GENETIC ALGORITHM FOR A UNITY CAR GAME

Group#5 Mini Project

1. Amjad Mohammed Alsharafi	17103443/1
2. Low Zheng Rong	17152121/1
3. Omar Abdelmomen Amin	17107261/1
4. Omar Ahmed Mohamed	17121293/1
5. Mohamad Najib bin Mohamed	17078332/2

Problem Definition

Car racing is an interesting problem, from creating a challenging race car opponent to testing different race track difficulty level.

We want to test whether an AI algorithms can drive a car through the track without having prior knowledge of the tracks and also its performance.

The solution?

GENETIC ALGORITHMS.

Related Works

Optimising the Performance of a Formula One Car using a Genetic Algorithm

Krzysztof Wloch, Peter J. Bentley (2004)

- Fitness testing is initialized by starting macro software to control the racing simulator and test each car setup in turn
- Evolved car tuning settings emerged with fastest lap time, beating other sets of parameter settings carefully tuned by experts

Related Works (cont.)

A Genetic Algorithm in the Game Racetrack
Robert Olsson, Andreas Tarandi (2011)

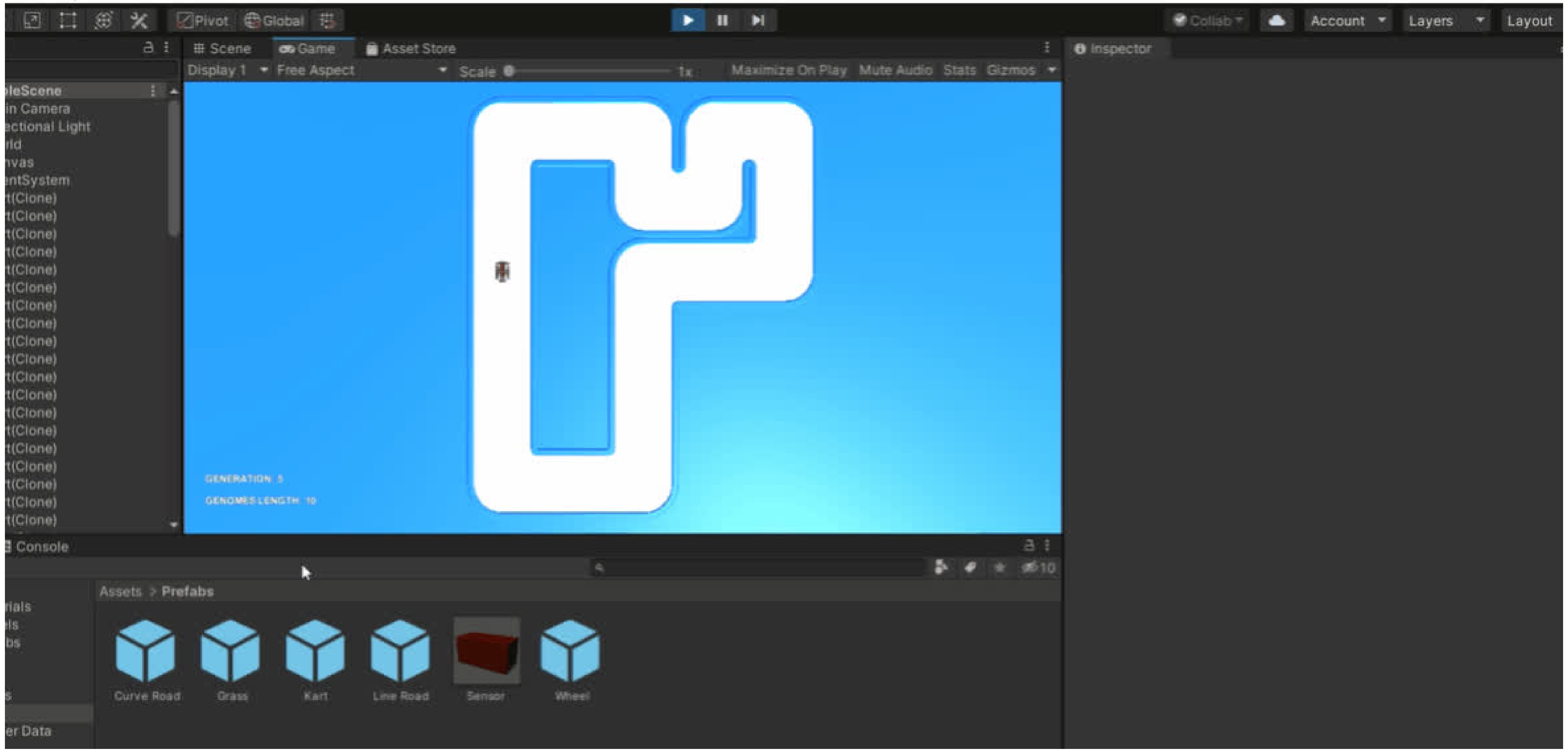
- Implemented in C++ with 3267 different possible state tuples taking inertia vectors and sensor values into account
- Fitness function is determined by measuring how close the cars are to the goal before crashing and how many moves required to get there

Environment and Simulation

The game was predeveloped in Unity 2019. It is simply a car that steers through a twisty road, with realistic physics rules that make steering through the path not an easy task even for humans.

Environment and Simulation

Video for human plays the game.



Genetic Algorithm

Used **Genetic Algorithm** to train the **Kart** to finish the level on its own.

Gene model:

Horizontal Input	**Vertical Input**
Steering	Acceleration

Each Gene is applied every 0.5 seconds

Genetic Algorithm

We use variable length genomes, After each generation an extra **N** genes will be added to each genome.

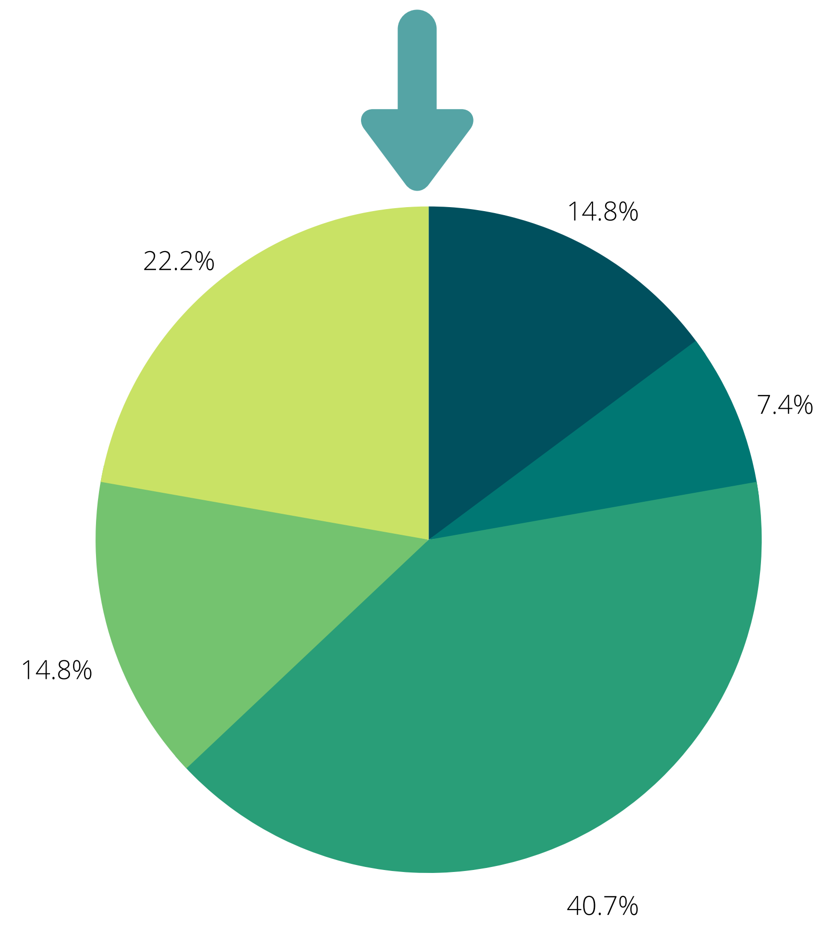
This is to accommodate the variable length of level, and for more dynamic environments.

Genetic Algorithm

After every generation is over a new generation is built using:

1. Pick two parents using **roulette** picking method.
2. Apply **crossover** and generate a new child.
3. Mutate the child.

We also add the fittest 10 from the previous generation to the new generation.



Experiment & Results

Experiment Parameters

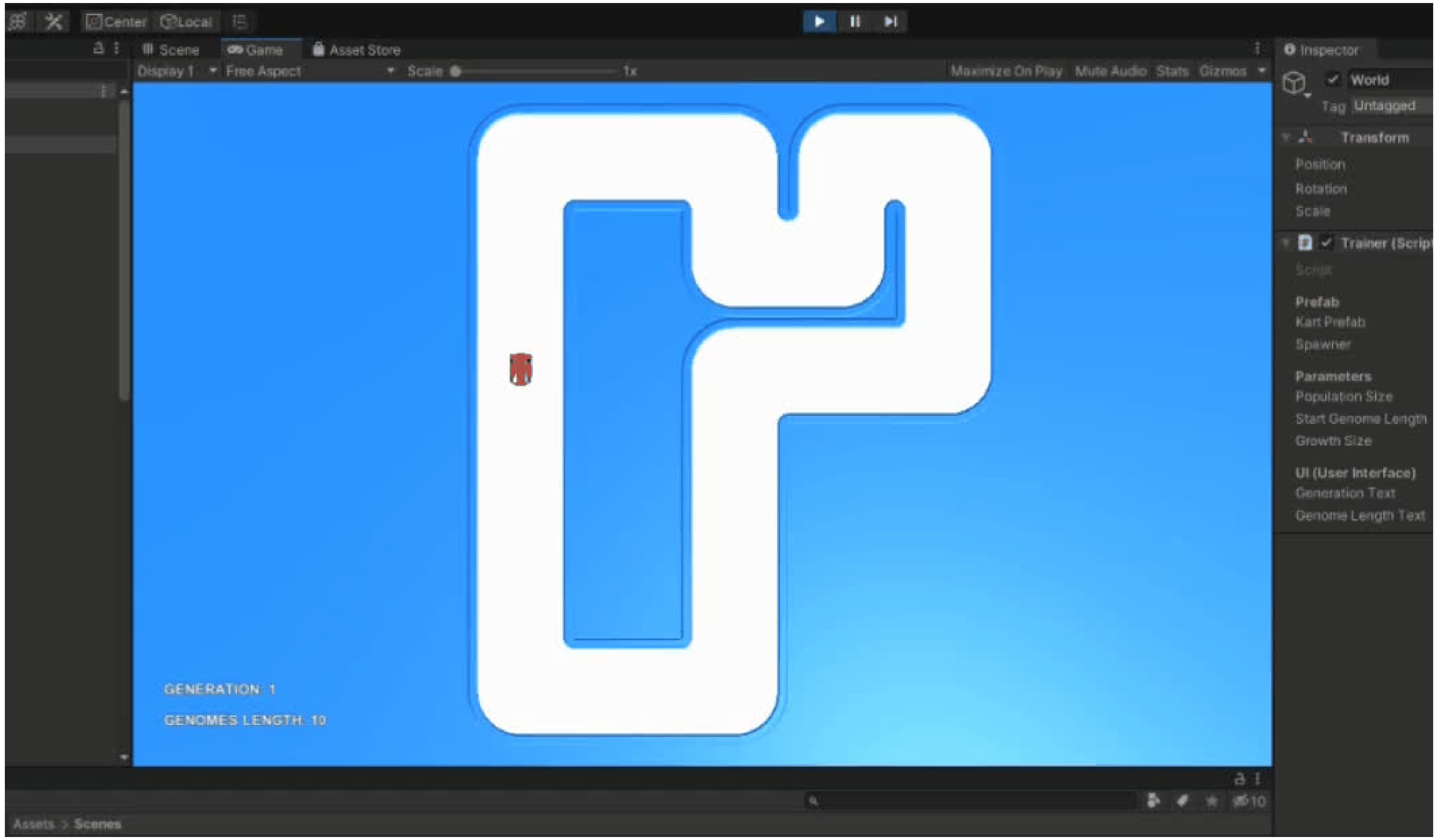
- Population size = 80

- Start Genome Size = 10 Actions

- Crossover (alpha) = 0.2

Experiment & Results

Training Simulation Video

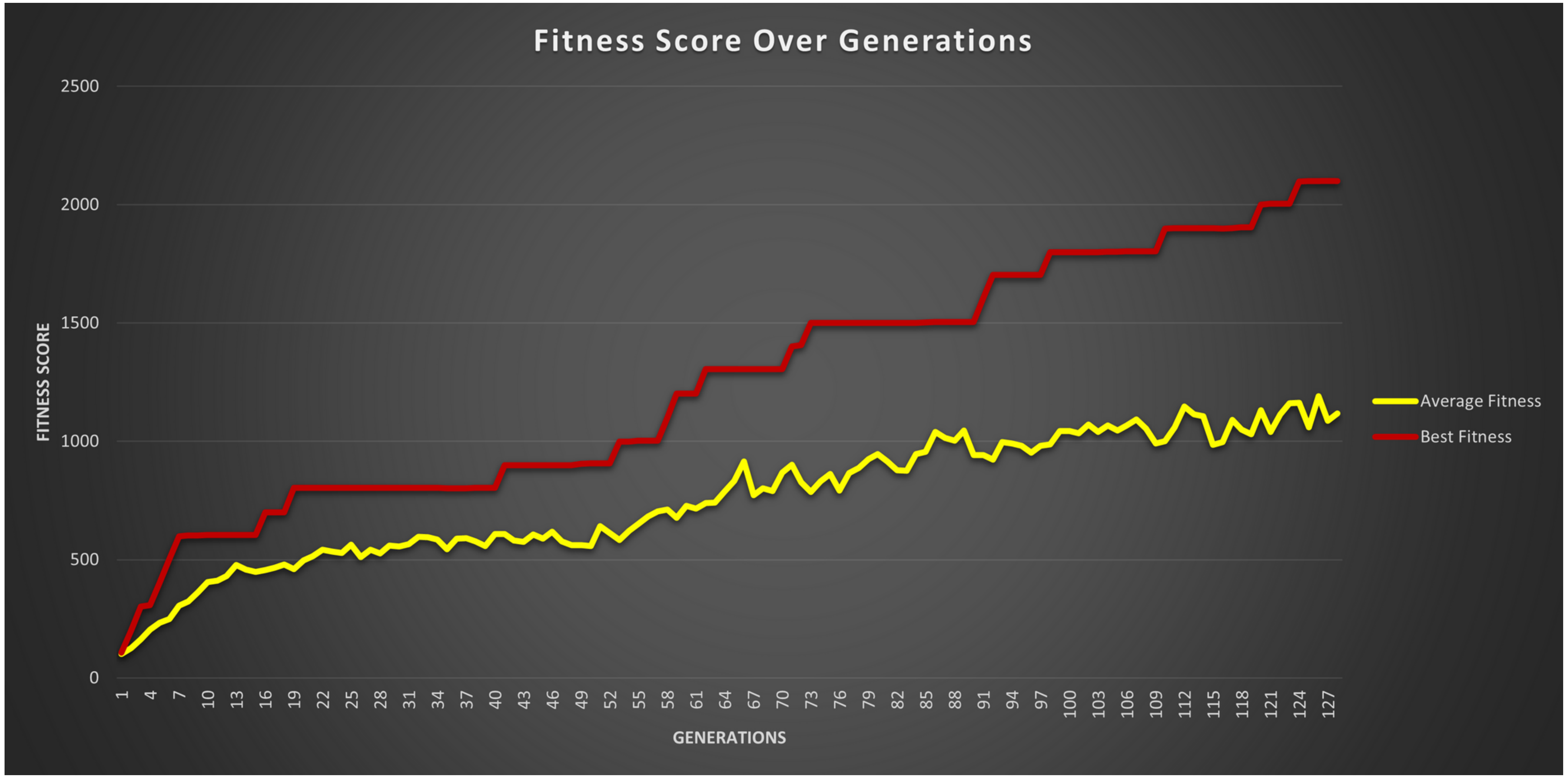


GENERATION: 1

GENOMES LENGTH: 10

Experiment & Results

Results


Fitness Score Over Generations
FITNESS SCORE
GENERATIONS
Average Fitness
Best Fitness