
updateVehicle(vehicleIterator: Iterator < Vehicle >)
Vehicle vehicle = vehicleIterator.next()
distance(vehicle.currentPosition, vehicle.toCoord) < vehicle.velocity
T
F
directedEdges.get(newDirectedEdge(vehicle.fromName, vehicle.toName)).decrement()
vehicle.currentPosition = add(vehicle.currentPosition, vehicle.direction)
entryPoints.containsKey(vehicle.toName)
T
F
vehicleIterator.remove()
String newDestination = intersections.get(vehicle.toName).getNewDestinationByProbability(vehicle.fromName, random)
! (! Objects.equals(newDestination, vehicle.fromName))
T
F
throw "vehicle is not allowed to turn"
∅
vehicle.fromName = vehicle.toName
vehicle.toName = newDestination
double rest = vehicle.velocity - distance(vehicle.currentPosition, vehicle.toCoord)
entryPoints.containsKey(newDestination)
T
F
toCoord = entryPoints.get(newDestination).coord
toCoord = intersections.get(newDestination).coord
vehicle.fromCoord = vehicle.toCoord
vehicle.toCoord = toCoord
vehicle.direction = subtract(vehicle.toCoord, vehicle.fromCoord)
vehicle.direction.normalize()
vehicle.direction.multiply(vehicle.velocity)
vehicle.currentPosition = add(vehicle.fromCoord, multiply(vehicle.direction, rest))
directedEdges.get(newDirectedEdge(vehicle.fromName, vehicle.toName)).increment()