LESSON 93

# Identifying a Problem and Initial Modelling

200 minutes (5 periods) | Applied Learning Tasks (ALTs) | Outcome 3.8 | Outcome 3.10 | Outcome 2.1 | Outcome 2.4 | Outcome 1.3 | Outcome 1.9 | Outcome 1.20

## WHAT THIS LESSON DOES

In this lesson, you'll explore modelling and simulation by identifying a real-world problem to simulate. Learn about abstraction and agent-based modelling, break down issues systematically, sketch initial plans, and consider ethical aspects while collaborating in teams.

## CURRICULUM OUTCOMES

| Outcome | What the specification asks for |
|---|---|
| **3.8** | develop a model that will allow different scenarios to be tested |
| **3.10** | explain the benefits of using agent-based modelling and how it can be used to demonstrate emergent behaviours |
| **2.1** | use abstraction to describe systems and to explain the relationship between wholes and parts |
| **2.4** | illustrate examples of abstract models |
| **1.3** | solve problems by deconstructing them into smaller units using a systematic approach in an iterative fashion |
| **1.9** | use modelling and simulation in relevant situations |
| **1.20** | collaborate and assign roles and responsibilities within a team to tackle a computing task |

## WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the process of identifying real-world problems suitable for modelling and simulation.
- Apply abstraction to simplify complex systems into essential components for effective modelling.
- Explore agent-based modelling to simulate interactions and predict emergent behaviours.
- Develop skills in breaking down problems systematically using computational thinking.
- Create an initial model plan integrating sketches, scenarios, and ethical considerations.

### BEFORE THE LESSON

- This is largely a planning and discussion lesson. Have a shared online document or word processor ready for each team and confirm students can open and save to it.
- Have plain paper or a whiteboard available for sketching environments and agent interactions before anything is typed.
- Decide team groupings and the roles in advance so brainstorming starts quickly rather than stalling on who does what.

# Identifying a Problem and Initial Modelling

### HOW THE LESSON RUNS

| |
|---|
| Introduction |
| Brainstorm |
| Understanding Abstraction |
| Exploring Agent-Based Modelling |
| Breaking Down the Problem Step-by-Step |
| Planning with Sketches |
| Creating Your Initial Model Plan |

### TEACHING MOVES

- 2. Brainstorm. Push teams past the first idea. Ask of each candidate: does it involve many things interacting, and can we test different scenarios? Reject one-off calculations. Traffic, rumour spread and predator-prey all pass because a group pattern emerges from simple individual behaviour.
- 3. Understanding Abstraction. Make abstraction concrete with their own chosen problem. Ask what detail they can throw away and still keep the behaviour they care about. A car keeps speed and gap to the car ahead. Colour, engine and driver mood are noise.
- 4. Exploring Agent-Based Modelling. Stress that no single agent is told to make the pattern. Flocking, jams and population booms emerge from every agent following the same simple local rules. Have each team name their agent and write two or three rules it obeys.
- 5. Breaking Down the Problem Step-by-Step. Have each team write one sentence stating the simulation's goal before listing any parts. A vague goal makes everything downstream vague. Then decompose into agents, environment, rules and what they will measure.
- 6. Planning with Sketches. Insist they sketch on paper before touching a keyboard. A grid for a city, a landscape for an ecosystem, arrows for interactions. Sketching surfaces gaps in the rules that pure text hides.
- 7. Creating Your Initial Model Plan. Have teams pull brainstorm, abstraction decisions, agent rules, sketches and ethical notes into one 2 to 3 page document. Check each plan actually lets you test different scenarios; that is the point of the model, not a single fixed run.

### WHAT TO WATCH FOR

- Students pick a problem that is really a single calculation, not a system, so there is nothing to simulate over time. Ask them to name at least two things that interact and change each other. If they cannot, the problem is not suitable for agent-based modelling. Steer them to one of the worked examples.
- Students think emergent behaviour must be programmed in directly, so they try to code the traffic jam itself rather than the driving rules. Return to the flocking example. No bird is told to flock. Have them write the agent's local rules only, then predict what group pattern those rules might produce.
- Abstraction is mistaken for leaving out things carelessly, so students discard detail that actually drives the behaviour. Ask, for each detail dropped, whether the behaviour they care about still happens without it. Gap to the next car matters for jams. Car colour does not.

**DIFFERENTIATION**

- Emerging: Give a ready-made problem such as rumour spread in a school and one agent with a single rule, so they start with structure rather than a blank page. Pair them with a stronger planner and ask them to draw the environment sketch while the partner lists rules.