PRINCIPLE:
SELF-ORGANIZATION

Self-organization is a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components. 

- Individual ants, bees, termites, and robots interact locally… yet a global pattern emerges

SO in Non-Biological Systems

Self-Organization can be observed in non-biological systems, for example in the formation of Bénard convection cells (left) or regularly space ridges in dunes (right).

The difference in biological systems is that the interacting components in biology are typically much more complex than in non-biological systems (oil molecules, sand grains). Also, physical laws are in effect for both, but biological systems also adhere to behaviours that are learned and/or genetically programmed.
Mechanisms for SO

SO is supported by the following mechanisms:

Positive feedback (amplification) Mechanisms that encourage certain quantities or patterns to grow. Recruitment of other members of the swarm to join in some activity is an example.

Negative feedback (inhibition) Mechanisms that encourage certain quantities or patterns to shrink. Negative feedback can be explicit such as when one agent causes another to inhibit their behaviour, or it can be an implicit result of resource limits.

Multiple interactions The global structure or pattern arises over time through multiple interactions between components/agents.

We will see instances of these mechanisms in the following examples...

Example: Honeybee Foraging

Honeybees gather nectar from flowers then return to the hive, give up the nectar to another bee. The bee will then do one of the following:
- Perform a ‘waggle dance’ (see right) indicating the direction and distance of the nectar source which tends to recruit other bees to that source.
- Continue to forage from her previous source without dancing.
- Abandon her previous source and follow another bee’s waggle dance, leading her to that source.

This schematic illustrates the choices ($c_1$ and $c_2$) open to a bee returning to the hive from a food source.

These choices are influenced by the perceived quality of the two food sources. Bees returning from high-quality food sources have a higher probability of dancing to support that source.

Performing the waggle dance to lead other bees to the same food source is an example of positive feedback that amplifies the selection of a large food source over a small one.

In this experiment food source B is more plentiful in the morning (8:00-12:00) but food source A is more plentiful in the afternoon (12:00-16:00). Consequently, B attracts more bees in the morning and A attracts more in the afternoon.
Biologists have noticed many fascinating examples of social insects organizing their environments.

- Colonies of ants will cluster dead ants together, seemingly without any planning or supervision.

A computational model was proposed to explain this behaviour:


Deneubourg et al’s model:

- Agents measure local object density by maintaining a short-term memory and counting the number of recent object appearances.
- Agents walk randomly and pick-up or deposit objects as a probabilistic function of local object density.

<table>
<thead>
<tr>
<th>Density</th>
<th>Pick-up Probability (if not carrying)</th>
<th>Deposit Probability (if carrying)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

Local object density (cartoon version)

\[ P_{\text{pick}} = \left( \frac{k_1}{k_1 + \text{density}} \right)^2 \]

\[ P_{\text{deposit}} = \left( \frac{\text{density}}{k_2 + \text{density}} \right)^2 \]
Experiments on the computational model closely match the results of biological experiments. But we have to be careful: similarity of results does not necessarily imply similarity of method.

Beckers et al. wrote a paper detailing their experiments in swarm robotic clustering. Unlike the [Deneubourg et al., 1990] model, the robot’s pick-up / deposit behaviour is implicit.

C-shaped gripper passively collects pucks. Infrared sensors detect obstacles (walls, other robots). Behaviour: Triggers random turn away from obstacle. Force sensor detects that gripper is pushing against three or more pucks. Behaviour: Triggers backup, then a random turn, resulting in the pucks being left behind (i.e. deposited). If no behaviour is triggered, the robot just moves straight.

Unlike Deneubourg et al. there is no explicit sensing of local object density. Yet when a high density area is encountered, the robot tends to further increase density by backing up and leaving its pucks behind.

Figure 1: Robot equipped with a gripper for object gathering. Experiments were carried out with from 1 to 5 robots of the same type.
The next task will be to consider a number of different possible sensor states and decide on reactions for each one.

We'll have to consider different numbers of pucks detected by the robot and decide whether they represent low or high density.

For simplicity, let's say that a single puck is low-density and two or more pucks is high-density.

Fill in the blanks with the choices the robot should make; e.g.

- Turn right
- Turn right
- Turn left
- Turn left
- Straight
- Straight
- Pick-up
- Turn away
- Deposit, then turn away

Why does this work?

The other behaviours are necessary but only these two have a direct impact.

Larger clusters attract more deposits than smaller clusters, leading to further growth and the gradual absorption of smaller clusters.

EXERCISE #6 / A6, TASK 2
OBJECT CLUSTERING

- Create a new controller that has the appropriate response for all of the conditions just described
- Beneath the simulation controls is a new plot called “Percentage Completion”
  - Details:
    - Computes the size of the largest cluster. Let this size be \( L \)
    - \( PC = 100 \times \frac{L}{\text{number of pucks}} \)
- CHALLENGES:
  - With 10 robots and 30 pucks see how fast you can reach 60% completion

OBJECT SORTING

- Sorting objects is a natural extension of clustering
- Clustering: One object type
- Sorting: Two object types
- It has potential applications in recycling, mining, and warehousing

SORTING IN WAGGLE

- Go to the main Waggle page (or just hit your browser’s back button):
  - http://bots.cs.mun.ca/waggle/
- Select the Sorting level. Note that there are now two different colours of pucks available: red and green
- We will use the same sensor and action blocks as before, but will need to customize the “___ puck held” and “Number _____ pucks” blocks:
• When sorting two colors, we have more choices to make
• Can you fill in the blanks?

Create a new controller that has the appropriate response for all of the conditions just described

CHALLENGE:
• With 10 robots, 30 red pucks, and 30 green pucks, reach 50% completion within 200 seconds

EXERCISE #7 / A6, TASK 3:
OBJECT SORTING

Conditions:
Any number of robots and pucks