LESSON 95

Refining, Testing, and Visualising the Model

240 minutes (6 periods) | Applied Learning Tasks (ALTs) | Outcome 3.8 | Outcome 3.9 | Outcome 3.10 | Outcome 1.19 | Outcome 1.20 | Outcome 1.23 | Outcome 2.21

WHAT THIS LESSON DOES

In this lesson, you'll refine, test, and visualise your model from previous work. Follow steps to analyse simulation outcomes, create graphs, modify models hands-on, collaborate with peers for feedback, and reflect on limitations to prepare for evaluation.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
3.8	develop a model that will allow different scenarios to be tested
3.9	analyse and interpret the outcome of simulations both before and after modifications have been made
3.10	explain the benefits of using agent-based modelling and how it can be used to demonstrate emergent behaviours
1.19	identify features of both staged and iterative design and development processes
1.20	collaborate and assign roles and responsibilities within a team to tackle a computing task
1.23	reflect and communicate on the design and development process
2.21	critically reflect on and identify limitations in completed code and suggest possible improvements

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the importance of iterative refinement in improving model accuracy and reliability.
- Develop skills in creating visualisations to interpret and communicate simulation data effectively.
- Gain proficiency in modifying and testing models to explore different scenarios and outcomes.
- Recognise the limitations of models and their impact on real-world applicability.
- Enhance collaborative skills through peer feedback and team review exercises.

BEFORE THE LESSON

- Confirm each student's Python model from earlier ALT 3 work is saved somewhere they can reopen, not left in a lost session.
- Test that matplotlib imports and installs in your room's Python environment. Check pip install works behind the school filter, or pre-install it, before class.
- Have a shared space ready for the team review, such as a shared document or folder students can already reach.
- This spans several periods. Plan where to break, ideally after the visualisation step and again after the peer review.

Refining, Testing, and Visualising the Model

HOW THE LESSON RUNS

- Introduction
- Analyse Simulation Outcomes
- Visualise with Diagrams and Graphs
- Hands-On: Run and Modify Models
- Team Review Exercises
- Emphasise Limitations
- Conclusion and Preparation

TEACHING MOVES

- 2. Analyse Simulation Outcomes. Insist they record baseline outputs before touching anything. Change one variable at a time and re-run, so any difference is traceable. A student who changes three things learns nothing about which caused what.
- 3. Visualise with Diagrams and Graphs. Have them sketch the graph they expect on paper first, then reproduce it in matplotlib. Push for labelled axes and a title on every plot. An unlabelled graph is not evidence in a coursework report.
- 4. Hands-On: Run and Modify Models. Frame this as controlled experiments, not tinkering. Each modification is a scenario with a stated question. Get them to write down what they expect before re-running, then compare.
- 5. Team Review Exercises. Give reviewers a focus: name one strength, one suggested change, one limitation. Require the author to make and re-test exactly one change from the feedback, so the review produces evidence rather than talk.
- 6. Emphasise Limitations. Ask each student to name the assumption their model most depends on and what breaks if it is false. Connect this to when a heuristic would be chosen for speed over accuracy.
- 7. Conclusion and Preparation. Have them assemble analysis, labelled graphs and limitations into one organised document now, while the reasoning is fresh. This becomes usable evidence for the ALT and for later evaluation work.

WHAT TO WATCH FOR

- Students treat a graph that looks plausible as proof the model is correct, without checking outputs against expected values. Make them state the expected result first, then compare the plotted data to it. A tidy graph of wrong numbers is still wrong.
- Changing several parameters at once and crediting the outcome to the wrong one. Enforce one change per run with results recorded each time. Reasoning about cause is impossible otherwise.
- Believing a limitation is a failure to hide rather than a finding to document. Reframe stated assumptions and limitations as marks-earning evidence of critical evaluation, which is exactly what the coursework rewards.

DIFFERENTIATION

- Emerging: If matplotlib or their saved model is fighting them, pair them with a working setup and get them analysing and interpreting outputs rather than losing the period to environment issues. Give them one specific parameter to change and one expected outcome to check, so the task is bounded.
- Developing: Ask them to predict the effect of a change out loud before running it, then explain why the actual result differed.
- Proficient: Push them to add randomness or a second variable and discuss emergent behaviour, and to compare their model against a heuristic approximation, noting the trade-off. This is strong evaluation evidence for the ALT.