

Technical University of Moldova  
Inginerical department S.A.

# Report

№: 1  
AMOO

Author:  
Prof:

Terman Emil FAF161  
R. Melnic

Chisinau 2017

# 1 Project description

In this project I want to work on a little API that will ease the use of some Machine Learning algorithms with some tools for Unity. The algorithms I want to focus most in this project are Genetic Algorithms. These are some reinforcement learning algorithms that basically find a good, but not the best solution to a problem.

These algorithms are inspired from real world evolution: a certain population is randomly generated, then their score (fitness) is calculated (how far did it jump, or how long did it walk), then, based on fitness, the individuals (genomes) of this population gets to mate (crossover) - the best individuals mates the most. From crossover, a new *child* is made - combine the data of the 2 parents. Finally, this child is mutated and a new generations starts. This loop continues until a certain fitness or goal is reached.

Some iconic problems that can be solved with these algorithms are:

- teaching a robot to walk, jump, etc;
- teaching a car to drive;
- teaching a game agent to play;
- evolving virtual trees;

So the user is aimed to be a developer. He is able to create a **IPopulation** and define the **IGenome** and its **Genes**. The population must contain a container of Genomes and have **Evolve** method, which will move the population to the next generation (the fitness of each individual must be calculated beforehand). The Genome must define a container for genes and a **Fitness** field.

The IPopulation comes with a **PopulationBase** class, which defines **IRandGenomeGenerator** and a **IGenerationGenerator** fields. The GenerationGenerator defines a *Generate* method, which creates the genomes for the next Generation.

The user is expected:

- to evaluate the genomes;
- define a list of genome producers. This can include mating, selection from previous generation and other;

The user can use Neural networks by defining synapses in Genes. For this, a **INeuralGenome** is specified, which the user can feed an array of floats as inputs and receive another array of floats as outputs from the network.

# 2 References

- [\*Robot walking\*](#)
- [\*Self driving car\*](#)
- [\*Evolving game agents\*](#)
- [\*Evolving virtual trees\*](#)