Spike: Task 11

Title: Emergent Group Behaviour

Author: Sam Huffer, 101633177

# Goals / Deliverables

* A group agent steering behaviour simulation that can demonstrate several modes of emergent group behaviour. It must:
  + Include cohesion, separation and alignment steering behaviours.
  + Include basic wandering behaviours.
  + Use weighted sum to combine all steering behaviours.
  + Support the adjustment of parameters for steering forces while the code is running.
* As extensions, consider and demonstrate:
  + What happens when agents can’t overlap and how does this affect the parameter settings discovered earlier in this task?
  + Adding a predator that other agents avoid.
  + Adding walls for agents to avoid, which will necessitate implementing feelers or another wall avoidance method. Consider what properties are needed to get agents circling around the game space.
  + Create different agent group types and investigate the behaviours that emerge between those groups.

# Technologies, Tools, and Resources Used

* Command prompt (for executing and testing the code)
* SublimeText (for editing code)
* Learning materials on Canvas (for instructions and sample code)

# Tasks Undertaken

To set up the project for this task, I copied the code used for Task 10: Tactical Steering to use as the basis for this task. I commented out several methods that, on a read through of the code required for the spike, seemed like they would not be useful at all, and changed the set up to generate one hunter, and allow the creation of prey agents with key input. I also shrank the prey agents to allow many more on the screen.

Next, I changed the key input logic to allow or disable various sets of key input according to keystrokes, and to allow the toggling of the display of various pieces of information pertaining to the prey agents (forces, wandering, avoidance etc.). I also updated the code so that instructions for keyboard input would be displayed on-screen, and that when prey agents existed, the values of their motion variables or associated weightings would be displayed on-screen, and would be changeable with keyboard input.

After that, I implemented the walls and code to agents allow them to avoid them, and to objects to prevent them spawning outside the wall margins.

Once the walls were set up, I implemented the code for the group behaviour methods, adapting the pseudocode in the lecture notes for this, and added a method to the world class to calculate all agents’ neighbours so they wouldn’t have to all do that themselves and double up on the work.

# Code Snippets

# Instructions for Operating the Code

* Key Backspace: exit current menu
* Key Escape: close the simulation
* Key A: open prey spawning menu
  + Key [N]: spawn [N] prey agents in random, valid positions
* Key I: open display info menu
  + Key A: toggle displaying of avoidance boundaries
  + Key F: toggle displaying of the force acting on the agent
  + Key N: toggle displaying of prey agents’ neighbourhoods
  + Key R: toggle displaying of agents’ radii
  + Key W: toggle displaying of wander circles
* Key O: open the obstacle menu
  + Key N: spawn new obstacle
  + Key R: randomise the positions of the obstacles
* Key P: toggle whether the simulation is paused or un-paused
* Key V: open the variable editing menu
  + Key Up: move selection up
  + Key Down: move selection down
  + Key Left: decrease the selected variable
  + Key Right: increase the selected variable
  + Key Plus/Equals: increase the increment step
  + Key Minus/Underscore: decrease the increment step

# In-Game Screenshot

# What I Found Out