Creating an AI for a Car Simulation

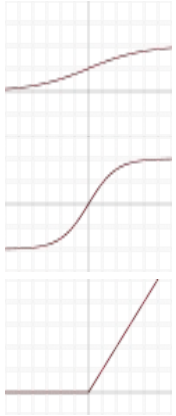
Using a Neural Network

Pedro Albarran 2151023
Alexandre Araújo 2151347

Neural Network

There are currently three activation functions available:

- Logistic
- Hyperbolic Tangent
- ReLU



Neural Network

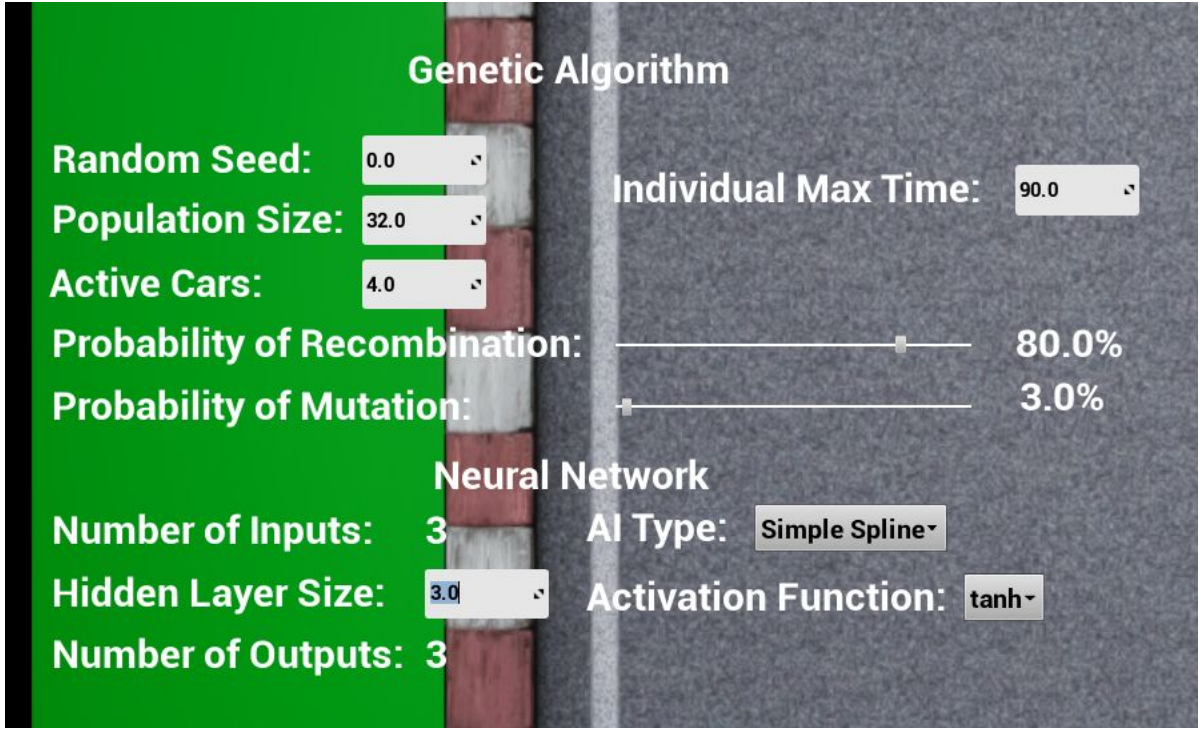
Our tests were done with the hyperbolic tangent since its image maps nicely to the vehicle's input range of [-1, 1].

Abstracted input and fitness function to CarAI class

Genetic Algorithm

Multiple car support

Number of cars on track at once, which CarAI to use, the activation function and hidden layer size



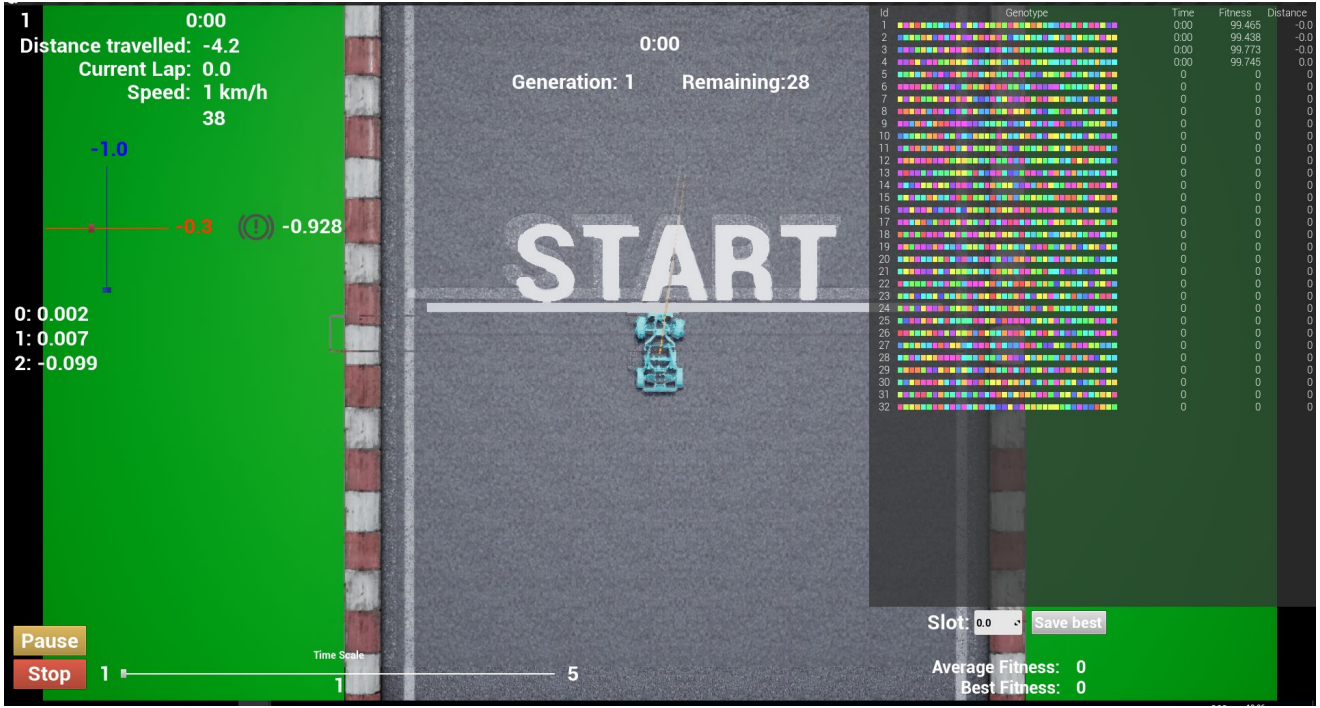
User Interface

Current leader's outputs and other info

Pause and stop buttons

Generation information

Quick save button



Genotype visualisation

The genes are represented by the colored squares, each color represents a value.

As the neural network evolves, the genotypes of different individuals start to converge into a recognizable pattern.

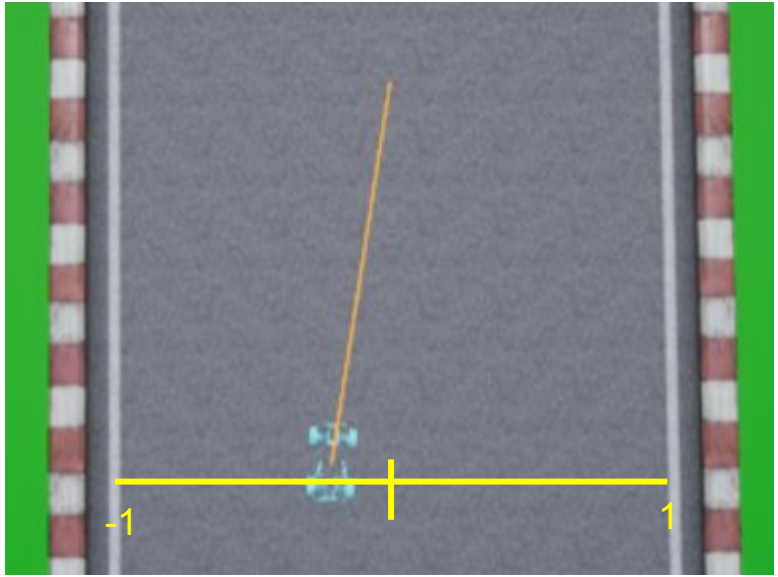
Kill conditions

- Box trace downward to kill the car when it exits the track
- Kill wall behind the starting point
- Maximum life time allowed on track before the car is killed
- Cars with negative fitness
- Cars that are too slow
- Kill the car after 90 seconds

Car AIs

Inputs

Several different AIs were made, each with different inputs.



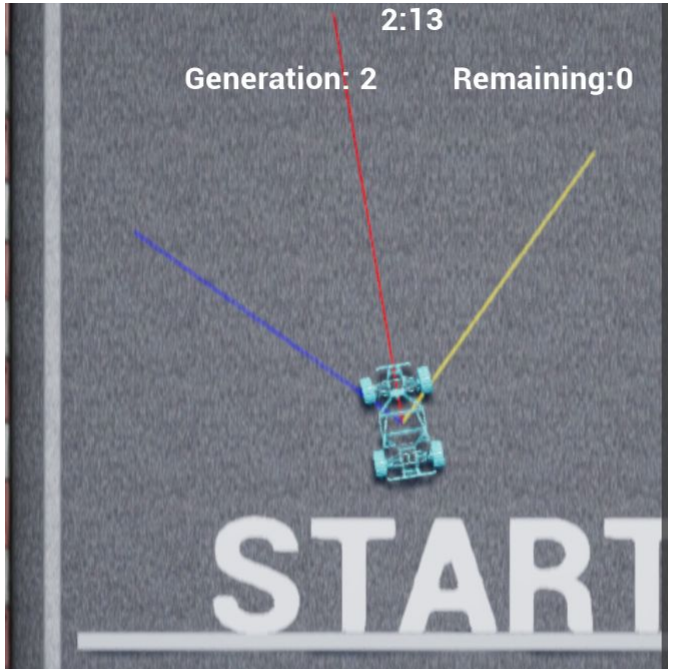
For SplineSimple: the inputs are

- Rotation of the car in relation to the spline of the track
- Rotation of the car in relation to a point some meters ahead
- Distance to track center (normalized to [-1,1])

Inputs

3rays:

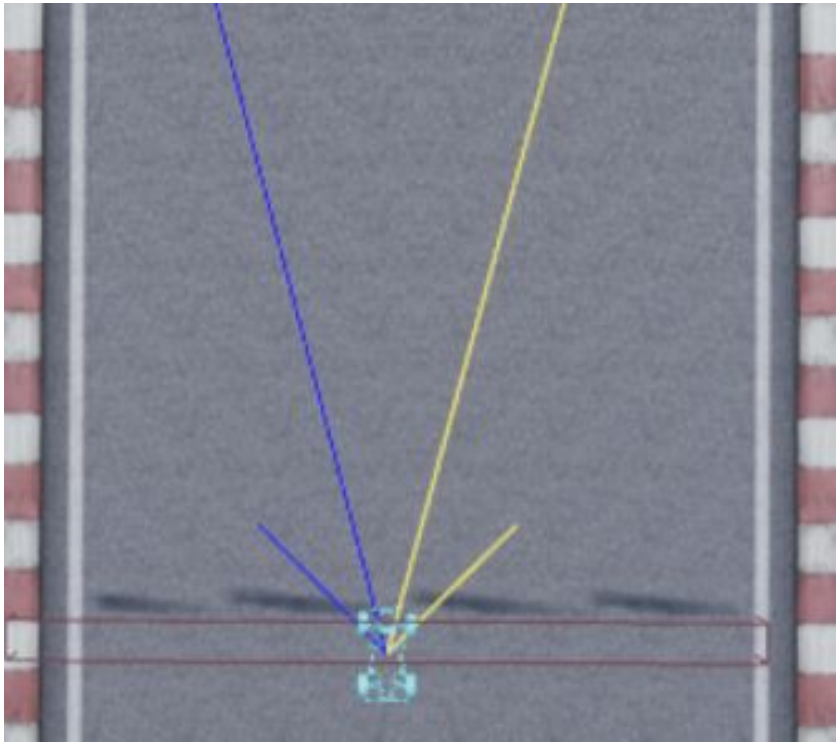
- two forward raycasts at an angle
- one raycast aimed forward, with its length depending on the car's speed.



Inputs

4rays:

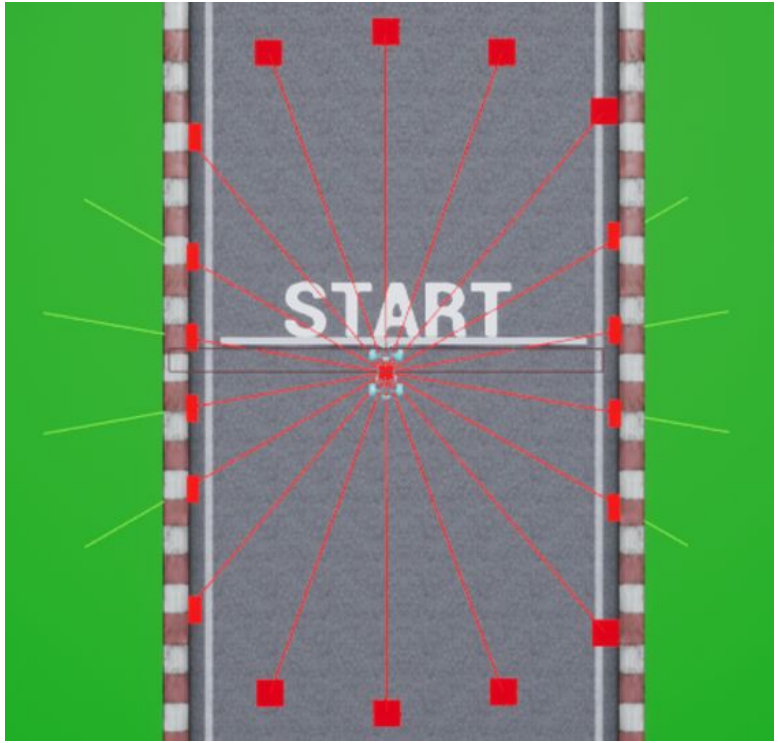
- two short raycasts
- two longer ones to detect upcoming turns. The forward rays are long enough to ensure at least one of them always hits a wall.



Inputs

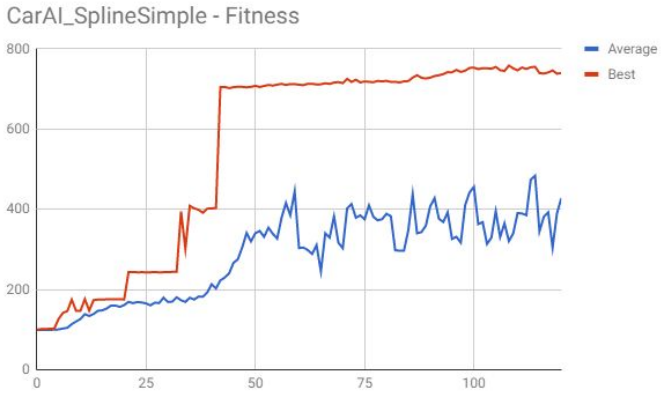
RaycastFest: just a bunch of raycasts in every direction.

Increasing the number of inputs isn't always better, they seem to have a "blinding effect" and make the AI take an extremely long time to learn.



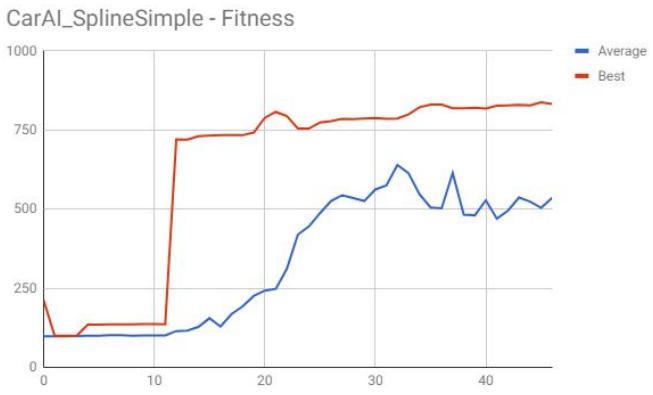
Results

Fitness



Generation number

Fitness



Generation number

What to do better next time

- ➔ Tweak the AI manually
- ➔ Be more consistent with inputs and fitness
- ➔ Get more computers to parallelize tests