

# Planisuss - Final exam project

## Computer programming Mod. 1

Alessandro Ghiotto

Università di Pavia

27 Settembre 2023



- 1 Introduction
- 2 Main structure
- 3 Visualization
- 4 A day on Planisuss
- 5 Conclusions

# Goal of the project

Design and implementation of a simulation of the **Planisuss world**

- object oriented programming;
- inheritance;
- provide data to be interactively visualized.

The **matplotlib** library

- the use of an external library;
- visualization of data;
- interactivity.

# Interactivity

## keypress events

- **"escape"**: close the simulation;
- **"spacebar"**: play/pause the simulation;
- **"r"**: visualize only the Carviz on the grid;
- **"g"**: visualize only the Erbast on the grid;
- **"b"**: visualize only the Vegetob on the grid;
- **"a"**: visualize the complete grid;
- **"u"**: update manually to the next day (only while pause is True).

# Interactivity

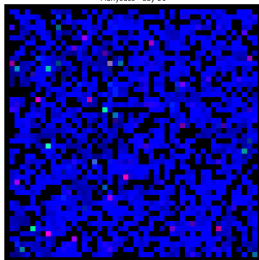
## Mouse events

"**click**" in the plot in which is visualized the world (the grid): select the cell in which zooming in and the cell chosen to be analysed in the table. You can click either in the complete grid, or in the zoomed one (for a more precise selection).

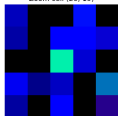
# Example of what we will visualize

x=19.87 y=28.06  
[0, 9.944, 9.675]

Planisuss - day 50

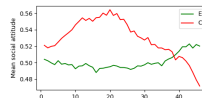
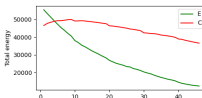
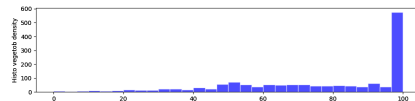
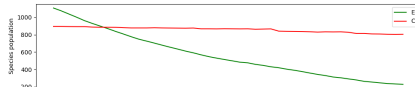


Zoom cell (20, 19)



Details cell (20, 19)

	Erbast	Carviz	Vegetob
t_population	17	0	35
t_energy	1151	0	0
m_lifetime	451.9	0	0
m_age	47.6	0	0
m_s.altitude	0.6	0	0



- 1 Introduction
- 2 Main structure
- 3 Visualization
- 4 A day on Planisuss
- 5 Conclusions

# Simulation

- **world**: the actual "Planisuss" world;
- useful variables for the methods, such as **pause**, **flag**, **zoom\_row** and **zoom\_col**;
- elements that constitute the figure, such as **GridSpec** and **axes** objects;
- the **FuncAnimation** object and the **cids**, in which we connect the interactive functions.



# World

- **grid**: a  $3 \times \text{NR} \times \text{NC}$  matrix, in each layer we have respectively the number of Vegetob, Erbast and Carviz;
- **herds** and **prides**: two dictionaries that have as keys the cells  $(i, j)$ , and as values a **herd** or **pride** object;
- **NR** and **NC**: number of rows and columns;
- **ground\_cells** and **n\_ground\_cells**: a list of the cells in which we don't have the water, and the len of this list;
- lists of data for the visualization of the plots, such as **time\_data**, **erbast\_population\_data**, **vegetob\_density\_data** and **erbast\_energy\_data**.

## Group: Herd and Pride

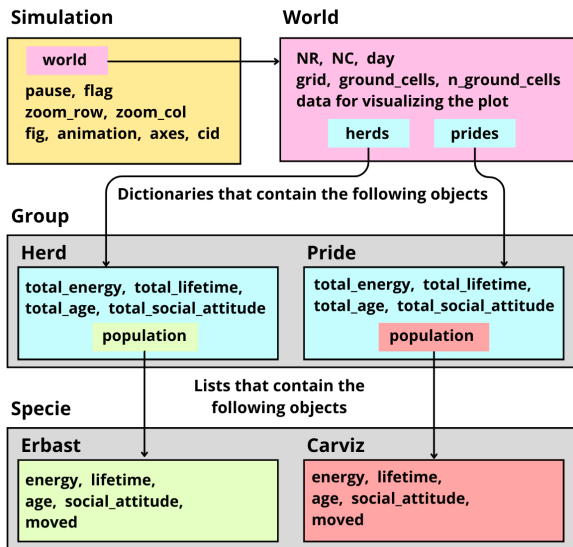
Herd and Pride are children class of the Group class.

- **population**: a list of **erbast** or **carviz**, that inhabit the cell in which is present the herd/pride;
- collections of useful data: **total\_energy**, **total\_lifetime**, **total\_age** and **total\_social\_attitude**.

## Specie: Erbast and Caviz

Erbast and Carviz are children class of the Specie class, and they are the animals that populate the Planisuss world.

- **energy**: the strength of the individual;
- **lifetime**: duration of the life of the animal in days;
- **age**: number of days from birth;
- **social\_attitude**: the likelihood of an individual to join to a group;
- **moved**: True if the animal moved today, False otherwise.



- 1 Introduction
- 2 Main structure
- 3 Visualization**
- 4 A day on Planisuss
- 5 Conclusions

# Figure

Thanks to the GridSpec objects we can divide in several parts the figure and create a complex layout.

## gs1 on the left

Here we have the grid, the zoomed grid and the table regarding the data of the selected cell.

## gs2 on the right

Here we have plots for visualizing the total population, the average age and lifetime, the total energy, the mean social attitude and a histogram that describes the Vegetob density.

# Update function

The **update** function is executed at each frame of the FuncAnimation object and is formed by:

- **world.a\_day\_on\_planisuss()**: update the Planisuss world;
- **create\_plots()**: update all the plots that are visualized;
- **display()**: display the new figure.

- 1 Introduction
- 2 Main structure
- 3 Visualization
- 4 A day on Planisuss**
- 5 Conclusions



## `__init__()`

Initialize a random grid. Then we create a random number of herds and prides with a random number of Erbast and Carviz with random stats.

## `growing()`

We update the Vegetob density by increasing each `ground_cell` by `const.GROWING`.

## `overwhelming()`

Eliminate the animals situated in a cell that is surrounded by cells with 100 of Vegetob density.

## movement()

For every group, the best cell is identified, the group decide if they want to move, and every animal decide if they want to follow or separate from the group. As result we will have list of groups, instead of a single one.

## unify\_groups()

We join the herds and prides that are present in the same cell. The herds are simply unified, instead the prides decide if they want to join or to fight.

## grazing()

Each Erbast obatin 1 point of Energy for 1 point of Vegetob density.

## hunting()

The pride identifies the stronger Erbast in the cell and combat with him. The Energy of the prey is shared by the pride individuals.

## spawning()

We update the age, decrease the energy (every ten days), and when the age reach the lifetime we spawn the offsprings.

## remove\_empty\_groups()

We check if there is a group with 0 animals and we eliminate it (useful to avoid dividing by zero).

## update\_stats()

Update the attributes `total_energy`, `total_lifetime`, `total_age` and `total_social_attitude` of each group and we update the attributes of the world object that works as data for the plot.

## neighbourhood(self, cell\_row, cell\_col, radius)

An auxiliary method that gives as result the list of the cells neighbours of my cell with coordinates `(cell_row, cell_col)`, within the distance radius.

- 1 Introduction
- 2 Main structure
- 3 Visualization
- 4 A day on Planisuss
- 5 Conclusions**

What could be improved:

- the decision functions of each phase of the day, that create a not very balanced simulation between the species;
- the trajectories of each herd and pride;
- time complexity and overall performance.

Thanks for the attention