Rough overview of project:

When program starts,

Takes arg NOGUI if on CS machine. Turns off gui.
Otherwise, will display as normal.

Initial population will either be ~200 or we could do one huge population of ~1000 and give threads subsets of that one population.

Will create so many creatures using genetic algorithm and randomly assign attributes.
Will store creatures in ArrayList.

Go through every creature and get their fitness level after they have “jumped”, and I use quotes because we don’t actually see them jump.

Get the top ~20 creatures from subset with highest fitness and make those shared between other subsets of the population.

Get one creature with highest fitness and display that on GUI.

All 32 populations (1 population per thread) are running in the background at the same time, calculating the fitness of their jumping creatures and recording the changes, but only one creature from one population is displayed at a time.

Once the population jumping rate starts to slow down, choose another subset and use crossover to create creatures with better genes.

Crossover should consist of taking two parent’s body lists and randomly choosing indices to swap. Attributes are connected to blocks so they’ll swap as well. Make sure axis is correct after crossover.

Display new child in Gui and record it’s fitness. If it got worse, delete creature. If it got better, then just keep repeating earlier steps.

Graphs should be recording all info over the course of time, saving creatures every hour.