Hexapod Walk

- Generating loopable walking animations for a virtual robot

Step 1: Defining a mathematical model
The most important thing to consider for a model was, that the walking animations must be loopable. This means that all joints of the robot must have the same rotation at the beginning and the end of the movement, which makes using a physics simulation impossible, because it is very unpredictable.
Therefore, the robot is moved by inverse kinematics, which means that a robot only knows, where it has to put its "feet" on the ground and moves the joints accordingly. Therefore the robot can also control, that after one cycle it moves all joints to the initial rotation, so it can start the walking cycle again.
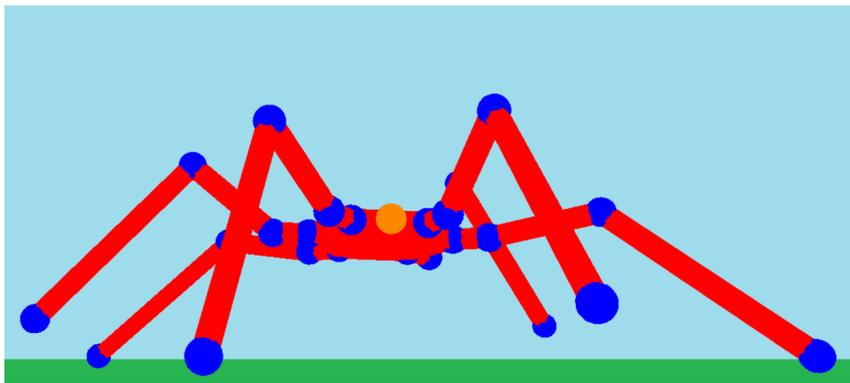
3 feet are always on the ground, so the robot stays in balance.

Step 2: Finding a gait
Hence hexapods have six legs, insects could be used as a guideline for the movement pattern. The tripod gait, in which three legs are on the ground and the other three legs are in the air to move to the new spot, seemed the best gait, because it is very fast, but at the same time guarantees a robot/insect not to loose balance.

The parameters which specify a tripod movement are:

- height
- rotation
- distance
- leg positions (1 per leg)

Step 3: Simulating
During the simulation, the fitness value of a movement is divided into two parts:
The first phase is the general part. The fitness value is decreased by one for everything which is wrong with the robot (wrong angles, self-intersections, etc.).
The second part is only calculated if there was no error with the movement. This part specifies what type of movement shall be learned (fast, flat, only lateral, etc.).
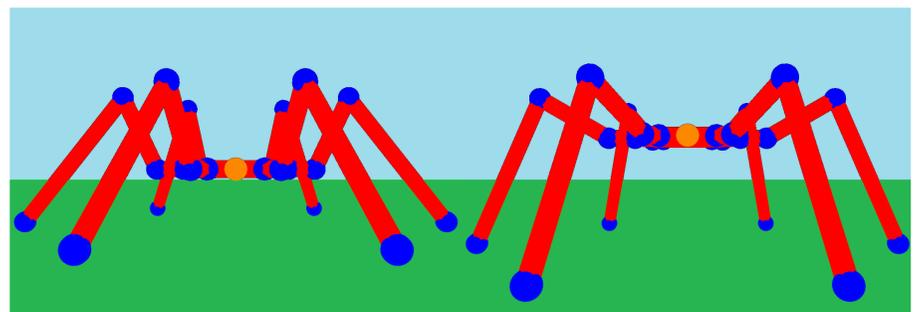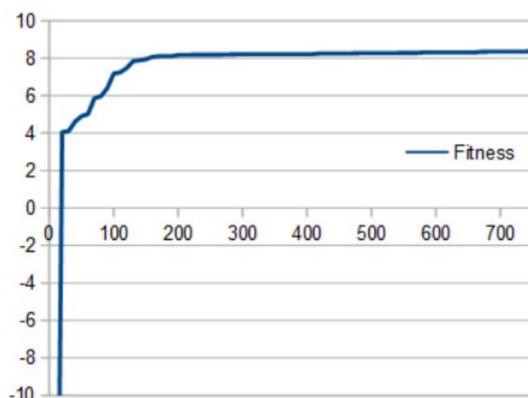
Different poses for different fitness functions

The fitness value converges very fast and then only increases very slowly, because now, several good mutations must happen at the same time, which is very unlikely.

Plans for the future
Other walking gaits should be implemented (especially one, where the front feet move first, then the middle feet and at last the back feet; this could be useful for overcoming little obstacles).
More (and especially more useful) fitness functions should be created.
Improving interaction with the program, inserting objects into the scene to find a sequence to navigate between points.

Used technology
Hexapod Walk is programmed in Java, for the visual representation jME 3.1 is used.

Related project
This project is related to **Hexapod Battle**, where the generated animations are used.

Tobias Wolf - tw067