Bipedal God

Victor Bianchi, Isa Blancett, Subeen Kim, Shreya Rangarajan

BIG IDEA

Generate, simulate, visualize, and test physical bipedal systems using an evolutionary algorithm to see how these bodies will move over several generations.

BACKGROUND

The complex physics of a bipedal is interesting to model. Several factors need to be considered in order to precisely simulate a walking bipedal. Our group was interested in learning more about evolutionary algorithms and the necessary steps involved in defining a dynamic physical system that interacts with its environment, but first we learned about how bipedals work.

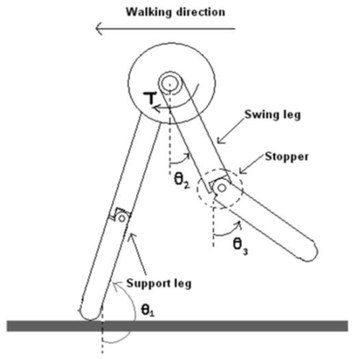


Fig. 1. Schematic of the biped mechanism.

Self-Excited Biped

Four motors:
- two at the hip joint
- two at the knee joint.

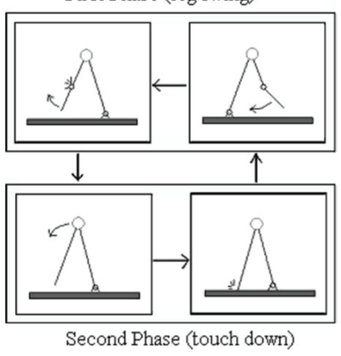
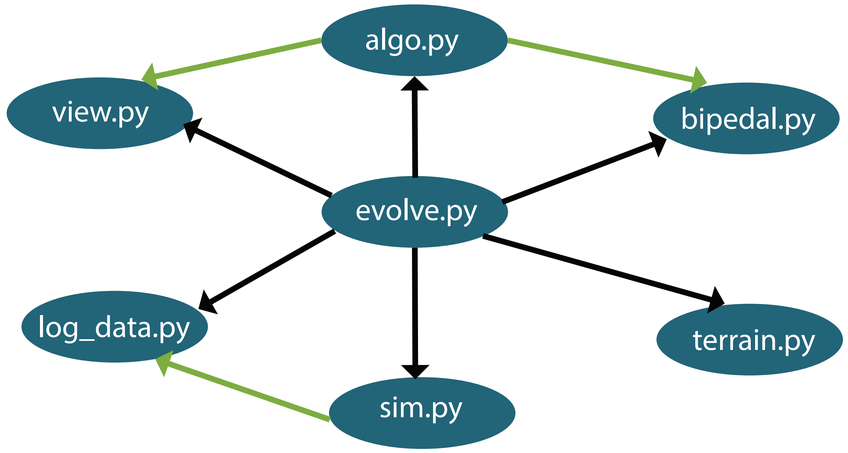


Fig. 2. Phases of biped locomotion [4].

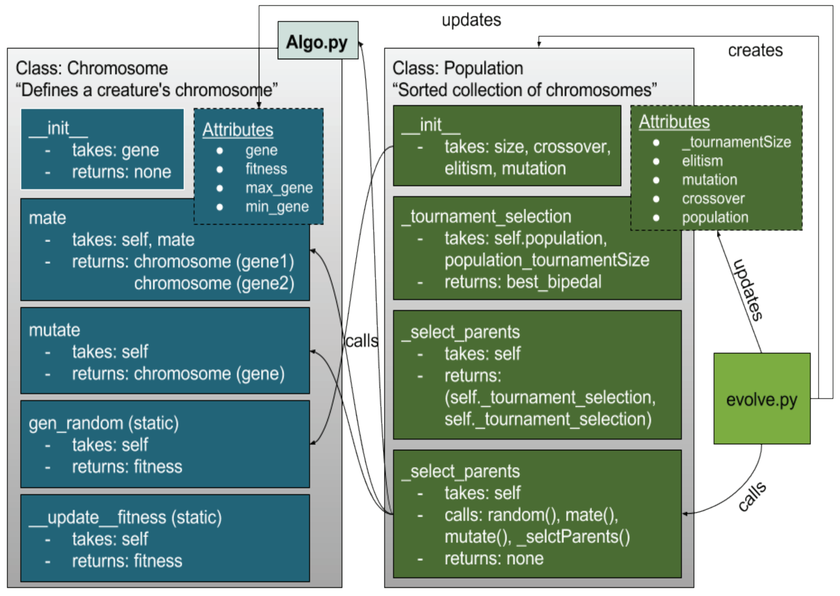
Biped Locomotion

Several important factors:
- joint movement
- angle/velocity of movement
- size of each thigh and shank
- Genetic information for evolutionary algorithm

IMPLEMENTATION

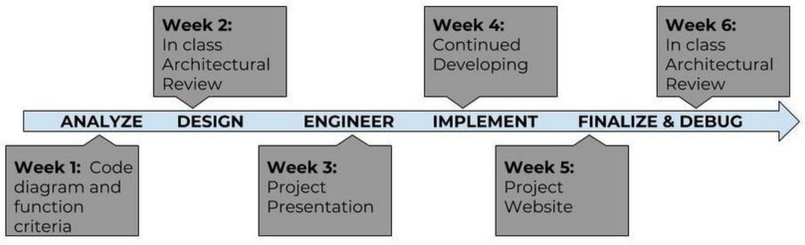


Our code is split up into seven files that are woven together and called in our main file known as evolve.py. The image to the left highlights an overview of what each file does and where they are called. Each blue circle represents a file name and the arrows represent how the files are connected. The arrowhead is pointed towards the file that the other file calls. For example, sim.py calls leg_data.py in the code.

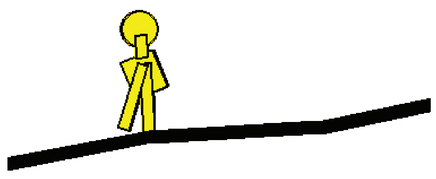


algo.py is the main file with all the code for our evolutionary algorithm, which has two main classes: Chromosome and Population. On th left is a basic structure of how we arranged the code for the algo.py file. Since this is where the evolutionary algorithm part of the code happened, it was the most interesting to structure. Note that not all the functions that are in the code are shown in this figure. Mainly the ones that we felt were important. Each class has an initializing function and and in the class Population, the init function actually calls the gen_random function. Using this algorithm, the bipedal is evolve overtime and get better at walking along the terrain.

TIMELINE



RESULTS



Here is our final product: a bipedal with terrain that was rendered using pygame. After being called in the algo.py file, the bipedal will have a certain fitness and will evolve overtime to become better at walking in any kind of terrain

ATTRIBUTES

we would like to acknowledge OpenAI gym resources and the evolution2D github for having code that helped us through this project. In addition StackOverflow was a big help for debugging.