INFO 6205 Final Project – Generic Algorithm for Game of Life

Team 102

Group Members:
Yifan He
Mingxin Liang

Instructor: Robin Hillyard

The topic of our project: Game of Life (cellular automata problem)

According to the Wikipedia's article: "The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970." Given a board with m by n cells, each cell has an initial state live (1) or dead (0). Each cell interacts with its eight neighbors (horizontal, vertical, diagonal) using the following four rules.

Team Details

Team number: team 102

What we Did:

- Create many patterns derive from a random number sequence
- Create a genetic algorithm to find the best pattern which can have max Generations
- Create **a fitness function** which is a measure of how good a candidate solution is for finding the best pattern which can have max Generations
- Create a **candidate selection function** to select individuals who can produce next generations. There is a condition that only if the individual's fitness is not less than the average fitness, they will generate next generations.
- Create **a mutation function** to in which a new pattern of chromosome expression
- Create an **expression function** to show the different between genotype and phenotype
- Create **unit tests** to test most of the methods to keep the project operating properly
- Add **UI** which users can upload a file and run the generation algorithm

Design of Genetic algorithm

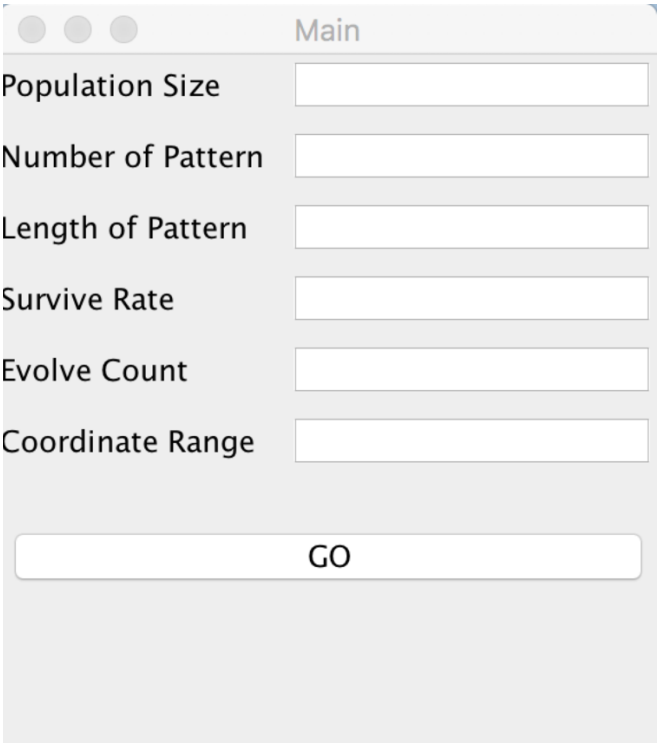
Evolve ()

1. sort old population by fitness
2. add survive population to new population
3. mutate
4. replace old population

Unit tests ()

1. Candidate selection Test
2. Expression Test
3. Fitness Test
4. Mutation Test

UI



Conclusion

By changing some parameters, we found many parameters have any parameters have influences to algorithm:

Population Number: If we choose large population number, the algorithm will get the best pattern earlier. That means to get the best entity in a single generate algorithm experiment, the larger the population is, the smaller generation number will be. And we need to choose larger population number for pattern, or we will not get the correct answer even the generation number is very large.

Survive Rate: If we choose too large a survival rate, the rate at which we get results will be slower, but if we choose a too small survival rate, we will most likely not get the optimal solution.

iter	population	generation	score	chromosome
0	120	1000	0.001	3 5, 5 3, 5 1, 8 3, 6 5, -1 8, 6 9
1	60	1000	0.001	9 5, 1 9, 7 6, 1 8, 3 3, 8 9, 3 6
2	30	1000	0.001	2 7, 8 4, 9 7, 4 6, 5 7, 6 5, 4 3
3	15	1000	0.001	0 1, 6 5, 8 8, 9 4, 0 4, 3 4, 6 9
4	7	1000	0.001	0 6, 0 7, 2 2, 4 3, 4 0, -2 9, 5 3
5	3	1000	0.001	0 1, 6 5, 8 8, 9 4, 0 4, 3 4, 6 9
6	1	1000	0.001	1 6, 1 3, 8 0, 9 4, 8 1, 6 4, 3 6