Overall Project Description

In a group of 2-3 students, you will design and write a software model of a physical, social, or business system and use it to conduct a basic study of that system.

Figuring out what to model

In choosing what kind of model to create, you will need to balance a number of considerations:

* Do you find the topic interesting?  If you find it interesting, that's a big plus.
* What background do you need to create the model?  If you need to teach yourself the dynamics of Calabi-Yau spaces in order to code the model, you need to find something closer to your current background.  On the other hand, if you implement a simulation of the card game War, you need to find something meatier.  Remember too, the "you" in this question includes all members of the team.
* How much time will it take to create the model?  I've allocated five class sessions to do Project-related activities.  Thus, your project should require something like 30+ hours of effort outside of class.  If you pick a project that is too simple or too hard, you won't be happy.
* What should I do with my time?:  Your model should be suitably complex, have suitable amounts of sensitivity analysis (i.e., you have some understanding of how your model behaves), and there is appropriate visualization/visualization analysis.

How to find ideas is a trickier question.  If there's a topic that you've always been interested in, that might be a good place to do a Google search on.  You also might find looking through peer-review articles (especially review articles) or pre-prints to be fruitful places to start.  The [arXiv.orgLinks to an external site.](http://arxiv.org/" \o "" \t "_blank) pre-print service might be one good place to look (because its coverage is broad and it's all free, so you don't have to see if the UW Libraries has a subscription).

To help jump-start your thinking, here are some ideas of possible phenomena to model.  You're welcome to use these ideas though some of them, in their current form, are a little too simple for this project:

* One or more of the project ideas described in S&S Chapter 14 (that we haven't done yet).
* Cellular automata traffic model:  See [Gould et al. (2007)](https://canvas.uw.edu/courses/1817127/files/132993763/download?wrap=1)

[Actions](https://canvas.uw.edu/courses/1817127/assignments/10260721)

.

* Markov model for boarding an airplane:  See [Steffen (2008)](https://canvas.uw.edu/courses/1817127/files/132993768/download?wrap=1)

[Actions](https://canvas.uw.edu/courses/1817127/assignments/10260721)

.

* A previous year's project:  A tool to assist in the scheduling of courses in the CSS Division for a given term as well as enables one to project future faculty and other resource needs as student population changes and the program is revised (e.g., new programs begun).  The model you develop can be used a tool to help analyze the resources needed under various development scenarios.  You are provided the following two documents:
  + A list of all current courses offered during a year:  [list\_of\_classes.xlsx](https://canvas.uw.edu/courses/1817127/files/132993707/download?wrap=1)[Download list\_of\_classes.xlsx](https://canvas.uw.edu/courses/1817127/files/132993707/download?download_frd=1).
  + A list of all current faculty and their areas of expertise and the number of courses they can teach:  [list\_of\_faculty.xlsx](https://canvas.uw.edu/courses/1817127/files/132993762/download?wrap=1)[Download list\_of\_faculty.xlsx](https://canvas.uw.edu/courses/1817127/files/132993762/download?download_frd=1).

Note that for both files, I have made changes to names, areas of expertise, etc. to obscure identities.  These documents only describe a starting point in terms of constraints.  Over time, the courses, number of faculty, etc. may change and your model has to incorporate that.  In fact, the purpose of your model is to enable one to see what the resource requirements will be under various scenarios.

**Number of hours of effort:**  Whatever you choose to do, remember that the total number of hours of effort should total the number of people on the team times 2.5 weeks times 15 hours/week.  Thus, a team of two people should be doing a project that reflects 75 hours of cumulative effort.  For comparison, recall I gave you around 1 week to do the cane toad model, so I am expecting you to be able to do a model of that level of complexity in around 20 hours of total effort.  Thus, your final project has to require considerably more effort than the cane toad model.  (Of course, the cane toad model was already pre-analyzed for you, so for your project, if you aren't choosing a canned model, a bunch of hours of effort will go into researching the topic and structuring the model.)

And, of course, feel free to see me to talk over your ideas!

Graded components

I will grade these components:

* Development Plan:  This is described in this document below.
* Milestone:  See the [Project: Milestone](https://canvas.uw.edu/courses/1817127/assignments/10260720) document for details.
* Final Presentation:  See the document for details.
* Final Software Product:  See the document for details.

Note that not only for the Final Presentation but also for the Development Plan and Milestone, you will be graded on your presentation of those items.  So, if you don't attend class the days you'll receive feedback on those items, you will lose points.

Development Plan

When working on a complex, long-term project, it is very easy to get lost, and hard to keep focused and motivated. Your Development Plan will tell both you and me what you will be doing and when you will do it. It will breakdown this large task into bite-sized pieces.  So, I want you and your group to create a Development Plan for your software.  Below is a list of the sections required in your Development Plan (all page counts assume your document is single-spaced):

* **Title:** Please provide a title for your project.
  + **D&D Dungeon Master Simulation (or something like that idk)**
* **Introduction (1 page):**  A general description of what you are modeling and why.  Here would be a good place to put in a brief (1/2 page) literature review describing the phenomena you're interested in (if applicable).  *Make sure you properly cite references and do not plagiarize!*

This project models a simulated Dungeons & Dragons (D&D) combat scenario using an agent-based, grid-based approach. The goal is to explore how different configurations of player characters (PCs) and enemy non-player characters (NPCs), along with varied strategic behaviors, affect the enjoyment and pacing of combat encounters. Specifically, the simulation focuses on measuring the length of combat rounds, damage dealt and received, and emergent patterns of interaction, to identify optimal combinations of enemy count, composition, and tactics that lead to the most engaging gameplay experience.

Dungeons & Dragons, a tabletop roleplaying game first published in 1974, relies heavily on the Dungeon Master (DM) to create and manage encounters. In combat, the DM orchestrates the behavior of enemies in response to player actions. While the game is deeply narrative and improvisational, combat sequences are structured around rules governing initiative, movement, actions, and damage, making them a viable candidate for agent-based simulation. In this project, a simulated DM agent will generate and execute combat encounters under different scenarios, enabling statistical analysis of their outcomes.

The phenomenon being modeled, combat enjoyment and balance in turn-based RPG systems, has been explored in various academic and game design contexts. Yee (2006) highlighted that combat pacing and strategic depth are key components of enjoyment in roleplaying games. Adams and Rollings (2007) describe combat mechanics as a central feature of player engagement, with balance and variety being critical to long-term interest. More recently, Togelius et al. (2011) have shown that simulations and procedural content generation can be used to optimize game design elements such as enemy placement and difficulty scaling.

Most existing research has focused on modeling players or generating content automatically, but this project shifts focus to the Dungeon Master’s role. By simulating a DM agent that runs battles using different enemy strategies and group compositions, the simulation can test how these changes influence key aspects of gameplay, like pacing, fairness, and engagement. For example, the simulation will analyze how tactics such as concentrating attacks on one player or spreading damage across the party affect the overall flow and enjoyment of combat. The goal is to identify patterns that consistently lead to satisfying and well-balanced encounters. Insights from this simulation could help game designers and DMs alike create more enjoyable and dynamic combat experiences, whether in traditional tabletop sessions or digital adaptations.

**Citations:**

Yee N. (2006). Motivations for play in online games. *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society*, *9*(6), 772–775. <https://doi.org/10.1089/cpb.2006.9.772>

Adams, E. (2025, January 28). *Fundamentals of Game Design*. Choice Reviews Online. <https://www.academia.edu/81247135/Fundamentals_of_game_design>

Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, *3*(3), 172–186. <https://doi.org/10.1109/tciaig.2011.2148116>

* **Model Description (1-2 pages):**  Describe what your model will do and be capable of.  What assumptions will you make in your model?  What constraints will you have (and do you have data/can you get data to constrain your model)?  What could you model be used for?  What kinds of problems could it help solve?  What use-case limitations would your model have?  What would you need to overcome those use-case limitations?  Do not be too general in this section of your Development Plan.  The more specific and detailed you are, the more guidance it will provide for your project.  Note that this description should utilize the kinds of analysis S&S model for us.  For instance, in Ch. 1 as well as in the models they describe throughout the book and we've implemented, they go through a specific process of analysis of the problem, the structure of the model, describing the routines involved, etc.  Your description should follow their "template."

Model Description

Problem Analysis

The objective of this project is to model and simulate combat encounters in *Dungeons & Dragons* (D&D) using an agent-based, grid-based system. The simulation focuses on evaluating how various enemy strategies and compositions impact the pacing, fairness, and enjoyment of combat. The central question is: *What configurations of enemies and strategic behaviors lead to the most engaging combat experience for a given party composition?*

The problem is inherently stochastic, as player and enemy actions involve probabilistic outcomes (attack rolls, damage variability), and different strategies may produce highly variable results even under identical initial conditions. The system includes both deterministic rules (movement speed, initiative order) and probabilistic processes (dice rolls, action outcomes). The simulation must account for these interactions to produce useful, generalizable insights.

The model is intended to generate repeated simulations of combat scenarios with varying parameters (number of enemies, strategy types), allowing for statistical comparison across multiple runs. Key performance metrics will include combat duration (in rounds), damage dealt and received, and party member survival. These will be used to evaluate which enemy setups produce challenging but enjoyable experiences.

Model Formulation

**a. Data Gathering**

Initial model parameters and rules will be based on the official 5th Edition D&D ruleset. These include character statistics, damage rules, turn structure, and enemy behaviors. Simplified stat blocks will be used for typical character classes (fighter, ranger) and a common enemy type.

**b. Simplifying Assumptions Made**

To make the model easier to work with, the following assumptions will be made:

* All characters act according to a basic decision tree based on their archetype (melee fighters prioritize close-range enemies, while ranged maintain distance).
* The environment will be a flat, grid-based battlefield without elevation, traps, or complex terrain.
* Player characters and enemies will have simplified, fixed stats with average HP and damage values representative of 3rd to 5th level characters.
* The only attack actions will be melee or ranged, there will be no magic.
* Enjoyment will be inferred from assumed measures: round length, damage balance, and number of surviving players. b

**c. Determine Variables and Units**

* **Stock variables**:
  + Number of enemies
  + Enemy strategy (focus fire, target spread, terrain positioning)
* **Output variables**:
  + Number of rounds until combat ends
  + Total damage received and dealt
  + Number of PCs surviving
* **Constants**:
  + Grid size
  + Turn structure and initiative rules
  + Character and enemy movement speed
  + Number of players
  + Player strategy (melee, ranged)
* **Units**:
  + Time: measured in rounds
  + Damage: hit points (HP)
  + Position: (x, y) coordinates on the grid (each grid square = 5ft)

**d. Establish Relationships Among Variables and Submodels**

The system is composed of interacting components (or submodels) that operate in discrete time steps (combat rounds). Each round represents a full cycle in which all agents (player and enemy characters) act in turn. The overall system can be viewed as a dynamic process in which the state of the simulation evolves over time according to deterministic rules and probabilistic outcomes. The submodels are defined as follows:

1. **Turn Submodel**: This submodel controls the initiative system, determining the order in which agents act. It updates the simulation clock by one round and activates each agent in order. Each agent's behavior and resulting actions are passed to the decision submodel.
2. **Decision Submodel**: This submodel represents the decision-making logic for each agent. For each turn, the submodel receives the current state of the system (positions, HP, etc.) and outputs an action: move, attack, or do nothing. Strategies are encoded as decision trees or conditional rules, such as "move toward nearest enemy and attack" or "focus lowest-HP target." While player strategies remain fixed, enemy strategies will be systematically varied across simulations.
3. **Combat Resolution Submodel**: Embedded within the turn and decision routines, this submodel applies the mechanics of the game to resolve attacks. It calculates hit/miss outcomes and determines damage dealt, based on probabilistic functions derived from dice rolls.

The interactions among submodels follow a cyclic structure:

* At the start of each round, the **Turn Submodel** sets the order of activation.
* Each agent, when activated, uses the **Decision Submodel** to determine its action.
* The chosen action is executed through the **Combat Resolution Submodel**, modifying state variables such as health and position.
* This process repeats until a terminal condition is reached (all enemies or all players defeated).

This modular structure enables experimentation by adjusting specific parameters (enemy behavior rules or number of agents) and observing their effect on the properties of the system.

**e. Determine Equations and Functions**

While no differential equations are required, the simulation will involve the following types of functions:

* **Movement function**: Uses Manhattan distance to determine legal movement paths.
* **Targeting function**: Returns list of valid targets based on strategy and constraints (range, line of sight).
* **Damage function**: Samples from a fixed probability distribution (1d8+3 damage = uniform distribution from 4–11).

These functions will be parameterized and run across multiple Monte Carlo simulations for each enemy setup.

* **Analysis (1 page):**  Describe what kinds of analysis you will do on the output of your model.  What metrics and quantities will you look at?  What questions will you ask of your model?  What can your model tell you about the phenomena you're interested in and its time evolution?  How will you validate/verify your model?
* **Testing (1/2 page):**  You should have a testing suite for your model.  In this section, describe the kinds of tests you will provide.  You do not have to use a formal Python unit testing framework like unittest, nose, or pytest, but you need to have some sort of testing suite for your model.
* **Personnel (1/2 page):**  A description of each member of your team, their areas of expertise, and *a rough description of how the various tasks involved in the project will be divided amongst the team.*  Note that I will assign groups with an eye to balancing areas of expertise within the group.
* **Technologies (1/2 page):**  List and describe the various technologies you will use, and their role in your project.  Please note that for this assignment, you *must write your program in Python.*Because this is a team software development project, you will need some way to do team development.  The technology you use must enable team development for both your code as well as all documents related to that code (e.g., your Powerpoint).  I recommend using GitHub.  You may use something else, but it *must provide a record of commits for all team members over time.*  I will use that record to help make any needed adjustments to the group grade to account for individual contributions.
* **Benchmarks (1/2-1 page):**  A list and description of benchmarks, with dates as to when each benchmark will be finished.  Of these benchmarks, one will be graded.  Call that benchmark to be graded "Milestone" in your Development Plan.

In addition to these sections, you can add other details.  These are optional.  Some of these optional topics include:

* A schedule of meetings with the instructor to discuss issues.
* A rough sketch of a possible API.
* A description of the tests and testing methodology you will use in your program's development (e.g., will you be doing continuous integration).
* Algorithms and modeling techniques you plan to learn or become more proficient in, with a schedule of what sub-topics will be learned for each.
* Additional collaborators you will consult with along with a list of their expertise and what their contribution to the project will be.

Each group will turn in one Development Plan.  Here is precisely what your group will turn in online in Canvas:

* A PDF document that is your group's Development Plan.  Do not give me a Word document nor give me more than one file for your Development Plan.
* If you are using GitHub, a commit history activity graph for each member of the group that shows me (roughly) how active each member was in creating the Development Plan.  Here's one for a repo I've been involved with:  [https://github.com/jwblin/aoslib/graphs/contributorsLinks to an external site.](https://github.com/jwblin/aoslib/graphs/contributors). Note that the GitHub activity graph does **not** include activity from merges.  If you do branch-and-merge development (as opposed to committing to master), you have to provide some other way of getting me activity information.  This is one way I figure out the equity of the contributions from each member of the group.  If you are not using GitHub, or the activity graph and lines of code counts do not adequately describe everyone's contributions, you need to turn in something similar that will give me similar information.

Please have the person whose last name comes**earliest in the alphabet** submit the work.  Again, **EACH GROUP WILL SUBMIT ONE COPY** of the deliverables.  Make sure it is submitted **ON TIME.**

On the day the Development Plans are due, as a class we will go through each group's Development Plans.  In the midst of that discussion, your group's Development Plan may be altered.  Thus, you should not assume that your Development Plan is complete until after we've all evaluated it.