**Worksheet 5 Report Analysis**  
**Hazem Elknany – 21010790**

**Task: Boid Flocking Simulation Code**

**Objective**

This will be a simulation of the emergent flocking behavior of a group of agents, the boids, moving in 2-D space. Each individual boid reacts to its neighbors by simple rules of movement. The pattern that emerges from such individual behaviors by agents is known as emergent collective behavior and simulates the real-world flocking phenomena in birds, fish, or whatever form of group dynamics.

**Code Explanation**

* Initialization: It instantiates num\_boids = 20 boids, and sets each with:
  + An arbitrary position in the 200x200 grid.
  + A random speed whose maximum is 2 units.
* Behavioral Controls: This motion of each boid is influenced by three behavioral rules:
  + Separation:
    - Boids avoid crowding by repelling from neighbours which are too near.
    - It is directly proportional to the number of other boids within half a radius of perception from it.
    - Separation, weighted by separation\_weight, 1.5.
  + Align
    - Boids move to align their velocity with the average of neighboring boids, within the perception radius of 20 units.
    - This helps ensure alignment within the group and smooth course corrections.
    - Weighted by alignment\_weight 1.0.
  + Coherence:
    - Likewise, Boids are attracted to the center of mass of their neighbours inside the perception\_radius.
    - Weighted by cohesion\_weight 1.0.
* Boundary conditions
  + The continuous boundaries are implemented using the modulus operator (% grid\_size).
  + Anything that exits on one side of the grid goes to the other side.
* Speed limitation
  + Boid velocities are capped at max\_speed (2 units) to ensure controlled movement and prevent erratic behavior.
* Visualization
  + Simulation relies on matplotlib's FuncAnimation to display animated movement of the boids.
  + Boids are represented as blue circular markers in a 2-D graph.
  + This continuously updates the positions on the grid for every frame, exposing emergent flocking over some time.

**Findings from the Simulation**

* Influence of Rule Weights
  + Separation weight:
    - Larger values disperse the flocks and the flocks are less cohesive.
    - Smaller values yield crowding and the simulation appears to be not very realistic.
  + Alignment Weight:
    - Higher values assure much easier direction changes in formations.
    - Smaller values create very discrete movements with little coordination.
  + Cohesion Weight:
    - Higher values are used for tighter clusters, while for loosely connected groups, lower values are used.
* Number of Boids
  + Increasing the number of boids will give greater, more complex flocking.
  + There are far smaller and less cohesive groups with fewer boids.
* Boundary Conditions
  + It uses a wraparound boundary so as not to interfere with the flow of movement in the flocking.
  + Other boundary conditions, such as bouncing, would lead to unnatural behavior around grid edges.

**Future improvements**

* Dynamic Perception Radius: Allow the perception radius to change with the velocity of the boid, or other property.
* Obstacle Avoidance: Obstacles can be thrown into the grid to simulate more complicated environments.
* Improved visualization:
  + Use directional markers (e.g., arrows) to show boid velocities.
  + Visualize neighbor interactions. For instance, draw lines between boids in the perception radius.

**Conclusion**

This code effectively simulates flocking behavior using simple rules. It demonstrates how individual agents following local rules can create complex and realistic group dynamics. The adjustable parameters (rule weights, number of boids, etc.) allow for extensive experimentation and provide insights into how collective behavior emerges in natural systems.