

COSC 3P91 – Assignment 1 – 6402176, 6186076

MICHAEL WISNEWSKI, AIDAN LAROCK, Brock University, Canada

The purpose of this paper is to highlight and cover the design choices made in creating a TrafficSim. This paper will discuss the architecture of the game (Classes, Interfaces, Abstract classes, etc.) and justify the design choices that were made.

ACM Reference Format:

Michael Wisnewski, Aidan Larock. 2022. COSC 3P91 – Assignment 1 – 6402176, 6186076. *ACM Comput. Surv.* 0, 0, Article 1 (February 2022), 5 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 TRAFFICSIM

The task of creating a whole game starts off as a daunting one. The first step in breaking the task down was to create a set of packages each with different aspects of the game.

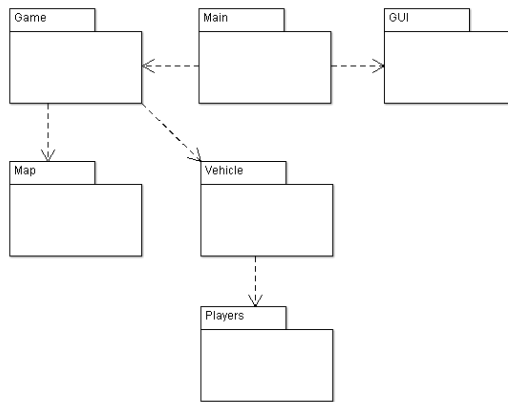


Fig. 1. The packages for TrafficSim (Main, Game, GUI, Map, Vehicle, Players).

1.1 Package Overview

Each package broke the task down into more manageable sub tasks. While the Vehicle package focuses on implementing different types of vehicles and the methods require for creating a vehicle object, the Map package deals with creating and implementing the graph which is the basis of our TrafficSim map. Packages such as Game and Main act as bridges between packages, with the Main class bringing the game and GUI together and the game package providing a class to have all of the vehicles and map objects as well as gambling classes. The Players class allows for a specialized type of vehicle object to become a type of player which allows for control over the object.

Author's address: Michael Wisnewski, Aidan Larock, Brock University, 1812 Sir Isaac Brock Way, St. Catharines, ON, L2S 3A1, Canada.

2022. 0360-0300/2022/02-ART1 \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

2 A PACKAGE BREAKDOWN

The following section will cover each package in more depth and breakdown the design choices of each package and class.

2.1 Main

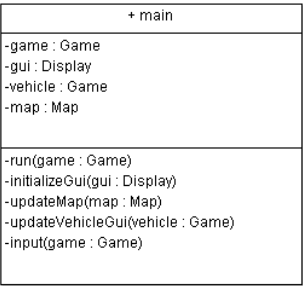


Fig. 2. The main class for TrafficSim.

The main package for TrafficSim is made of one singular class which initializes and calls the game engine and GUI. The game engine (which consists of maps, vehicles and players) sends information to the main class, which acts as a bridge between the GUI and game engine allowing both classes to interact and send and receive information without being able to directly effect the other. The main class is the central class between the GUI and engine.

2.2 Game

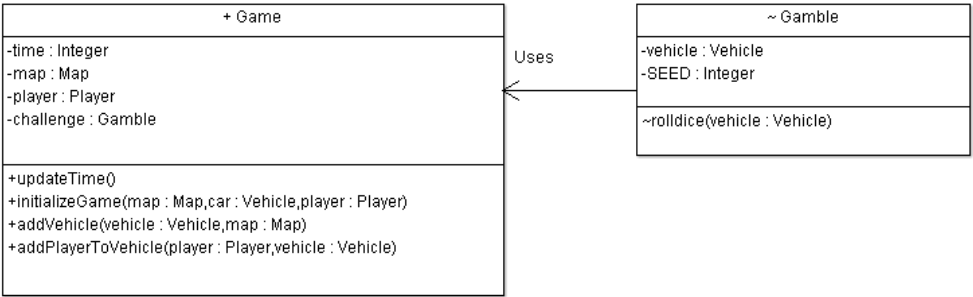


Fig. 3. The game class engine and gambling class.

The game package in the main hub for the engine in TrafficSim. the game class runs vehicle and map updates every timestep. The game class calls to create a map object and

creates vehicle objects adding each to the map as well. The game engine creates map and vehicle objects and brings them all together to create a functional game. The game class also uses a Gamble class. The gamble class uses a randomizer and dice roll to compute random elements in the game. The gamble class is separated from the game class to provide security and help break down the code as the game does not need to know how the random elements in the game are calculated, just t

2.3 Map

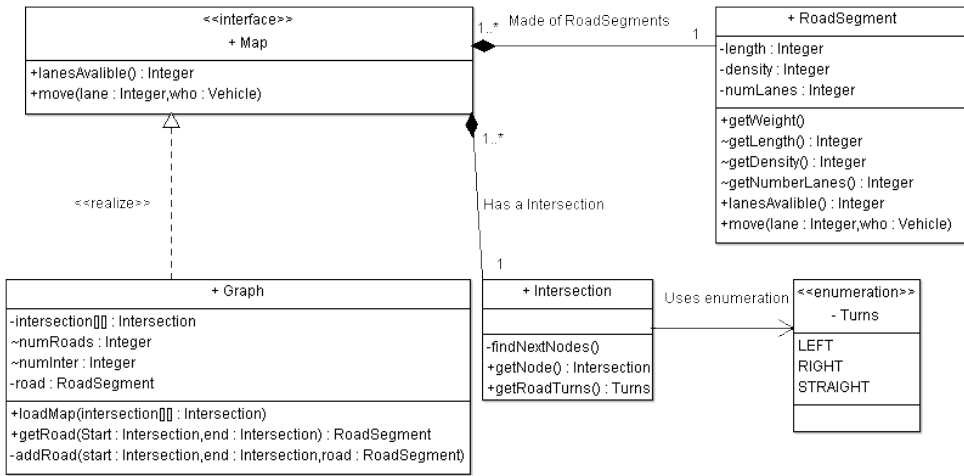


Fig. 4. The map package contains roads, intersections and a graphical representation.

The map package is broken down into an interface and a main Graph class which realizes said interface. A map is made up of at least 1 Intersection and RoadSegment components, each which use a map interface. The Intersection class also uses an enum type of turns (Left, Right, Straight). The Graph class loads and stores a map into a graph (adjacency matrix). An adjacency matrix was chosen as it is faster and better at handling densely populated graphs than an adjacency list. The graph is made up up Intersections (nodes), and RoadSegments (edges). The Map was selected as an interface because each section of the map would need its own implementation, thus we created contracts for each implementation to follow the interface specifications such as the Road Segment class.

2.4 Players

This package is comprised of a single class, Player. this package is a special type of vehicle. The player class adds functionality to a vehicle object, and allows for a player to control a vehicle.

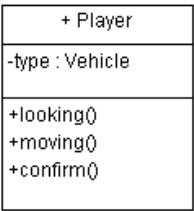


Fig. 5. The player class which is a type of vehicle.

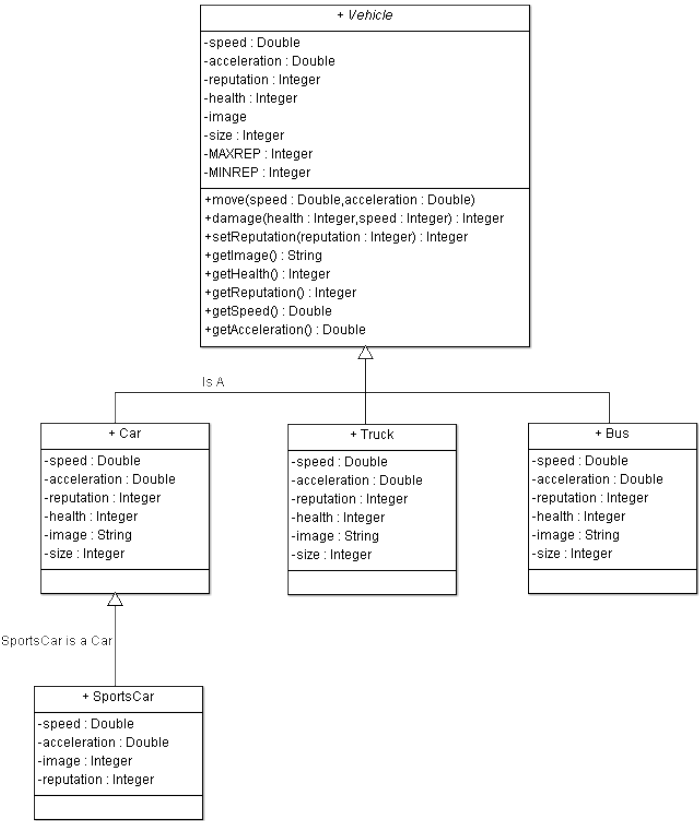


Fig. 6. The vehicle class which is an abstract class and has many types (super car, truck, etc.).

2.5 Vehicle

The vehicle package is centered around a main abstract class (Vehicle). Each different type of vehicle uses the methods from the main Vehicle class and are all a type of vehicle. Each vehicle differs with only their attributes (some vehicles are slower than others, more health, etc.). Vehicle was made an abstract because many of the vehicles (i.e Truck, Bus) are a type of vehicle and will use the same functions from the Vehicle class. By using an abstract

class, the code in Vehicle can be reused in the sub-classes (car, truck, etc.) with each class allowing for distinct attributes when needed.

2.6 GUI

The GUI Class is not yet implemented but is recognised and is separated from the game engine. The GUI will receive updates from the main class and display the updates to an interface. It will also send events that the player inputs to the main class. The GUI does not need to know how the information its passed is generated, just that it receives displayable information. Thus, The GUI class is separate from the engine by the main class.

3 WRAP UP

Overall, the TrafficSim is broken down into 6 distinct packages. Each package is key in providing functionality, visuals, or bridging gaps and hiding attributes and methods. Abstract classes and interfaces are used when possible in order to reduce code and allow for quicker run time as well as easier and faster updating and additions to the code. Visibility is kept strict with attributes and methods being hidden to classes or packages when necessary, and allowed public when needed to improve security and safety of attributes and methods. The TrafficSim implements all of the UML from this design and the Java code for the game can be found attached in the ZIP file.