

Karim Elzanaty

Jeff Timanus

AI Programming

October, 2018

1. Overall Architecture

For the Flocking architecture I created a Steering classes for Cohesion, Separation, Group Alignment, and Flocking. For each class I iterated through the Unit map in the UnitManager to check if any boids were in range of the current one. The range was radial, rather than angular, so boids have more of a tendency to move in on direction as a group rather than turn. If they were in range, it would add their respective data to a 2D Vector that would be passed to another Steering behavior. Cohesion used the Arrive Steering to get towards a groups center mass. Separation used Seek to get away from the center mass by reversing the seek. Group Alignment used Face Steering, but a different getSteering() function. This way, I didn't have to create a separate Align class, and instead I could just pass it a 2D Vector rather than make it calculate its own. The Flock Steering got Steering from each of these behaviors, as well as Wander Steering, and put them through a Weighted Blending algorithm. The only behavior that wasn't blended with weights was the Wander, as I wanted it to affect everything, and if the Alignment was high enough, it overpowered it anyway. I didn't create a separate function for the blending, as I didn't feel it was necessary. Each behavior was blended together and combined into a final acceleration/rotational acceleration value to be used.

2. Challenges faced in development

I had a huge problem getting the Group Alignment working. Boids would consistently match their alignment, then do a full 360 out of nowhere. Turns out the other behaviors were retaining old rotational acceleration and would return that to be used even though the Group Alignment was done. I was combing through my face trying to solve this, and discovered a problem in it along the way, which had nothing to do with the previous problem, but could have caused trouble down the road.

3. Areas where further improvements could be made

The “UI” for showing the Flock values is a bit rushed and messy, and I would like to put all of the values in a map for ease of use. It would make loading and saving them neater, as well as changing modes during simulation. Also getting rid of the large if statements would be nice.

