INFO6205 Final Project Report

Find the best starting pattern for Game of Life

Group 303
Yen-Hua Huang 001376848
Ziyu Zhou 001492338

Logic of finding best starting pattern

We use genetic algorithm to find the best starting pattern which can last longest in Game of Life.

Generate Genotypes

We use Jenetics to generate a population of individuals, we set the population size to 20 initially.

- **Genotype**

Our genotype is a set of bit chromosome. We use 1 byte(8 bits) to represent a cell and create patterns with size 20.
The first 3 bits are used to decide the direction, and the rest of the bits represent the steps of moves.

- **Phenotype**

Our phenotype is a list of points which we convert as a string pattern.

Example

Genotype:
“01100010111101100001111111001100010001011100110111101000111011000001100010111011010010110010101001100101”
Phenotype:
“1 -1,-3 0,0 5,0 -2,2 0,0 -3,-1 0,-2 0,0 2,-4 -4,3 0,-2 2,2 -2”

Fitness Function

Our fitness function is based on the generation which the pattern can last. The more generation it has, the higher fitness score it can get. We will give each a double value score which calculated by following.

```
double d = (1000/(double)(10000-i))*(double)100000;
```

Selection

1.Sort those patterns according to the Fitness score.
2.Select the better half of the population as the fittest individuals and do mutation of those.

Mutation

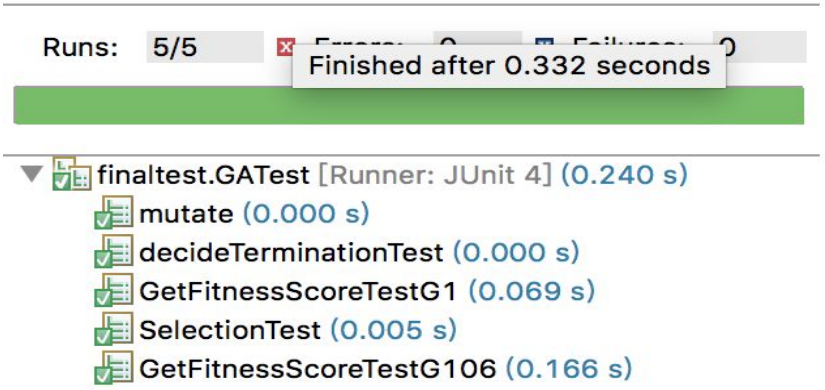
The mutation rate means the mutate possibility of each gene(each bit). We set the mutation rate as 0.0001 and we gave each bit a random float range from 0-1, if the float>mutation rate , it will do mutation, if not, the bit remains the same.

How we do evolution

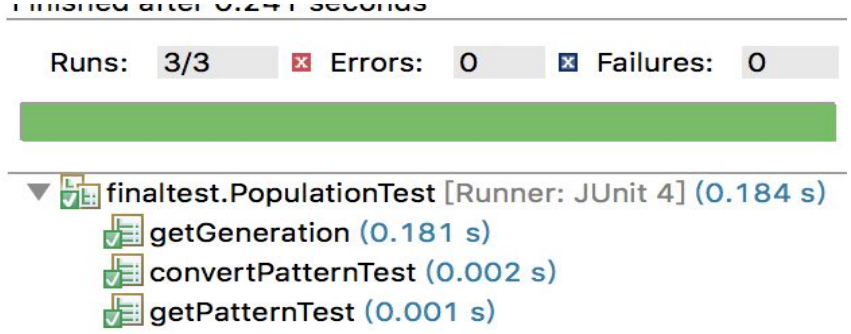
1.Generate genotypes and convert them to points, we have 20 individuals at first.
2.Run the game and get each pattern's generation, then give Fitness score for each individual.
3.If nobody's generation exceed max generations, which has been set to 10,000, then use Selection to get the top half of the population according to their Fitness score and do Mutation.
4.Combine the new borns which have been mutated with the parent patterns and run the game again.
5.Repeat 2 to 4 until we find the one which can last in the game for more than max generations.

Unit Test

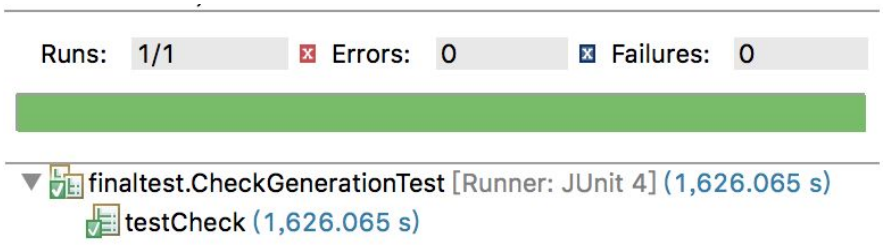
1.unit test class 1: GATest()
Mainly do the test for GA, including mutation, selection, fitness function, decideTermination test



2.unit test class 2: PopulationTest()
Mainly do the test for the expression of genotype and the termination point for evolution



3.unit test class 3:CheckGenerationTest():
To check pattern's generation and decide if we have to keep doing the evolution or not



Conclusion and Findings

The best pattern we found:

- **Population = 20**
- **Mutation = 0.0001**

- **numberOfPoint 5-8**
- **>10000 generation.**
- **[10010101001101001110000111100110010100101110101110100010]**
- **[3 3,-2 2,-1 0,-2 0,2 0,-3 0,-1 -1]**
- **i:14**
- **Seed: 1575760089581 (Set as final value in Main.class)**

We do the experiment many times and find some factors that may have influences on our best starting pattern:

1. Number of individuals: given fixed amount of individuals initially and running the game will terminate with fewer generations than random amount. In our case, it will need to do selection and mutation more time for getting the qualified pattern.
2. Steps the individual move: if the moving steps of the cell is too large, then it will terminate with less generation because of extinction.
3. Population size: if given smaller population size, we will need to do more selection and mutation to get the best pattern.
4. MutationRate: We will need more time to find the best pattern if possibility of mutation is lower.