Assignment 8 ReadMe

Nikolai Medvedev (The Artist)

1. Summary

In this assignment, I sought out to design a random 3D creature's morphology and behavior for locomotion using the parallel hill climber.

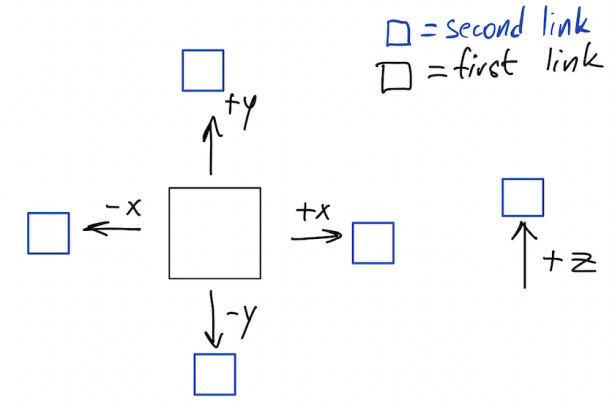
2. Reproducing Results

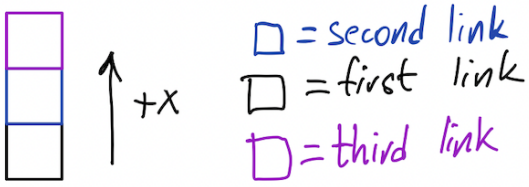
To run the simulation, you can use a terminal to enter the directory where the files are, typing in "python3 search.py". Alternatively you could also simply just run the file button.py or search.py. Since there is a rare error that sometimes pops up due to a UTC encoding bug, simply close and rerun the simulation if it crashes at any point.

3. System

1. **Creating a randomly sized creature with arbitrary sizes**
    a. In order to achieve this, I used random.random() to select a random number between 0 and 21 for the length of the creature and I then added 1 or 2 for width and height to create a randomly sized shape in order to have variability.
2. **Assigning neurons connections randomly and choosing colors**
    a. I accomplished this by creating a sensorList and then randomly choosing the number of sensor neurons (numSensorNeurons) and randomly assigning whether a link became a sensor or not in constants.py.
    b. Moreover, I calibrated my Create_Body() function to appropriately assign the correct color depending on whether the link was a sensor (green) or not (blue).

3. **Choose a random direction and then attach the link to the randomly selected branch's previous link**
    a. I generated a random number between 1 and 5 using random.randint() and then subsequently chose a direction that corresponded to that number in order: -y, +y, +x, -x and +z. Since I wanted to avoid collision with the floor I avoided choosing a -z direction. In keeping up with avoiding collisions, I also made sure to prevent links and shapes from forming within each other and expanding in the same direction. Therefore, I created a loop that checked if there was already a link coming out of the selected link in that direction. If the direction was the same, I would then pick another direction instead to expand in, significantly reducing the chances for unnecessary collisions during generations.





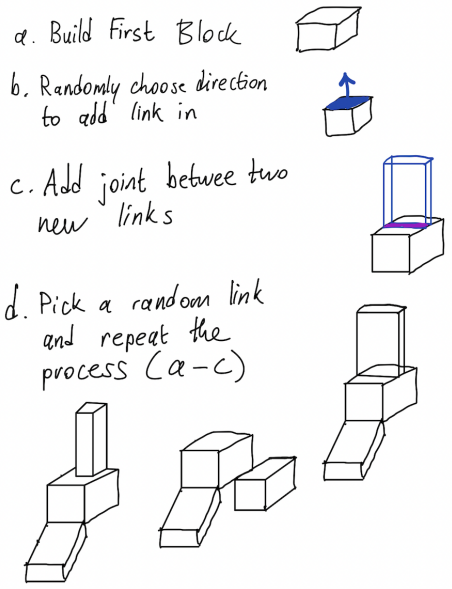
Since nextPos(□) == nextPos(□), there is collision and therefore adding to +x doesn't work.

4. **Constantly updating the links lists to keep track of any new link additions**
    a. The moment the first block of the entire body is generated, I use three lists (allSizes, directions and limbs) to keep track of the joint, position, and joint axis information of the first link and any other additional links in order to accurately adjoin the parent link to the child link and not mess up the creature's generation in the process.

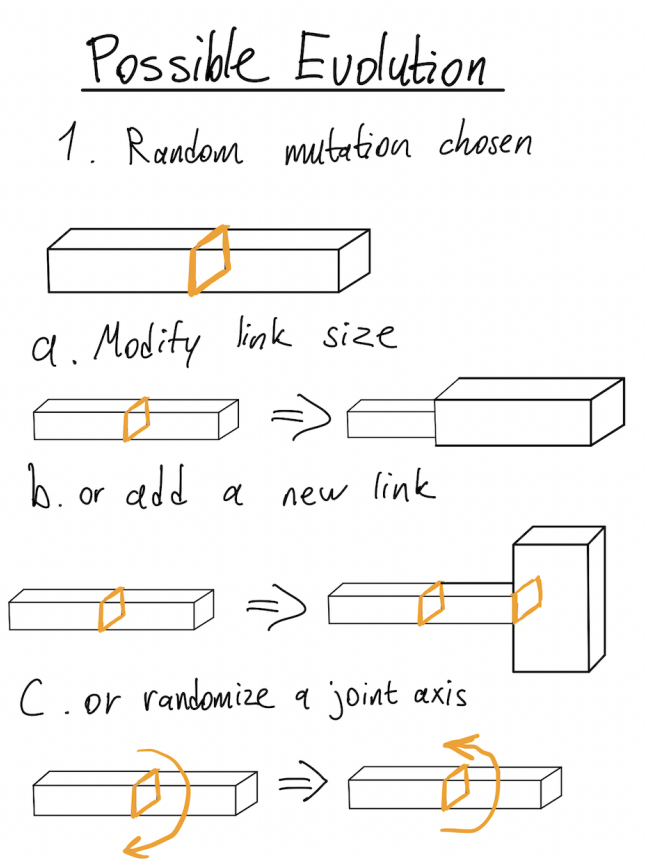
5. **Evolving and Mutating creature**
    a. With every generation, the child would undergo some sort of random mutation and its fitness would be subsequently tested, if the fitness of the child was better than that of the parent, then that child would become the new parent and the evolution would continue up until the end of the simulation. Possible mutations include: changing weights of motor neurons, link size, joint axis, or adding a link to the last made link. As a result, the creature would experience mutation in its physical body (shape, size or movement direction) and brain (change of neuron values).

4. Diagrams

1. Designing and Generating 3D Creatures

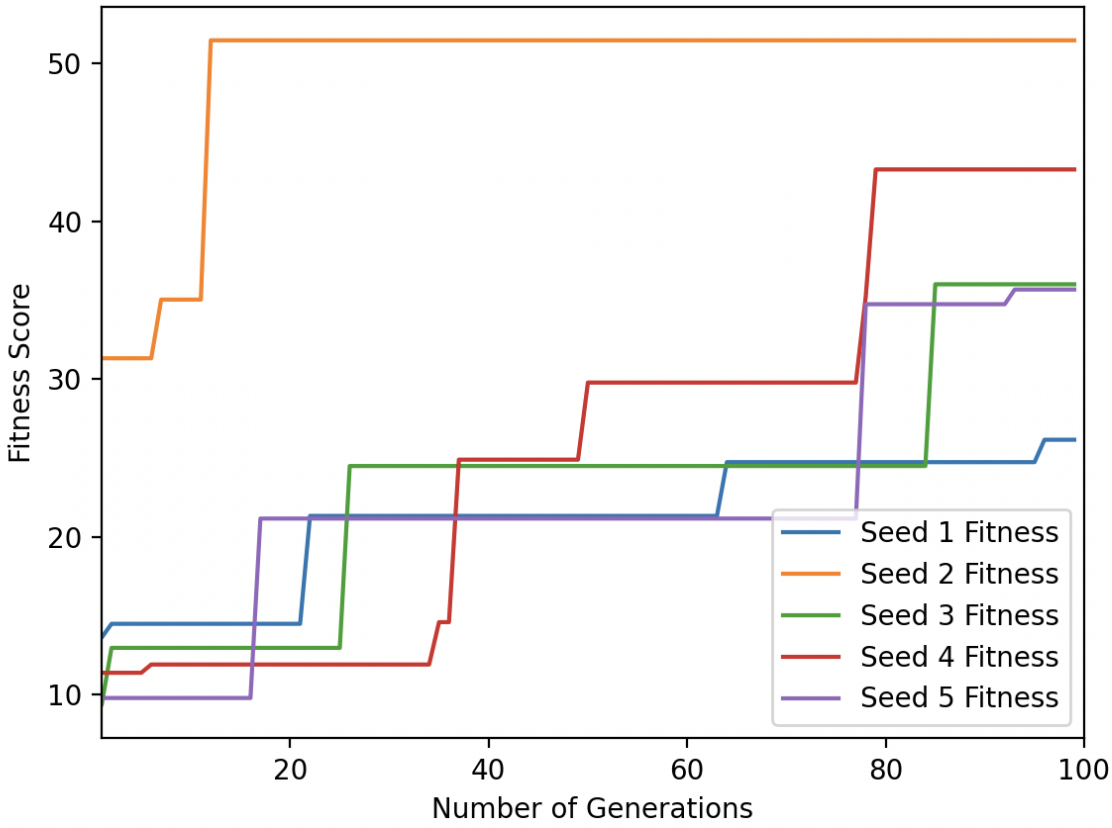


2. Evolution and Mutation of 3D Morphologies



5. Results

The following graph portrays an example of the various 3D morphologies. Starting with five different unique seeds, the graph shows the fitness of the best evolved creatures that came from 5 separate individual runs.



6. Sources

The following assignment could not have been completed without the following resources:

a. Ludobots - https://www.reddit.com/r/ludobots/
b. PyroSim - https://www.thunderheadeng.com/pyrosim