Map.py contains the universal map class.

The Map class is a structure for storing the most basic variables and functions needed in our driverless code. The racetrack is defined by cones, so the map class holds all cone lists. The pathfinding is (at least at time of writing) stored as an ordered list of Targets (simple objects), so the map class holds this as well. Other basic elements like finish line data, functions for adding/removing cones and functions for checking cone positions/distances/connection-chains are in the map class as well.

The ‘car’ object (attribute of map class, as well as class within Map.py) holds variables, like dimensions, wheelbase and position (within map), as well as functions, for updating position, getting the shortest distance to the chassis and getting the position of the center of rotation. The Car class does not however, include information that depends on the testing scenario. If the ‘instance’ is a simulation, then another class will envelop the Car class while adding things like acceleration parameters and timer values. If the ‘instance’ uses a real-life car, then functions for communicating with the car’s MCU (microcontroller) are added. In those cases, the (initialized) car attribute is overwritten with a new class (which has the Car class as its parent).

We decided to implement a unified map class because we have several programmers working in parallel, on completely different things, which all need to apply to the same driverless ‘instance’. Without a singular map class, we would all make individual (temporary) map classes, which would require conversion to each other’s functions later in the project. By standardizing the map class early on, we can avoid this hassle and simply merge code without rewriting it.

On the specifics of merging code (and applying function-classes (like coneConnecting.py, pathfinding.py, etc.), please review the ‘merged driverless instance’ document)

Currently, the types of data stored by the functions that apply to the map class (like coneConnecting.py, etc.) is somewhat uncertain. As a result, each subclass (Cone, Target & Car) has undefined variables ‘coneConData’, ‘pathFolData’ and ‘slamData’. For now, these attributes will be overwritten with specific mini-classes (functionally just structs), like ‘coneConnection’ (see coneConnecting.py).