pathPlanning(temp).py is a function class (to be used on Map class objects) for auto-driving and calculating cubic splines.

The pathPlanner class contains functions for calculating the steering (and throttle) parameters required to reach target coordinates, like calcAutoDriving() and targetUpdate(), as well as functions for generating cubic splines, makeBoundrySpline() and makePathSpline().

calcAutoDriving() will update the target the car is driving to and calculate the steering control required to reach it. (at time of writing) The steering angle calculation is very simple; the between the car position and the target coordinate, with respect to the cars current heading, constrained to match the cars maximum steering angles. The target is updated whenever the current target is hit, passed or missed, with the distinction lying in the distance and angle to the target. If the distance to the target is below a low threshold, it is considered to have reached it, if the distance is below a slightly higher threshold, and the angle is such that it would not make sense to try to reach it, it is considered passed, and if the distance is above both thresholds and the angle is above a high threshold, it is considered a miss.

The code for finding an appropriate first target is to be improved in the future, as the current code just selects the first target in the Map.target\_list.

Cubic splines can be used to increase the number of points in the cone boundaries and target-list, as well as provide a smoothing effect to them. The splines are stored as an attribute of the pathPlanner class (and therefore any child class with it) to save CPU resources (instead of recalculating them repeatedly). For further information about the merits of cubic splines in autonomous path planning, see the report enclosed as part of the documentation, ‘Autonomous Path Planning using Self-Evolving Cubic Splines’ (Hill, 2021).