Final Project Report

-Game of Life through GA

Lihang Zhou 001302072, Zihan Li 001083359, Xinyue Cui 001344423

github.com/catduck2010/final\_game\_of\_life

**Introduction**

Game of life is a zero-player game. It includes a two-dimensional rectangular world, and each square in this world is inhabited by a living or dead cell.When there are 2 or 3 living cells around a square, the living cells in the square continue to survive at the next moment.If there are no living cells in the square at this moment, the living cells will be "born" at the next moment and the cells died before will revival if it has exactly three neighbors.

Our project used the Genetic Algorithm (GA) to solve the question of getting a pattern that able to grow and last for long in Conway’s Game of Life.

**Genetic Algorithm**

Genetic algorithm is a global optimization probability algorithm with many advantages:

1. The genetic algorithm does not have too many mathematical requirements for the optimization problem to be solved. Due to its evolutionary characteristics, the intrinsic nature of the problem is not needed in the search process. For any form of objective function and constraints, whether linear or non-linear yes, discrete or continuous.
2. The ergodic properties of evolutionary operators make genetic algorithms very efficient in searching for global meanings of probability meaning.
3. For a variety of special problems, genetic algorithms can provide great flexibility to mix and construct domain-independent heuristics, thereby ensuring the effectiveness of the algorithm.

**LANGUAGE USED**

This project was programmed based on Netbeans with JDK8 development kit.

**GA Code**

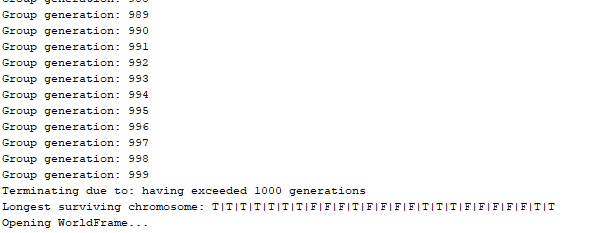
We used Jenetics for the frame of GA.As Jenetics does not have a type of a implemented gene that stores Boolean values, we defined a new gene using class AnyGene to implement it. We used this type of gene to implement the simplest method to encode and decode, which is to use every value of the chromosome to express a cell is alive or dead. We would set a value of the length of the sequence first.

Furthermore,we wrote an explainer to convert from genotype to phenotype and we set a fitness function to measure population (in Jenetics, it refers to a set of phenotypes), value is calculated as the product of the generations of the pattern survives and the difference of number of surviving cells of the pattern of last generation and the initial pattern. The bigger the value, the better the phenotype.

We also designed the UI interface. The UI interface can show the evolution of the pattern with the longest survival time and the increase in the number of cells. We used Gamepanel to create a visual interface to display patterns,World class to caculate the evolution of the pattern,WorldFrame JFrame to show what Gamepanel shows and control its evolution process manually or automatically.

**Result**

The longest surviving chromosome output obtained by the genetic algorithm is shown as Figure 1:



The evolution of the pattern with the longest survival time and the increase in the number of cells are showed as below figures.

Figure 1.

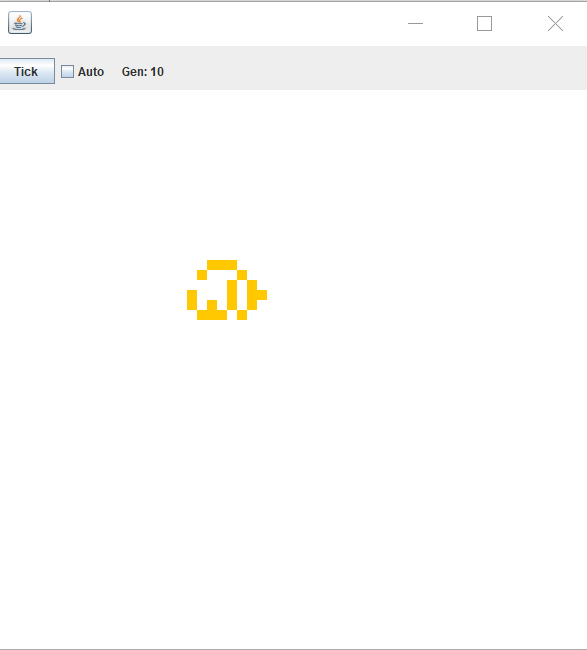
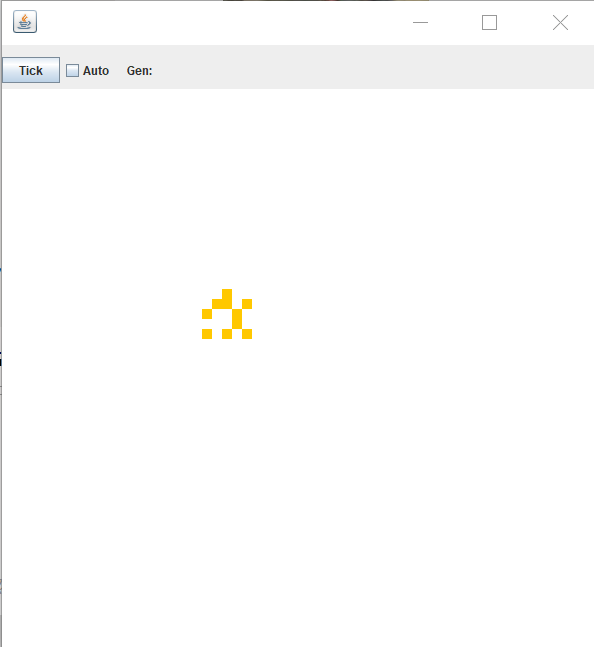


Figure 2 Generation=0 Figure 3 Generation=10

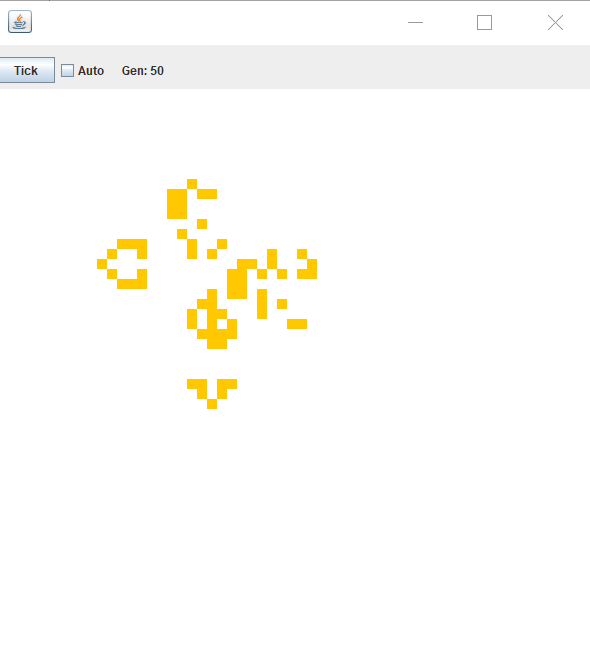
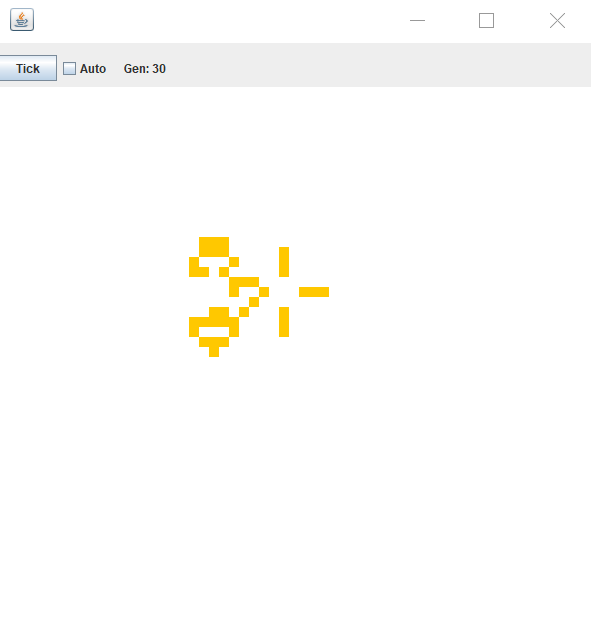


Figure 4 Generation=30 Figure 5 Generation=50

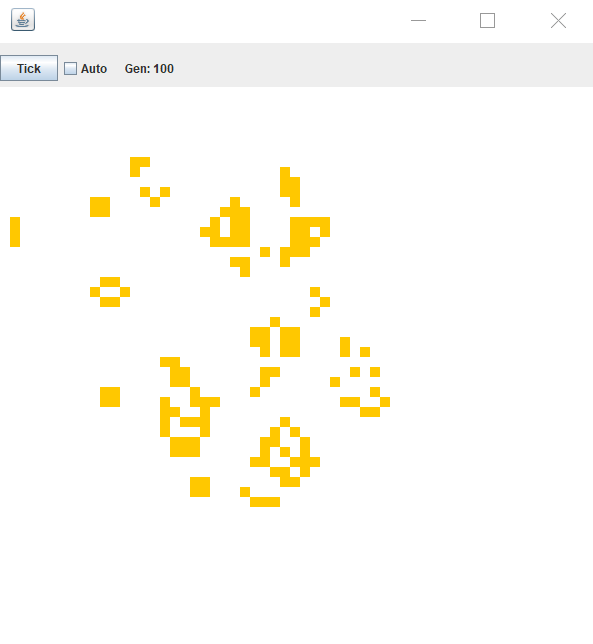
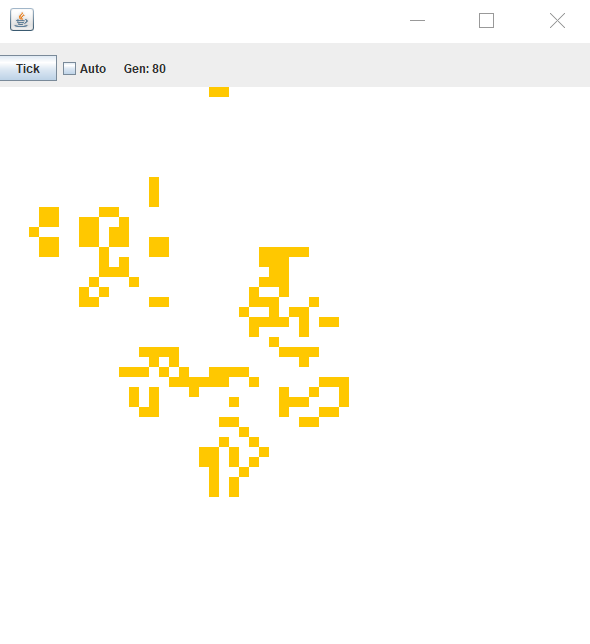


Figure 6 Generation=80 Figure 7 Generation=100

As we can see from those figures, during the iteration, this chromosome has been alive all the time, and the more iterations, the more cells in it.

We finally realized using the Genetic Algorithm (GA) to solve the question of getting a pattern that able to grow and last for long in Conway’s Game of Life.