

Predator-Prey Simulation Using Boids Model

Matija Ojo, Miha Krajnc, Marko Adžaga, and Janez Kuhar

Collective behavior course research seminar report

January 6, 2024

The collective behaviors observed in nature, such as flocking, herding, or schooling, often serve as adaptive strategies that enhance the survival chances of individuals within a group. Understanding these natural behaviors serves as inspiration for designing autonomous agents capable of sophisticated interactions within a simulated environment. Our goal is to simulate prey and predator with different predator tactics (attack center, attack nearest, attack isolated, attacks from various directions, constant bearing hunting), escape maneuvers (split, hourglass, herd, vacuole, flash expansion, fountain) and parameters (perception radius, moving speed, turning speed) in order to conclude how different escape maneuvers affect predator's success.

This research paper however, will only focus on the basic boid (bird-oid object) simulation. This simulation will serve as a foundation for the more complex simulations in the future.

Collective behavior | Boids | Simulation | Prey-Predator | Escape patterns

Introduction

One of the most striking patterns in biology is the formation of animal aggregations. Classically, aggregation has been viewed as an evolutionarily advantageous state, in which members derive the benefits of protection, mate choice, and centralized information, balanced by the costs of limiting resources [2]. We would like to experimentally determine which flocking behaviors help the prey best defend itself against a predator.

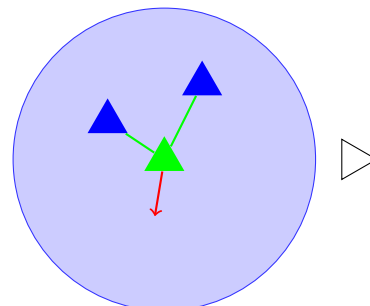
The flocking behavior can be simulated in different ways. For example, Heppner and Grenader [3] were modeling birds behavior with stochastic nonlinear differential equations. Oweis, Ganesan, and Cheok [4] took a different approach and modeled birds with a centralized logic (as in the server-client architecture). In 1987, Reynolds [5] proposed a simple algorithm, which was groundbreaking at the time, to model the flocking behavior of birds, herding of sheep, and similar phenomena, known as the Boids (Bird-oid objects) model. In contrast to controlling the interactions of the entire flock, the Boids simulation focuses on dictating the behavior of each individual boid. Despite consisting of a few simple rules, this algorithm produces complex and lifelike behaviors similar to those observed in nature.

Our research is based on a paper by Papadopoulou and others [1], which we will extend with the results of our predator and prey simulation. Although we are not using fuzzy logic to set the direction and speed of our boids, which makes the movement less natural, we have taken some elements for our model from [6]. Specifically, we've set the field of vision for our boids to 300° and implemented occlusion for the predator.

Methods

A boids model is implemented, along with some additional checks to add more realism. Boids model has 3 rules:

1. Avoid collisions.

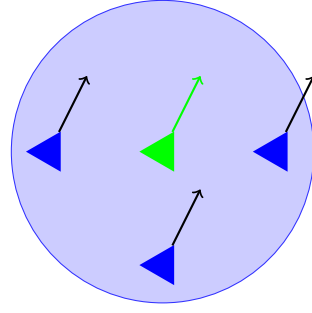


2. Maintain the same heading and speed as the neighboring boids.

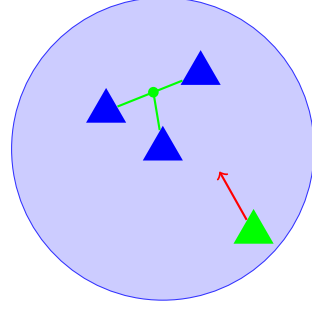
Collective behavior course research seminar report

Predator-prey interactions is of significant importance in biology and nature itself. The insights gleaned from this research can offer more than a theoretical understanding; they pave the way for the design and optimization of autonomous agents capable of adaptive and context-aware behaviors. The applications range from research in biology to simulations of large amounts of boids found in computer graphics.

Collective behavior | Boids | Simulation | Prey-Predator | Escape patterns



3. Gravitate toward the center of the flock.



Boids implementation. Each boid B has the following properties:

1. position - a vector in \mathbb{R}^2 , denoted by $position(B)$,
2. velocity - a vector in \mathbb{R}^2 , denoted by $velocity(B)$,
3. acceleration - a vector in \mathbb{R}^2 ; used exclusively for the internal computation of the boid's velocity and not for behavioral logic.

With regards to behavioral logic, we also assign the following attributes to all boids:

1. perception radius (denoted by r_P)
2. separation radius (denoted by r_S , also note that $r_S < r_P$)
3. perception angle (denoted by fov)

The *Euclidean distance* (1) is used for computing the distance between boids.

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}; \quad p, q \in \mathbb{R}^2 \quad [1]$$

To avoid the computation of the costly square root of a real number, we utilize an equivalent formula (2):

$$d(p, q)^2 = (p_1 - q_1)^2 + (p_2 - q_2)^2; \quad p, q \in \mathbb{R}^2 \quad [2]$$

Each step of the simulation loop updates the direction of a boid, which is then applied to its acceleration, determining the actual velocity for all boids.

Before applying the 3 mentioned rules, the neighbours (3) of boid B are all B_i for which the following formula holds:

$$d(B, B_i)^2 < r_P^2 \wedge \text{angle between}(B, B_i) \leq fov \quad [3]$$

Taking perception angle (fov) into account adds more realism, especially when interacting with predators.

The direction for collision avoidance, also known as *separation*, for boid B is computed by the following formula (4):

$$\text{direction} = \sum_{i=1}^n position(B) - position(B_i) \quad [4]$$

where i -th boid B_i is a neighbor of B and: $d(position(B), position(B_i))^2 < r_S^2$. This effectively means that boid B will move away (in the opposite direction) from the boids which are too near (closer than the specified r_S).

The direction for *alignment* for boid B is computed as (5):

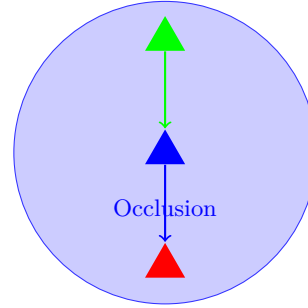
$$\text{direction} = \frac{\sum_{i=1}^n velocity(B_i)}{n} - velocity(B) \quad [5]$$

where i -th boid B_i is a boid such that: $d(\text{position}(B), \text{position}(B_i))^2 < r_P^2$ (i.e., B_i is B 's neighbour).

The direction for *cohesion* for boid B is computed like so (6):

$$\text{direction} = \frac{\sum_{i=1}^n (\text{position}(B_i) - \text{position}(B))}{n} \quad [6]$$

Occlusion algorithm. In order to add more realism, the occlusion algorithm was implemented. Because the algorithm is quite long with some edge cases, we will only explain the idea of the algorithm, which is quite simple. Given a list of potential neighboring boids, we must determine which are occluded and in turn take only the nearest (non-occluded) boids as neighbours. This is done by iterating through the list of neighboring boids (of boid B) and computing the angle between i -th (B_i) boid and all the others (in j -th step B_j). If the angle difference is below a threshold, boids B_i and B_j are considered occluded. To determine which boid occludes the other (is the actual neighbor), minimum distance is computed: $\min(d(B, B_i), d(B, B_j))$.



Turn speed. In order to add even more realism, the **turn speed** of a boid is limited. Whenever the acceleration of a boid is computed, the angle between the acceleration vector and the current heading vector is checked. If it exceeds a threshold, the old heading is rotated by the maximum amount in the given direction and scaled by the magnitude of the acceleration. Therefore boids have a maximum value in which they can turn at each step of the simulation.

Escape maneuvers. The 2 simplest escape maneuvers are implemented: escape based on position and based on velocity (direction) of the predator.

The position based escape maneuver works exactly the same way as separation, except the position of predator is taken into account.

The direction based escape maneuver works such that it determines on which side of prey lies the predator, and going into a perpendicular direction in the opposite side.

Predators. 2 simple predator behaviors (strategies) are implemented: attack centroid, attack random prey.

Results

By combining all the features adding upon realism and the escape maneuvers, we can obtain a realistic pattern **split**. This pattern is obtained by using the **position** based escape maneuver and any predator tactic.

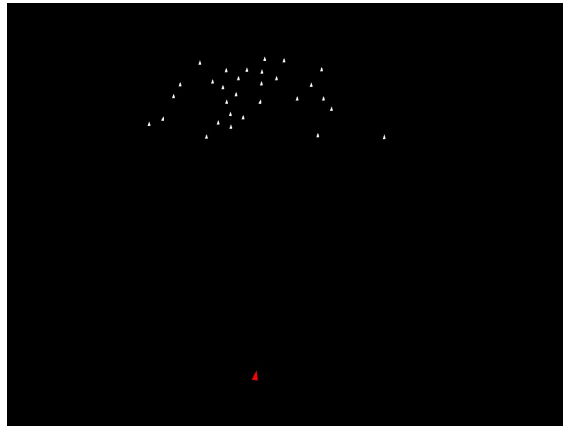


Figure 1. Predator chasing a group of boids (step 146)

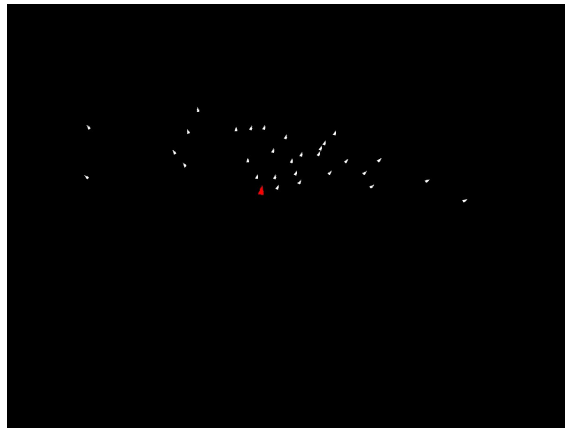


Figure 2. Start of split pattern (step 233)

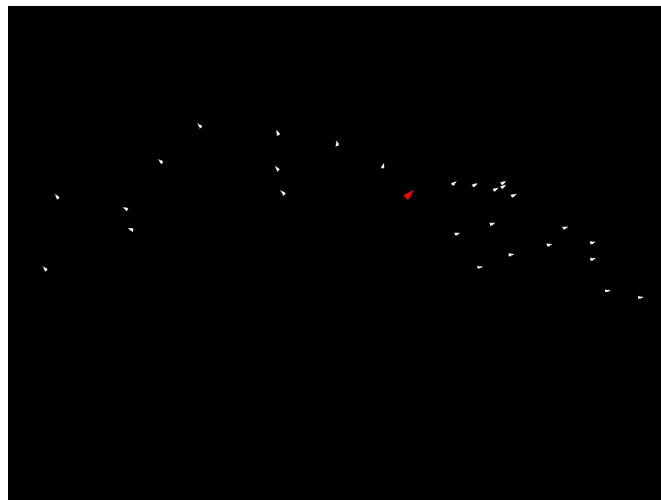


Figure 3. Predator caught a prey, split continues (step 261)

Discussion

This concludes boids implementation with additional realistic features. The simulation itself was heavily parameterized as well, making tweakings easier and more reproducible.

There is still room for improvement in the visualization of the simulation (add traces, add predators target, ...).

The most important aspect which remains is the implementation of different escape maneuvers and predator tactics and the comparisons of the latter.

CONTRIBUTIONS. Matija Ojo: Add realistic features, fix escape maneuvers, report, Miha Krajnc: Escape maneuvers, Janez Kuhar: report, Marko Adžaga: Researching sources and report

Bibliography

1. Papadopoulou M, Hildenbrandt H, Sankey DW, Portugal SJ, Hemelrijk CK (2022) Emergence of splits and collective turns in pigeon flocks under predation. *Royal Society Open Science* 9(2):211898.
2. Parrish JK, Edelstein-Keshet L (1999) Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science* 284(5411):99–101.
3. Heppner F, Grenander U (1990) *A Stochastic Nonlinear Model for Coordinate Bird Flocks*.
4. Oweis S, Ganesan S, Cheok K (2014) Illustration of centralized command and control for flocking behavior. *International Journal of Handheld Computing Research* 5:1–22.
5. Reynolds CW (1987) Flocks, herds and schools: A distributed behavioral model in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. pp. 25–34.
6. Demšar J, Lebar Bajec I (2014) Simulated Predator Attacks on Flocks: A Comparison of Tactics. *Artificial Life* 20(3):343–359.