Assignment 2

My architecture has 3 steerings, cohesion, separation, and alignment. All of the steerings check each unit in the map to see if it is within a certain distance near them and if they are, reacts to them accordingly. If there are no units to flock with, the units wander until they come in to contact with another unit. The weighting system is modeled off of normalizing a vector3, and then taking the x, y, and z values and multiplying them to the steerings. Since the function works this way, for adjusting the steering dynamically all I did was slowly add to or subtract from whatever steering the user chose and normalized right after. It currently takes away from both other steerings equally to give the desired steering more weight, which I assume would be the best way to do it. For actually changing the weights, rather than having 6 buttons, I have the user hold down a button to indicate which steering, and then have add and subtract buttons from there. For writing to a file, my current solution just continuously appends a file stored in "component steering SDL" with the name "WeightSave," so the user has to dig a little bit to read the values.

Ideally the steerings would only check within somewhere around a 180 degree ray in front of them, but in trying to implement that I got an error where the steering would disregard other units while turning, I assume because the rotation went from 360 to 0 or 0 to 360, as I had rotation set to a mod of 360, So my solution checks a full radius.

I had a very strange error that was messing with my steering where the data of the steering of each unit after getting the steering would be filled with nonsense values that messed with my steerings a lot. I solved it with an if statement that turns itself off after one run with a Boolean, but I have no idea where those garbage values came from.