

Assignment 8 ReadMe

Nicket Mauskar

February 28, 2023

1 Summary

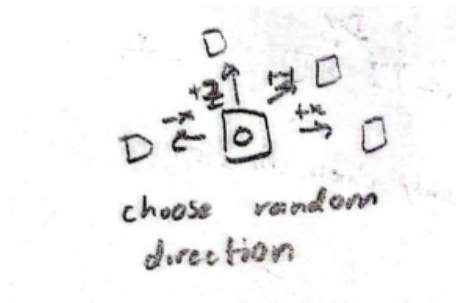
In this assignment, I was tasked with using the parallel hill climber to design and evolve a random 3-d morphology and behavior for locomotion.

2 Reproduction Steps

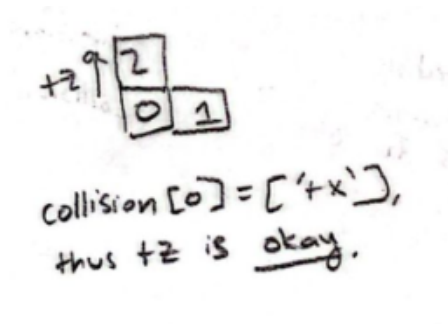
To run the code, simply run 'python search.py'. Rarely, an error pops up upon running with an out of bounds issue with np.random, but if you 'ctrl + c' and run it again, it will work.

3 Process

1. **Create randomly sized creature with variable sizes:** I did this by specifying the length/size of the creature to be a random number between 5 and 15. Any higher number would work, but just for the purpose of this submission, I chose 5 and 15. Also, I made the sizes of the cubes to be a random number between .2 and 2 to create some variability.
2. **Assigned neurons randomly and color coded accordingly:** I did this by creating a method in SOLUTION which takes the random number of total links(lets call this *numlinks*), and returns a list of a random length from 1...*numlinks*, in increasing order, of random numbers from 1...*numlinks*. I then used this list as a guide to show which links will have sensor neurons and motor neurons connected to it. I also calibrated the *send_cube* function to assign the colors of the sensed links to green and the nonsensed links to blue. Joint Axis' were randomized and implemented on each joint to provide variability as well.
3. **Generate a random direction and attach the link to a randomly chosen previous link with a collision check.**

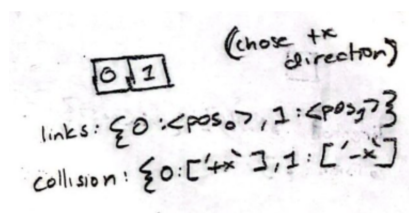


Generated a random number 1-5 which signified the directions of expansion (+x, +y, +z, -x, -y). After this, it randomly chooses a link from a dictionary of already established links (every time I add a link, I add the linkname: position into a dictionary). I also have a collision checker dictionary which keeps track of all the directions that a certain link has expanded to, and before adding another link, I check if the cube has been expanded in the randomly chosen direction already. If it has, then I choose another direction and try again. If not, then I continue to expand the creature with less chances of collisions.



4. Constantly update the links dictionary and collision dictionary.

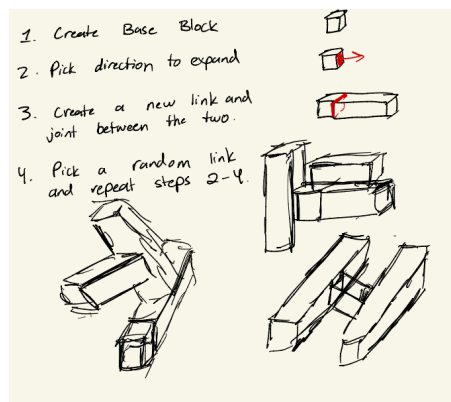
Everytime a new link is placed, I make sure to update these two dictionaries such that any time I add a new link to a random link from before, I have all the position, joint, and axis information of that parent link such that my child link will have the most optimal position and connection with said parent link.



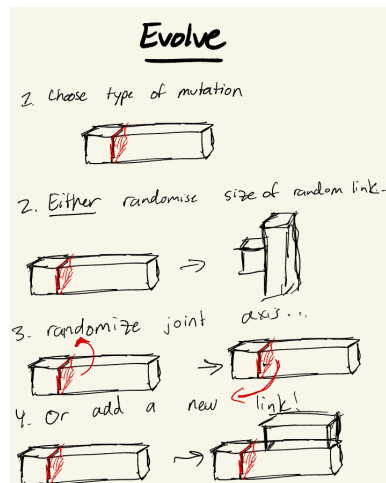
5. **Evolved/Mutated my creature across generations** In between every generation, the child had to undergo a mutation. This would include either changing the weights of the motor neurons as in past assignments, changing the size of a random existing link in the previous iteration, changing a joint axis among one of the sensor neurons, or adding a new link to the last made link. This way, the creature will undergo mutations in both it's brain(neuron values), as well as it's actual body composition(size, shape, movement).

4 Diagrams

1. Design and Build of 3-d Morphologies



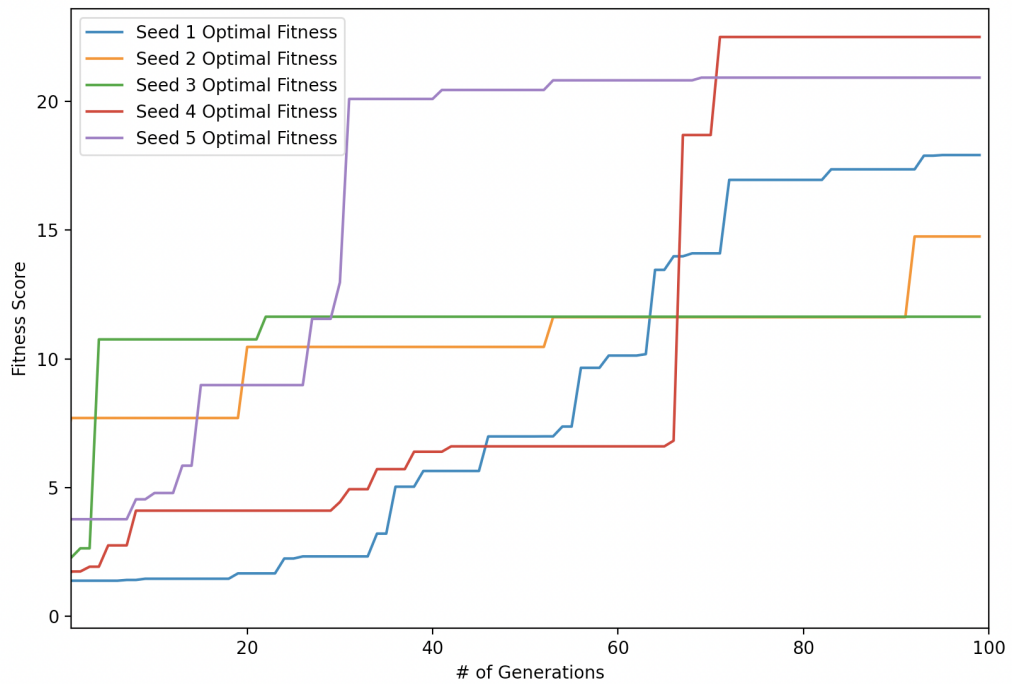
2. Evolution/Mutation of 3-d Morphologies



Note: Inspiration for designs from Karl Sims

5 Results

Below is a graph that portrays various 3-d morphology's, each starting from a different random seed, showing the fitness of the best creature in the population at each generation across 5 different runthroughs.



Note: this is **not** all from one run. It is a accumulation of 5 different runs in which the line shows the optimal fitness from the best creature at each generation (in most cases it is the same creature).

6 sources

1. r/ludobots - <https://www.reddit.com/r/ludobots/>
2. pyrosim - <https://www.thunderheadeng.com/pyrosim>