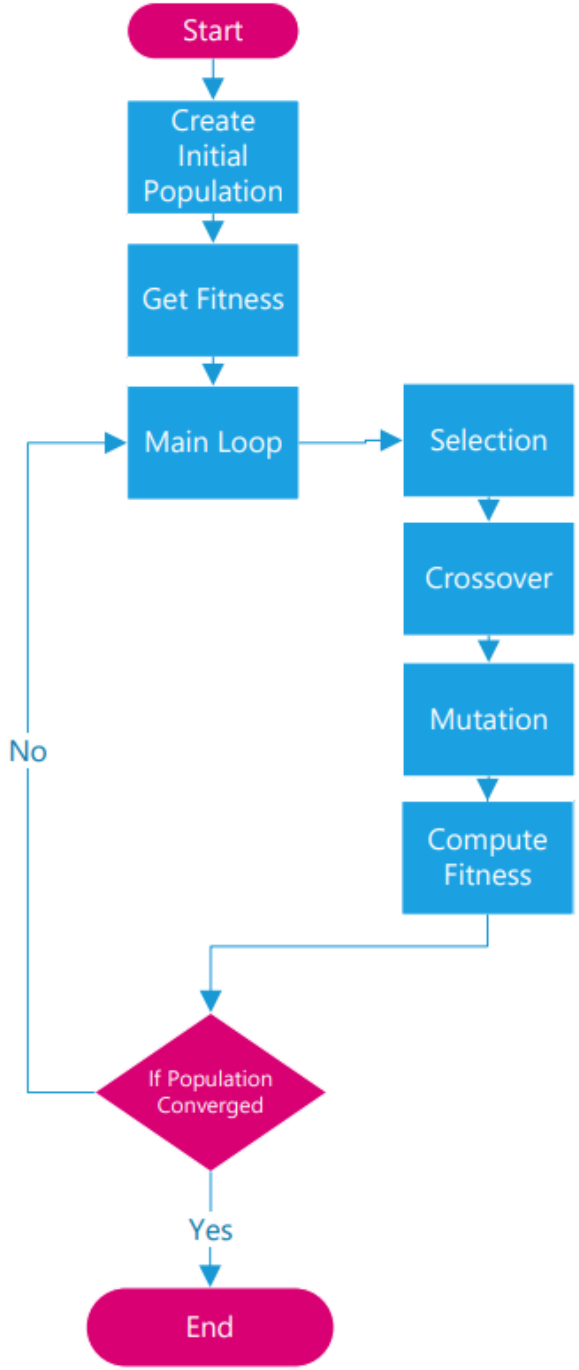
CSSE220 Genetic Algorithm

Cai Yuchen and Logan Manthey

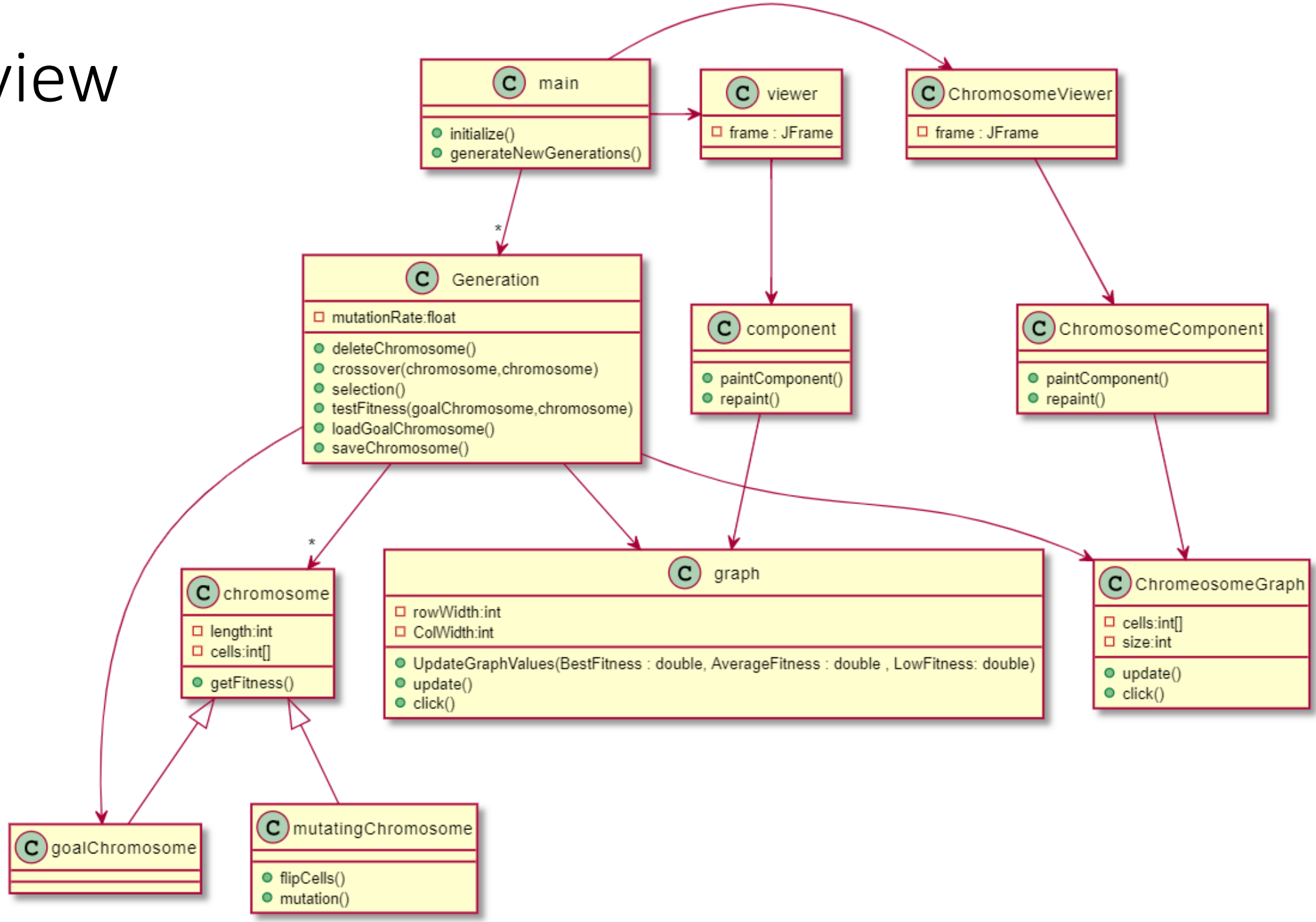
Introductions and Basic Idea

- Cai Yuchen
- Logan Manthey

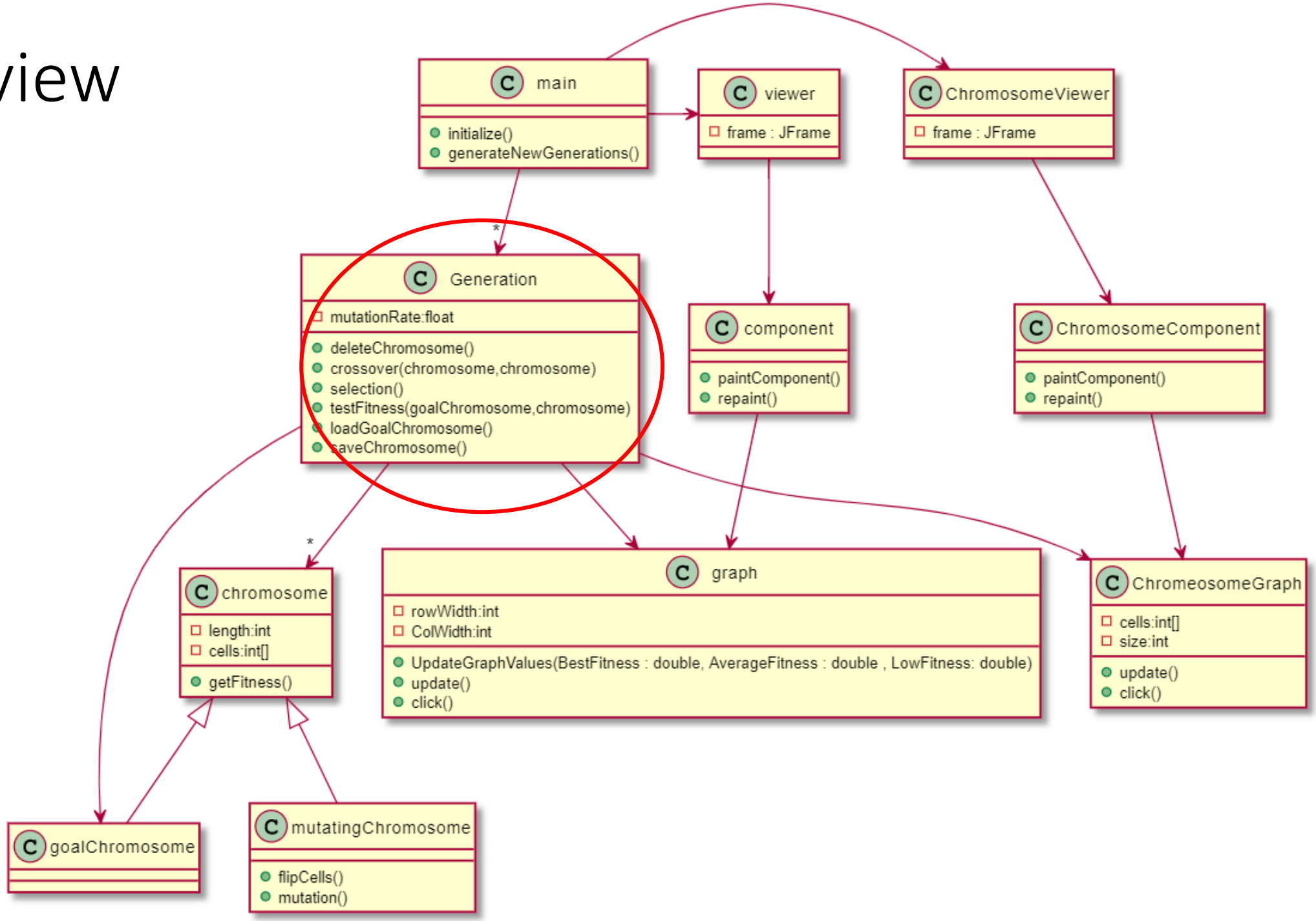
- Basic Idea of Project
  - Initial Population
  - Fitness Function
  - Selection
  - Crossover
  - Mutation



UML Overview

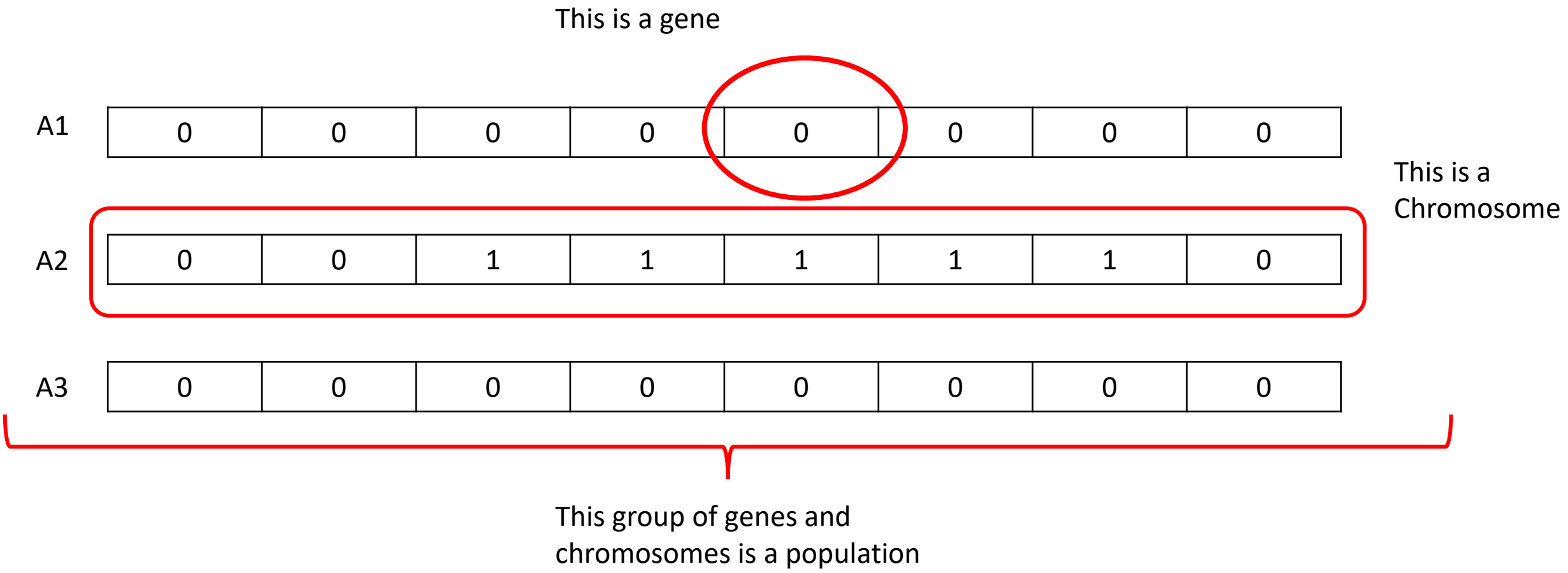


UML Overview

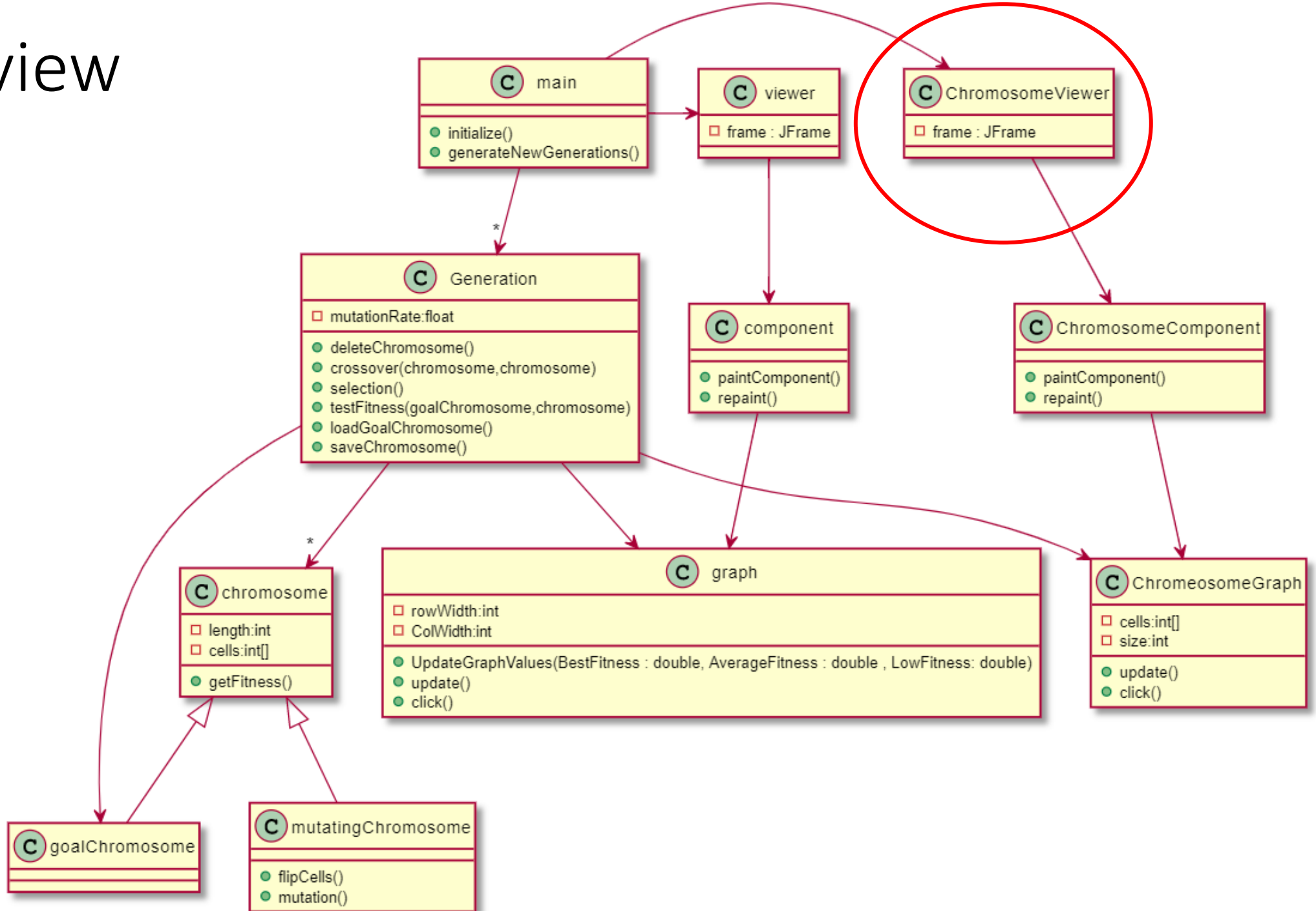


Initial Population/Generation

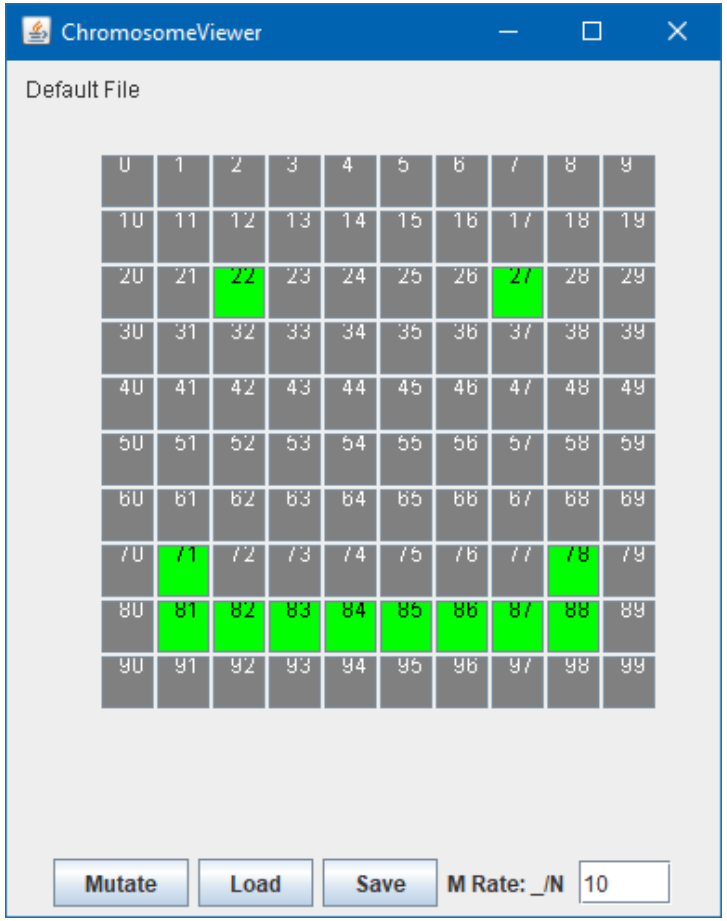
In Biology Chromosomes carry genetic information to be passed down to offspring

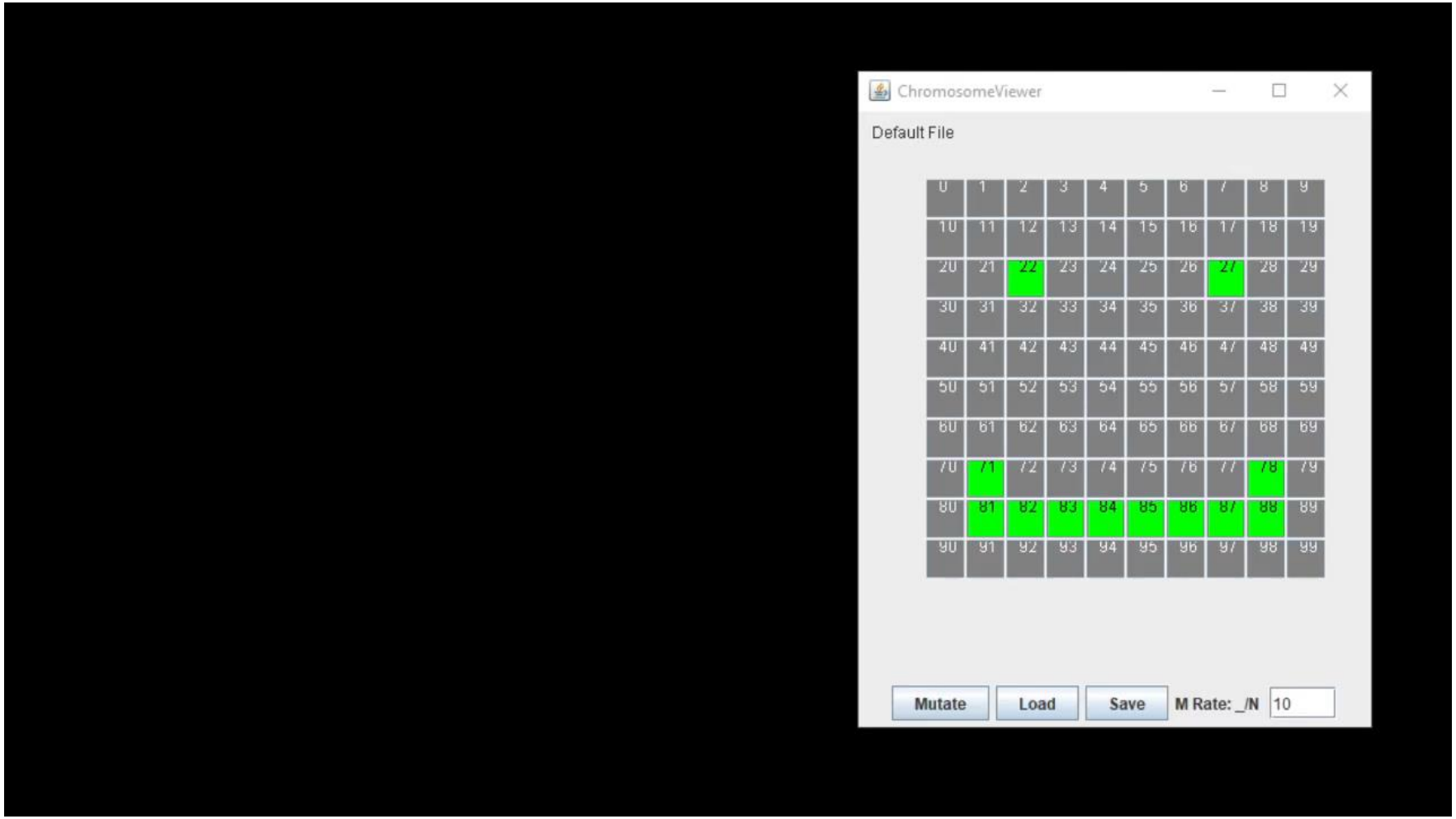


UML Overview



Chromosome Viewer in Program





Mutation

- These offspring from before can become mutated based on a probability defined in the parameters of the evolution viewer.
- This mutating flips bits in the chromosome
- This creates a diverse population and delays the convergence (the end condition)

A1	0	1	0	1	1	0	0	0
A2	0	0	1	1	0	0	0	0

Fitness and Selection

- **Fitness**
  - How fit each induvial is
  - Fitness Functions assign fitness score to each induvial
- **Selection**
  - This is where the fittest individuals are selected and allowed to pass their genes to the next generation
  - Individuals with high fitness have more change to be selected for reproduction.

For example, one could assign a fitness score based on the number of consecutive ones. This process of assigning a value for this fitness is a fitness function

A2	0	0	**1**	**1**	**1**	**1**	**1**	0

Fitness Functions

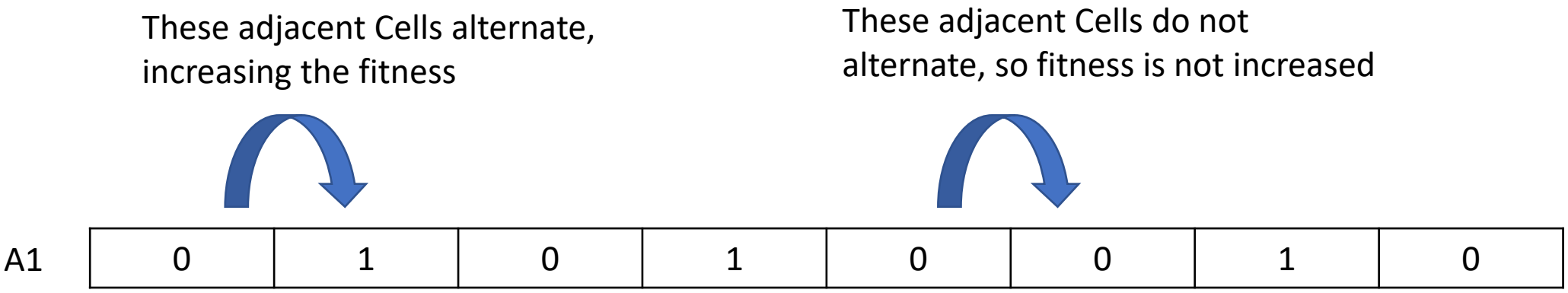
- Number of Ones
- Number of Max Consecutive Ones
- **Alternating Adjacent Values**
- **Target Chromosome**

This data is then used later to compare fitness between the chromosome and the goal chromosome

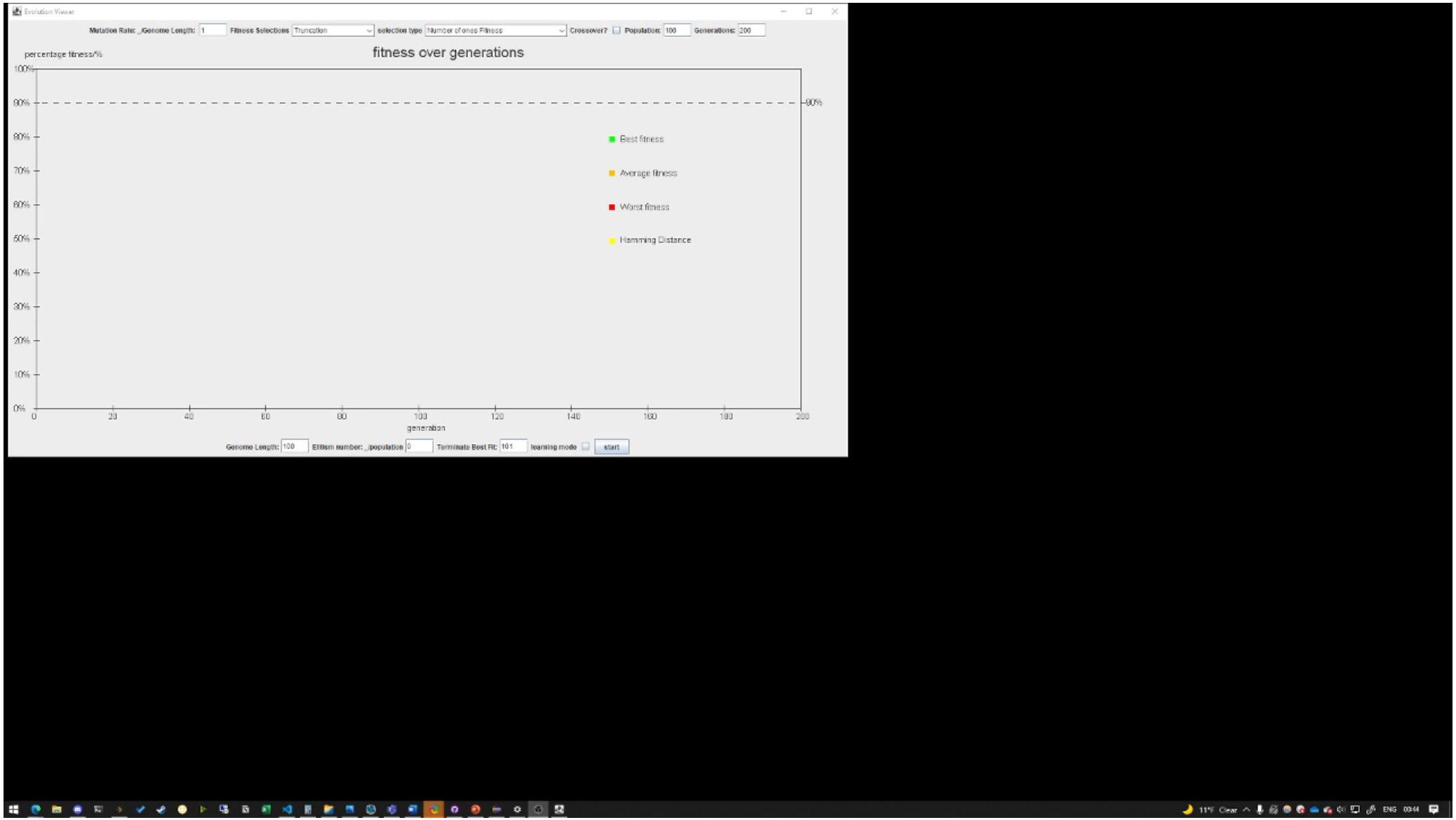
Object Oriented Design Principles
By being able to apply various fitness functions to the chromosomes these chromosomes can then tell their fitness which then allows the selection to take place (Design Principal 2b)

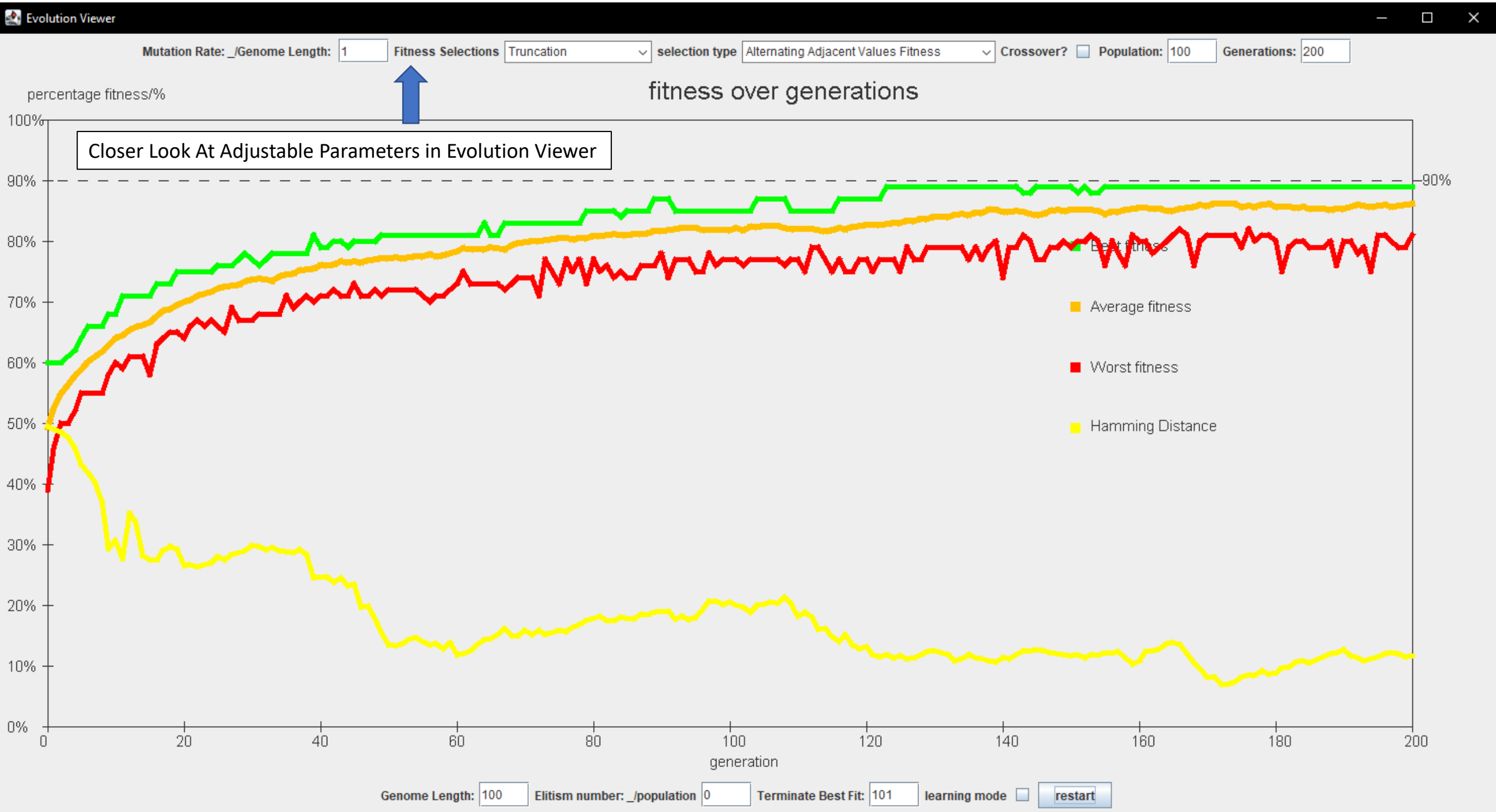
Alternating Adjacent Values Fitness Function

- This function looks at each one spots and then the next one to it and sees if they are alternating. If that is the case the fitness score is increased.



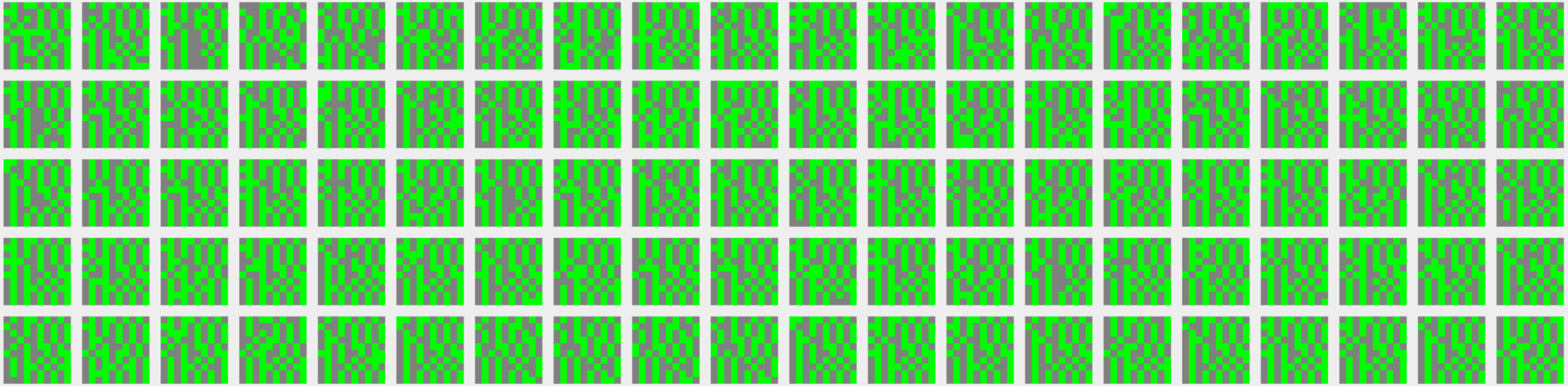
Alternating Adjacent Values Fitness Function Demo



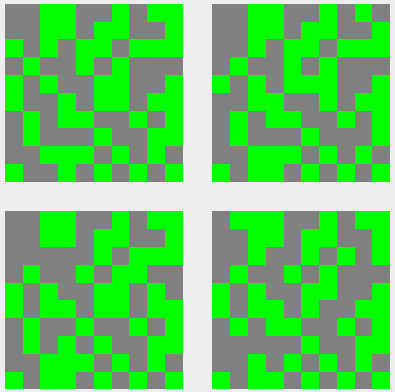

Evolution Viewer
Mutation Rate: _/Genome Length: 1
Fitness Selections Truncation
selection type Alternating Adjacent Values Fitness
Crossover?
Population: 100
Generations: 200
Closer Look At Adjustable Parameters in Evolution Viewer
fitness over generations
percentage fitness/%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
90%
Best fitness
Average fitness
Worst fitness
Hamming Distance
0
20
40
60
80
100
120
140
160
180
200
generation
Genome Length: 100
Elitism number: _/population 0
Terminate Best Fit: 101
learning mode
restart

Chromosome Viewer

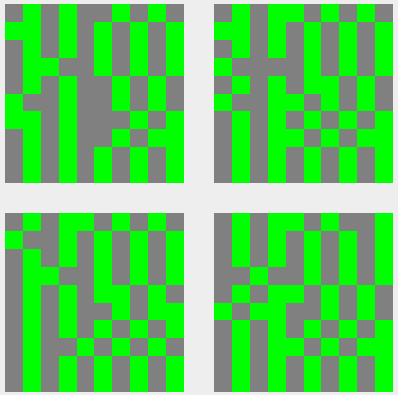
generation 201



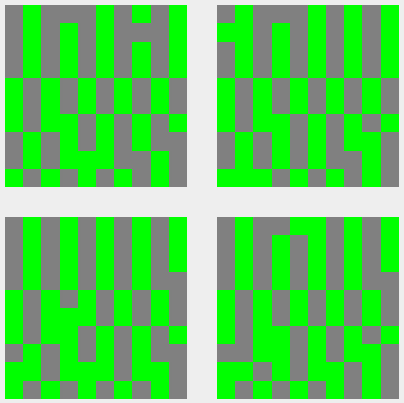
Generation 9



Generation 200

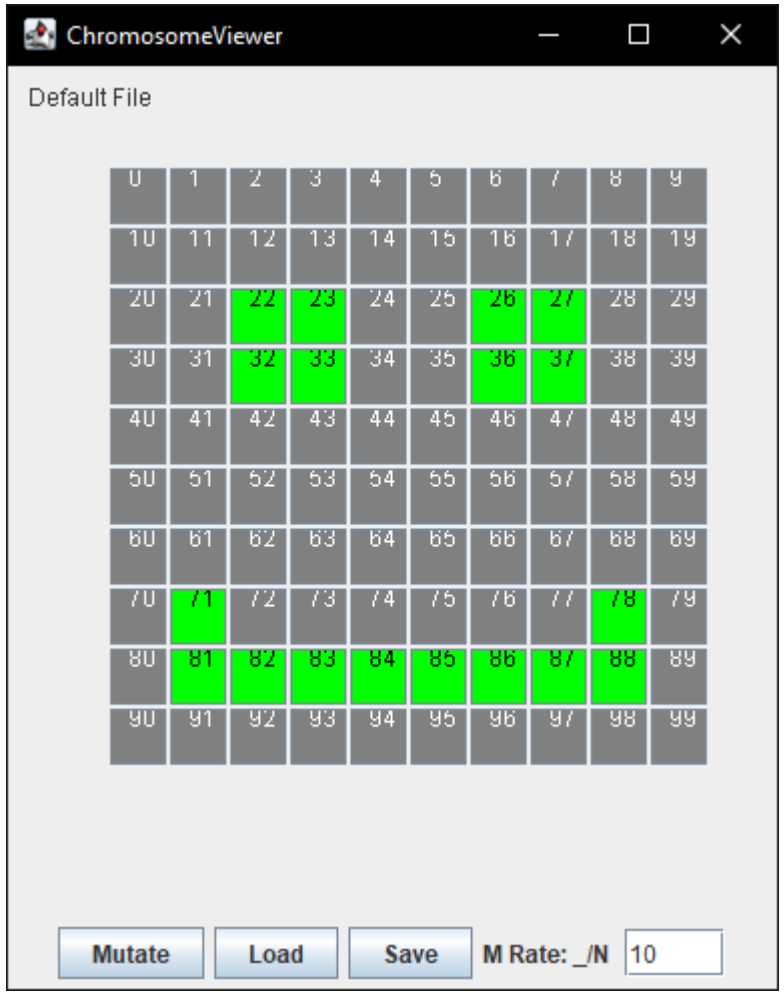


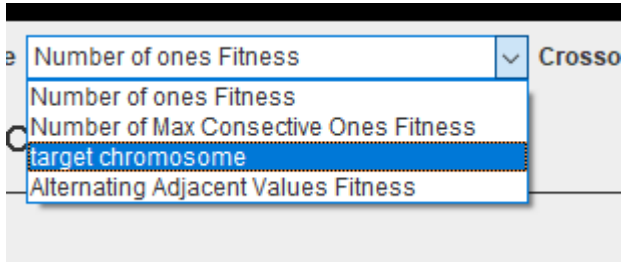
Generation 500

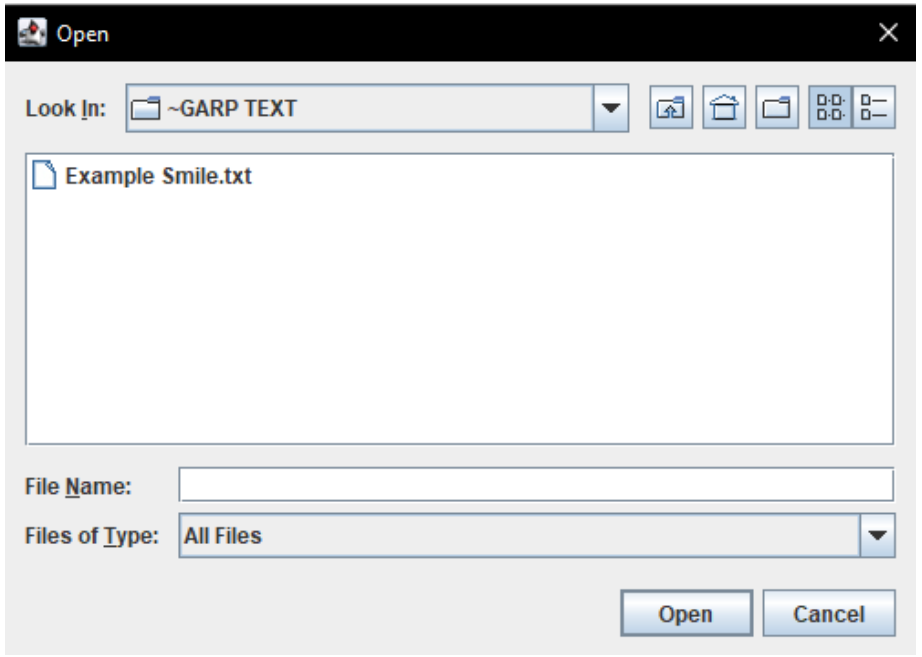


After running for 500 Generations this pattern becomes very apparent

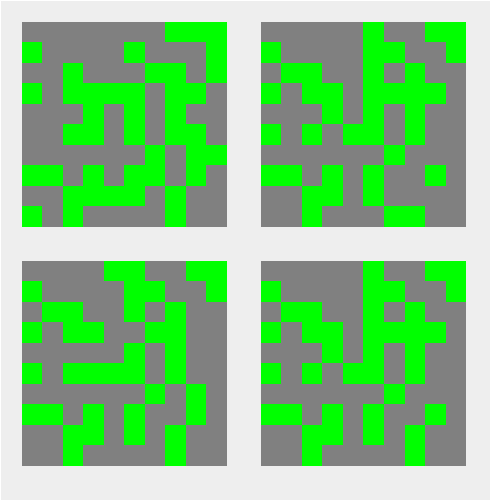
Target Chromosome Fitness Example



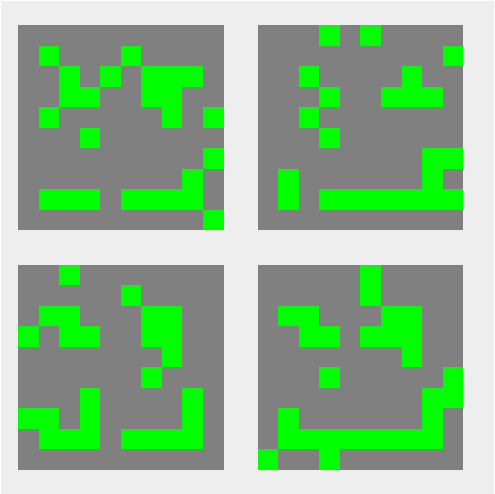




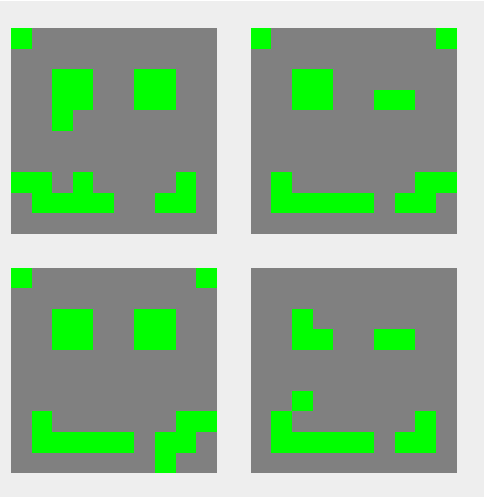
Gen 11



Gen 168



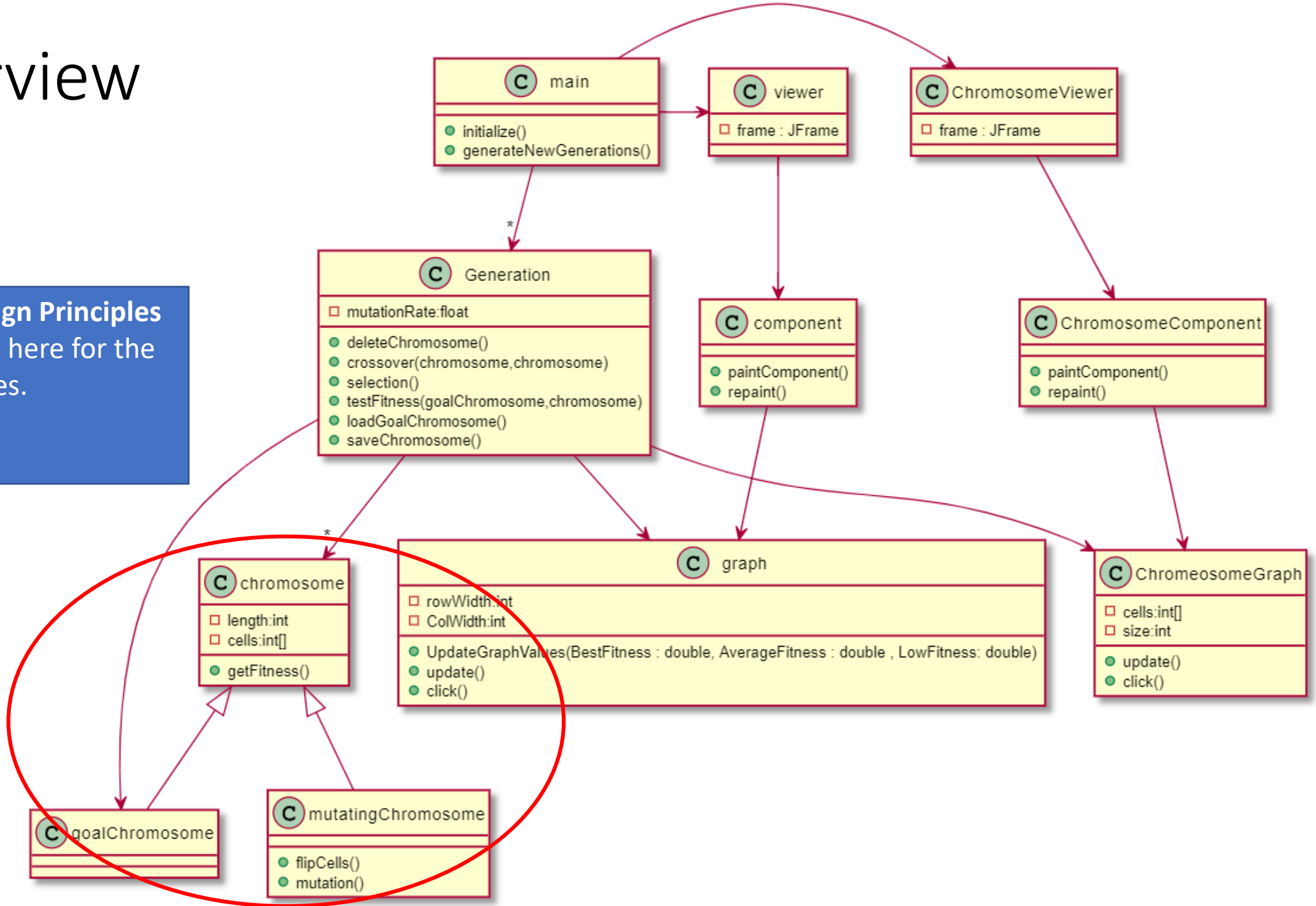
Gen 200



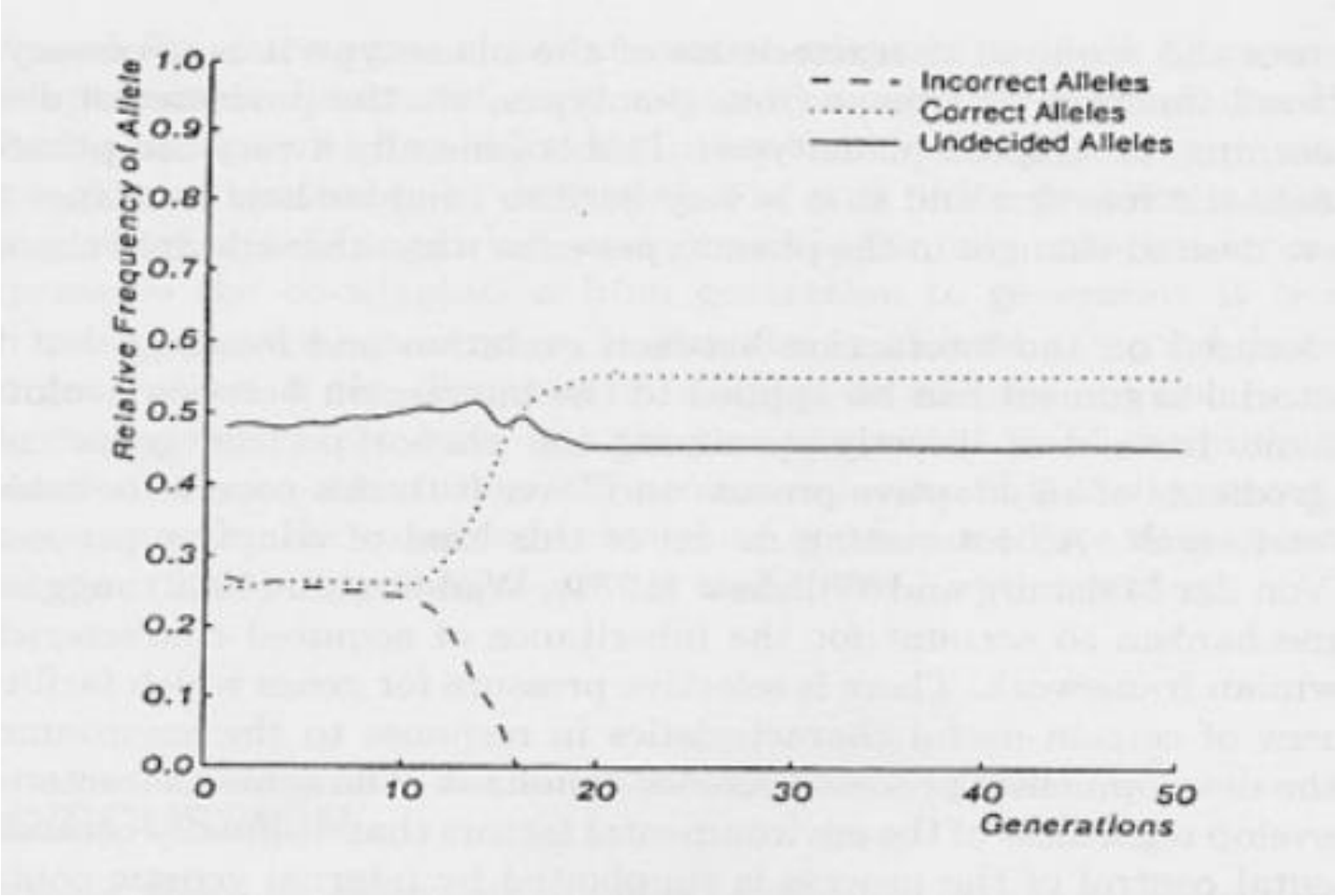
One can see these Generations are converging to a point. This convergence is the end condition of the simulator

UML Overview

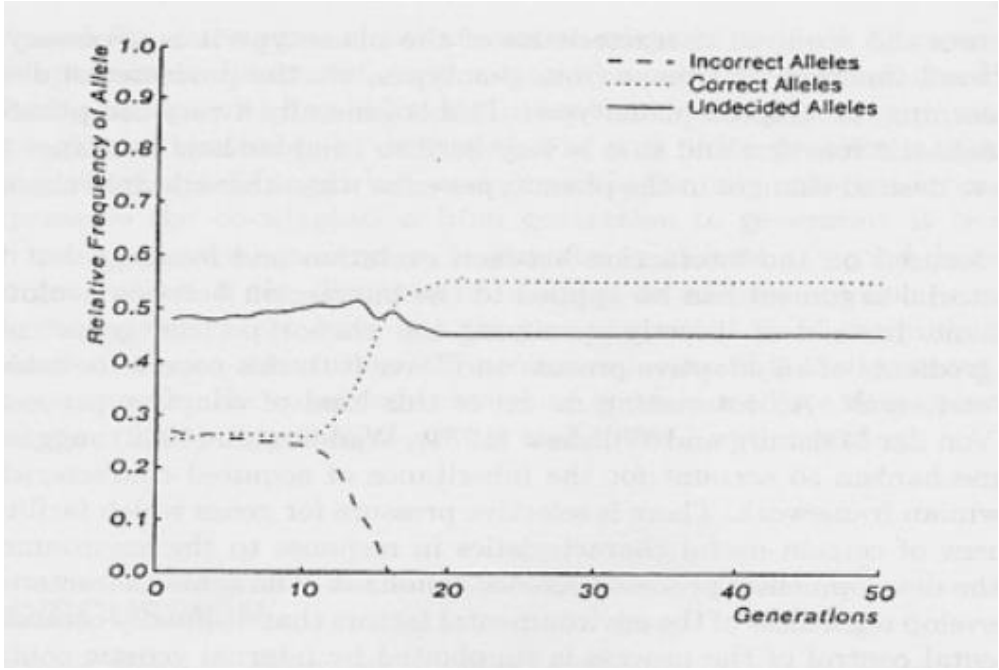
Object Oriented Design Principles
Inheritance was used here for the types of chromosomes.
(Design Principal 5d)

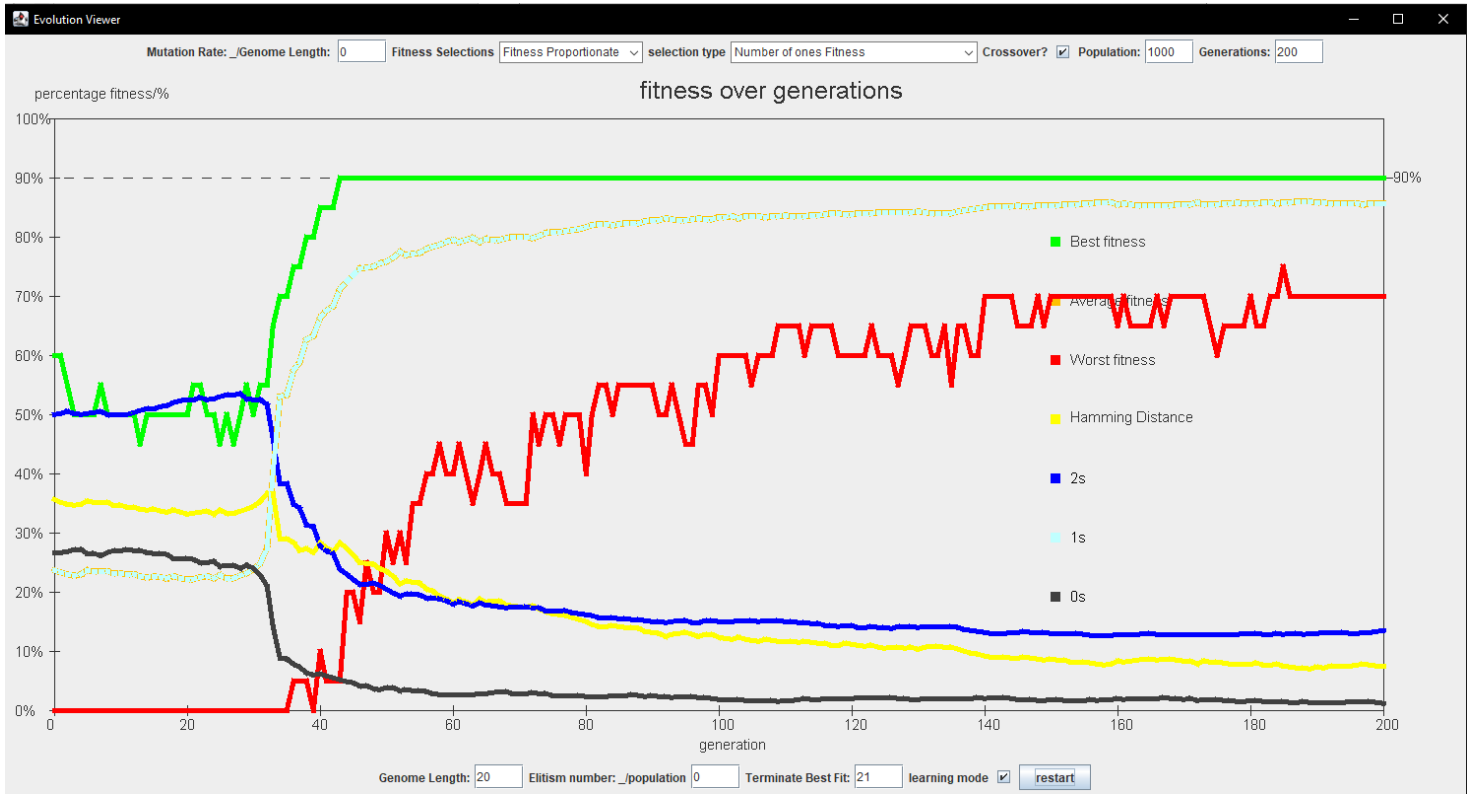


Scientific Paper Replication



Scientific Paper Replication

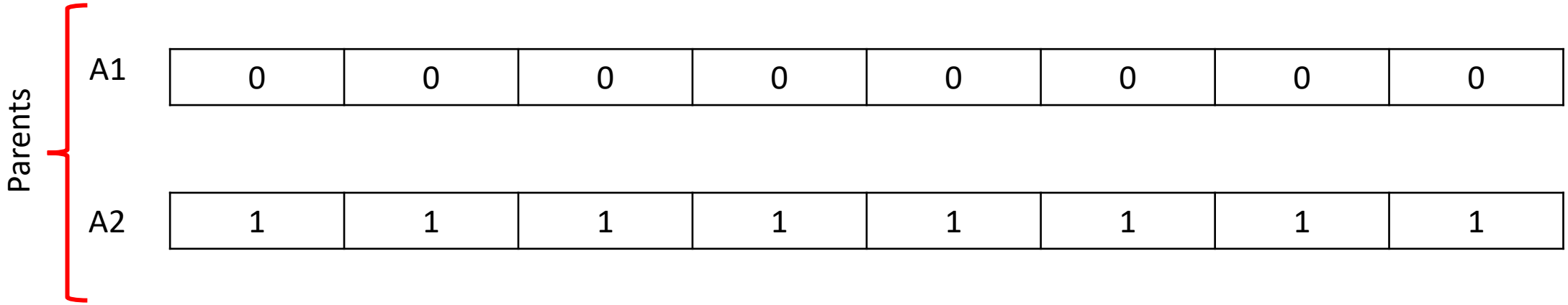




Any Questions?

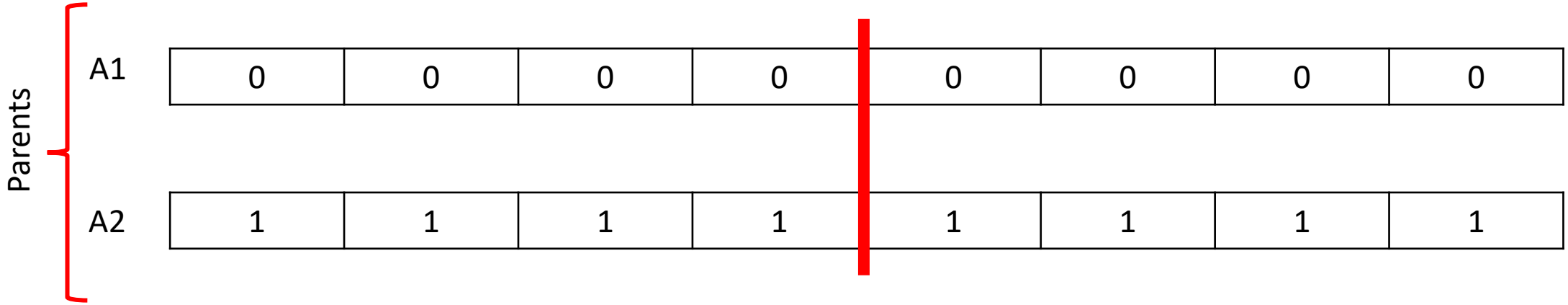
Crossover

- This is a point that shows how the parents of the individual will pass their genes to the off springs. This point is to be chosen at random



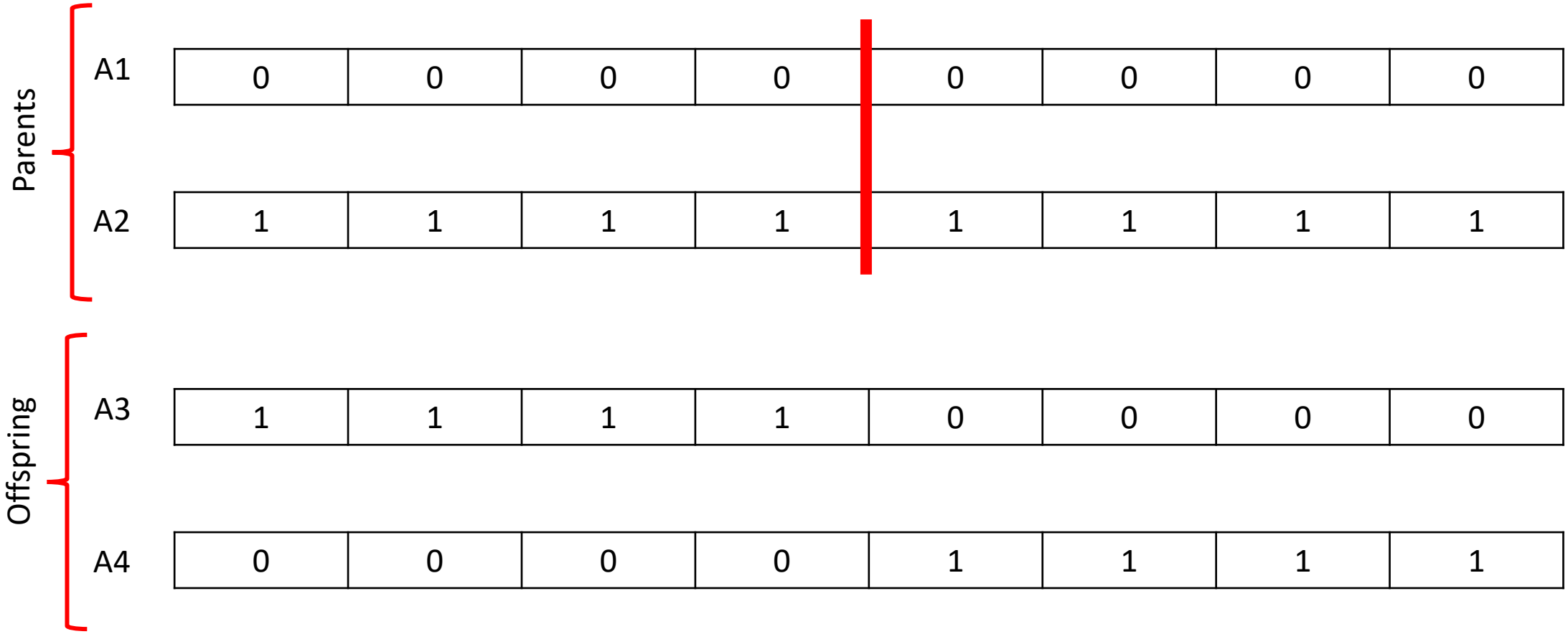
Crossover

- This is a point that shows how the parents of the individual will pass their genes to the off springs. This point is to be chosen at random



Crossover

- This is a point that shows how the parents of the individual will pass their genes to the off springs. This point is to be chosen at random



Crossover

- This is a point that shows how the parents of the individual will pass their genes to the off springs. This point is to be chosen at random