Project 1

Subsymbolic Methods in AI
Håkon Gimse

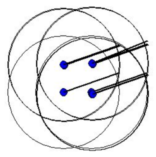
Architecture and Implementation

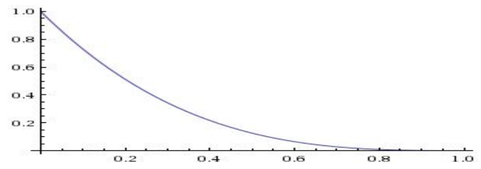
My architecture consists of one GUI class, one flock controller, and several different objects (boids, predators and obstacles). The GUI class is the main class, in charge of rendering the simulation. This class instantiates a flock controller class. This class is used for controll of several objects. It stores a number of boid instances, predators and obstacles. The flock controller serves as a middleman between the gui and the objects rendered. If you want to add a obstacle, you click the button in the GUI, and the gui asks the flock controller to create a obstacle object and store it with the other obstacles so every predator and boid becomes aware of it when nearby. It is also used to calculate the different forces when each boid or predator should look at each other boid or predator. The obstacle class is the most basic, with just a radius and a position. Boids and predators are more advanced and has a velocity and some methods for calculating new behaviours. The predator class is extending the Boid class as use mostly the same calculations.

Forces

Separation

I wanted the separation force to be very large when objects were very close, and then drop fast as they became further away. To achieve this curve as shown below I used the formula $Sep = maxVelocity * ((1 - relative\ distance)^3)$ where the ralative distance number is between 0 and 1 found by dividing the distance between the position of the boid and the position we want to separate from and the maximum distance of neighbours.
This maximum is the radius of the neighbourhood + the radius of the object.
As the point to separate from I have tried to use both the average position of neighbours and just the closest object, and I found that the best behaviour in my system came from using only the closest object. When using the other solution I more often than not got boids in certain patterns, laying on top of each other as this was a flocks optimal configuration. This is illustrated here with 12 boids:





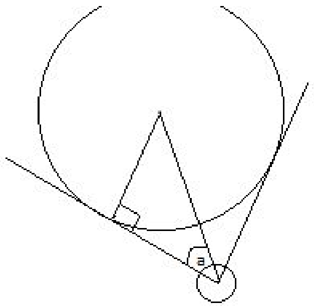
Alignment

To calculate the allignment I simply find the average velocity vector of all the neighbours, than I find the difference between the current velocity vector and the average and this result multiplied by the alignment weight is returned as the new force. I have found this to work very good, and the boids conform relatively smoothly to each other as they are supposed to.

Cohesion

I wanted this gradient to be more even than the separation gradient. To calculate this I used the formula $Coh = maxVelocity * relative\ distance$. The ralative distance is the same as the one I used in separation, but the “attracting” point is the average position of neighbour boids.

Obstacle Avoidance
I did this with focus on the smooth avoidance as described in the assignment. To achieve this I found the angle between the boid and the optimal point to pass the obstacle. Then I used this angle to get a velocity vector that matched. This vectors size match the old velocity. Before doing this I check if the boid is on collision course, this is done by comparing the angle of the current velocity against the angle between the obstacle center and the boid plus-minus the angle needed for avoidance. The perfect angle is found using simple trigonometry as shown here:



Here the angle $a = asin(obstacle\ radius \ /\ center\ to\ center\ distance)$

Fleeing from predators
When a boid is to close to a predator it should flee wildly. When this happens, I use the exact same formula for calculating cohesion, but reverse it. When calculating this cohesion the "attracting" point is the average of close predators.

Emergent Behavior

Scenario	Separation	Alignment	Cohesion	Result
1	Low	Low	High	Concentrated flocks with many boids. Low velocity as direction is conflicting. Becomes aligned after a while.
2	Low	High	Low	All boids same direction.
3	High	Low	Low	Entire screen evenly divided between the boids. A lot of movement back and forth. No flocks.
4	Low	High	High	Many small flocks very quickly. Perfectly aligned and laying on top of each other. Takes a while to find other flocks.
5	High	Low	High	One large flock emerges, and includes every boid. This flock has no direction.
6	High	High	Low	Every boid perfectly aligned and with a perfect distance on each side, so the entire environment is covered by someones neighbourhood.