Package

Map

+ intersections:ArrayList<Intersection>
+ roads:ArrayList<Road>
+ vehicles:ArrayList<Vehicle>

+ createIntersection(intersName:String, intersID:int):boolean
+ createIntersection(intersName:String, intersID:int, roadAttachedTo:Road):boolean
+ attachExistingIntersectionToRoad(intersName:Intersection, roadToAttachTo:Road):boolean
+ createRoad(roadName:Road, roadID:int, maxVehiclesPerLane:int):boolean
+ createRoad(roadName:Road, roadID:int, maxVehiclesPerLane:int, topIntersect:Intersection, botIntersect:Intersection):boolean
+ addVehicle(vehicle:Vehicle):boolean
+ removeVehicle(vehicle:Vehicle):boolean
+ removeVehicle(vehicleName:String, vehiclID:int):boolean
+ getNumberOfVehiclesOnMap():int
+ getAllVehiclesOnMap():ArrayList<Vehicle>
+ getAllIntersectionsOnMap():ArrayList<Intersection>
+ getAllRoadsOnMap():ArrayList<Road>
+ getBoardState():Intersection, Road
+ doesIntersectionExist(intersectName:String, intersectionID:int):boolean

Intersection

- intersectionName:String
- intersectionID:int
- roadsAttachedTo:Road[4]
- trafficLight1:TrafficLight
- trafficLight2:TrafficLight
- currRoadsGoing:int[2]
- vehiclesHere:Vehicle[*]

+ getIntersectionName():String
+ getIntersectionID():int
+ changeTrafficLight()
+ attachRoad(road:Road)
+ removeRoad(road:Road):Road
+ removeRoad(roadID:int):Road
+ isRoadAttached(road:Road):boolean
+ isRoadAttached(roadID:int):boolean
+ getRoadsAttached():Road[*]
+ addVehicle(vehicle:Vehicle):boolean
+ removeVehicle(vehicle:Vehicle):Vehicle
+ removeVehicle(vehicleID:int):Vehicle
+ isVehicleInThisSection(vehicle:Vehicle):boolean
+ isVehicleInThisSection(vehicleID:int):boolean
+ getAllVehiclesHere():Vehicle[*]

TrafficLight

+ currentLight:TrafficLightColor = RED

+ changeLight(lightColor:TrafficLightColor)
+ setLightColor(newColor:TrafficLightColor)
+ getLightColor():String

«use»

«enumeration»
TrafficLightColor

• RED
• YELLOW
• GREEN

Road

- roadName:String
- roadID:int
- maxLanes:int = 3
- maxVehiclesPerLane:int
- lanes:Lane[*]
- topIntersection:Intersection
- bottomIntersection:Intersection
- vehiclesOnThisRoad:Vehicle[*]

+ getRoadName():String
+ getRoadID():int
+ getMaxLanes():int
+ getMaxVehiclesPerLane():int
+ isVehicleOnThisRoad(vehicle:Vehicle):boolean
+ addVehicle(vehicle:Vehicle, lane:int):boolean
+ removeVehicle(vehicle:Vehicle):boolean
+ setTopIntersection(intersect:Intersection):boolean
+ getTopIntersection():Intersection
+ setBottomIntersection(intersectIntersection):boolean
+ getBottomIntersection():Intersection
+ getAllVehiclesOnRoad():Vehicle[*]

1 .. maxLanes

Lane

- listOfVehicles:LinkList

+ addVehicle(vehicle:Vehicle):boolean
+ doesVehicleExistInThisLane(vehicle:Vehicle):boolean
+ removeVehicle(vehicleName:String):Vehicle
+ removeVehicle(vehicleID:int):Vehicle
+ removeVehicle(vehicle:Vehicle):Vehicle

<<Imports>>

Package

mainGameLogic

+ gameWindow:GUI
+ map:Map
+ vehicle:Vehicle
+ player:HashMap<Player>

+ doesVehicleExist(Vehicle):boolean
+ turnAIVehicle(Vehicle, Intersection, int):boolean
+ changeAILane(Vehicle, Road, int):boolean
+ moveGameForward()

<<Imports>>

Package

GUI

- window:Frame

+ drawIntersections(Intersection[ ])
+ drawRoads(Road[ ])
+ drawVehicles(Vehicle[ ])
+ updateMap()

<<Imports>>

Package

User

userName:String
userID:int
userVehicle:Vehicle

+ getUserName():String
+ getUserID():int
+ chageVehicle(Vehicle)
+ getCurrVehicle():Vehicle
+ lookAround(roadOn:Road)
+ lookAround(intersectionName:Intersection)
+ changeLanes(roadName:Road)
+ turn(laneToTurn:int)
+ gamble()

Player

AI

<<Imports>>

Package

<<Vehicle>>

topSpeed:int = 100
currSpeed:int = 0
health:int = 100
inLane:int = 1
carID:int
carColor:String
carMake:String
currRoad:Road
currIntersection:Intersection

+ increaseSpeed(incBy:int = 5):boolean
+ decreaseSpeed(incBy:int = 5):boolean
+ getCurrSpeed():int
+ getHealth():int
+ getLane():int
+ changeLane(laneNum:int):boolean
+ changeVehicleColor(color:String)
+ changeVehicleID(newID:int):boolean
+ getVehicleID():int
+ getVehicleMake():String
+ getCurrRoad():Road
+ updateCurrRoad(newRoad:Road)
+ getCurrIntersection():Intersection
+ updateCurrIntersection(newInter:Intersection)
+ decreaseHealth(dmgTaken:int)

Bus

- topSpeed:int = 90
- health:int = 120
- passengers:Player

+ addPassengers(passenger:Player)
+ removePassengers(passenger:Player)
+ doesPassengerExist(passenger:Player)

Car

- topSpeed:int = 150
- health:int = 90

Truck

- topSpeed:int = 80
- health:int = 130

+ attachTrailer(trailer:Vehicle)