pathFinding.py is a function class (to be used on Map class objects) for finding a (rudimentary) path through (previously established) boundary cones.

The pathfinder class contains only 1 function; makePath() generates a new Map.Target object in the Map.target\_list. (this may change entirely, or have a less direct effect in the future) Targets are placed between a left- and a right cone. If there are no previously established targets, it will attempt to find suitable cones for the starting Target. If any finish line cones exist (/have been found), they will be used. Otherwise, it will search for cones near the car. If a first Target has been established, the process for finding a suitable next one is relatively simple; the next target can either be between one of the current (latest) target’s cones and a connected cone in front of it, or between the cones connected to either side of the current (latest) target’s cones. The choice between these (3) options is made by calculating a strength value for each option and selecting the highest one. The strength value is calculated using several parameters: the width of the track (distance between cones) and the angles between the potential target heading (angle car will take) and both the left- and right track boundary, with the idea being that the car’s heading should be similar to the boundary.