

AI Simulation Creation

Creating AI-Generated Simulations as a Mode of Scientific Inquiry

Scientific Inquiry with AI

Student Guide

Detailed Instructions & Course Material

Why This Matters Now

Simulations let learners do things the physical world may not permit: pause and repeat an event, vary one condition while holding others fixed, explore inaccessible scales, practice a high-stakes situation safely, and see relationships that are otherwise hidden. They already support learning in virtual laboratories, flight and clinical training, interview practice, and role-playing environments. Historically, however, most students could only operate simulations that someone else had built because creating one required substantial programming and modeling expertise.

Generative AI changes that division of labor. A model described in ordinary language can become an interactive simulation in minutes. The AI can write the code, but the human creator has the responsibility to determine the purpose of the simulation, what model to use, what variables the user should control, what assumptions are being made, and what accuracy is required to meet the goals. Building therefore becomes a way to learn: you must externalize your mental model, discover what you have left vague, make predictions, debug contradictions, and revise your understanding. The finished application matters, but much of the learning occurs during the building cycle.

In this lesson, you will experience both sides of this new capability. First, you will use an established simulation as an instrument of inquiry. Then you will create a bounded, rule-based simulation of your own and determine what it is trustworthy for. This first build focuses on models whose relationships can be stated explicitly and checked independently. Later in the course, you will apply the same habits to a wider range of simulations and create tools for specific scientific investigations.

Learning Goals

Upon completion of this investigation, you should be able to:

Phenomena:

- Operate an established simulation as an instrument of inquiry: predict, vary one parameter at a time, observe the response, explain discrepancies, and obtain quantitative results for scenarios that are difficult to produce experimentally.
- Identify the assumptions that bound a model's intended use and domain as well as predict conditions under which its representation of the phenomenon will become inadequate.

Experimental Practices:

- Translate a conceptual or mathematical model, grounded in at least one independent source, into a natural-language specification precise enough for an AI system to implement.
- Evaluate an AI-generated simulation at three layers: artifact operation (does the application run and respond?), implementation verification (did the AI implement the specified equations or rules?), and model validation (is that model adequate for the intended purpose and domain?).
- Use limiting cases, an independently calculated interior point or trend, invariants, units, and order-of-magnitude expectations to build evidence for trustworthiness without claiming that a finite set of tests proves the simulation correct everywhere. Refine through targeted prompting and repeat affected tests after making changes.

Habits of Mind:

- Predict before prompting and before running, so that every output is compared against a commitment made in advance.
- Treat AI output as a candidate result and verify it against an independent check before accepting it as correct and reporting it.
- Use a build as a route into a topic you have not yet mastered while keeping an authoritative source—not the AI alone—as the anchor for the governing model.
- Compare your initial and final mental models and identify which specification choice, failed test, or observed discrepancy changed your understanding.

What You Will Need

- **Your university-provided premium ChatGPT account.** You may use another capable assistant if you already have access, but no personal paid subscription is required.
- **A laptop or tablet** with a current web browser.
- **One professionally built simulation** for Step A — an entry from PhET Interactive Simulations (<https://phet.colorado.edu/>) or an equivalent in your own field.
- **Access to Gemini Notebook**, or an equivalent tool.
- **Prelab_Tutor_Building_to_Understand_Creating_Simulations.docx** — you will upload this file to your AI assistant for the tutoring session.
- **Simulation_Gallery.html** — nine simulations that provide a quick view into model features and complexity that is possible to create using natural language prompts with AI models.

The Two Modes of Simulation Work

Simulation work in this course serves two purposes. You practice each one separately in this lesson and think about how you might use the different approaches in your learning journey.

Mode 1 Simulation as instrument	A validated simulation is used to investigate: predict a response, sweep a parameter, locate a threshold, find where behavior changes, or reach a regime the available equipment cannot produce. PhET Interactive Simulations fall into this mode.	Pre-class Step A; Stage 5; post-class extension
Mode 2 Creation as learning	In this mode you are creating the simulation and the learning results in externalizing and revising the model: defining variables and causal or mathematical relationships, identifying assumptions, making predictions, debugging contradictions, and deciding what evidence is sufficient to validate the simulation.	Pre-class Step D; Stages 1–4

Part 1 — Pre-Class Preparation

~1.5–1.75 hours

The pre-class work is done individually; discussing it with classmates is fine. This pre-class activity will give you a chance to explore pre-built simulations and create a specification for a simulation you will construct and test in class.

A. Simulation as an Instrument (~15 minutes)

Open one professionally built simulation in an area that interests you, such as an entry from PhET Interactive Simulations (<https://phet.colorado.edu>).

Change one variable at a time and observe how the simulation responds. Record your responses to the four bullets below in your notebook and bring them to class:

- What is the simulation attempting to teach? State what you take its learning goals to be, and be specific about what you were able to learn from it. Most simulations support several distinct investigations.
- One predict–observe–explain cycle. Write a directional or quantitative prediction before moving the control, record what happened, and explain any difference between the prediction and observation.
- One thing the simulation lets you do that would be difficult or impossible in the physical world.
- Two features of its construction that you would want in the simulation you build later in the lesson.

Why this step first. An established simulation gives you a reference for usability and inquiry. It also lets you notice that even a professionally built simulation is trustworthy only for particular goals and conditions.

B. Socratic AI Tutoring Session (~20 minutes)

Upload `Prelab_Tutor_Building_to_Understand_Creating_Simulations.docx` to your AI chat assistant and work through the session. If you conduct the prelab via the `knowledgewithai` website, it will save a summary of your chat which can be viewed by your instructor. Otherwise, share your chat conversation with your instructor.

The tutor asks which of the two modes your chosen phenomenon leans toward, draws out your initial mental model and uncertainty, and helps you identify what kind of independent source could anchor the governing relationship. If you want something to refer back to, ask for a one-page summary when the session ends.

C. Assigned Reading (~20 minutes)

Load the publication titled “Leveraging generative artificial intelligence for simulation-based physics experiments: A new approach to virtual learning about the real world” (Ben-Zion, Carroll, West, Wong and Finkelstein, *Physical Review Physics Education Research*, 2026) into Gemini Notebook. Use the tool to summarize the findings and to ask your own questions about the approach. Two questions are assigned; record a one-sentence answer to each in your notebooks:

- Which of the two modes above did the study measure?
- What does the study report about students who generated their own simulation, compared with students who were given a finished one?

D. Simulation-Specification Exercise: Write, Survey, Critique, Revise (~40 minutes)

You are now ready to develop the specifications for your simulation.

First-build scope. Choose a transparent mechanistic, mathematical, or rule-based model whose inputs, outputs, and relationships can be stated explicitly. Good choices include a physical/biological/chemical process, a simple population model, an engineering system, or a bounded economic rule set. Your topic must have (1) an independent source for the governing model and (2) at least two predictions you can check without the simulation. Do not use a historical counterfactual, election forecast, complex social system, or fact-rich scenario for this first build; those may be valuable later, but those types of scenarios are difficult to validate.

Develop your scientific specification (~15 minutes)

Use the worked example below as a guide to the level of specificity to include for creation of your simulation. Without AI, and before opening the example gallery, write a one-page scientific specification of your phenomenon. You will learn that an AI model can create a detailed specification for you with minimal input, but the goal here is to go through the process on your own first. As you develop the specifications think about what choices belong to the scientific model and what choices belong to the interface. Don't worry if your specifications are not perfect—you will have time to improve it later. This is one of those places where it is so much easier to ask the AI to

do it, but you will learn more by trying it on your own first. The example should provide the extra support to think it through.

- **Scientific specification — driving question and scope.** State the driving question, intended use, and range over which you want the simulation to be informative.
- **Scientific specification — model.** Give your initial mental model: a short explanation or sketch showing which variables affect which others, plus the part you are least certain about. Then list the variables and parameters with units. State the governing relationships and key assumptions, including what is neglected and why. Cite at least one independent anchor—a textbook relation, authoritative reference, documented empirical result, or derivation—and state exactly what it supports. AI cannot be the only source for the governing model.
- **Scientific specification — independent checks.** Plan at least three checks: one known or limiting case, one independently calculated point or trend inside the intended range, and one invariant, unit, or order-of-magnitude check where applicable. For each, state the expected result and how you will obtain it without the simulation.
- **Brief interface specification.** State what a user should be able to change and what the simulation must display so that your scientific question and checks can be investigated. Do not spend time specifying colors, fonts, or decorative features unless they matter to the inquiry.

Keep the scientific specification and interface specification visibly separate in your notes.

Worked Example: Radioactive Decay

The concept. Radioactive decay is the spontaneous transformation of unstable nuclei. For a particular isotope, each nucleus has a constant probability of decaying per unit time, so the number remaining in a large sample decreases exponentially. The half-life, $T_{1/2}$, is the time required for half the original sample to remain.

Scientific specification

Driving question. How does the number of radioactive nuclei remaining change with time, and how does that depend on half-life?

Intended use and range. Explore decay and test the meaning of half-life from 0 to 5 half-lives for a single isotope with constant $T_{1/2}$.

Initial mental model. N decreases as time increases; the decrease is rapid at first and more gradual later. A longer half-life means slower decay. The part I am least certain about is how the smooth curve represents random decay of individual nuclei.

State variable. $N(t)$, the number of undecayed nuclei at time t .

Parameter. $T_{1/2}$, the half-life; t and $T_{1/2}$ must use the same time units.

Governing relationship. $N = N_0 * \left(\frac{1}{2}\right)^{\frac{t}{T_{1/2}}}$

Key assumptions. $T_{1/2}$ is constant; nuclei decay independently; the exponential law describes the large-sample average; environmental changes of temperature and pressure are neglected.

Independent model anchor. A standard chemistry or physics reference on radioactive decay supports the governing relationship.

Independent checks

- **Known case.** At $t = 0$, $N = N_0$.
- **Half-life.** At one half-life, 50% remains; at two half-lives, 25% remains.
- **Interior calculation.** At $t = 1.5 T_{1/2}$, $N/N_0 = (1/2)^{1.5} \approx 0.354$, so about 35.4% remains.
- **Invariant.** N remaining + N decayed = N_0 at every time.
- **Unit check.** The exponent $t/T_{1/2}$ must be dimensionless.

Interface specification

Controls. Change N_0 , $T_{1/2}$, and t ; include reset.

Outputs. Show $N(t)$, fraction remaining, number decayed, and a graph of fraction remaining versus time.

Presentation. Keep it simple and readable; the model and checks matter more than decoration.

Survey the gallery (~10 minutes)

Once you have written down your draft specifications, open `Simulation_Gallery.html`. Each entry shows a finished simulation together with some additional information about how it was made: the prompt, the build time, the builder's prior coding experience, and its validation status. Three tasks:

- Find one example whose scope is larger than what you specified, and name the one thing it does that you did not think to ask for.
- Find one outside your own discipline and identify why its underlying claims would be easier or harder to validate than those in your first-build topic.
- Note the build times. Students built each of these in 10-20 minutes. Validation takes longer than the build.

The prompts are worth reading closely, because they show how much the assistant fills in on its own. Several builders were exploring how little they could specify and still get something that ran, then modified it from there. We want you to practice including more detail for your first simulation.

One entry carries a known error, left in place. The Knicks Shot Probability Simulator reports a base probability of 31%, a defense penalty of 7.0 percentage points, and an adjusted probability of 27%. $31 - 7 = 24$. The missing 3 points are a per-player clutch bonus applied to every shot and displayed nowhere in the panel. The builder checked that the results looked reasonable across players, and this defect passes that check. The check that catches it is arithmetic: read the numbers off the panel and confirm that they close.

AI critique and revision (~15 minutes)

Now that you have seen the gallery, you can either start a new simulation or build upon your initial simulation specification. Once you have your new specification, give it to an AI for critique—completeness, missing assumptions, ambiguity, unsupported governing relationships, quantities without units, and checks that are not genuinely independent. Ask for critique only; make no build request. Revise where the critique identifies something useful. Record the first draft and revised specification; you will build from the revision in class and refer to both in your final report.

E. Student Guide Review (~10 minutes)

Skim Part 2 below and note the decisions you will be making live. Bring the revised specification to class.

Part 2 — In-Class Investigation**110 min**

The Investigation, Stage by Stage

The class works through the stages in parallel, each student on their own simulation, with the instructor opening and closing each stage. The session begins with a brief introduction to using and creating simulations and a discussion of the prelab activities. You then build (Stages 1–3), test one another’s work (Stage 4), and operate your own bounded model as an instrument (Stage 5).

Stage 1 — Predict, then prompt (~10 minutes)

What to do. Before prompting, re-read your scientific specification. Briefly write what the simulation should show in your notebooks: a sketch of the curve, a limiting value, or a directional trend, together with the expected results of your independent checks. Then turn the specification into a generation prompt, including both the scientific specification and the interface specification. Use the prompt below, filling in the information from your specification.

Build a single self-contained HTML file that simulates [phenomenon].

Scientific specification:

Driving question: [the question it will answer].

Variables and parameters: [each one, with units and allowed range].

Governing relationships and model anchor: [the equations or rules, plus the independent source or derivation that supports them].

Assumptions: [what the model neglects and where those assumptions may fail].

Independent checks: [known/limiting case, interior calculation or trend, and invariant/unit/order-of-magnitude check where applicable; state expected results].

Interface specification:

Controls and outputs: [what the user can change; what must be displayed, including readouts needed for the independent checks].

Presentation: [keep the interface clear and simple; specify visual details only when they matter to the inquiry].

Format: one self-contained .html file with no external libraries, fonts, or data sources, which runs offline when opened from a local copy.

Anything left out of the prompt becomes a choice the AI makes for you. You can also add a statement at the end of your prompt to have the AI ask you for any additional information that it needs to build the simulation. Usually, it will just make decisions and tell you what additional decisions it made after it creates the simulation.

Note: HTML file, Artifact, GPT, or Gem. An AI can deliver a simulation as a live in-app view, such as a Claude Artifact, a ChatGPT GPT, or a Gemini Gem. These support fast iteration, and each one requires its own AI model to run. The deliverable here is a single self-contained .html file that you download and open in a browser: it runs offline, renders the same way regardless of which AI built it, and can be validated and submitted as a file. Build and iterate in the live view if you prefer, then ask the AI to place everything in one self-contained HTML file and download that file.

Stage 2 — Test artifact operation (~10 minutes)

What to do. Download the file and open the downloaded copy, not the preview pane in the chat. Then confirm three things: that it launches with no errors and with the network disconnected; that every control produces a response when moved across its full range; and that every output named in your specification appears and updates when its inputs change.

What to look for. A build that pulls a charting library, a map, or a font from the internet works in the preview and fails when the downloaded file is opened offline. One gallery entry, Africa Explorer, has that failure and is kept as the counter-example: its map area reports itself unavailable on a machine with no connection. (This could be fixed by pulling the information into the simulation code.)

Record. What you tested, what failed, and the prompts used to fix it.

Stage 3 — Verify the implementation; validate the model (~20 minutes)

What to do. Probe two different questions when testing the model. 1) Implementation verification asks whether the code carries out the equations or rules in your specification. 2) Model validation asks whether those specified equations or rules are an adequate representation for your intended purpose and range. A program can implement an inadequate model perfectly. Run the checks from your specification and record the expected and observed result. Different simulations will require different test procedures but consider some of the following:

- A known or limiting case. Set a driver to zero, reproduce a documented case, or approach a limit. This often exposes implementation errors quickly but does not prove the will run correctly over the full range of values.
- A quantity that must stay fixed. Something that must sum, balance, or conserve.
- Compute one result inside the intended range, or derive the expected direction/shape of a response, and compare it with the simulation.
- Model adequacy and scale. Confirm units and order of magnitude; and ask whether the assumptions are reasonable for the intended use.

Record errors you identify in your notebook. Work with the AI to identify the underlying cause of the error and the implementing a fix. Iterate on the errors you identify until the class is ready to move on to Stage 4.

Stage 4 — Peer test and revise (~20 minutes)

What to do. Trade simulations with a classmate and use theirs without asking the builder how it works. The reviewer should start by testing the model relationships, assumptions, and adequacy for the stated purpose. The review should also look for unclear instructions, hidden assumptions, broken controls, and unexpected behavior. The simulation creators should take the feedback, fold them into v2, and re-run to check the revisions have addressed the issues.

What peer testing can establish. If we had more time, we would try to find multiple reviewers that were both knowledgeable of the topic and were from other disciplines. Cross-disciplinary testing is strong evidence about usability, ambiguity, and robustness. Domain specific reviewer can evaluate the model and confirm the accuracy of the simulation output.

Stage 5 — Operate your instrument (~20 minutes)

What to do. The build has reached a tested version; now use it as an instrument within the purpose and domain you designed. Consider some of the following approaches for using the simulation as a learning tool.

- Before moving a control, choose one or two informative settings or trends and predict the response. Then observe what happens and explain any discrepancy. A disagreement may reveal a bug, a poor model, or a gap in your own understanding.
- Answer the driving question by sweeping one parameter across its supported range, holding others fixed, and recording the response at 5 or more settings. Identify one quantitative result that the governing relationships alone did not make evident.
- Push toward an edge and determine if the model stops adequately describing the phenomenon. Explain the breakdown in terms of a stated assumption and distinguish failure of the real-world model from failure of the numerical or software method.

Reflect: What Changed in Your Understanding? (~10 minutes)

Review the following questions and discuss in groups of 4.

- How did the process of creating and validating your simulation helped you extend your understanding of the topic?
- How does using a validated simulation advance your understanding of a topic.

Part 3 — Post-Class Work

~1.75–2 hours

A. Create and Audit Your Simulation Record (~30 minutes)

Create a concise 2–3 page simulation record of your development process. It must show: (1) the driving question, intended use and range, scientific model anchor, relationships, assumptions, and interface specification; (2) evidence from all three layers—artifact operation, implementation verification, and model validation—plus failures, fixes, and untested regions; (3) your predictions, observations, explanations, parameter sweep, and breakdown case; and (4) the before-and-after

account of your understanding. You may ask AI to organize the record, but audit it line by line against your notes to assure accuracy in the record. Assessment emphasizes the evidence and your audit, not report length or polish.

B. Extension: Take the Model Where the Equipment Cannot Go (~35 minutes)

Modify your simulation to answer a question that arose during testing or discussion. Ask the AI to build upon your current simulation and then operate it to answer that question. Consider choosing a question that would be impractical to investigate experimentally—changing gravity, stretching time, running a process for a million years, or operating outside available equipment limits. Predict the response before running the extension, then observe and explain. Report the prediction, result, and whether each changed part was verified and validated.

Then plan the next build (~10 minutes). Name a topic you are trying to learn now and write a three-line specification for a simulation that would show it to you: what varies, what the simulation must display, one independent model anchor, and two checks you could perform without it. No build is required – unless you want to.

C. Retrieval Practice / Course Review (~30 minutes)

Spend approximately 30 minutes on integrated course review using the customized AI-enhanced retrieval tool, which supports spaced repetition and interleaving. Items from this lesson cover the two modes of simulation use, the three evidence layers, the difference between verifying an implementation and validating a model, the limits of edge tests, external model anchors, predict–observe–explain, and bounded claims of trustworthiness.

Summary of Deliverables

Your key deliverables are listed below with the time they should be completed. The prelab items are to be completed before the in-class session because the in-class work depends on them. The **bold** items will be **turned in**, and the other items should be available to show the instructor upon request.

Due before class, with your prelab

- The Step A prebuilt simulations evaluation: learning goals, your predict–observe–explain cycle, one thing that would be difficult in the physical world, and two construction features worth copying.
- **The exported prelab tutor transcript.**
- One-sentence answers to the two reading questions.
- The first-draft and revised specifications, including the initial mental model, intended use and range, scientific model anchor, independent checks, interface specification, and record of the AI critique.

Due at the end of the in-class investigation

- The generation prompt, plus a change log of the follow-up prompts used to debug, adjust layout, or correct calculations, including the revisions made after peer feedback.
- The evidence record separated into three layers: artifact operation, implementation verification, and model validation. Include inputs, expected and observed results, failures and fixes, peer-review roles, the supported purpose and range, and what remains untested.
- The instrument-mode result: predictions, observations and explanations; the parameter sweep; the breakdown case; and one quantitative finding the governing relationships did not make evident.
- The assessed before-and-after account of your mental model: what changed, the evidence that changed it, and what remains uncertain.
- **The simulation itself: a single self-contained .html file that runs immediately in a browser with no local build steps. An in-app link does not satisfy this requirement.**

Due after class

- **The concise audited simulation record, with your corrections marked.**
- The extension prediction, observation and explanation, plus its implementation-verification and model-validation status.
- The three-line specification, model anchor, and two independent checks for your next build.
- Your answers to the reflection prompts.
- **30 minutes of practice with the retrieval tool.**

Acknowledgements

Created by the LLNL Summer 2026 STEM Education Research Team.

Student Team: Ramina Amino, Jahanvi Chamria, Arya Ferozy, Bryanna Gonzalez, Tai Le, Zedikiah McAdams, John Navarra, Joshua Sarabia, Abdurrahman Raza

Faculty Team: Praveen Pathak, David Rakestraw, David Strubbe, Brian Utter

This lesson was developed as a partnership between Lawrence Livermore National Laboratory and the University of California, Merced.