Sub symbolic AI methods: Project 1

Flocking and Avoidance with Boids

Bjørnar Walle Alvestad

2016

Norwegian University of Science and Technology

NTNU

Introduction

This is a technical report for project number 1 of the course IT3708 "Sub symbolic AI methods". The report will cover technical details about the architecture and implementation of the assignment, and describe key behavior of the simulation.

Architecture

A class diagram for the software is shown in figure 1. The entity class is the superclass of boid, obstacle and predator. It contains a position vector, a velocity vector and methods for calculating forces, which is used by the boid class to simulate flocking behavior. Predator inherits the boid class, because of their technical similarities. A few modifications in the predator class ensures the distinct predatorily behavior.

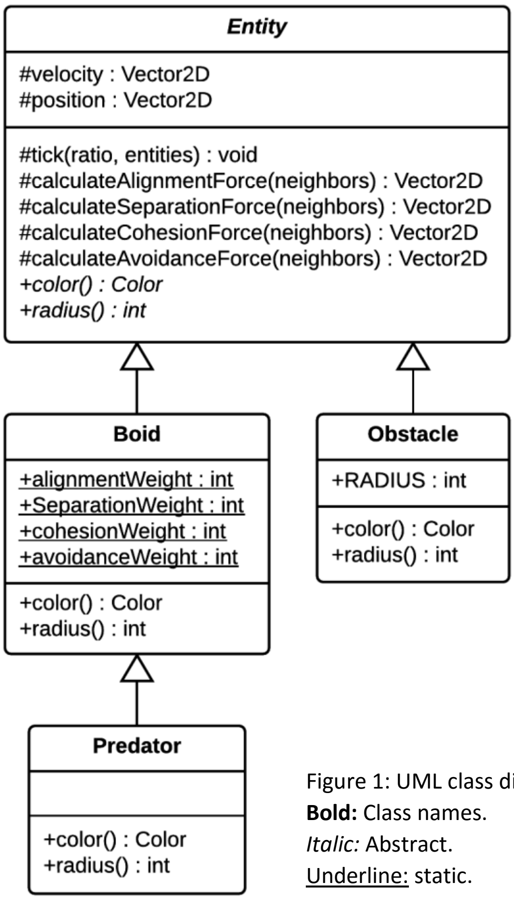

Figure 1: UML class diagram.
Bold: Class names.
Italic: Abstract.
<u>Underline:</u> static.

Implementation

This assignment is implemented in the Java programming language. Graphics and the user interface are powered by the Java swing toolkit.

The main loop of the application is designed to simulate the boid behavior in regular sized time steps, which can be changed by the user while the simulation is running. The simulation thread will call the tick method of each entity, supplying the time step ratio in seconds. That way, the simulation can run in a constant pace, regardless of framerate.

Calculation of forces

Separation

Separation of boids is calculated by forming a vector, representing the distance between the two boids in question (red arrow). This vector is normalized to length 1, then scaled by the inverse ratio of the distance between the two boids. This ratio is equal to 0 when the boid is located on the edge of the local radius, and 1 when it is located in center of the local radius. This ratio is scaled exponentially, so that the repulsive force is far greater near the center of the local radius. This calculation is carried out for every boid inside the local radius, and the resulting vector added to the result vector.

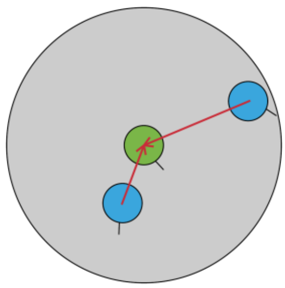

Figure 2: Boid (green) in the process of calculating the separation force.

Alignment

Because alignment is formed solely by the direction of the velocity vector, the alignment force is based on the average velocity vector of the surrounding boids. It is calculated by iterating through all boids inside the local radius, adding the velocity vector to a result vector. The result vector is then scaled by $1/(number\ of\ neighbouring\ boids)$, and subtracted by the velocity vector of the center boid. This forms the delta velocity between the center boid and the average velocity of nearby boids, and is used as the alignment force for the center boid.

Cohesion

The cohesion force is calculated in the same way as the alignment force, but instead using the position vector. The average position of all local boids is calculated, then subtracted by the position of the center boid. This forms the delta position between the average position and the center boid.

Obstacle avoidance

The obstacle avoidance force is based on the distance between the velocity line, and the center of the obstacle. First, a line that is parallel to the velocity vector is formed (velocity line). Then the perpendicular distance between the line and the center of the obstacle is calculated (d). If this distance is less than the radius of the obstacle and boid combined, it's certain that the boid is going to intersect the obstacle in the future. A vector parallel to the velocity line is then formed, and rotated 90 degrees so that it will point away from the obstacle. The direction of rotation is determined by the sign of the distance d, which emerges from the equation of perpendicular distance. This vector is scaled based on the distance between the obstacle and the boid, making the avoidance force stronger if the boid is very near an obstacle.

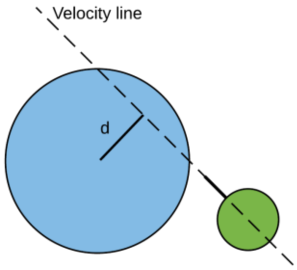

Figure 3: Illustration of the perpendicular distance between the velocity line, and the obstacle center point.

Fleeing from predators

When a predator enters the local radius of a boid, the boid will respond by fleeing in chaotic way. This is implemented by greatly increasing the separation force generated from predators. Additionally, the boid will enter a "panic" state, which induce higher movement speed and a small, random rotation of the velocity vector. This will create a panic effect, which wears off gradually over 1 second as the predator leaves the local radius of the boid.

Emergent behavior

Scenario	Separation	Alignment	Cohesion	Behavior
1	LO	LO	HI	Boids form tightly packed clusters, resembling a "swarm" of flies. After a while, the clusters stabilize to a unified ball of boids, moving in one direction. Clusters are easily de-stabilized by other clusters.
2	LO	HI	LO	Boids moving in a unified pattern. The boids don't collide, because the movement direction of the boids is changed instantly due to the high alignment force.
3	HI	LO	LO	All boids are moving freely and independent, and are spread evenly across the map due to the high separation force. After a while, the boids move in a common direction.
4	LO	HI	HI	Boids form tightly packed "balls", moving in a unified direction. Balls of boids merge smoothly when approaching each other.
5	HI	LO	HI	Boids form swarms resembling flies. They behave in a chaotic manner, but doesn't collide with each other. Swarm will move in a common direction after a while. Also, a big swarm will easily split into two smaller swarms.
6	HI	HI	LO	Unified movement direction. Boids are evenly distributed across the map, because of the high separation force. A very stable movement of the boids.

References