# (R)Evolution Simulator

Federico Malnati, Corso Sebastopoli 306, Torino (TO)
Matteo Palmieri, Strada Caldano 59/61, Caselle Torinese (TO)
Alessandro Sosso, Corso Vittorio Emanuele II 20, Torino (TO)

Liceo Scientifico Galileo Ferraris
Reference teachers: Andrea Doveri, Annalisa Gratteri

# Contents

**1 Introduction 3**

**2 Experimentsi 3**

2.1 System design . . . 3

2.1.1 The chunk . . . 3

2.1.2 The creature . . . 4

2.1.3 Graphical user interface . . . 5

2.2 Materials and methods . . . 5

2.3 Variables . . . 5

2.3.1 Variables of creatures . . . 5

2.3.2 Variables of chunks . . . 6

**3 Discussion 6**

3.1 Genetic characteristics and natural selection . . . 6

3.1.1 Numerical characteristics: agility, bigness, speed, fertility . . . 6

3.1.2 Mendelian-genetic characteristics: resistance to temperature . . . 6

3.1.3 Control-genes: Hardy-Weinberg principle and Genetic Drift . . . 7

3.2 Territory distribution of creatures considering the food . . . 7

3.3 Demographic trend . . . 7

3.3.1 Birth and death cycle . . . 7

**4 Conclusions 7**

**5 Thanks 8**

# 1 Introduction

In every field of research, the usage of computers is growing very fast, since they permit to simulate natural environments, in which it possible to analyse and observe particular events, in an easy way. Using software, spatial flights, astronomical motions and ecosystems can be simulated as videogames. Our project consists of a Python program simulating the processes of evolution of a specie in a virtual world. Each creature behaves as in the real world: they look for the most comfortable habitat, they feed and reproduce themselves. On the contrary if they are in an uncomfortable zone, they will probably die.

Looking at data, we can observe the mechanisms of population genetics, the large-scale transmission of genetic traits and the natural selection of creatures of a specie. The different traits (both numerical and Mendelian ones) and their behaviour will be analysed comparing them with the Hardy-Weinberg principle and the genetic drift.

Our creatures represent very well any specie of herbivores, whose simple behaviour, shows the sinusoidal trend of the population and the diversified distribution in the territory. The graphic interface, made with Python, permits to display everything in a very simple and efficient way.

# 2 Experimentsi

## 2.1 System design

The virtual land where our simulated creatures live has a rectangular shape and is divided into small square portions named *chunks*. The flow of time is measured in discrete units named *ticks*. Every time the clock ticks, the features of every creature and *chunk* are updated.

### 2.1.1 The chunk

Each chunk is characterized by: a position in the world; a total-quantity of producible food (between 0 and 100); a food-growth coefficient; a temperature, between -100 (cold) and + 100 (hot), that doesn't change over time.

When the program starts, each chunk is assigned an initial quantity of food and a temperature using the Perlin Noise function, which generates similar values for nearby *chunks* and unrelated values for *chunks* which are far away (the result is a more credible randomization, as can be seen in the Figure 1 in the Appendix).

Figure 2 shows two images representing the world without creatures: the first one from a climatic point of view (red zones are hotter, blue zones are cooler); the second one from the point of view of the land fertility (in dark green *chunks* there is more food, in the lighter ones there is less food).

We can observe that the temperature changes gradually from chunk to chunk, as well as the quantity of food, thanks to the Perlin Noise function quasi continuity.

However, when creatures reach a given chunk, its quantity of food starts reducing, as a consequence of their feeding.

If all the creatures die, the food starts growing again, going back to the situation in Figure 2.

### 2.1.2 The creature

The creatures, at each tick, calculate which is the best chunk to go to, in a range of 7 x 7 *chunks*, considering the quantity of food and the distance from their current position. Once they reach the chosen one, the creatures eat, increasing their energy level and decreasing the quantity of food in that chunk. At each tick, creatures also lose a fraction of their vital energy.

The sexual reproduction between two creatures takes place also in a range of 7 x 7 *chunks*. In particular, if two of them have enough energy, they reproduce, giving birth to a new creature that inherits a combination of parents' genetic characteristics with some mutations. During the reproduction, parents lose a part of their energy and give it the new creature. (Figure 3 shows the flowchart of creatures behaviour)

Creatures die for three reasons:

- Old-age: age limit is calculated at birth, based on a random number with a gaussian trend (the average is 1000 ticks and the standard deviation is 200 ticks)
- Temperature: each creature has its own genotype of the resistance to temperature (similar to fur length). In particular there are 3 different alleles: a dominant allele (N), and two recessive alleles (l and c), respectively fitted for lower and higher temperatures. Consequently the possible genotypes are homozygous (NN, ll, cc) or heterozygous (Nl, Nc, lc). NN, Nl, Nc and lc genotypes are fit for mild zones (coloured with white, Figure 2) and usually die in chunk with extreme temperatures; ll and cc genotypes respectively resist to lower temperatures and higher temperatures. At each tick, every creature has a determined probability of death based on its genotype and the temperature in its chunk.
- Starvation: if a creature cannot find enough food for a certain number of ticks, the energy drops under a minimum level and the creature dies.

The creatures are characterized by different genes, transmitted during the reproduction. There are three different types of genes:

1. Numerical genes: they are expressed by a number that represent the phenotype. During the reproduction, numerical genes of the new creature are calculated one by one multiplying the initial value with a random number (with a gaussian trend of average 1). They are:
   - $agility$, the moving attitude of a creature;
   - $bigness$, the size of a creature;
   - $fertility$, the number of minimum ticks that should pass between two consecutive reproductions, it is inversely proportional to the minimum energy level needed for a reproduction;
   - *num_control*, a numerical gene used as control-gene (it has no phenotypic effects);
2. Mendelian genes are characterized by a genotype made up of a couple of alleles, and a phenotype, calculated acoording to the laws of genetics. In the reproduction the two alleles of each gene are trasmitted choosing them one per parent. They are:
   - *temp_resist*, the ability of a creature to resist to a certain temperature; the alleles are c,l (recessive) and N (dominant);

- *mndl_control*, a control-gene without phenotypic effects, the alleles are a (recessive) and A (dominant);

3. The secondary genes are numerical genes, calculated on other numerical genes of a creature. They are:

    - *speed*, defined as $2 \cdot agility/bigness$.

### 2.1.3 Graphical user interface

In the graphical user interface, *chunks* are represented as small squares (Figure 2) and creatures as small circles. The dimension of circles may represent different features of the creatures. In Figure 5, the bigger the circles, the more energy the corresponding creatures have at that moment, because they have just eaten some food.

Also the colour of the creatures shows different features of them:

- cyan/yellow represent the two possible sexes;
- red/grey-white/blue represent the different phenotypes of temp-resist. In particular, red creatures correspond to c phenotype (fitted for hotter zones), blue creatures correspond to l phenotype (cooler zones), white and grey creatures correspond respectively to homozygous N phenotype and heterozygous N phenotype (fitted for mild portions);

## 2.2 Materials and methods

The program has been initially written with a Python extension for Processing, that permitted us to represent the world quite easily and to have an immediate feedback of our work. Then we readjusted it and translated to Python 3.6 language for a larger flexibility. We made different versions of the program, improving every time a different part of it (temperature, death for old-age, reproduction) and changing the many coefficients to balance the system.

Once we finished the program, we started collecting data using simulations of 10000 ticks in a $60 \times 45$ *chunks* world. In the graphical user interface each chunk is a square sized $10 \times 10$ pixels. All simulation data is saved and used for our following analysis.

## 2.3 Variables

We are going to observe the characters of each creature (speed, bigness, fur length, etc), analysing their variation, generation by generation, in different environments.

### 2.3.1 Variables of creatures

- *ID*, a progressive number that identifies each creature;
- *birth_tick*, the birth tick of a creature (for living creatures at the beginning of the simulation, this number is negative);
- *parents_ID*, the ID of the parents;
- *sex*, 0 or 1;
- *death_tick*, the tick in which the creature dies;

- *death_cause*, the cause of the death;
- *genes*, the characteristics of the creature;

#### 2.3.2 Variables of chunks

- *foodmax*, the maxium quantity of food of a chunk (between 0 and 100);
- *growth_coeff*, the growth rate of food (directly proportional to foodmax);
- *food*, the current quantity of food in a chunk; la quantitá di erba presente sul territorio
- *temperature*, the temperature in the chunk (between -100 and +100);

# 3 Discussion

## 3.1 Genetic characteristics and natural selection

### 3.1.1 Numerical characteristics: agility, bigness, speed, fertility

We saw that with time there is a natural selection between the creatures, and consequently their characteristics evolve (Figure 6): all numerical genes converge on a certain value, and the percentile distribution of values decreases.

From an evolutional point of view, creatures tend to develop their speed (faster creatures can reach more quickly food, even if it is more distant) and become at the same time bigger, to eat more food each time. Because of faster creatures have a greater survival ability, the speed-graph clearly increases.

On the contrary, the bigness-graph shows a relevant decrease at the beginning of the simulation. This is caused by the fact that bigger creatures are slower (as it can be seen in the equation: speed= agility/bigness) and so are disadvantaged. However, bigness values do not drop under a certain level, because bigger creatures can eat more food in the *chunks*.

Fertility generally increase with time, even if it represent the minimum number of ticks that there must be between two consecutive reproductions. This is explained by the fact that creatures with a greater value of fertility need less food to reproduce themselves, because this is inversely proportional to the gene itself.

### 3.1.2 Mendelian-genetic characteristics: resistance to temperature

We divided the territory into 8 classes of climatic frequency e we studied how many creatures were in each class, drawing the graphs for each temperature phenotype (c, l, N) (Figure 7). Each bracket is coloured with 8 different shades from red (hot climate) to blue (cold climate).

It has been observed that creatures with c phenotype, that are fitted for hot climate, usually placed in hotter chunk; on the contrary creatures with l phenotype placed in cooler ones. The others (N, n) lived in intermediate zones. Figure 8 shows the climate map of the territory and the disposition of coloured creatures coherently with their phenotype.

The division of creatures based on temperature can be considered a sort of allopatric speciation, because, even if in theory able to reproduce themselves, a creature with c phenotype and a creature with l phenotype cannot meet each other, because living in temperature-conflicting zones.

### 3.1.3 Control-genes: Hardy-Weinberg principle and Genetic Drift

A different discussion mast be done with the two genes that have a control function with no phonotypical effects.

The numerical gene $num_control$ has not a univocal behaviour (always increasing, decreasing or constant), but varies in each simulation [Figure 9].

Moreover there is no reduction of the percentile distribution as in other numerical genes. Considering the Mendelian gene $mndl_control$, it can be observed that its distribution in the territory is homogeneous on temperature varying, for both the two phenotypes, A dominant and a recessive, as in Figure 10. We also controlled if it respected the Hardy-Weinberg principle. The condition for this equilibrium are: no mutations, casual reproduction, a big population, no gene flow and absence of natural selection with that gene. We saw that this character does not respect this equilibrium, because of the small population (only 150 creatures in the world) and the unstable trend due to the random food consumption in some of the areas at certain ticks. We can say that this character is subjected to a genetic drift.

## 3.2 Territory distribution of creatures considering the food

As we did for temperature, we divided the territory into 8 classes of $food_max$ quantity in each chunk. We represent the graph with the number of creatures for each class over time. We coloured the brackets with 8 different shades: darker ones has less food, lighter ones has more. The graph in Figure 11 shows that creatures are generally in lighter brackets. Figure 12 shows the distribution of creatures on the territory in a certain tick.

## 3.3 Demographic trend

### 3.3.1 Birth and death cycle

The population graph follows a sinusoidal trend with a creatures increase of about 100 organisms. (Figure 13). There is a mechanism of auto-regulation that effects when there are less creatures than that world can sustain, causing many reproductions between creatures and an increase of population number. But, at this point, there are much more creatures than food availability, and we can observe a demographic fall. Then the cycle restarts.

# 4 Conclusions

We observed that our system acts like a natural ecosystem, in which genetic traits are subjected to natural selection through renewal and genetic mutation; in which there is a selection on the territory based on its natural characteristics; and in which is present a cycle of births and deaths comparable to the real world. We have thus proved that a system based on simple laws can provide a simulation of systems as complex as natural ones.

Our project allows to analyse a number of cases higher than what is observable in nature, in a lower amount of time. A 20000 tick, $60 \times 45$ *chunks* simulation allows to watch around 16000 creatures, something difficult to do in nature for the vast majority of species, unless through many generations.

The code, useful also for educational purposes for visualizing apparently abstract concepts, has a considerable flexibility and allows numerous expansions and developments: for example the insertion of a species of predators can be able to show the prey/predator cycle described by Lotka-Volterra equations; or the insertion of a contagious illness can highlight the dynamics of an epidemic and permit to find the minimum number of immune individuals to prevent the disease from spreading.

## 5 Thanks

We thank our teacher Riccardi, who held a Python course, for his creative and important suggestions about the code.

## References

[1] D. Shiffman: *The Nature of Code, simulating natural systems with Processing*

[2] David Sadava, H. Craig Heller, Gordon H. Orians, William K. Purves, David M. Hillis: *Biologia.blu, le basi molecolari della vita e dell'evoluzione*, Zanichelli

[3] *http://py.processing.org/reference/*

[4] Wikipedia, L'enciclopedia libera, *http://it.wikipedia.org/*

[5] *http://evolution.gs.washington.edu/pgbook/pgbook.pdf* , Joseph Felsenstein,"Theoretical evolutionary genetics"