# AI Progress Report #3

Thomas Guerena

Last week I decided to redesign my project as a user-parameterized bacteria colony simulation. My main focus is to reduce randomization and give a user control over almost every aspect of the simulation. I'll provide an incomplete list of those parameters below.

## New Mechanics

The simulation will now include two important new mechanics, one important replacement for an existing mechanic, and two mechanics have been removed.

1. **Energy** (new) - Each bacteria cell has an *energy* value. Every action that a bacteria cell can take requires energy (move, replicate, mate) with the exception of *eat*.
2. **Food** (new) - I'm adding an additional entity into the matrix which bacteria cells can *eat* to gain energy.
3. **Priorities** (new) - Limiting a bacteria cell's energy limits how much it can do, slowing down the simulation and making things much less chaotic. The cells will no longer be trying to move, replicate, and mate all in a single generation. Instead, they will have prioritized actions, influenced mostly by their current energy levels. I want to allow users to order these priorities and adjust the threshold between them. These prioritized actions include *eat (search), mate (search), replicate*. Movement is inherently part of any action requiring search.
4. **Diversity** (replacement) - Diversity is a replacement of cell mutation and serves as a fitness against the antibiotic. After doing more research on the biological aspect of this problem, I've realized that it's not a good idea to try to model the way in which bacteria evolve to resist antibiotics. There are too many ways this occurs, and trying to model just one feels pointless. Instead, I'm focusing on a general theme between all the ways bacteria become resistant - *diversity*. In general, if a colony becomes genetically diverse enough, it will have a greater chance of surviving a antibiotic (the same way a more diverse human gene pool leads to better disease immunity). To model this, I've made bacteria diversity an integer value in each bacteria cell. Now, mutation and mating has been simplified to different methods of increasing diversity - mutation being rare, and mating being expensive (energy). This inadvertently adds a game aspect to my simulation. My user can now explore the simulation's various parameters and discover how to form a bacteria colony that can best resist the antibiotic.
5. **DNA** (removed) - Genetic encoding is no longer useful in the simulation. It adds too much complexity to the simulation, and creates chaotic results. I think the chaos came from DNA-influenced decision making. I think I've come up with a more useful, two-part approach.

a. *Live or Die*: use simple “Game of Life”-style rules to determine whether a bacteria cells survives the current generation.
b. *Goal Oriented Actions*: Behavior related to prioritized actions such as searching for food, searching for a mate, replicating, etc. will be hard coded. Meaning, all bacteria would act the same given an identical matrix state. I think this is important. I’m trying to do away with as much randomization as possible.

# Parameterizable Features

- Bacteria starting energy.
- Initial bacteria population size, and position of each bacteria cell.
- Number and position of initial food.
- Rate of food generation.
- Number of generations before the antibiotic enters the matrix (short, moderate, long).
- Potency of the antibiotic (weak, moderate, strong).
- Order of prioritized bacteria actions and the threshold at which a bacteria would switch from one to another.
- Rate of mutation (influences rate of diversity).

This list is likely to grow/change as I discover different ways to give the user more control.