Project 1 IT3708:

Flocking and Avoidance With Boids

Implementation

The Simulation is implemented in Python, using Zelle's graphics library[1]. A boid is represented by a class, which has predator as a subclass. Additionally, there is a data class for obstacles. The boid class consists of one function to calculate each force, as well as the function *updateBoid*, which sums up the forces and updates velocity and new position of the boid. The subclass predator only adds a function *updatePred*, which replaces *updateBoid* for predator instances.

The main function creates a hardcoded number of random boids in a window which represents the world and draws a GUI with adjustable scales for the weights as well as buttons to control predators and obstacles in a seperate window. In the main loop it is searched for every boid the neighbors and predators that are within its radius, as well as the obstacles, that are in front of it and within its radius. The same is done for each predator and the lists of objects, together with the via the GUI set weights and radiuses are given to the *updateBoid* and -*Pred* functions. Afterwards the graphics are updated.

The forces are calculated as follows:

- Separation

$$f_{sep} = \sum_{n \in neighbors} \left(\frac{distance_n - radius}{radius}\right)^2 \cdot (\vec{n} - \overrightarrow{boid})$$

where d_n is the distance between the boid and its neighbor n, $radius$ is the radius in which the boids can sense. $(\vec{n} - \overrightarrow{boid})$ is the vector between the boid and its neighbor.

This effects in the boid steering away from neighbors, the closer the stronger the force.

- Alignment

$$f_{align} = \sum_{n \in neighbors} velocity_n$$

This forces the boid to steer in the average of it's neighbors steering direction.

- Cohesion

$$f_{coh} = \left(\frac{\sum_{n \in neighbors} velocity_n}{|neighbors|}\right) - \overrightarrow{boid}$$

This results in a force directed to the midpoint of all neighbors positions.

The fleeing and obstacle-avoidance force are implemented analog to the separation force.

1 http://mcsp.wartburg.edu/zelle/python/

Behaviour in the given scenarios

- Scenario 1: high cohesion

  The boids start moving towards each other until they are on top of each other. They move around in stacks, that merge.

- Scenario 2: high alignment

  The boids form unorganized flocks that move in the same direction. After a short time, all flocks move to the same direction.

- Scenario 3: high separation

  The boids move around unorganized, keeping at least their sight radius as a distance to each other. If there are enough boids, they form a grid-like pattern without much movement.

- Scenario 4: high alignment and cohesion

  Same as in scenario 1 but they tend to walk to the same direction more and less straight towards each other (it takes them longer to merge).

- Scenario 5: high separation and cohesion

  If both are equally high, the boids are moving unorganized, keeping a distance of at least the sight radius. If the separation weight is lower, they stay in very loose small flocks, keeping the radius as a distance to each other.

- Scenario 6: high separation and alignment

  The boids form loose big flocks that move in the same direction, keeping the set radius as a distance to each other. They seem to never merge to one flock.