Modelling and simulation

Use the credentials below to logon the computer. If it says that someone else is already logged in, you can just click on “**Log in as another user**”.

Username: ohvd

Password: Exeter2020

Let us now open the Terminal application and get the source files from a Github repository using git

cd Desktop <Enter>

git clone <https://github.com/hsbarbosa/ohvd-modelling> <Enter>

A screenshot of a cell phone

Description automatically generated

Now we will load a module to make the agent-based modelling platform NetLogo available for us.

module load netlogo <Enter>

NetLogo <Enter>

You should see something like this:

A screenshot of a cell phone

Description automatically generated

Activity 1 – Flocking behaviour

In this activity we will try to mimic the flocking behaviours of animals (e.g., birds, fishes, insects). The formations that might appear in this model are not created or led in any way by special leader. Rather, each **agent** is following exactly the same set of rules, from which flocks emerge.

Now let us open the Flocking model.

File -> Open -> Desktop -> ohvd-modelling -> Flocking.nlogo

# How it works?

The **agents** follow three rules: “**alignment**”, “**separation**”, and “**cohesion**”.

* Alignment - means that an agent tends to **turn** so that it is moving in the **same direction** that nearby agents are moving.
* Separation - means that an **agent** will turn to **avoid another one** which gets too close.
* Cohesion - means that an agent will **move towards other nearby agents** (unless another one is too close).

1. When two agents are too close, the “separation” rule overrides the other two, which are deactivated until the minimum separation is achieved.
2. The three rules affect only the agent’s heading. Each agent always moves forward at the same constant speed.
3. Agents have a limited vision, which means that they can only *see* within limited distance **around them.**

# What to test?

## Experiment 1

* Set
  + All slides to 0
  + Press setup and go
  + max-cohere-turn =20
  + increase vision in steps of 0.5 from 0 to 4.5
* What do you observe?
* What happens when you change vision back to 3.5?

## Experiment 2

* Set
  + All slides to 0
  + vision = 10
  + max-cohere-turn to 5, 10, 15, 20
  + Set it back again to 10
* What happens to the sizes of the flocks?
* What animal do you think this behaviour looks like?

<Swap seats>

## Experiment 3

* Find if one rule by itself enough to produce some kind of flocking.
* What is the most important parameter in your opinion for the emergence of flocking?

## Experiment 4

* Set
  + vision=5
  + minimum-separation=1
  + max-align-turn=5
  + max-cohere-turn=3
  + max-separate-turn=1.5
* What is the difference between this flocking behaviour and the one in Experiment 1?
* How is the travel direction defined?
* What if you now set minimum-separation=0

# Discussion

How these behaviours emerge in nature?

Activity 2 – Natural selection

Now let us open the Bug Hunt Camouflage NetLogo model.

File -> Open -> Desktop -> ohvd-modelling -> Bug Hunt Camouflage.nlogo

In this activity we will explore how natural selection plays a role in shaping the characteristics of a species to make them more adapted to the environment in face of the existing evolutionary pressures. This is a model of natural/artificial selection that shows how a population hunted by a predator can develop camouflaging. For example, in a forest with green leaves, green bugs may emerge as the predominant bug colour.

When a predator uses colour and shape to identify the location of prey in an environment, then the colours and patterns in the environment provide additional selective pressure on the prey. If some prey tend to blend into the background better, they tend to survive longer and reproduce more often. If this continues over many generations, the distribution of colours in a population may shift to become better camouflaged in the surrounding environment.

# How it works?

In this game, you are a bird (the predator) and has to hunt butterflies. Each bug is represented by three **genes:**

* red-pigment-gene-frequency
* green-pigment-gene-frequency
* blue-pigment-gene-frequency

As you can guess, each gene will control one component in the colour of the butterfly in (R,G,B). When you capture a butterfly, one of the remaining ones (randomly chosen) will produce an **offspring** of similar characteristics with a small **mutation**.

# What to test?

## Experiment 1

Press setup and go

* Hunt as many butterflies as you can **as fast as possible**.

What do you observe in the **Bugs caught vs. Time** plot?

Compare the current hue distribution with the initial hues.

## Experiment 2

* Set the max-mutation-step to 100
* Press setup and go
* Hunt as many butterflies as you can **as fast as possible**.

What do you observe?

<Swap seats>

## Experiment 3

* Pause the simulation pressing the go button again.
* Change the environment selector from poppyfield.jpg to glacier.jpg.
* Press the change-environment button
* Set max-mutation-step to 26 again
* Resume the simulation pressing the go button

What do you observe?

## Experiment 4

Let us all hunt together…

Go back to the terminal and run the commands below, paying attention to capitalisation.

module load netlogo <Enter>

HubNetClient <Enter>

* Enter your name in the User name box.
* Enter the server IP address and port
* Press the Enter button