Colorboids

Genevieve Waldorf and Matthew Finlayson

Goals

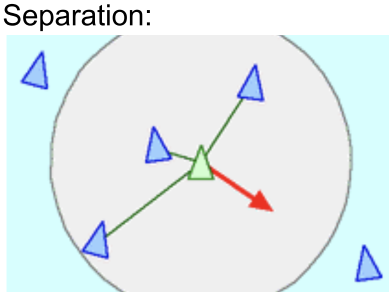
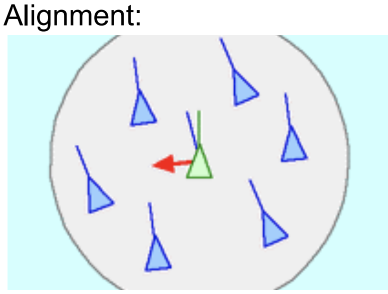
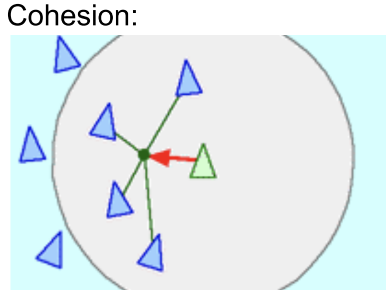
The goals of our project are to
- Learn some Unity
- Implement the boids algorithm to produce emergent flocking behavior
- Extend boids with additional features to make them more interesting

The boids algorithm

The boids algorithm is a distributed algorithm for generating emergent flocking behavior from three simple rules. This algorithm was developed by Craig Reynolds in 1986 and can be used to simulate flocking organisms like birds and fish.

The simulation requires a number of objects called boids. Each boid is given three rules:

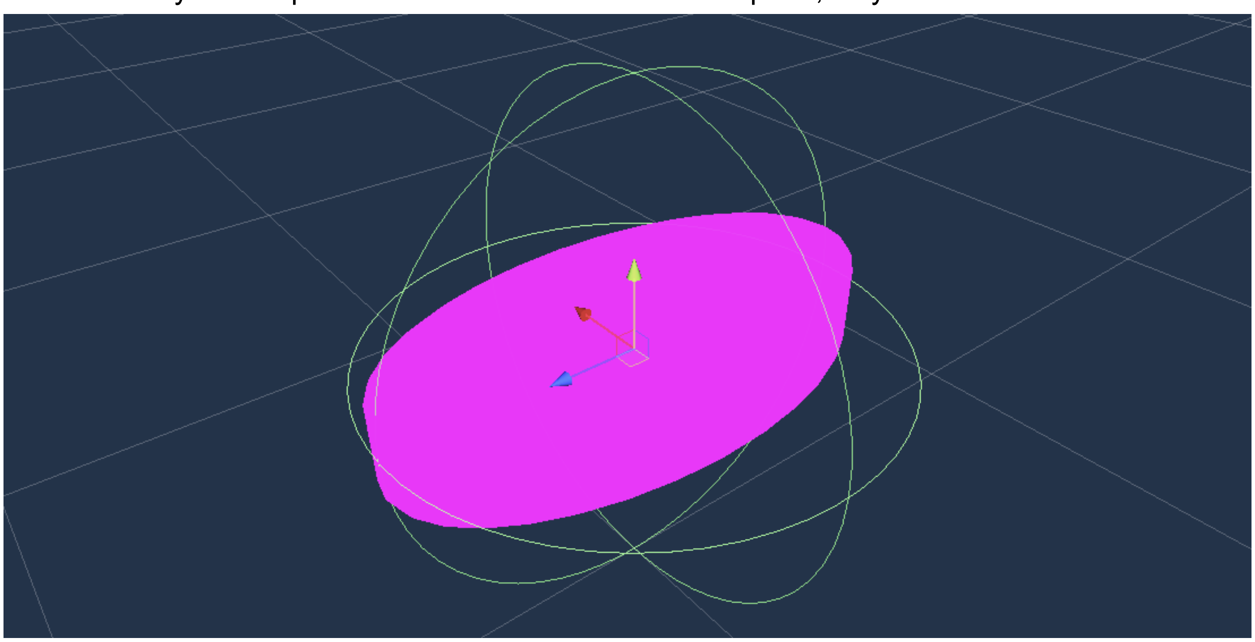
- Separation: Do not get too close to other boids
- Alignment: Align your direction of travel with your fellow boids
- Cohesion: Move towards the average position of other boids

Separation:	Alignment:	Cohesion:
		

Each boid has a limited sensing area, so they are not aware of all the other boids in the flock, only those closest to them.

Our implementation

To implement the boids algorithm, we created an environment in Unity and used a prefab to generate ~30 identical puck-shaped boids. The user interface provides 3 sliders that control the strength of the boid's preferences for separation, alignment, and cohesion. The boids are free to move about the environment, and move forward the same distance every timestep. The boid rules do not affect the speed, only the direction of the boids.



To turn the boids, each rule provides a vector for the direction the boid should travel. The vectors are converted to quaternions that preserve the right-side-up-ness of the boid's rotation, and then are linearly interpolated (according to the rule strength) with the boid's current direction, one at a time.

Separation is implemented by finding neighboring boids that are within a close sensing distance of the boid. We evaluate the direction of the boid by negating the average position of the neighboring flockmates.

Alignment is implemented by finding neighboring boids that are within a sensing distance of the boid. (This sensing distance is larger than the distance used in separation). We evaluate the direction of the boid by slerping between the boid’s direction and all of its neighboring flock directions.

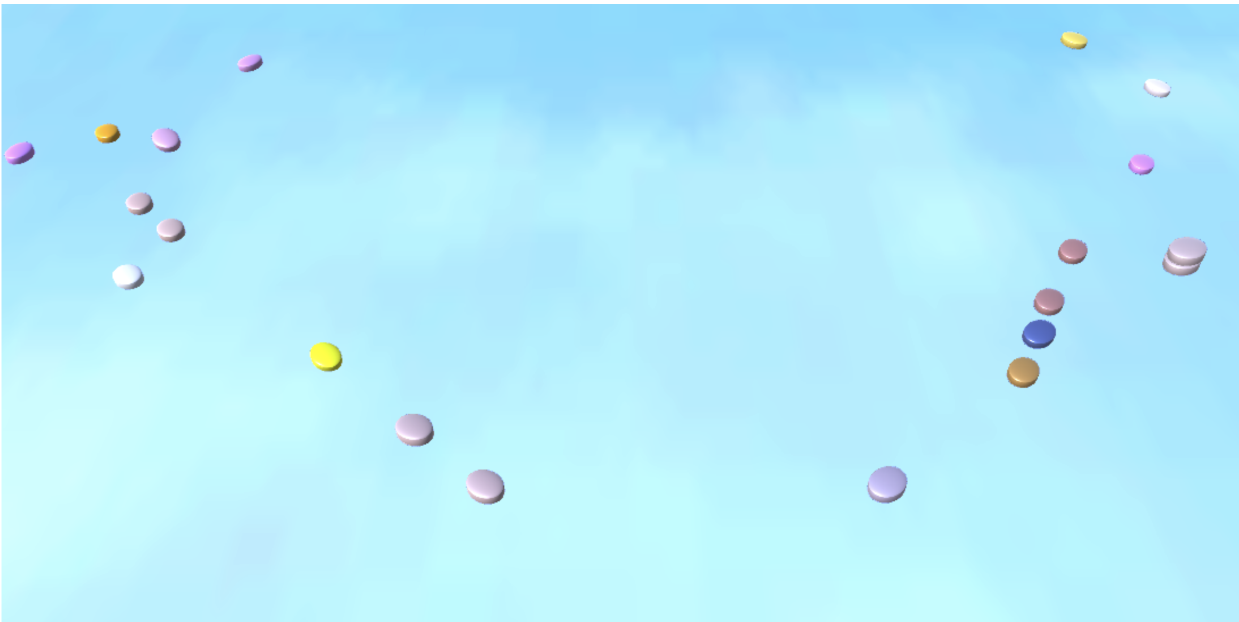
Cohesion is implemented by finding neighboring boids that are within a sensing distance of the boid. (This is the same sensing distance used in alignment). We evaluate the direction of the boid by taking the average position of the neighboring flockmates.

We began with an enclosed environment with walls and a ground. When handling the boids behavior when colliding with the walls, we found the interactions to look unnatural. We decided upon a more realistic environment where the boids flew around the sky, free to move on and off the screen without harsh collisions with boundaries. In order to keep the boids from moving infinitely far away, we implemented an attraction component to the boids behavior. When boids wander too far from the attractor, another rule is added which strongly encourages them to turn towards the attractor. This gives the effect of the boids gently “bouncing” off some invisible walls, while allowing them to still leave the area and move around freely.

Boid behavior

By adjusting and combining the boid rules, we were able to produce a number of interesting behaviors. We describe and illustrate a few below.

No rules: The boids end up bouncing around the edges of the boundary. They move through each other and do not affect one another.

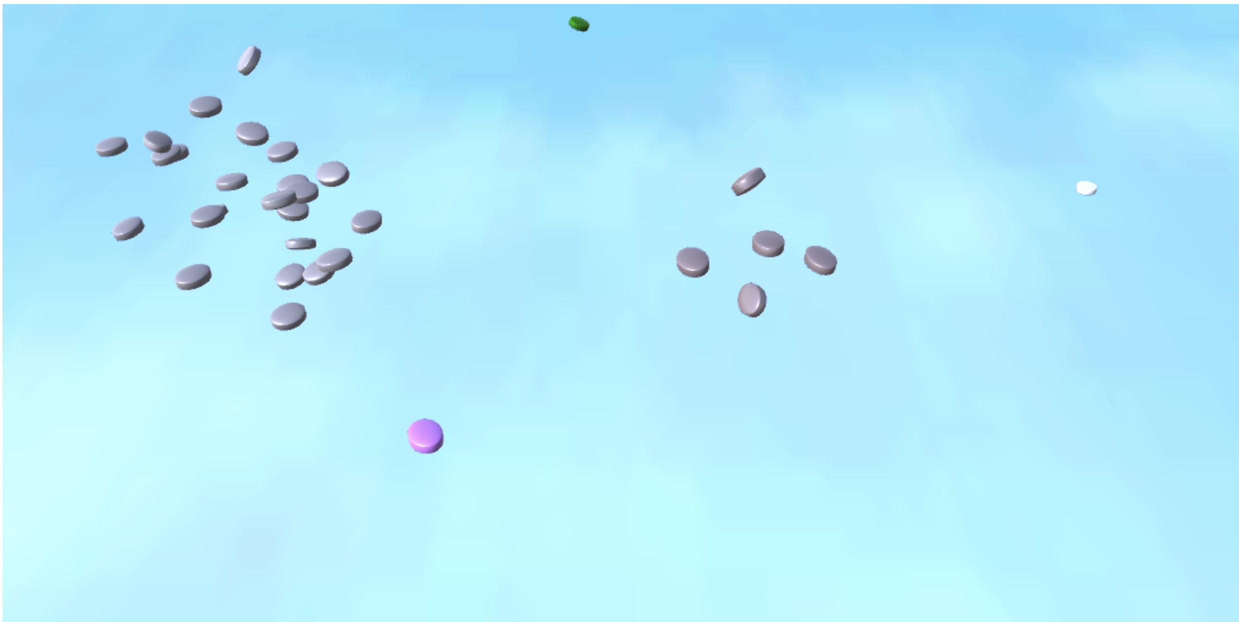


Separation: The boids separate from each other and no longer move through each other. This causes them to distribute about the environment much more evenly because they push each other away from the edges. The boids are seen randomly moving throughout the space without colliding into one another.

Alignment: The boids group up into bunches as they encounter each other, but stay together only because their trajectories match up.

Cohesion: The boids form small swirling groups that circle each other endlessly. The stronger the cohesion, the tighter the groups.

All rules: The boids form flocks that come together and break apart as the rules interact and fail to converge. The result is reminiscent of schools of fish, or flocks of birds.



For ideal, interesting flocking behavior, we suggest setting separation and alignment all the way up, and cohesion about half-way up.

Extending boids

To make our simulation more interesting, we decided to add some extensions.

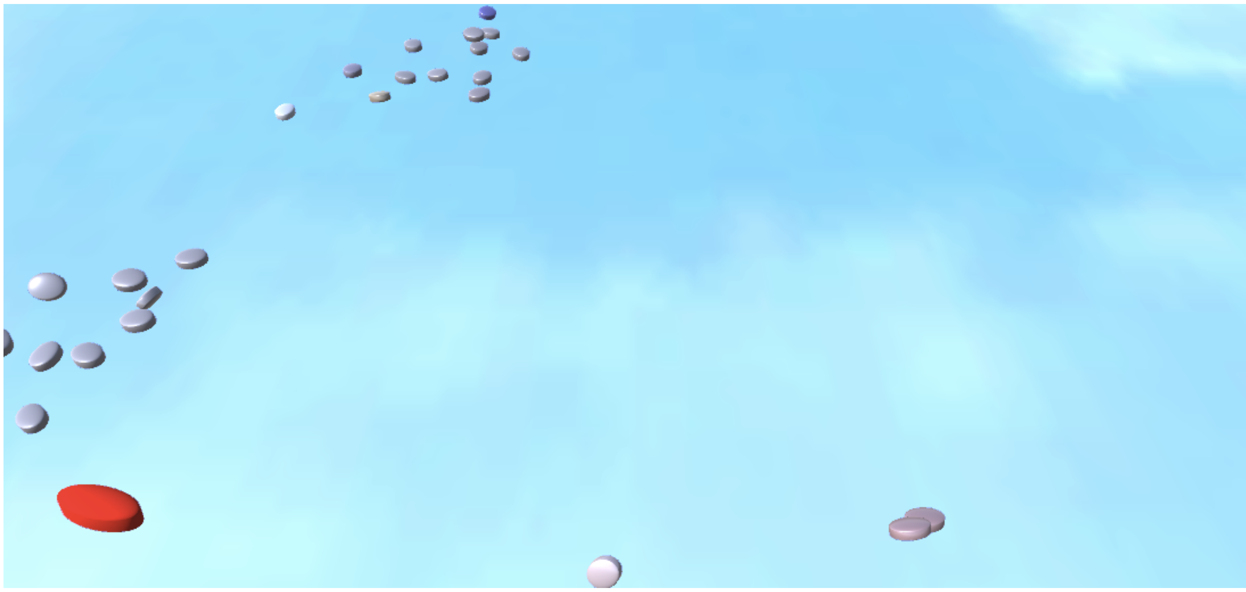
Predator

We added another game object, the predator, which pursues the boids. The predator has a greater sensing distance than the boids and follows these rules:

- Move towards the average position of the boids you can sense
- Move towards the attractor if you are too far from it.

Adding a predator, we observed some interesting interactions. The stronger the flocking behavior of the boids, the harder it is for the predator to get near any single boid. This is because when approaching a group of boids, the boids scatter and regroup, causing the predator to split its attention between multiple boids. Without flocking, the predator is able to pursue individual prey and get closer to them.

In the image below, we see the predator (red) chasing a small flock of boids, while another flock happily swirls in the background.



Color sharing

Each boid begins with a distinct color. The preset color sensing distance determines which neighboring objects are part of that particular boid's color flockmates. Once the color flockmates are determined, the color values of all of its members are averaged. The boid then alters its color slightly in the direction of the averaged color. This is done by taking the averaged flockmates color minus the current boid color divided by a preset constant.

$$color\ change\ =\ (averaged\ color\ -\ current\ color)\ /\ colorDiv$$

The subtle change in color is evaluated by taking the color change RGB values and adding them to the RGB values of the boid's current color. This creates the visual effect of incrementally sharing colors with the boid's color flockmates.

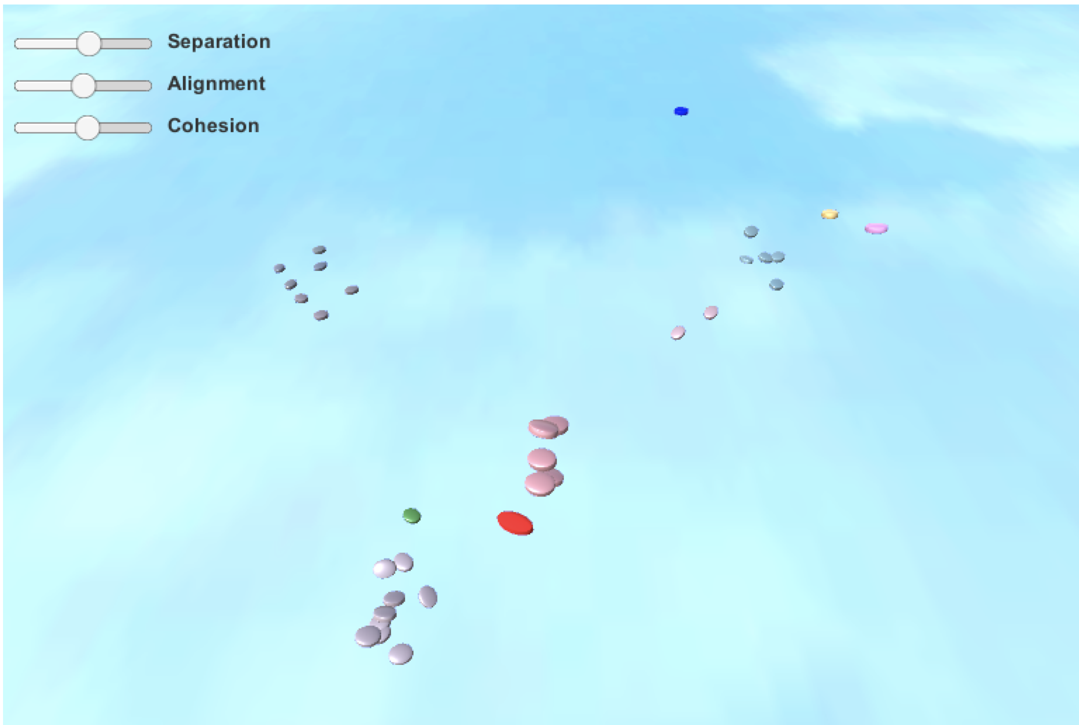
If a boid leaves its flockmates and begins to travel alone (the boid is the only member of its color flockmates), the boid slowly returns back to its original color. Similar to the aforementioned color change value, this behavior is done by taking the original color minus the current boid color divided by a preset constant.

$$color\ change\ =\ (original\ color\ -\ current\ color)\ /\ colorDiv$$

The subtle change in color is evaluated by taking the color change RGB values and adding them to the RGB values of the boid's current color. This creates the visual effect of the boid incrementally returning back to its original color.

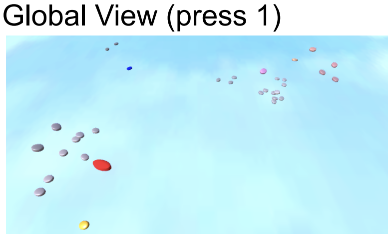
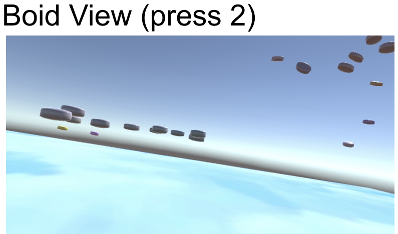
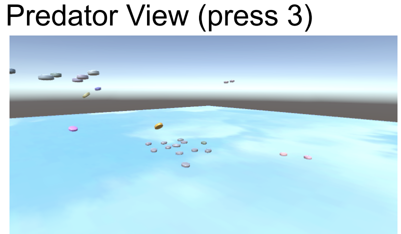
We reached this method of color sharing via experimentations with various methods of color sharing. First we simply changed the current boid color to the averaged color flock color. This resulted in a rapid change of color that made it difficult to observe the color sharing behavior. We then decided upon these incremental changes in color which slowed down the rate at which the flocks morphed into one color. After some period of time passed, all of the boids were the same grey color. To improve upon this, we implemented the boid's return to original color behavior when diverting from the flock.

When viewing the simulation as in the image below, this color sharing feature enables the boids to begin as distinct objects and then merge into flocking groups that share the same colors. When approached by the predator or diverting from the flock, the boids regain their individuality and return to their original distinct color.



First person cameras

To add to the excitement, we decided to add first-person cameras for the predator and the boids. There are 3 cameras in the scene. Press 1 to see the “global view,” a stationary view of the environment. Press 2 for the “boid view” which is a camera anchored to a boid, looking forward. Press 3 for the “predator view” which glides around from the predator’s perspective as it pursues the boids. The boid and predator views are a bit hard to watch because the boids and predators can turn quite abruptly (which looks normal from afar, but is a bit jarring from their perspective.)

Global View (press 1)	Boid View (press 2)	Predator View (press 3)
		

References

[1] Eater, Ben. *Boids Algorithm Demonstration.* https://eater.net/boids

[2] Price, Adam. *Optimising Boids Algorithm with Unsupervised Learning.* Towards Data Science. 2020. https://towardsdatascience.com/optimising-boids-algorithm-with-unsupervised-learning-ba464891bdba

[3] Sajwan, Mohit et. all. *Flocking Behavior Simulation: Explanation and Enhancements in Boid Algorithm.* International Journal of Computer Science and Information Technologies. 2014.

[4] Weerd, Harmen. *Boids: Flocking made Simple.* 2016. http://www.harmendeweerd.nl/boids/

[5] Wong, Timm. *Boids.* 2008. https://cs.stanford.edu/people/eroberts/courses/soco/projects/2008-09/modeling-natural-systems/boids.html