Kelly Herstine

Jeff Timanus

AI Programming

October 8, 2018

Assignment 2

Overall Architecture of your Assignment:

All Flocking behaviors are inside of their own classes. These classes include: Separation, Cohesion, Group Alignment, and Flocking. Separation consists of a constructer and a getSeperation() function that returns a Vector2D. The class has a map iterator that moves through the entire map in the Unit Manager class and checks to see whether any of the units inside of the map are close to one another. At the end of the class, it reverses the direction of the unit and returns the direction for the use in the blended weights function. Cohesion and Group Alignment both have the same start at each of their classes. In their own functions, they have a map that includes the entire Unit Manager list and checks to see whether any of the units are close to each other. In Cohesion, the class averages out the locations of each of the units and creates a local direction. It then returns this direction. In Group Alignment, the class averages out the velocity of each unit to be used in the weight blended function and then returns that value. In the Flocking class, there is a getSteering() function that includes references to each of the different flocking behaviors. Inside of the class, it updates the blended weights and assigns them to a Vector2D. After it does this, it gets the facing direction of each unit and normalizes them. It then takes the direction that they are heading and sets it to the target direction. The target direction is then normalized and the blended weights are added to it. The class returns the acceleration of the object. There is another function in this class called updateBoidWeight(). This is used purely to update all the weights inside of the class and then cap them at 0 and 1. This makes sure that there are no odd values to the weights.

Challenges Face in Development:

I really struggled with trying to understand how flocking would work in a coding sense. I think I tried about 4 or 5 different algorithms before I finally came upon one that worked. It was hard to think of all the different flocking behaviors and how I would be able to blend them all together in the flocking class.

Areas where further improvements could be made:

I think that I could still definitely improve on how my Separation class works. Sometimes the units will still go through one another if they are not in the same group, but if they are in the same group, they make sure that they aren’t colliding with one another.