1 Introduction

This is a collection of different programs that are intended to make it possible for someone to create a race track and use a genetic algorithm to train an AI to drive around said track; even if they have little to no experience with programming or machine learning.

2 trackDesigner

This program is designed to help make creating a race track as simple as possible. The created tracks can be tested using *single_player.py* where the use can drive around the track, and then AIs can be trained and drive around the track using *AI_drivers.py*.

The initial creation of a track can be done by simply clicking in three different locations and then closing the loop by clicking on the first point. Alternatively, you can click on various points to help give the track some defined shape.

Once the basic shape of the track is created we need to add points and move them around to get it into the desired shape. Adding a new point can be done by clicking on the grey line that runs along the track. To move a desired point simply click on a point and drag it to the desired location.

By default the start-finish line is located at the first point created for the track. The start-finish line can be moved by selecting a the desired point and pressing the 'f' key. Check points can be created by using a similar manner. To add a check point, select the desired point and press the 'c' key. Adding at least one check point fairly close to the start-finish line is important. During AI training, it will be assumed that a car is not performing properly and driving too slowly if it does not cross a check point within 15 seconds. Only one check point is required to ensure that each AI driver is actually driving and progressing along the track.

Tracks can be saved by pressing the 's' key. This process can take a minute, a message will be displayed in the console indicating when the save process is complete.

Loading a previously saved track can be done by pressing the 'x' key. The track that is loaded is based on the value for *track_name* in the file *trackDesigner_class.json*.

If you are unhappy with the results of your track and wish to start again, press the 'r' key. This will remove everything that has been done so far and start you over again with a blank canvas to work with.

3 trackDesigner_class

This class object is used during the track creation process. When the track is saved this class will create a *pkl* file that stores all the necessary information for the track to be used in other programs.

3.1 Default Parameters

track_name - This is the base of the file name that will be used for saving and loading. For example, if the track_name is "raceTrack," the files that will be saved/loaded will be named "raceTrack_params.pkl" and "raceTrack_image.png".

display_max_resolution - This indicates the maximum size of the image that will be displayed on screen. If the track image is larger than the max_resolution, the image will be scaled down.

track_resolution - This sets the resolution of the track. The first value indicates the tracks width and the second value sets the tracks height. Note, this is not the resolution that will be displayed on screen. This can be larger than your display, allowing for the track to be larger, having more detail and a greater length.

m_per_pxl - The conversion factor to convert image pixels to the race tracks size in meters. Using a smaller number allows the track edges to be smoother at the cost of reducing the length of the track.

track_width_m - This indicates how wide the track should be in meters. Real race tracks tend to be about 8 meters in width.

4 single_player

This program allows the user to drive a car around the track to see how the track feels and ensure they are happy with the setup before using for training AIs.

4.1 Default Parameters

use_AI_drivers_params - This parameter can specify a parameters file for the AI_drivers. If specified, the values will be used from the AI_drivers file instead.

track_file - This is the base of the file name that will be loaded to select a race track.

display_max_resolution - This indicates the maximum size of the image that will be displayed on screen. If the track image is larger than the max_resolution, the image will be scaled down.

car_file - This is the base of the file name for the car that is to be loaded. If the car you want is "gt86.json" then car_file should be "gt86".

FPS - This sets a maximum value for the games frames per second.

number_vision_rays - The number of "vision rays" that will be cast out from the car. This is to show what an AI driver will "see."

5 track_class

This class object is where the loaded track data is stored. When the game is running, the *car_class* object will query this class to see if a track wall has been hit or if a check point or the start-finish line has been crossed.

6 car_class

This class object contains the information and functions necessary to progress the car. During run time the function *update_kinematics* is run. This function is responsible for determining how the car is moving around.

6.1 Default Parameters

t_0_60 - This is the 0 to 60 mph time for the car, in seconds.

full_brakes - This is the maximum deceleration rate of the car when the brakes are fully applied.

max_corner_g - This is the maximum cornering g, or lateral g, that the car can do around a corner. The unit is in g where g= 9.81 m/s^2.

top_speed - This is the top speed of the car as determined by gearing. The unit is km/s.

max_speed_drag - This is the top speed could potentially reach as limited by aerodynamic drag. This is used to simulate aerodynamic drag on the car as it accelerates. If the parameter is left as null, max_speed_drag will be set to 1.1×top_speed. if this parameter is less than 2, max_speed_drag will be set to the parameter×top_speed. Otherwise max_speed_drag will be set to be this parameter. The unit should either be none if it is just a factor, the first two conditions, or km/h if it is an actual speed.

turn_radius - This is the turning radius of the car at full steering lock. The units are meters.

car_dimensions - This parameter refers to the dimensions of the car in meters. The first value should be the car's length and the second value the car's width.

updates_per_frame - This parameter is a unitless number and refers to the number of times the model should update the car's motion per frame.

7 create_all_defaults

If you wish to get all new default parameter files you should run this program. In some cases you can obtain a single, new default parameter by simply running a program itself. Programs that allow this are: *trackDesigner_class*, *car_class* and *AI_class_genetics*. Default parameter files for *single_player* and *AI_drivers* can be created by running the function $create\_default\_params()$ in their respective *py* files.

The names of the files for the default parameter for *trackDesigner_class*, *car_class* and *AI_class_genetics* can all be changed. Just be sure to update the files for *single_player*

and *AI_drivers* to refer to the default parameters you wish to use.