Nathan, Duncan, Serena, David
3/23/16

Project Proposal

1. **The Big Idea:**

- Main Idea: Genetically evolving blobs to survive given different physical and social conditions, using natural selection (rather than picking the blobs with the highest fitness evaluation)
- Topics: Genetic algorithms, neural nets (maybe?)
- MVP: Blobs that are genetically trained to find food
- Stretch goals: Social-greater good, social interaction, seeding

2. **Learning Goals:**

- David: Learn how to implement multiple types of evolutionary algorithms (genetic/neuro). Compare with existing libraries and optimize. Practice good large-scale programming habits.
- Duncan: fuel this newfound obsession for natural evolution paralleled programmatically, and create a portfolio piece I'm passionate in the process. I also really want to come out of this project understanding every line of code, or at least feel confident that I could walk someone through it without being hand-wavey.
- Nathan: Explore genetic algorithms/ machine learning. Obsess over computational complexity and distributed systems for algorithms we build from the ground up.
- Serena: Learn how genetic/machine learning algorithms are implemented, and (maybe possibly) learn how to optimize my code in significant ways.

3. **Implementation Plan:**

- Generate initial population and environment (scattered food, fuel, obstacles, whatever...)
- Evaluate fitness of individuals in population.
- Repeat:
    - Select most fit individuals.
    - Breed most fit individuals.
    - Evaluate new fitness of new individuals.
    - Replace unfit population.
- Possibly DEAP, though we're all interested in at least trying to implement from scratch, even if it's less efficient

4. **Project schedule:**

- To start, evolve a blob that can go fetch food particles in a field. Hopefully we can achieve this in the first week, week and a half
- Health timers
- Collisions
- Social/greater good behavior?
- Config file to control parameters

5. **Collaboration plan:**

- Individual and team programming time.
- Define clear parameters to work with so that code can be written individually.
- Use (and figure out) git, obviously
- Trello
- Meet once or twice a week.
- Weekly on Sunday afternoon from 2 to 5 or later

6. **Risks:**

- Inability to implement algorithms
- Not figuring out how to effectively work as a four-person team
- Little time to work together (scheduling)
- Lack or organization/consistency between four people's codes
- Aadfs

7. **Additional Course Content:**

- Collaborative git
- Neural network implementation? We can do that ourselves mehbe
- Structuring code (multi-script patterns)
- Best data structures to use (ex. Classes, matrices), pros and cons for code optimization