Final Project Instructions

CSE 6730/CX 4230, Fall 2023

Your goal: For your final project, your goal is to build your very own computer simulation, a.k.a., a “simulation model.” That sounds vague but hang in there — if you choose well and devote yourself to it, it will be more fun than you think, and it’s a great opportunity to enhance your project portfolio for your future job search with something that might help you “stand out” from the crowd!

Scope: Simulation models vs. data models. As a reminder, you can divide the world of models into two broad camps: simulation models and data models.

- In a simulation model, you design a set of rules that express how you believe the real-world system behaves. You can think of these rules as capturing your prior knowledge about the system.
- In a data model, you use data to try to learn the model. You try to minimize your use of prior knowledge, letting the data tell you what specific form the model should take.

In reality, a model may be a combination of these two styles. For instance, the rules may be parameterized and you use data to estimate those parameters. Or, you might design a data model in a way that you can inspect its structure and infer the rules that might be governing its behavior.

The focus of your project should be to create a simulation model, rather than a data model. It doesn’t mean you can’t use data in your project, given the hybrid nature of real-world modeling. But since it’s a class about simulation, your project should skew its emphasis accordingly.

The structure of a typical project. A typical project has these components:

- (**Inspiration**) It starts by “taking inspiration” from natural or engineered time-varying systems from the world at large. It’s often helpful to pose one or more questions about a system that you think a simulator could help to answer. Of course, real-world systems are complicated! Part of the “art” of your project is deciding what aspects of elements of that system you will focus on modeling and simulating.
- (**Model**) It then develops an abstract model of this system’s behavior. Think of this phase as the “rule generation” part or “conceptual modeling” task. Data might inform your design.

- (**Simulator**) Create a computer simulation of this abstract model, i.e., code up your simulator! Some projects lend themselves naturally to a team developing additional bells and whistles, like a frontend for interacting with the simulation or a backend visualization tool.
- (**Experiments**) Conduct computer experiments with your simulator. Your goals are to validate your model and, assuming that process looks reasonable, try to answer the questions you posed about the system or phenomena of interest. You should use data in some way, either qualitatively to inform the design of your abstract model, or quantitatively as part of your experiments and validation. Data may again play a role in validating your results.
- (**Reporting**) To help us understand what you did, you'll write up a report and what you did and create a short video that explains your project.

Teaming. You'll work in small teams of 2-5 students each.

Inevitably, conflicts within the team may arise. Please communicate with one another, and if needed, work with the teaching staff to resolve problems early on. The deadline for "divorce" is Checkpoint 2 — after that point, we will consider all teams "final" and you will just have to "make it work" until the end.

Milestones and grade weights. The following list is a summary of what you need to turn in and when. For "how" to submit, see later sections of this document. *(The percentage values shown indicate the fraction of your final course grade and add up to 50%.)*

1. (1%) Form project teams — due **Sep 29**: You need to form a team and "declare it" on Canvas.
2. (2%) Literature survey — due **Oct 13**: You need to search for papers, books, or materials related to your project topic and summarize them in a 1- or 2-page review.
3. (3%) Project checkpoint 1 — due **Oct 27**: You need to show preliminary progress on your project by explaining your abstract or conceptual model, as well as any preliminary progress on implementing this model.
4. (5%) Project checkpoint 2 — due **Nov 17**: You need to show that you have made significant progress on implementing your simulator. If you've hit any snags, you need to describe them at this point.
5. (40%) Final project (due **Dec 5 — final deadline, absolutely no extensions!**): Your project is due on this date and will be worth 40% of your final grade. You will need to submit a written report, your *documented* simulation code (via your GitHub repo, above), and a short (2-3 minute) video summarizing your project.

Where to find inspiration. Here are a few examples of projects (reports only) from a previous year:

[https://gatech.instructure.com/courses/349628/files/folder/project/old-projects-hall-of-frame]

Do not feel limited by what you see there, however! There were many interesting ideas not included above.

Here are some reference materials you can consult to help you brainstorm with your team about project topics. Except where indicated, these should be available electronically either by direct link (e.g., Smith? book) or through the GT Library in electronic form (search for the item via library.gatech.edu).

- Sayama's book (2015) -- lots of examples, exercises, and mini-projects that you can go deeper on or extend
- *Scientific computing with case studies*, by Dianne O'Leary (2009), SIAM.
- *Mathematical modeling of zombies*, by Robert Smith? (2014). (Yes, the question mark at the end of Smith?'s name is for reals.) You might find inspiration from some of the mathematical techniques for conceptual modeling that Smith? applies to consider various aspects of how zombies might behave "in the real world."
- *The structure and dynamics of networks*, by Mark Newman, Duncan J. Watts, and Albert-László Barabási (2006).
- *Mathematical models: mechanical vibrations, population dynamics, and traffic flow*, by Richard Haberman (1998), SIAM.
- *Neural ordinary differential equations* (NeurIPS'18 best paper). Some of you have asked about how to connect data or ML to the topics of the first part of this course. Here is one way (lots of related cited papers contained therein) that makes one perhaps-unexpected connection!
- *Tutorial on the dynamics of biological neurons* (spiking models), by Jack Terwilliger (2018).
- *A primer on mathematical models in biology*, by Segel & Edelstein-Keshet (2013).
- *Cellular automata tutorial*, by Jarkko Kari (2008?). A good summary of key technical results related to CA models.
- *The nature of code*, by Daniel Shiffman. Section 7.9 has an interesting list of application ideas for cellular automata (as does Sayama's book).
- *Traffic and related self-driven many-particle systems*, by Dirk Helbing. [Link via GT Library Proxy]
- *Programmable matter*:
  - Talk by Dana Randall @ GT — https://www.youtube.com/watch?v=nPCjWIoK5KI
  - A recent article on self-organization in this context – https://science.sciencemag.org/content/371/6524/90
- Examples of discrete-event models — [folder on Canvas]

Evaluation. Roughly speaking, when we assess your project report (and summary video) and code artifact, we'll be considering these dimensions:

- Exposition -- How well does the report and video explain the problem(s) and solution technique(s)? Is it concise, but also clear, readable, and precise? Does a prospective reader take away insight?
- Code -- Is it clear and readable? Does the implementation pay attention to efficiency considerations?
- Results -- Does the project contain simulation examples to help illustrate the problems and techniques? Does it use visualization appropriately?

There is no hard "weighting" among these categories, but do try to pay equal attention to all criteria. Your report should highlight where you put a lot of effort and acknowledge any weak points in what you did.

Peer assessment. The entire team receives the same base grade. However, at the final submission, we may ask each of you to submit your own "peer assessment" in which you evaluate the contributions of your teammates. In the event of discrepancies, a small additional component of the grade may be based on teammate evaluations. The purpose of this assessment is to create an incentive for you to find ways to work well together and contribute equally to the overall product.

Submission instructions. We will post more detailed guidelines for each submission as we get closer to each milestone's deadline and update the document here.

For all milestones: You will use the GT GitHub instance (github.gatech.edu) to submit some of your milestones. Start by creating an initial repository there to hold the materials for you and your teammates. You can make this private if you wish, but then you will need to share it with all of the teaching staff so we can grade it (more detailed instructions to come).

Team Formation (due **Sep 29**)

For this milestone, you just need to "declare" your team. Go to Canvas and complete the following two steps:

1. Under "People," look for the "Groups" or "Project Groups" tab, find an empty Project Group, and sign yourselves up!
2. Under "Assignments," submit the Form Project Teams assignment. State whether you are in the graduate, undergraduate, or online/distance section, and list your teammates and contact info. Optionally, if you have a topic idea already, please mention that.

1st milestone: Literature Review (deadline of **Oct 13**)

For your literature review, write up a short document (1- to 2-pages, plus unlimited additional space for references) that summarizes what others have done related to your project. That is, are there existing models, simulators, or simulation techniques? Will you adopt one of these approaches as a basis for your project?

- (**Graduate projects only**) What will you do that will be new or different from what you've found? You don't have to do completely new research, but if you are attempting to replicate an existing simulator or simulation study, you should distinguish what you will attempt to do from what has been done before.

Project Checkpoint 1 (Due **Oct 27**)

Submission: A single PDF document, which includes a pointer to your GitHub repo at https://github.gatech.edu/ **GT GitHub Enterprise**

There are three components for the submission for the checkpoint:

1. A clear and detailed description of your project (~ 2-4 pages, excluding references).

At this stage, you should have a clearer idea of the project and the details that you expect would be part of the final submission.

Some things to include:
a) An abstract summarizing the system and the goals of the project
b) Description of the system being studied
c) A conceptual model of the system
d) Platform(s) of development
e) Literature review (paste in what you submitted before, possibly updated if you discovered new things in the interim — the literature review does not count against the 2-4 page guideline)

2. An update of the current state of the project and initial results, if any (max 2 pages)

Some things to include:

1. A "show of progress" via some working code, analysis, or initial modeling attempts
2. If there have been any major changes in direction or "course corrections" since your original proposal, you can describe them here.
3. Division of labor: How will you divide up the remaining work among your team? In particular, we will be looking to see that you've given thought to how to ensure your project justifies a multi-person effort.

3. A Git repository for the final submission

This component of the checkpoint would ensure that you have set up a git repository on the Georgia Tech GitHub repository. This is where you will be sharing your final project implementation with the instructors.

Things we will evaluate:
a) A GT GitHub link to the project repository is included in the report

b) The repository has correct permissions, that is shared with all the teammates, instructors and TA. **See step 8 below.**
c) At least one file, e.g., README or the Project Checkpoint report is in the repository.

NOTE: Although we recommend it, you don't have to use git for your development. You just have to share your work with us on Git.

In case you have not interacted with GT GitHub or Git in general. The following is a brief tutorial for setting up a GT git repository and pushing (read, saving) your local copy on the GitHub server.

1. Log on to GT GitHub (https://github.gatech.edu/) using your GT credentials.
2. Name your new repository (or "repo").
3. Choose the privacy setting for the repo. If you choose to make it public, anyone with a link and a GT account would be able to access your repo and its contents. This can be updated later.
4. Follow the instruction on the next page to set up a local repo or for a Linux (terminal) based system use the following.
   For other systems follow this link (https://hackmd.io/s/B1Tgv0bNE):
5. **NOTE: For Windows users, download Git for windows and install it. You can type the following commands into the git for windows terminal. Alternatively, you can download and use GitHub desktop if you want a UI like repo management instead of all these commands.**

   ```
   1. cd <project_folder>  # go to you project folder
   2. vim README.md         # optional: create a readme file for your repo
   3. git init              # initialize local git repo
   4. git add README.md     # optional: add file to local git
   5. git commit -m "first commit" # describe the state of the repo.
                                   # examples could be:
                                   # "added feature X", "completed part Y"
   6. git remote add origin https://github.gatech.edu/<gt_username>/<repo_name>.git
   7. git push -u origin master    # upload the current repo state to GT GitHub
   ```

6. Once the repos are set up, work as usual in your project directory. When it is time to push changes:
   1. `git add --a`   // to add all the new files to git.
      ProTip: use `git status` to view the current state of all files added to git or updated since the last commit.
   2. `git commit -m "second commit"`
   3. `git push`
7. Collaborators can be added through the settings section on your repo at **GT GitHub Enterprise** `https://github.gatech.edu/<gt_username>/<repo_name>/settings/collaboration`

ProTip: You can search and add people using GT usernames.

8. Add **the TAs** *and* **Peng** as collaborators on your repo. Their IDs are:
   1. psi6 — Phillip Si

2. yliang329 — Yulong Liang
3. sjaiswal41 — Shikhar Jaiswal
4. yqiu327 — Yuan Qiu
5. pchen402 — Peng Chen

You are always welcome to request a discussion with the TAs via Piazza, in case you find something unclear.

Project Checkpoint 2 (due **Nov 17**)

For this checkpoint, we would like to see that you have made significant progress toward creating your simulator. The TAs will inspect your code repos to see how much progress you've made. Please submit a short (1- to 2-page) summary of how you've divided the work so far and what you have completed, and what work remains.

Project–Final Report (due **Dec 5 @ 11:59 pm ET — no extensions**)

Here is what you need to do for your final project submission. There are three parts.

Part 0: Summary video. Create a short (2- to 3-minute video) summarizing your project. You will upload your videos to Georgia Tech's MediaSpace site. We've created a channel to hold your project videos, which is here:

https://mediaspace.gatech.edu/channel/Project%2BSummary%2BVideo%2Bfor%2BCSE-6730-A_CX-4230-A/324015442

Please upload your video there and use the title convention, "[Team ##] Title of your project" (without quotes), e.g.: *[Team 2940] Simulating the universe in 3 easy steps*. If you can embed a direct link to your video from your project report, that would also be helpful.

Notes:

1. You will be able to see other team videos on this channel! This "feature" is a way to mimic the poster session that we would have normally arranged.

2. If you have issues uploading your video, please check the MediaSpace FAQ: [link]

Part 1. Each team will upload a single PDF into the assignment on Canvas named [Final project submission link here] for your final project submission. This PDF should contain the following information.

- A single cover page that shows the title of your project or tutorial, your team number, a list of your team's members, and a link to your GitHub repo so we can verify what you implemented. (Presumably, this repo is the same as what you indicated in your Project Checkpoint.)

- The project report or tutorial itself
- After the report or tutorial contents, provide a brief description of how you split the work among the team members.

Please also place a copy of this PDF in your repository.

Suggested outline: Most of you are doing projects that involve simulating a real-world system. In such cases, here is a suggested outline for your report. (The top-level bullets are suggested section headings for the report; the sub-bullets are only there to explain what goes in the corresponding section.)

- **Abstract** — 100-200 word summary of the overall project
- **Project description** (see also [BA19] Chapters 2.3.1 & 2.3.2)
    - What is the goal of your project?
    - If you are modeling some real-world phenomenon, what aspects of that phenomenon are most relevant to capture in your model?
    - Assume a reader does not know much about this phenomenon. Use clear language to explain it and minimize jargon or define terms as needed. Illustrations or sketches are always helpful!
- **Literature review** (include from your earlier submission)
- **Conceptual model** (see also [BA19] Chapters 2.2 and 2.3.3)
    - Describe the abstract or mathematical (conceptual) model you have developed. Clearly explain how features of the model reflect features of the phenomenon of interest as described under the "project description."
- **Simulation / simulator / simulation model** (see also [BA19] Chapters 2.3.4, 2.3.5, and 2.4)
    - Summarize the simulation model you implemented.
    - Verification: Explain the procedure you used to verify your implementation relative to your conceptual model
- **Experimental results and validation** (see also [BA19] Chapters 2.4 and 7)
    - Explain what studies you did that use your simulator. Clearly describe your experimental procedure.
    - How did you attempt to validate the simulator? Justify how you modeled the inputs to the program.
    - Clearly explain how you analyze your outputs. Report confidence intervals as appropriate.
- **Discussion, conclusions, summary**
    - What did you learn about the system you were modeling and simulating?
    - What would you suggest they do if someone were to extend this work?
- **References** (from your literature survey)
- **Appendix: Division of labor**
    - Briefly explain how you divided the work. Who did what?

Part 2. Separate from the above submission, which only needs to be done once for the team, each of you must **individually** submit a [Teammate assessment]. It is simply a text entry box in which you will enter the following information:

- Your team number and list of teammates
- Assign a letter grade (A through F) to yourself and each of your teammates based on assessing their contributions to the project. In each case, explain your grading. (Note: Do not interpret this grade as one for your project; instead, think of it only as a grade based on the effort expended to complete the project.)
- We will not share your teammate assessment with anyone else on your team. It will only be used for us to identify issues and try to resolve them after the fact.