



# Homework # 4

## Project title

Master Mind

## Name of the group members

- Ivan Mosquera
- Kleber Díaz
- Edgar Carvajal.

**Group number:** 8

**AN ANALYSIS OF THE SELECTED DEVELOPMENT TOOL (SHELL) WHICH WILL BE USED TO IMPLEMENT THE SOLUTION.**

## Library/Language

- **JGAP/Java**

JGAP (Java Genetic Algorithms Package) is a component of genetic programming and genetic algorithms provided as a framework in JAVA. It provides a set of basic genetic mechanisms that can be easily used to apply the evolutionary principles of solutions to problems.[5]

Some of the advantages of using JGAP:

- Introduce Distributed Computing.
- Graphical support (e.g. statistics, configurability).
- Allow for easy extensibility by carefully building up the architecture.
- Extensive documentation
- Detection of changes and errors in the code by means of unit tests. Allows testing of JUnit test.

- **PyEvolve/Python**

Developed to be a complete genetic algorithm framework, written in pure Python, Its main advantages are [6]:

- Multi platform
- Easy-to-use Api
- It allows to visualize the statistical evolution by means of graphs.

- High performance
- **ECJ/Java**  
Written in Java, it is highly flexible; All structures in the system can be easily modified. It is efficient and allows dynamic runtime. [7]
  - Graphical user interface with graphs
  - Multi thread
  - Hierarchical parameter files
  - Abstractions for implementing a variety of EC forms.

## **A DESCRIPTION OF A PROTOTYPE**

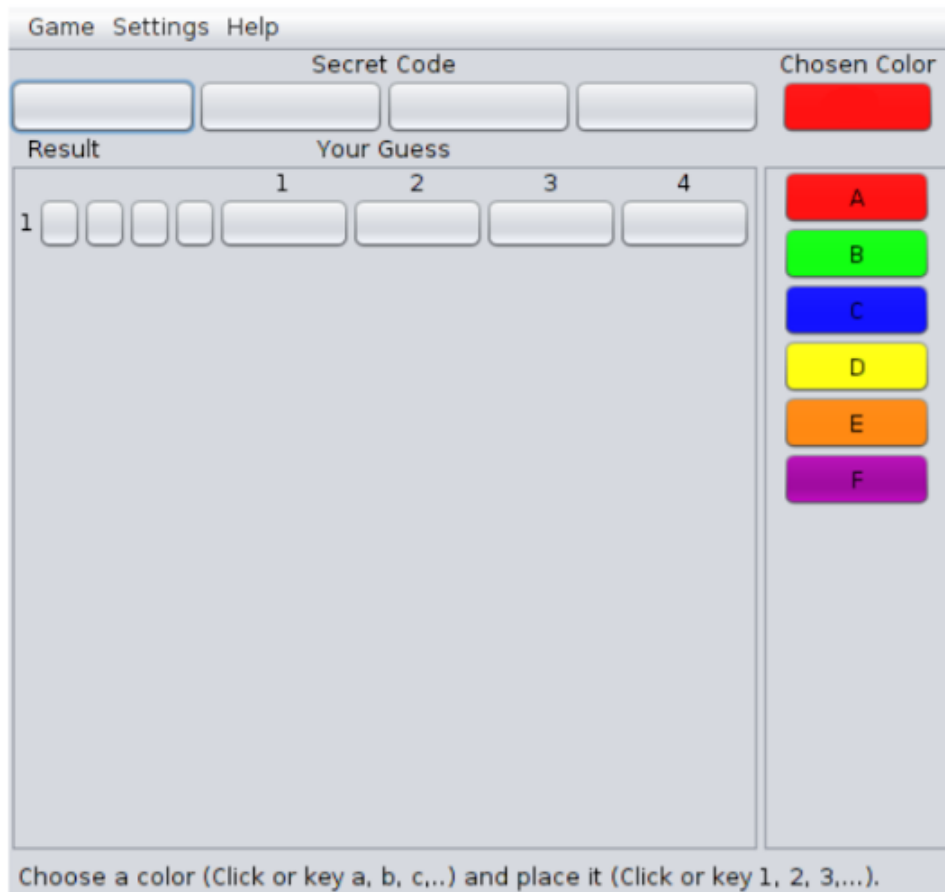
We discussed in group and we decided to develop a solution for this problem using Genetic Algorithm in JAVA environment.

The prototype that we see in *Picture#1* is a demonstration of the final desktop application, developed with Netbeans IDE version 8.1 using the Java Programming Language.

In the final report project we will have an interface where the user can play Mastermind in an easy way to understand.

It will contain:

- The main screen of the Game, that means, to set the colors that will be part of the secret code which the machine or the user will guess.
- User will also put in the board the colors that are thought to be the secret code.
- User will also obtain a small result panel to check which colors are going well in the correct position and which are not, so then with that information try to decipher which is the secret code.



*Picture#1. The main screen of the application.*

On the other hand we can also have the experience of playing with the machine, since we set a secret code (codemaker), then the machine (codebreaker) using genetic algorithms could find a solution to find out the secret code in a fast way and win the game.

And also we can have the experience of playing against another User, where the computer will help us or show us a better way to get the answers and win the game.

## A DEMO OF THE PROTOTYPE OF THE IMPLEMENTATION

First we start a new game, and we select the settings of the game: the number of colors, the width of the game, the max number of tries, to enable or not double colors and the codemaker and codebreaker.



*Picture#2. Settings of a new game.*

During the game the codemaker tries to find out the secret code that the codemaker setted using the different colors availables, the codemaker can also asks for a help to the computer, which will give the user a good possible guess to find out the secret code.

## SOME PRELIMINARY RESULTS

The game ends when the codebreaker finds out the secret code as we see in *picture#3*



Picture#3. End of the game.

In this case, we see that it took five tries for the codebreaker to find out the secret code.

## 5. REFERENCES

- [1]Temporel, A., & Kovacs, T. (2003, September). A heuristic hill climbing algorithm for Mastermind. In UKCI'03: Proceedings of the 2003 UK Workshop on Computational Intelligence, Bristol, United Kingdom (pp. 189-196).
- [2]Runarsson, T., & Merelo-Guervós, J. (2010). Adapting heuristic Mastermind strategies to evolutionary algorithms. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 255-267.
- [3]Berghman, L., Goossens, D., & Leus, R. (2009). Efficient solutions for Mastermind using genetic algorithms. *Computers & operations research*, 36(6), 1880-1885.
- [4]Khalifa, A. B., & Yampolskiy, R. V. (2011). GA with Wisdom of Artificial Crowds for Solving Mastermind Satisfiability Problem. *Int. J. Intell. Games & Simulation*, 6(2), 12-17.
- [5] JGAP Philosophy and Visions, <http://jgap.sourceforge.net/doc/philosophy.html>
- [6]Pyevolve v0.5 documentation, <http://pyevolve.sourceforge.net/intro.html>
- [7]ECJ, <http://cs.gmu.edu/~eclab/projects/ecj/>
- [8]Signal Processing Toolbox, <https://es.mathworks.com/products/signal.html>