INFO 6205
Program Structures & Algorithms
Summer Full 2018
Project: Genetic Algorithms

Team Number: 112
Team member: Zhichao Pan (001493794), Jing Zhou (001441740), Ke Yuan (001491111)

1. Abstract

Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. In this project our team focus on dealing with images.

Given a target image, the application "evolves" a new image from a population of random pixels. It is intended to be a gentle solution using genetic algorithms, using Java. The images are supposed to be Black and White, and more complex topics like RGB pixels, high resolution images, etc., have not been included.

2. Implementation Details

#1 Chromosome: Each chromosome has a gene that represents one possible solution to the given problem.
Each gene is simply an array of pixels.

#2 Gene expression: In our case, each gene represents an array that strives to match the target pixels array. The trait is represented as the color of the pixels (BLACK or WHITE).

#3 The fitness function: Each chromosome has a fitness attribute that is a measure of how close the gene is to the target. This measurement is just a simple sum of the difference of each pixel in the gene to the corresponding pixel in the target array.

#4 The survival function: We use a constant float, elitismRatio, as the portion of the population that will be retained without change between evolutions. (Our population is sorted by the fitness.)

#5 The evolution driver: GAHelloWorld is the driver of this application. The driver code simply instantiates a new Population instance with a set of values for the population size, crossover ratio, elitism ratio and mutation ratio, as well as a maximum number of generations to create before exiting the simulation, in order to prevent a potential infinite execution.

#6 Track the progress of the evolution: The population is sorted by the fitness, so the best candidate from the final generation is the first element of population list.

#7 Unit tests: please see Part 4.

3.Architecture

The genetic algorithm is broken up between two logical units: a Chromosome and a Population.
The image source includes: an ImageSource and a DrawImage.
The driver of the application: GAHellowWorld.

I.Genetic Algorithm

Population

The Population has 3 key attributes (a crossover ratio, an elitism ratio and a mutation ratio), along with a collection of Chromosome instances, up to a pre-defined population size. There is also an evolve() function that is used to "evolve" the members of the population.

Evolution

The evolution algorithm is simple in that it uses the various ratios during the evolution process. First, the elitism ratio is used to copy over a certain number of chromosomes unchanged to the new generation. The remaining chromosomes are then either mated with other chromosomes in the population, or copied over directly, depending on the crossover ratio. In either case, each of these chromosomes is subject to random mutation, which is based on the mutation ration mentioned earlier.

Chromosome

Each chromosome has a gene that represents one possible solution to the given problem. In our case, each gene represents an array that strives to match the target pixels array. Each chromosome also has a fitness attribute that is a measure of how close the gene is to the target. This measurement is just a simple sum of the difference of each pixel in the gene to the corresponding pixel in the target array above. Each gene is simply an array of pixels of BLACK or WHITE.

The functions operating on Chromosome include mutate() and mate().

mutate()
The mutate() function will randomly replace one character in the given gene.

mate()
The mate() function will take another chromosome instance and return two new chromosome instances. The algorithm is as follows:

Select a random pivot point for the genes.
For the first child, select the first n < pivot characters from the first parent, then the remaining pivot <= length characters from the second parent.
For the second child, repeated the same process, but use the first n < pivot characters from the second parent and the remaining characters from the first parent.

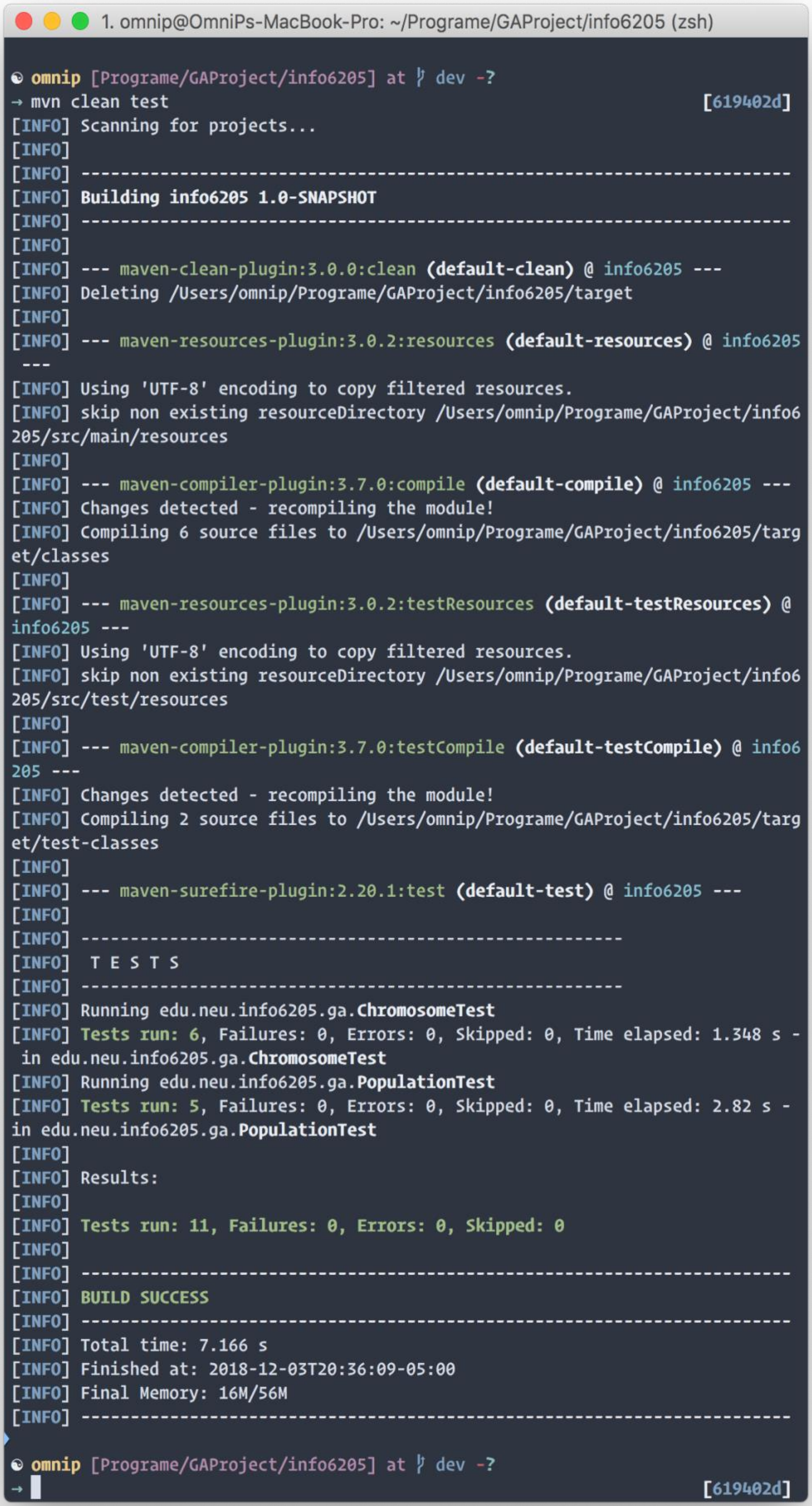
II.Image Source
ImageSource
The ImageSource is intended to maintain and generate image format data from array of pixels.

DrawImage
The DrawImage prints the image as .jpg file in file system.

III.Driver
GAHelloWorld
GAHelloWorld is the driver of this application. The driver code simply instantiates a new Population instance with a set of values for the population size, crossover ratio, elitism ratio and mutation ratio, as well as a maximum number of generations to create before exiting the simulation, in order to prevent a potential infinite execution.

4.Unit Test

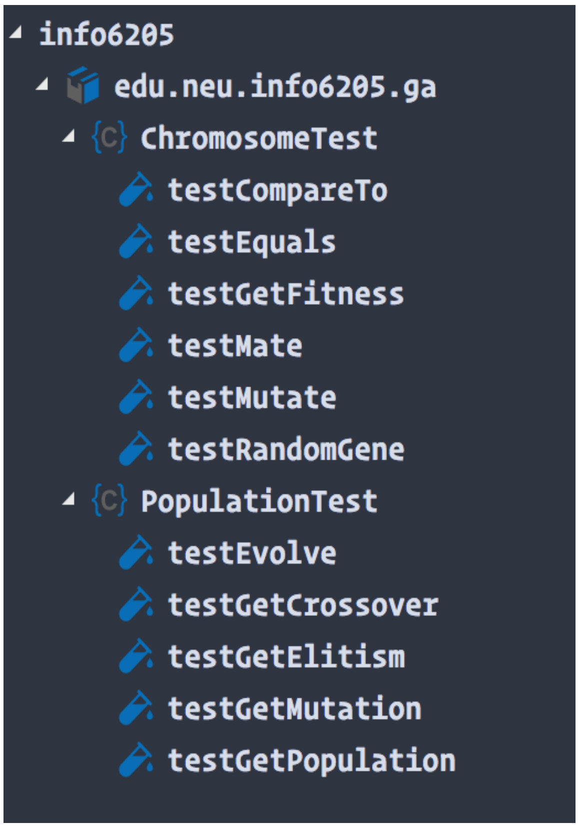
I.Test Results


1. omnip@OmniPs-MacBook-Pro: ~/Programe/GAProject/info6205 (zsh)

```
omnip [Programe/GAProject/info6205] at dev -?
→ mvn clean test                                                   [619402d]
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building info6205 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:3.0.0:clean (default-clean) @ info6205 ---
[INFO] Deleting /Users/omnip/Programe/GAProject/info6205/target
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ info6205
 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/omnip/Programe/GAProject/info6
205/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @ info6205 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 6 source files to /Users/omnip/Programe/GAProject/info6205/targ
et/classes
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @
info6205 ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/omnip/Programe/GAProject/info6
205/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:testCompile (default-testCompile) @ info6
205 ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 2 source files to /Users/omnip/Programe/GAProject/info6205/targ
et/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.20.1:test (default-test) @ info6205 ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running edu.neu.info6205.ga.ChromosomeTest
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.348 s -
 in edu.neu.info6205.ga.ChromosomeTest
[INFO] Running edu.neu.info6205.ga.PopulationTest
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.82 s -
in edu.neu.info6205.ga.PopulationTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 7.166 s
[INFO] Finished at: 2018-12-03T20:36:09-05:00
[INFO] Final Memory: 16M/56M
[INFO] ------------------------------------------------------------------------

omnip [Programe/GAProject/info6205] at dev -?
→                                                                  [619402d]
```

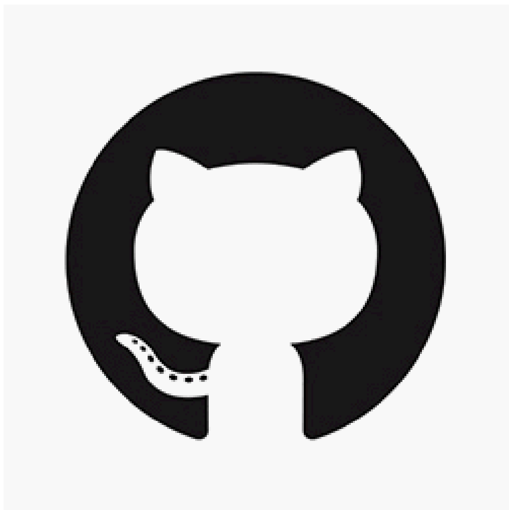
II.Test Methods



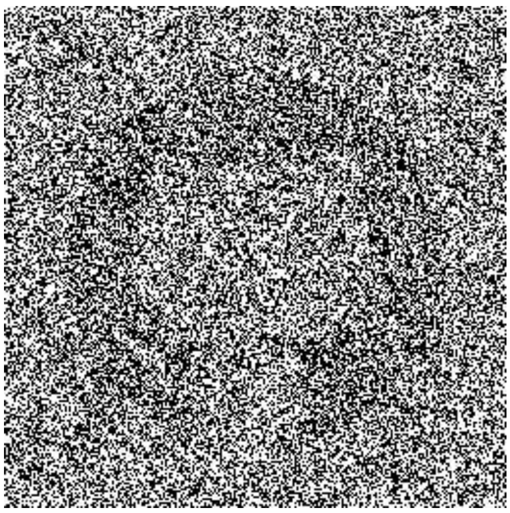
```
info6205
  edu.neu.info6205.ga
    ChromosomeTest
      testCompareTo
      testEquals
      testGetFitness
      testMate
      testMutate
      testRandomGene
    PopulationTest
      testEvolve
      testGetCrossover
      testGetElitism
      testGetMutation
      testGetPopulation
```

5.Solution: Generate image

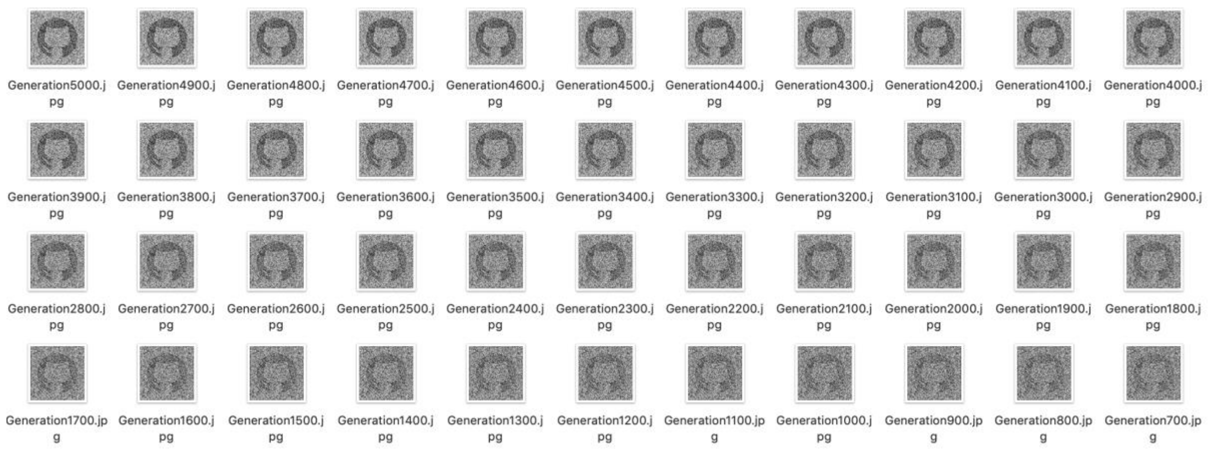
I.GitHub Icon



Target Image



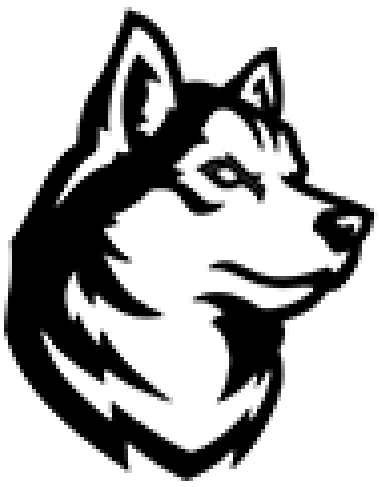
Sample of Generation 1000

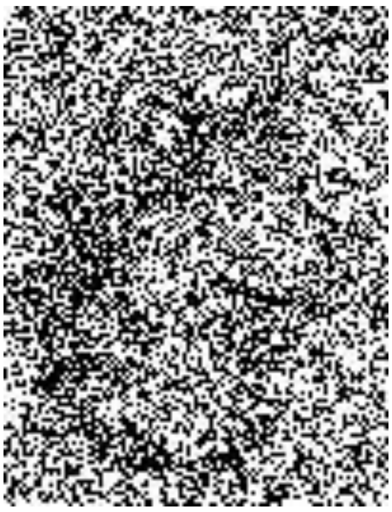


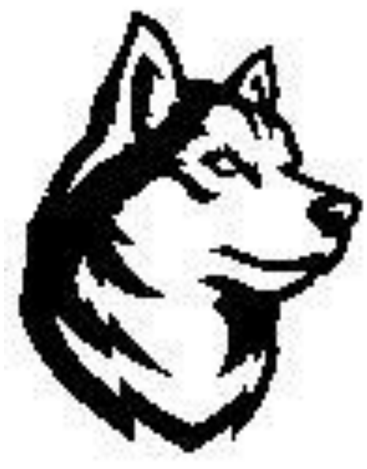
Generation Map

Since the resolution of the target image is too large, it takes an extremely long time to evolve into the exact answer. The execution stops when the generation reaches 5000.

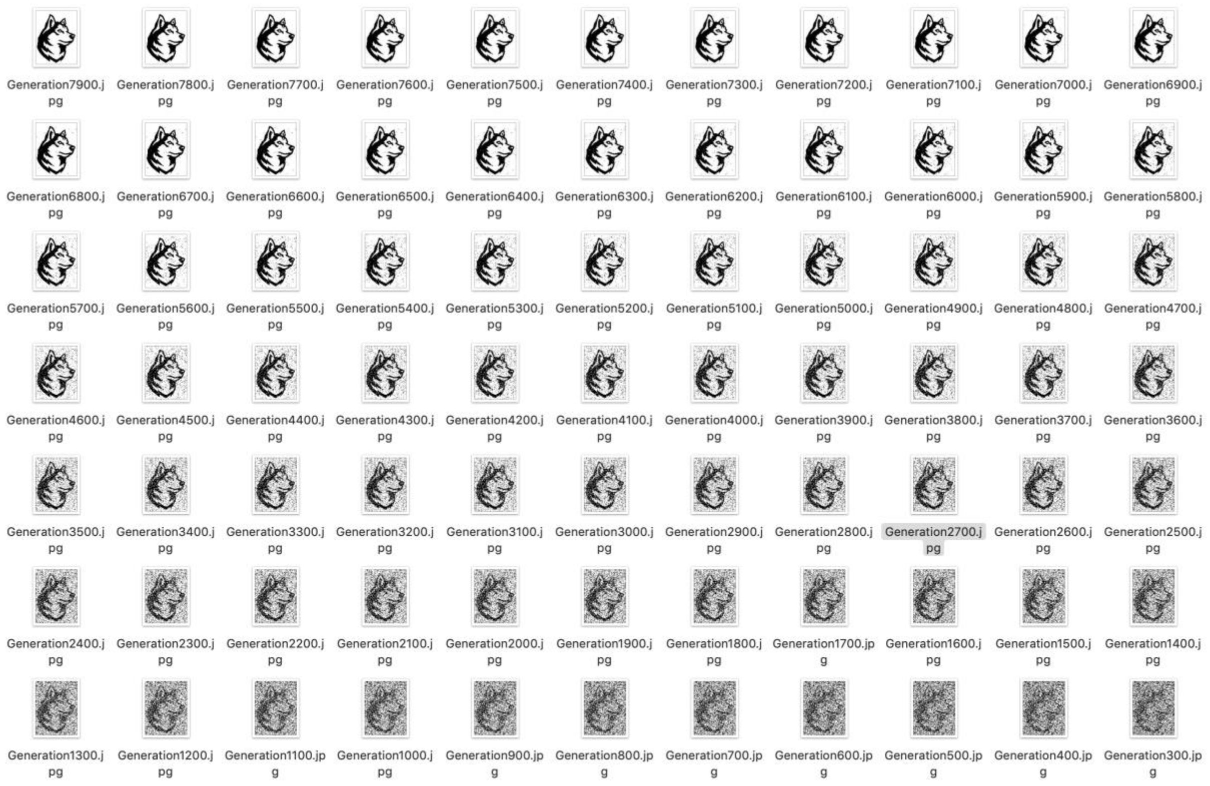
II. KingHuskyHead







Target Image Generation 100 Generation 9000



Generation Map

The population finally reaches the target at generation 9009.
Total execution time: 468664ms.

6.References

1. https://en.wikipedia.org/wiki/Genetic_algorithm
2. https://github.com/jsvazic/GAHelloWorld