

# COMPUTATIONAL INTELLIGENCE IN VIDEOGAMES AND VIRTUAL REALITY

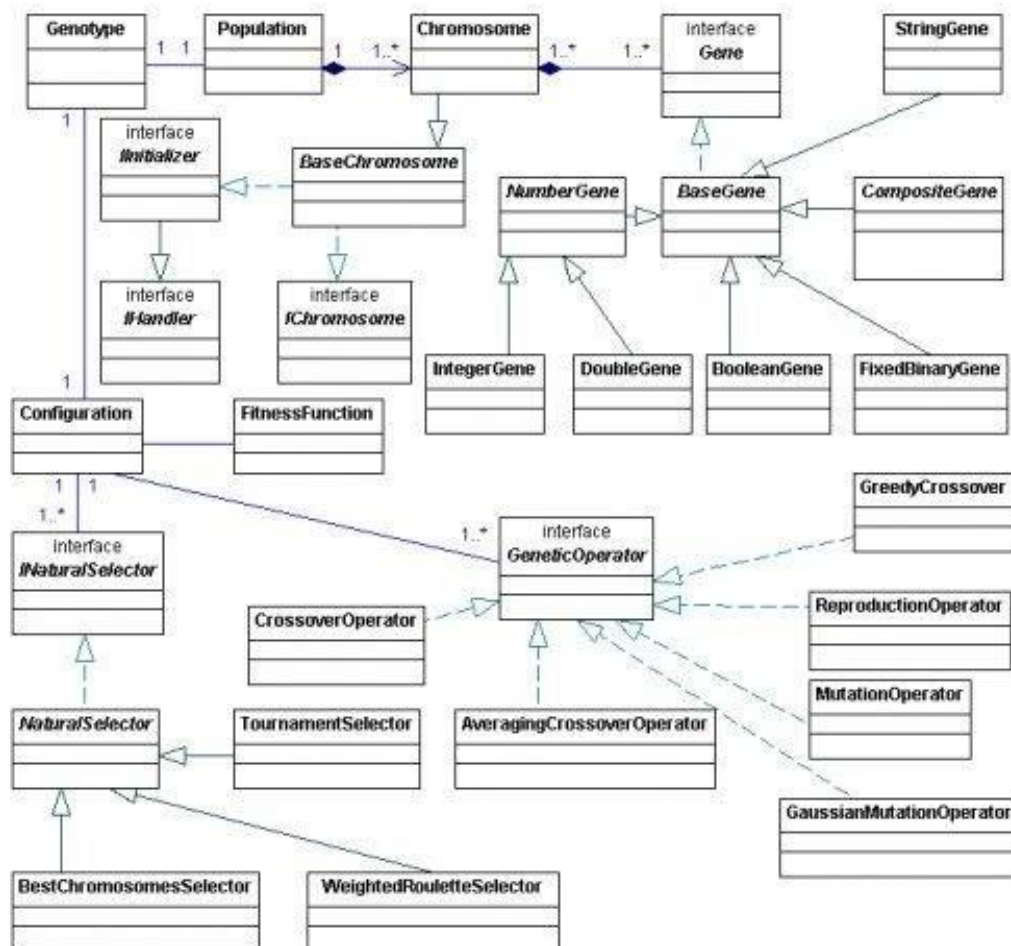
## MODULE 1 – PRACTICAL PART 2: Libraries/Frameworks for EAs - JGAP

This part is just devoted to show an example on how to build a Genetic Algorithm using one of the existing libraries/frameworks for creating Evolutionary Algorithms in Java.

We have selected **JGAP** (<https://sourceforge.net/projects/jgap/>) as it is one of the most extended and easy to use.

*JGAP is a Genetic Algorithms and Genetic Programming package written in Java. It is designed to require minimum effort to use, but is also designed to be highly modular. JGAP features grid functionality and a lot of examples. Many unit tests included. It was created by Klaus Meffert.*

This is the **Diagram of Classes** of JGAP:



We could say that the main component in this diagram is **Configuration**, as it is in charge of defining all the features of the EA to implement. But of course many others are also important:

**Genotype** (used to define the population), **NaturalSelector** (define the selection mechanism), **GeneticOperator** (Crossover and Mutation methods), and **FitnessFunction** (evaluation function).

## **EXAMPLE OF BASIC GA IN JGAP**

Let's see an example which aims to optimize a String, transforming it from random characters to a given sentence. In this case "Hello GAs World!".

Steps to run it:

1. Download the file `JGAP_helloworld.zip` from Moodle.
2. Unzip it wherever you want. Then you will have a structure of folders with the source files (`src`), compiled ones (`classes`), readme to install and run it (`docs`), and libraries JAR files including JGAP (`libs`).
3. Follow the steps in `readme.txt` to install it in your favourite IDE.
4. Now you can build (compile) and run it.

There are two classes:

- `HelloWorldFitnessFunction.java` → This class defines the evaluation of the individuals for solving the Hello World problem. It extends `FitnessFunction` class from JGAP, and overrides `evaluation` function.
- `GaHelloWorld.java` → This class implements the GA loop (`evolve` method), but previously defines and configures the algorithm (`setupGenoType` method).
  - o It uses a population of Vectors of Integer values (the ASCII code of the characters)
  - o The Genetic Operators are those default ones
  - o There is elitism implemented
  - o The fitness function is the one defined in previous class

## **OTHER EXAMPLES ON JGAP**

They can be found and downloaded here:

<http://jcraane.blogspot.it/2009/02/introduction-to-genetic-algorithms-with.html>

<https://code.google.com/archive/p/jc-examples/>

## **OTHER RECOMMENDED LIBRARIES/Frameworks**

There are many other libraries/frameworks for Java, such as:

**ECJ:** Probably the most extended. It is extremely complete, but also quite tedious to be used (at the very Beginning).

Web: <https://cs.gmu.edu/~eclab/projects/ecj/>

**JCLEC:** A very good library, but not as famous as ECJ. Very focused in Machine Learning tasks, such as classification. Not easy to be used.

Web: <http://jclec.sourceforge.net/>

**Jenetics:** The most modern one. It is based on Java Streams, so it profits the new programming advantages. However using it requires the programmer to have extensive knowledge on these techniques.

Web: <http://jenetics.io/>