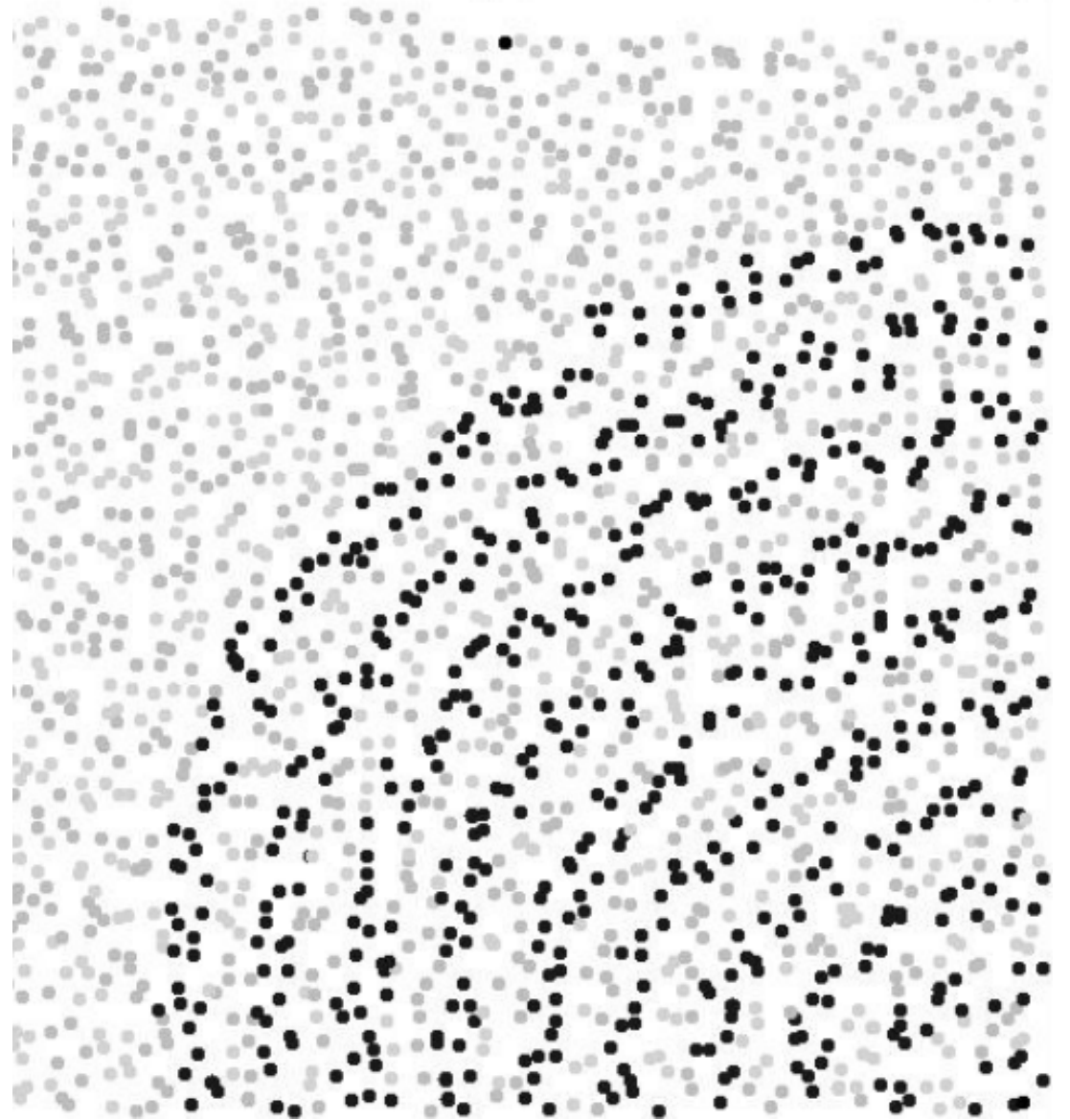# Lab 2

- Due Oct 28 noon

- Coordinate system based on hopcount

- Based on paper: "Organizing a global coordinate system from local information on an ad hoc sensor network."

- Must work by yourself, NO partners

Academic honesty and plagiarism

- Plagiarism is unacceptable.
  - Code you submit must be your own. No copying, adapting, or submitting code you did not create is allowed. Working together and presenting variants of the same file is not acceptable. Here are some specific guidelines to make sure you don't cross the line:
    - Do not exchange programs or program fragments in any form on paper, via e-mail, photos, or by other means.
    - Do not copy solutions from any source, including the web or previous quarters' students.
    - Do not discuss code with other students at the level of detail that will lead to identical programs or program fragments.
  - Ask me if you are uncertain.

Goal

- Display the Northwestern "N" in 256 robots
  - Display by adjusting robot's color (no need to match exact color of "N") based on coordinates
  - No robot motion.
- Use standard and "smoothing" approach for coordinates
- Seed robots in bottom corners
  - Given unique id 1,2
  - All other robots given id 0

Simulation updates

- You **<u>cannot</u>** use in submitted code:
  - robot.get_pose()
  - robot.id
  - robot.set_vel()
- You can use new variable for robot ids
  - robot.assigned_id

- Use new code updated to canvas files, lab 2
  - New start commands, and initialization code.

Hints:

- Break problem into manageable pieces and check each step
  - Generate 1 hopcount
    - Use modulus 2 of the hopcount to display color for debug
  - Generate 2 hopcounts
    - Use modulus 2 of the hopcount to display colors for debug, with one hop count for one color, and the other hopcount for the other color.

  - Generate coordinates for all robots
    - Again use color to debug, for example red if x>constant, off in x<constant
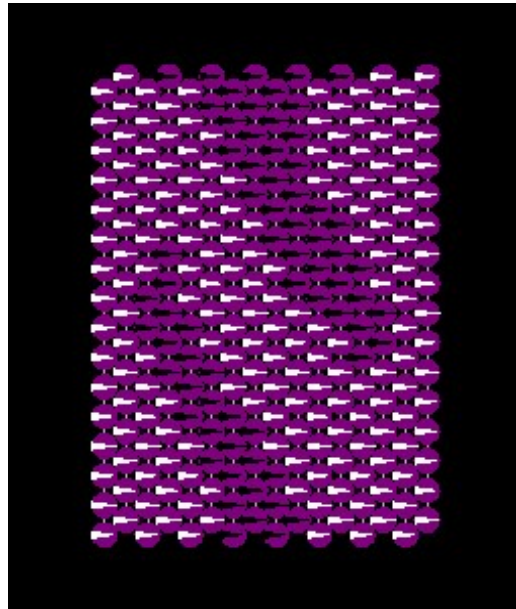 - Display shape on all robots.

Submit the following

- Image (.jpg) of the swarm displaying "N" for:
    - regular
    - smoothed
- Well commented usr_code.py

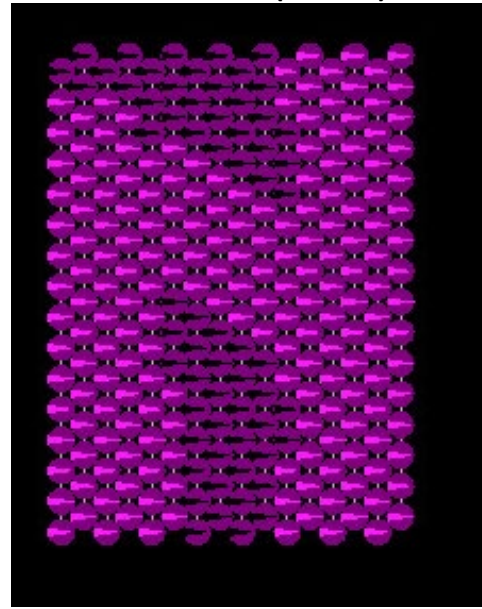# "N" quality for best scenario

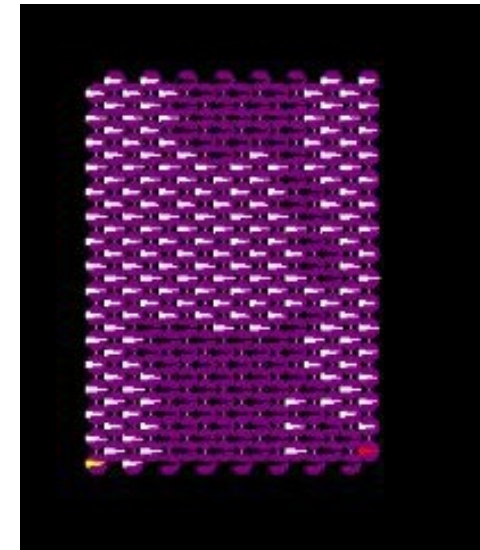10% extra credit for "best" N
5% for second best N

Extra credit winner 2020

Good-ish (9/10)

Not ok (5/10)

Bad (0/10)