Last modification: January 22, 2023

RBE595/CS525: Swarm Intelligence
Spring-Term 2022/2023
Homework 3

Exercise 1 (The Schelling Model) [100 points]

Make program that performs a simulation of the Schelling Model. You can use any programming language as long as it's Python.

- The world is a 50×50 grid, for a total of $G = 2500$ cells.
- The population P is expressed as a fraction of the cells. Consider $P = \{0.6, 0.8\}$.
- The agents are always 50% `X` and 50% `O`.
- The satisfaction threshold is set to $t = \{3, 4, 5\}$. Specifically:
  - Run 10 simulations with all the agents having $t = 3$.
  - Run 10 simulations with all the agents having $t = 4$.
  - Run 10 simulations with 80% of the agents of each type having $t = 3$ and the rest $t = 5$.
- Agents are updated according to the cell they occupy, left-to-right, top-to-bottom. Start at the top-left corner in the grid, move left-to-right along the first row, and update all the agents you encounter. Once done with the row, move down to the leftmost cell of the second row, and repeat the above steps.
- Unsatisfied agents move to the closest cell that makes them satisfied. Use 8-distance to find the closest cell.
- You can run the simulation for as many steps as you see fit. A good value is $1,000$ steps, but you might want to try more for some configurations and less for others.

If you find online sources for your code, **cite them in your report**. We know they exist!

Deliverables

- A PDF document in which you report, for each combination of the parameters P and t:
  - 4 intermediate grid printouts from your code (selected among the 10 runs by you to discuss interesting phenomena you see) and the final grid;
  - A discussion of how fast segregation is reached;
  - Any interesting experiments you tested that you wish to mention.
- The complete code in an easily executable form, along with instructions to execute it in a README file. It is useful to include language specific information (Ex. Python 3.7). Also please include any outside libaries used with these languages (Ex. Matplotlib, Seaborn). A 'requirements.txt' file is useful for Python but not required.

Deliverables

Create an archive called `LastnameFirstname.zip` structured as follows:

```
LastnameFirstname/
   <your code files>
   README (a *pure text* file that explains how to run the code)
   report.pdf (please include your name at the top)
```

Make sure to **follow these guidelines accurately**: any mistake will entail a **10% penalty** on the final score. Example mistakes: the files are not named as requested (no, we won't accept `docx` files); the archive is not a `zip`, but some other format; the structure in the archive is incorrect...