has a speed of 65 (mph) and a time step is .1seconds, then it would travel 10-15ish ft (I did very estimated math) and would move over to the next cell as each cell is defined by a cell\_size variable which is 10 ft (or some other size of a car)

So if a vehicle was in a cell, but thenis going fast enough and time\_step\_duration is long enough, it could move 2 cells in 1 step. Or it could be adjusted to move 1.5 cells in a step, so every 2 steps it moves 3 cells.

This way the simulation can be tuned to have shorter time steps = more work, or any other comination