HW2:

Arrive and Align: I have 3 versions currently for Arrive and Align. I struggled immensely with becoming familiar with c++ as well as the transitions from degree to radians and back again. As such, the part2 folder is my earliest version and is extremely rough in its implementation[1]. Part2copy[2] is much better, with parameters tuned the boid can approach smoothly and will come to a stop as it should. I believe this in part because of smaller max velocity and max turn. However, I was surprised to find the biggest difference for me was that if I increased the radius of deceleration, it can get stuck looping around the point[3] as opposed to stopping. This may be in part due to a combination of the turn radius per frame and the radius of deceleration allowing for the boid to get stuck like this. Overall, however I believe part2copy is the most successful implementation and I will move forward using this version.

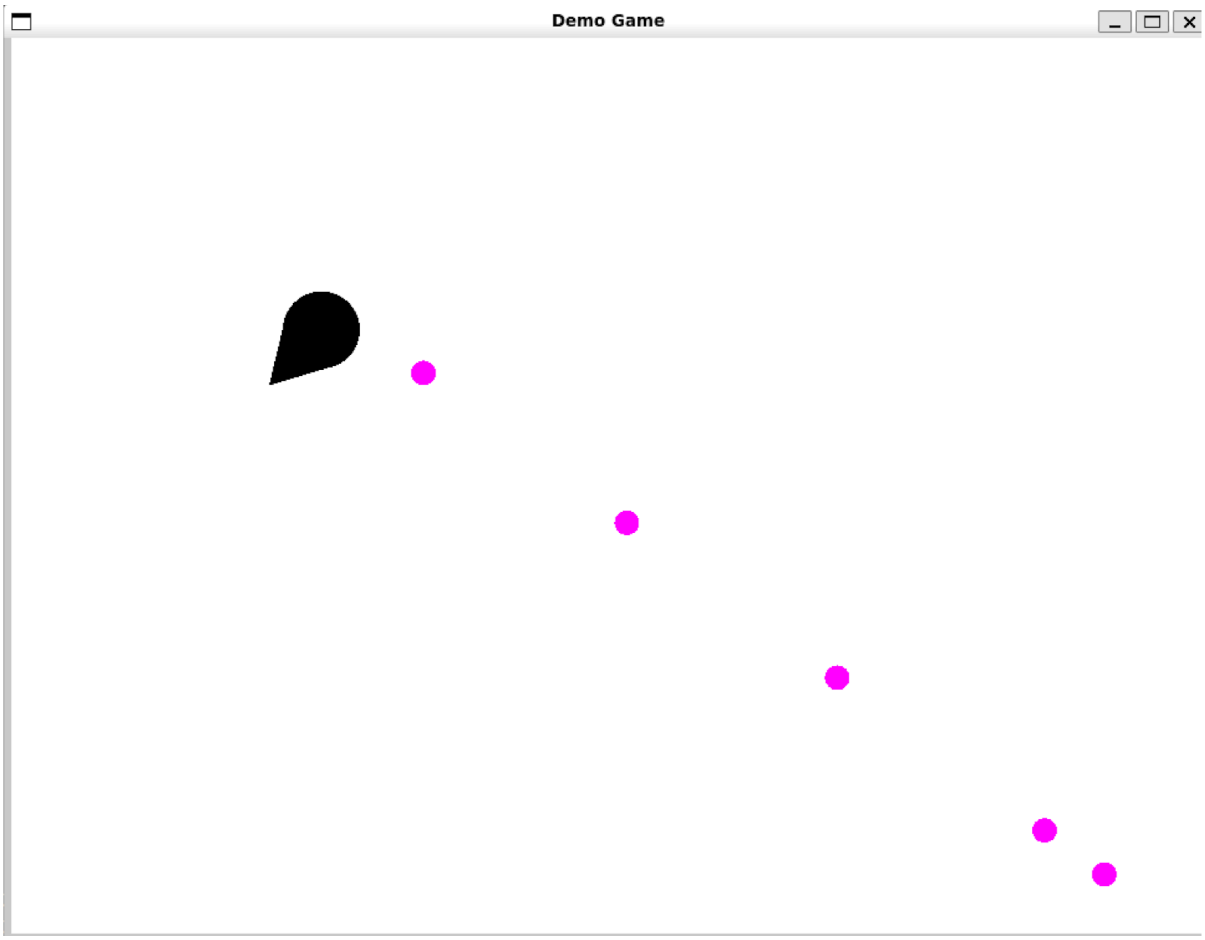
Wander: I found wander to be difficult as my boid would frequently simple head in the same velocity and the tweaks wander made were not enough to cause meaningful “wandering” . I tried adjusting parameters in wander multiple times but did not find a satisfactory implementation using the book's suggested algorithm. At the time I moved on and implemented my Flocking algorithm. As I was reviewing my part 3 however I realized I had implemented a random position on spawn for boids in order to help flock. I was able to create a second wander class in order to set goals which were much further out and created a wander which looked much more random. I found that the smoothest implementation resulted from actually combining the two wanders with a loop counter so that every ten updates a further goal would be chosen and every 4 updates we would use the original wander algorithm to make smaller adjustments. I feel that the current wander implementation in part3copy[4] looks the best. However, I am disappointed because it feels like the speed of the boid should be slower and look more like a meander and less like a panicked running around. I found however when I made the boids movement slower on this implementation that it looked much choppier than I was happy with. I also feel that sometimes the alignment of the boid during wander looks off. I think this is my largest complaint with my entire project if we're honest. It does a little loop when it isn’t necessary and despite my best efforts I could not seem to eliminate it. I tried out various combinations of parameters as well and found that increasing the max velocity made it look much jerkier and increasing the max rotation made it more likely to do the little loops. Overall, I found a max velocity of about 4 seemed about right for wander.

This brings me to an overall frustration with the organization of my project. I kept variables such as max rotation, and velocity and max acceleration in my character class which is called Kinematic. However, sometimes I wish I had kept them as part of the individual behaviors. It makes sense to clip max speed during wander more than you might if you were trying to velocity match, align, or seek. I also wish that I had implemented more randomness towards the actual speed than I did.

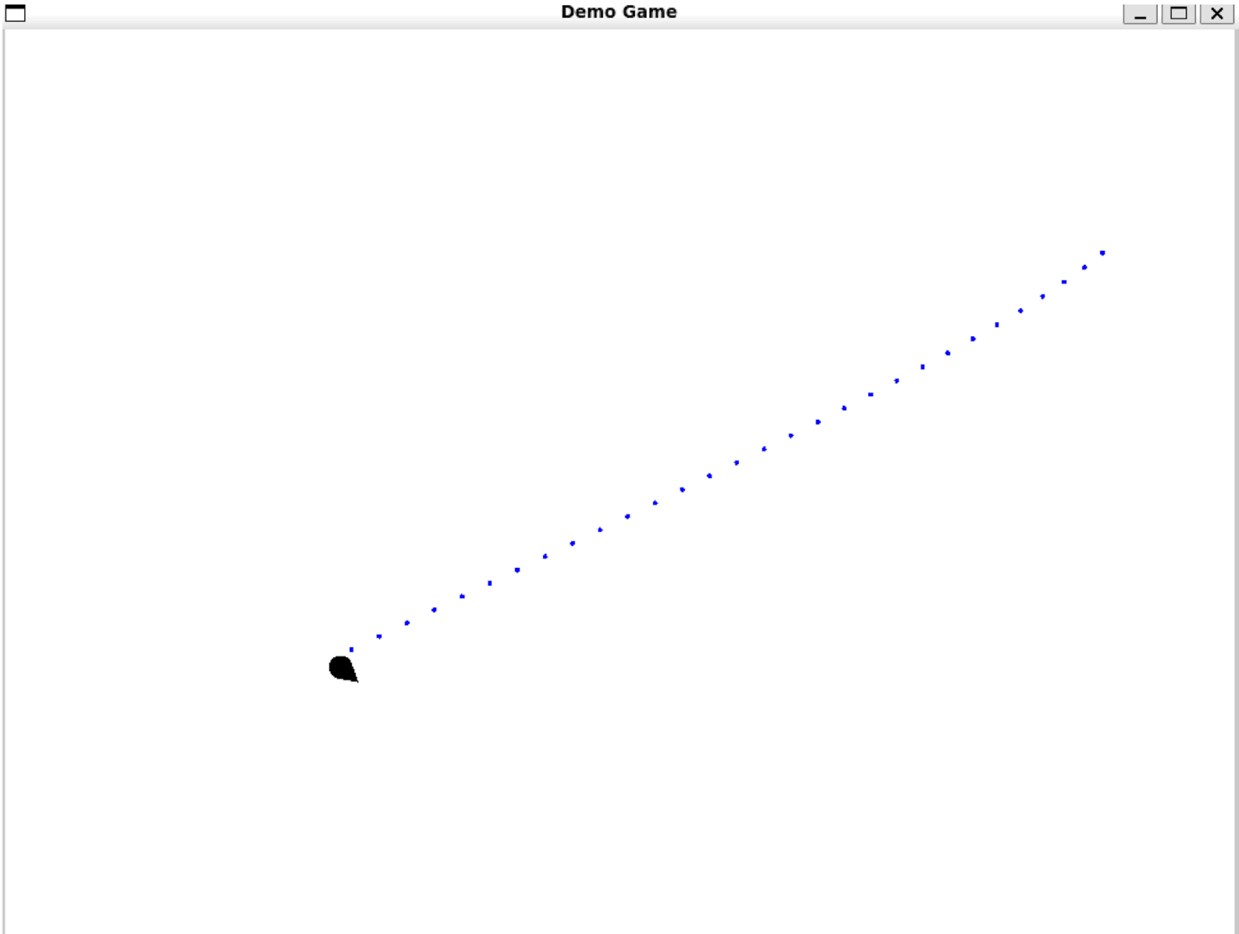
I chose during the project to implement boundary violations as if the boid wrapped around to the other side. I played with the idea of trying to make it avoid having boundary violations by adjusting velocity as a boid approached a boundary and having the boid bounce off the boundary. Neither looked quite right and as such I ended up simply treating it as a contiguous space. This became a slight issue however during flock as the look ahead cannot consider boids which will be close post warp and so increases the likelihood of collisions.

Flocking: As usual I was thoroughly unprepared for the challenge that flocking would present. First attempting to find a thorough explanation of the algorithm was more difficult than I initially expected. While Craig Reynolds site for boids is still up many of the links are bad so I consulted google. I attempted two different implementations and honestly I am happy with neither. It is possible that like with the wander behavior, I may have better results combining my efforts differently. Flocking is much better than flocking two however I believe. Flocking is largely adapted for use from **V. Hunter Adams** from Cornell University. His code also allows for boids to have bias and is used for predator avoidance modeling. I have disregarded the bias. His pseudocode made for easy understanding and implementation in theory but of course it is necessary to adapt to fit the code base. As such, I worry some things have been mis-implemented. He has implemented both cohesion and alignment and separation but the separation seems to fail. I feel this is due possibly in part to the separate pieces of the algorithm not being normalized before being weighted. I find it very odd that Adams does not average the vector representing the sum of the position vectors of boids that were too close. It seems this way he is more likely to over represent separation and yet the opposite is in fact the case. I did attempt to normalize all vectors before multiplying them by their weight and this seemed to make the avoidance issues slightly worse, however the flock looks significantly less “bird-like” after doing so. Tuning flocking was extremely difficult. Variables are extremely interdependent and different max/min velocities might require entirely different adjustments to the behavior parameters. It was interesting to see how different configurations might take longer to become more cohesive and I also found it interesting that the weights he gives for edge avoidance are small enough that they aren’t observed unless the boid was already going slowly. It is interesting to note that in the video he has of his implementation, boids do not always avoid each other. I was unclear if ours should or not but have opted regardless to weight avoidance much more heavily than his suggested parameters. Overall, I have had to increase the parameters a good bit. I am unclear how much of this may be because of his boids being much smaller, but whereas he recommended eight for the “protected radius” I found at least 30 was necessary to stop most collisions. I have opted to change the weights to avoid, match, and center so that they add up to essentially one. I weight avoidance most heavily at about .6 and matching .3 and centering at .1. For flocking I found a max velocity of five seemed to work well and a minimum velocity of 3. Flocking 2 was inspired by a github repository hosted by Jorge Yanar. Despite my best efforts it never seemed to really look like I would expect flocking to. This was very disappointing as I found his implementation to make a lot of sense. He calculates each of the rules separately and then adds them together after each has been individually normalized. His method of normalizing and multiplying by the inverse of the distance between the boid being inspected and current boid seemed to really encourage separation regardless of how far down the weight of the parameter was set. It actually seemed to overemphasize avoidance which I believe is what keeps it from exhibiting a true flocking behavior. Most of my testing is done on 30 boids. I did test at 60 and 10 as well however. I found that at 30 Flocking seemed to be much better able to simulate more than one flock in existence. At 60 it seemed as it the flock was essen

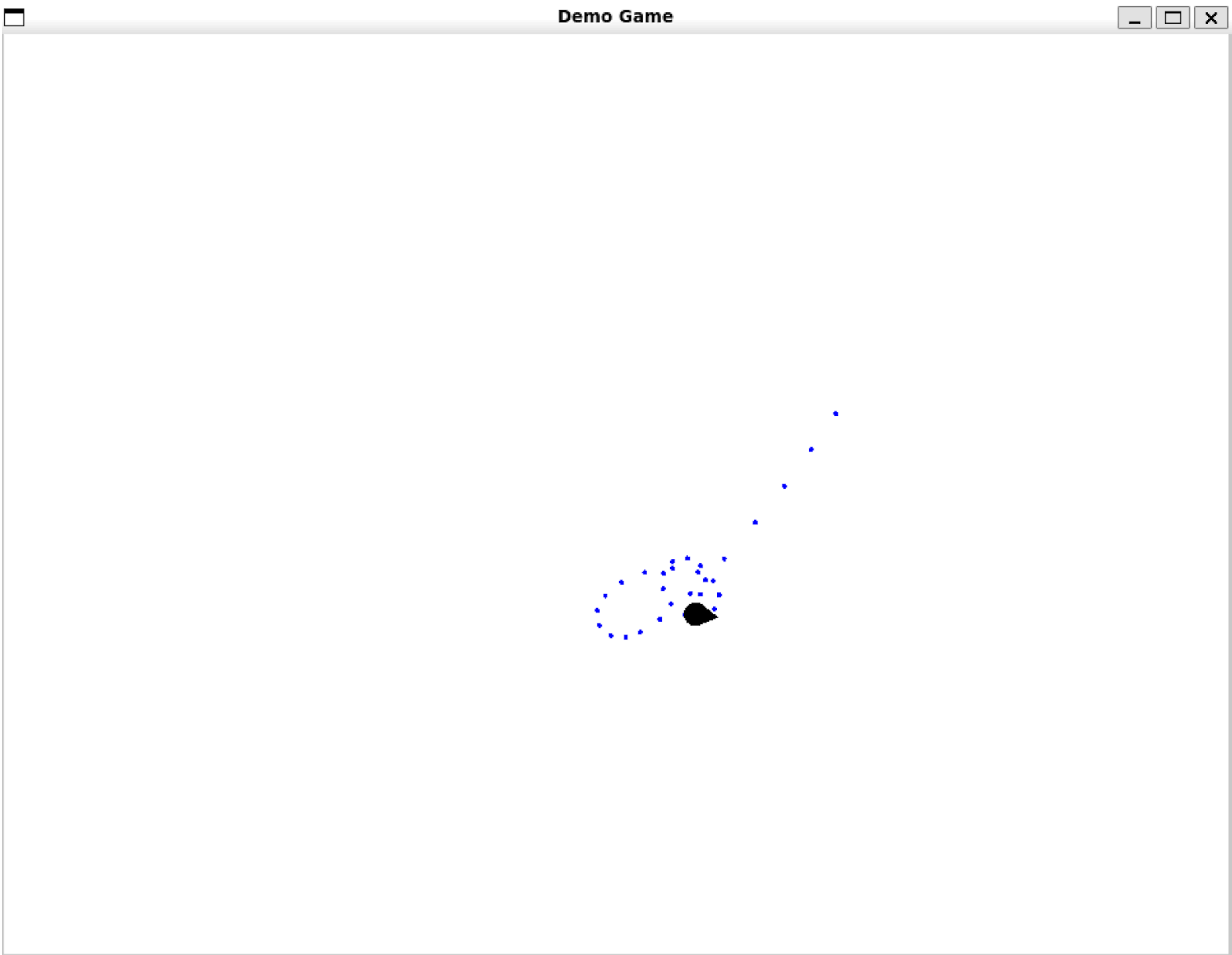
Interrupting itself, and seemed to take much longer to become cohesive. It seemed to settle eventually into 2 flocks but seemed to struggle with changing positions after it settled.



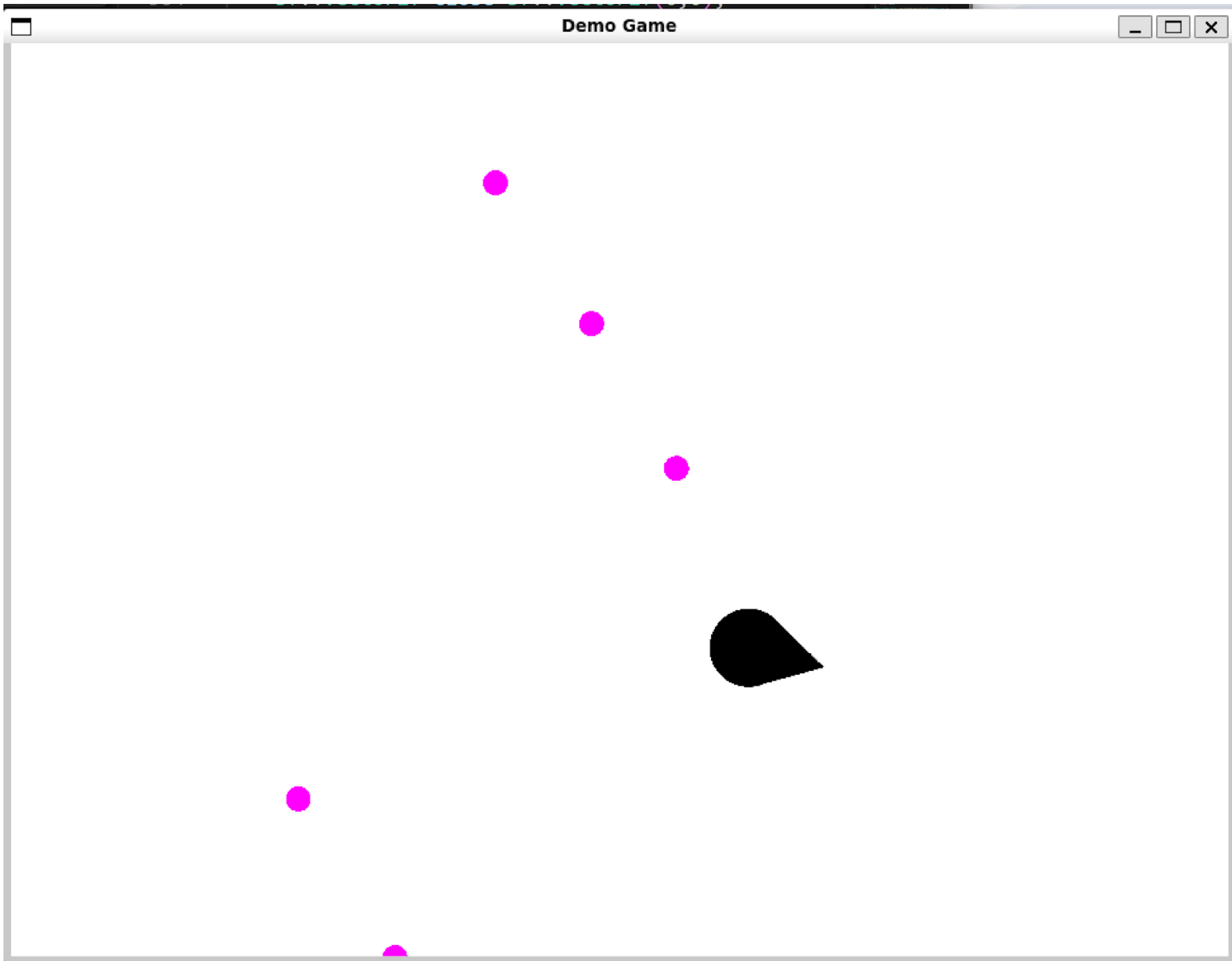
Part2 early arrive was clunky and rotations were awkward.



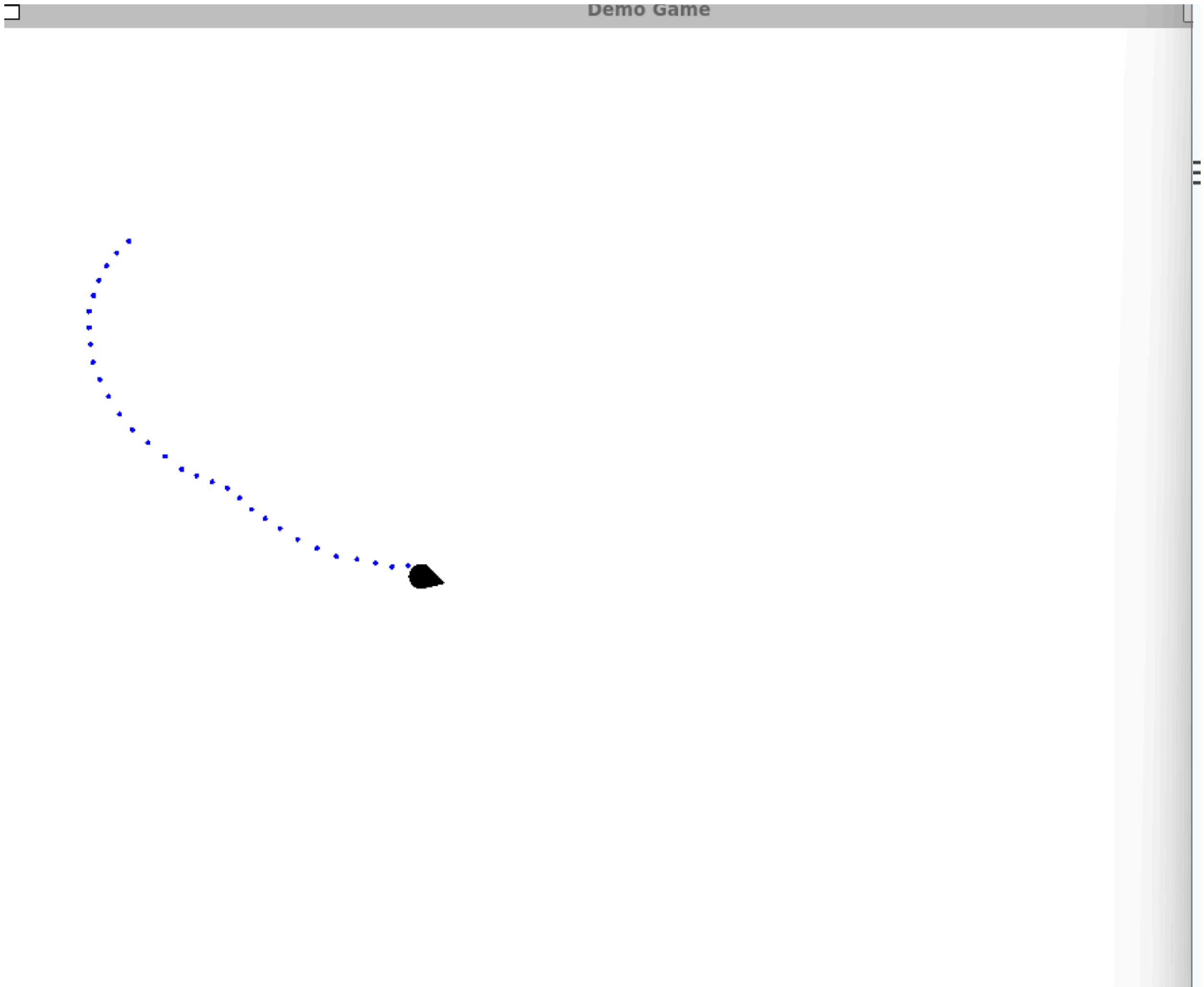
Part2copy works well and smoothly rotates and slows to a stop



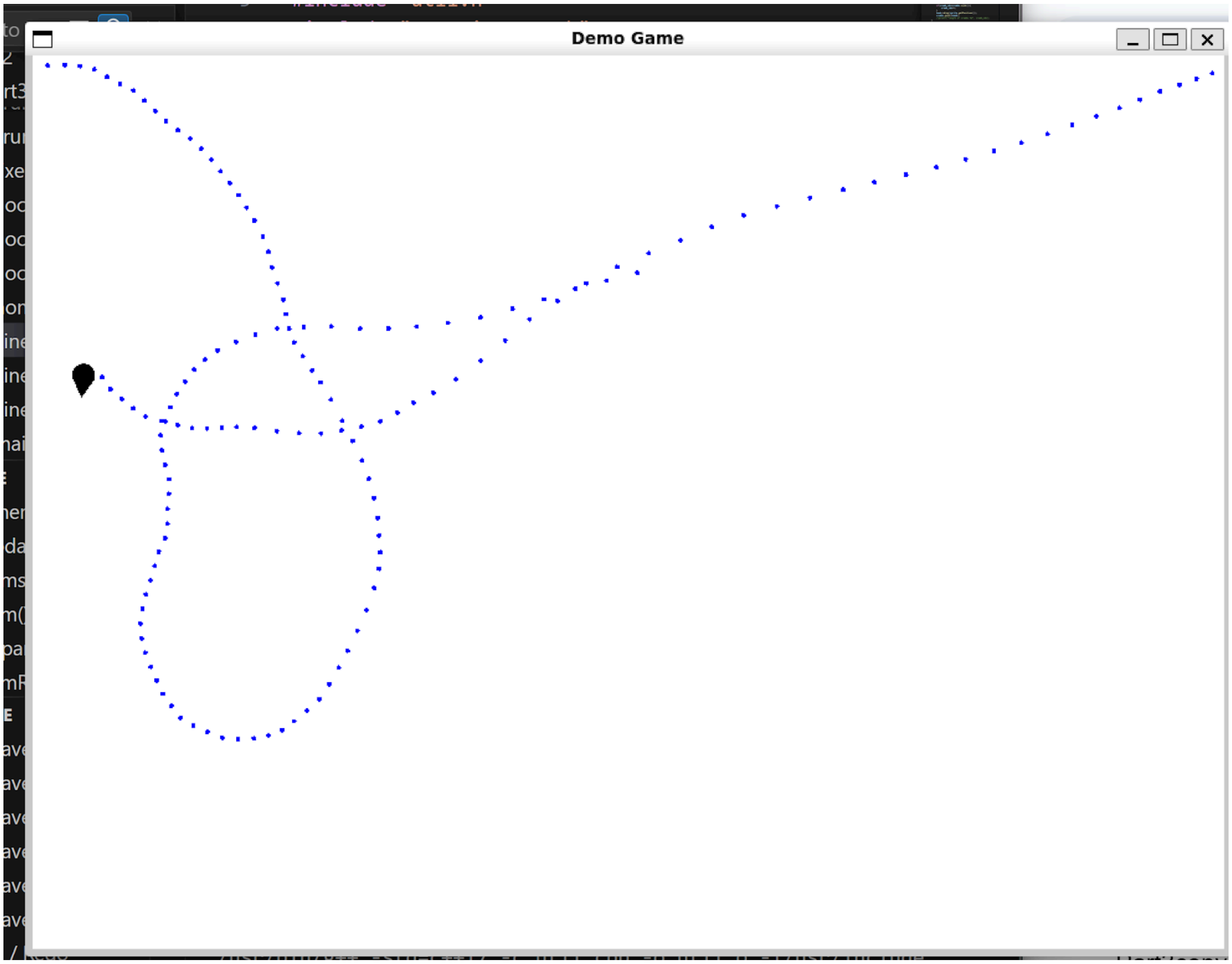
Part2copy2 with large radius of descent, can end up looping sometimes instead of arriving.



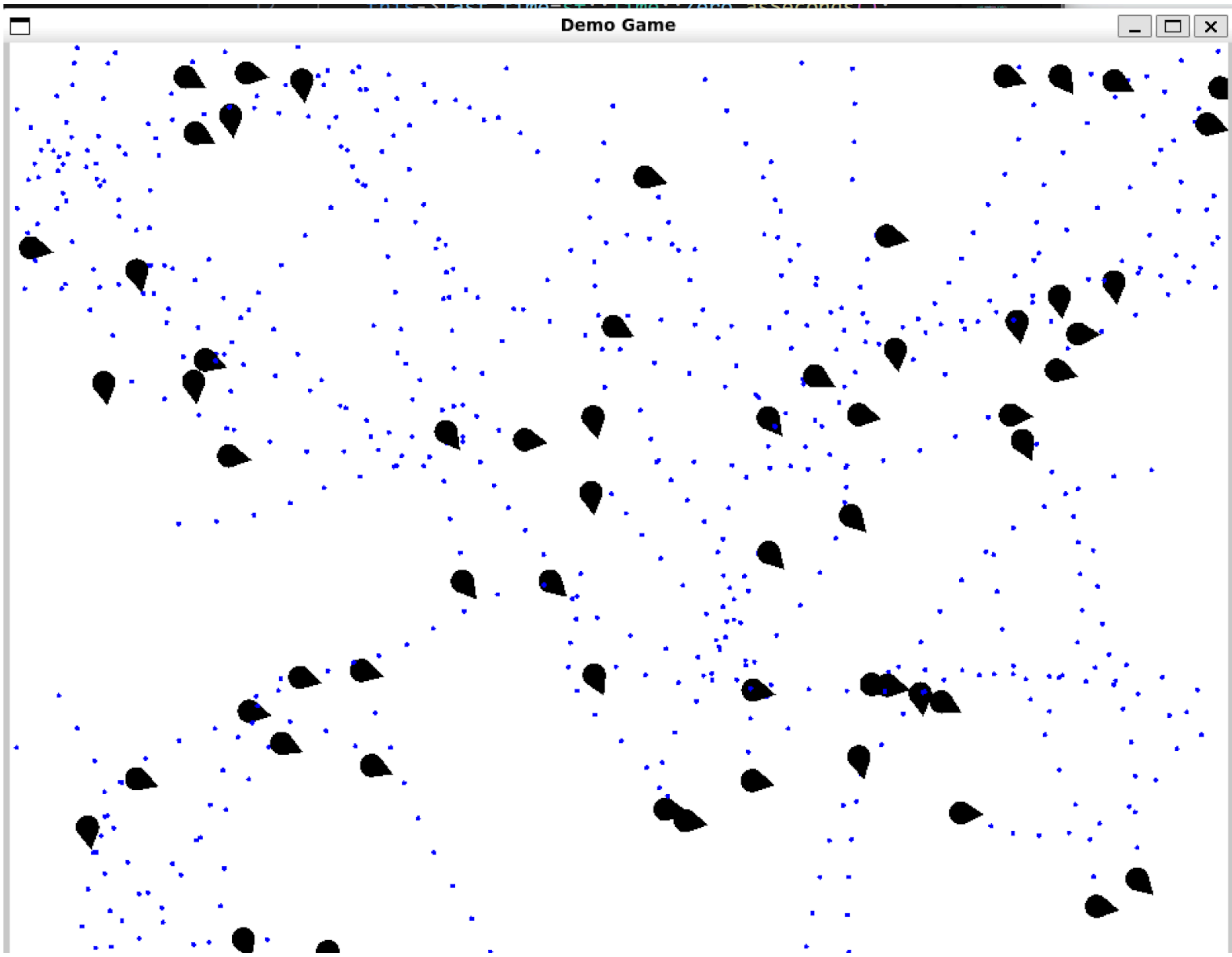
Part3 early wander



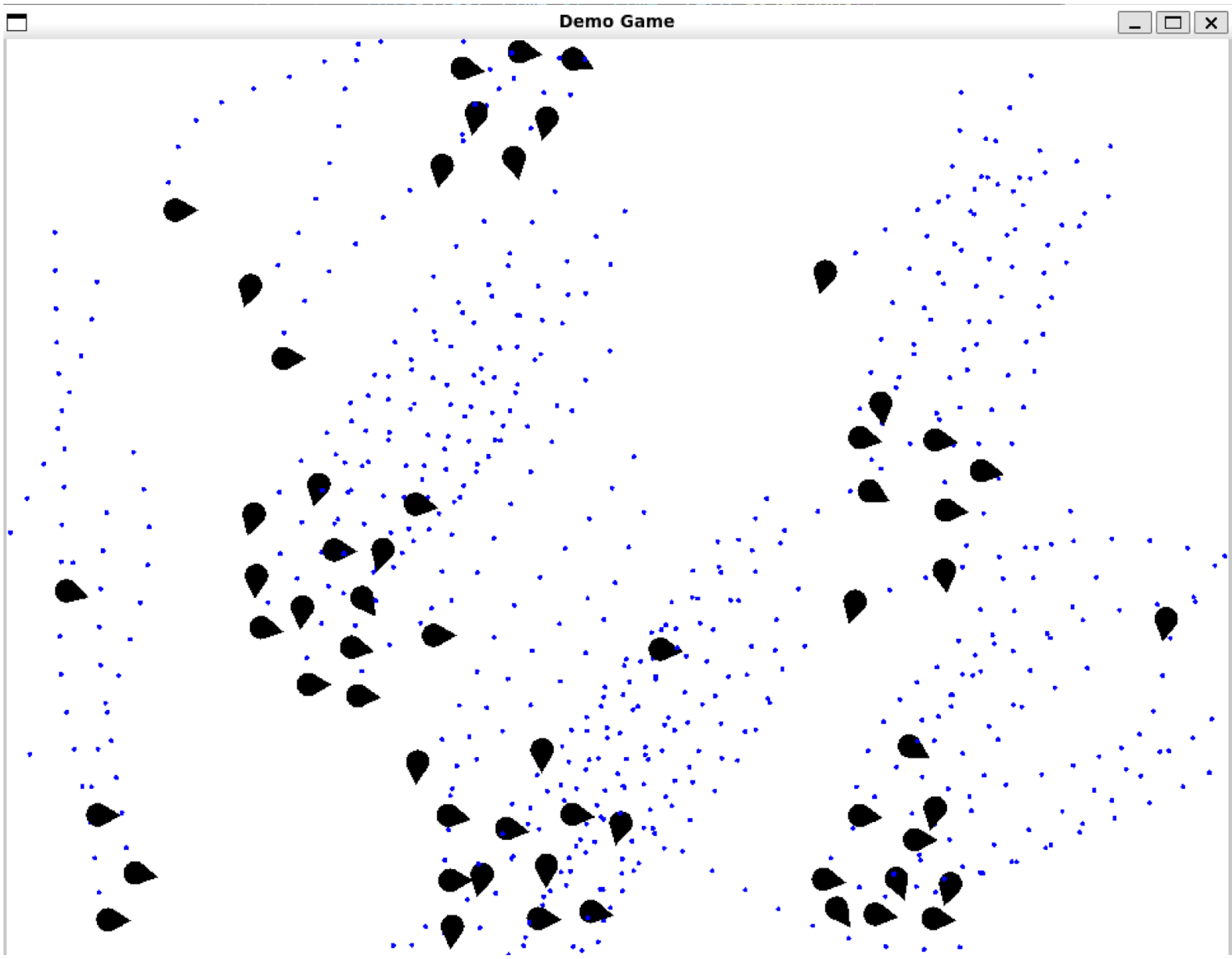
Part3copy wander seems to be much more of a wander and less of a back and forth pacing.



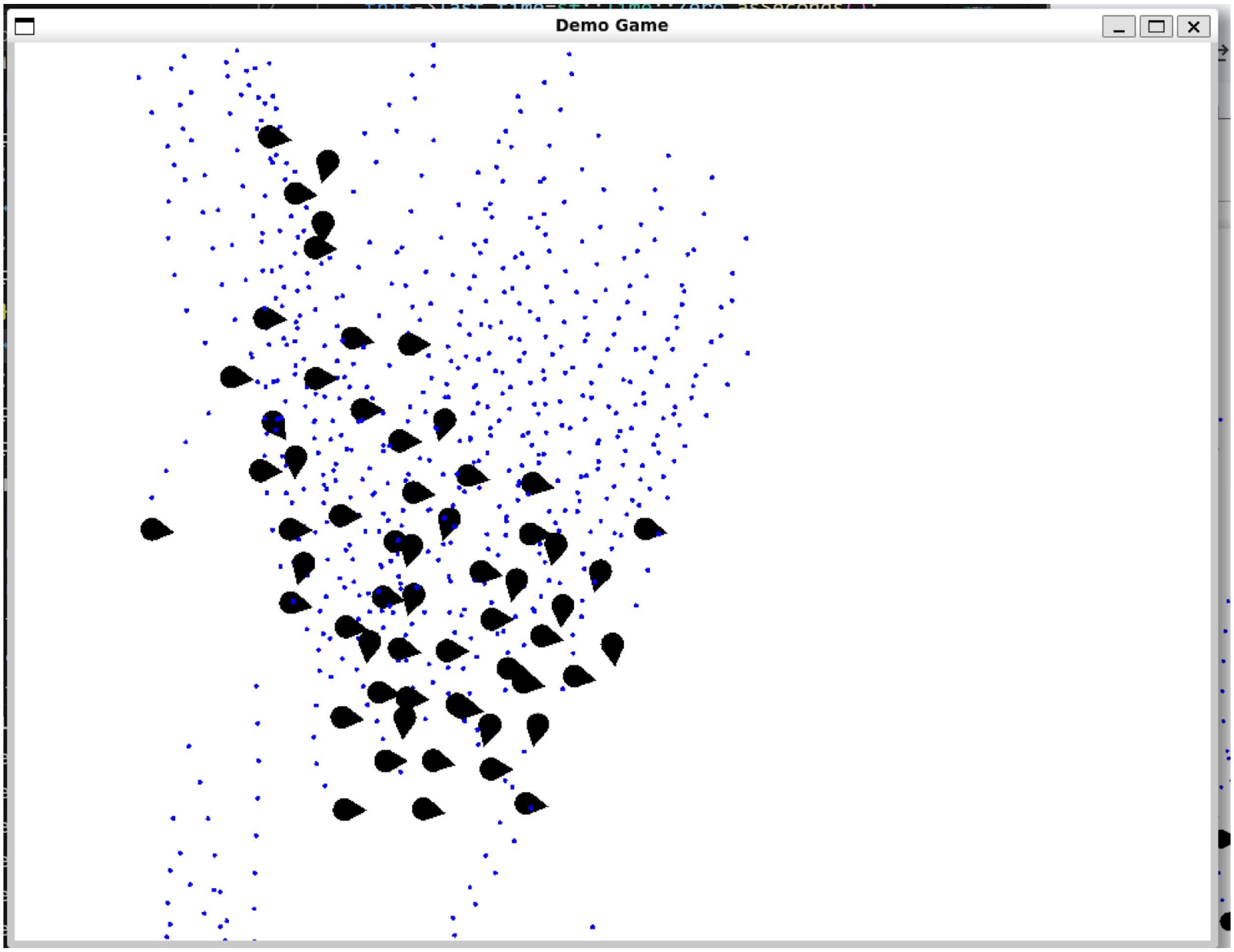
Part3 copy wander with more breadcrumbs.



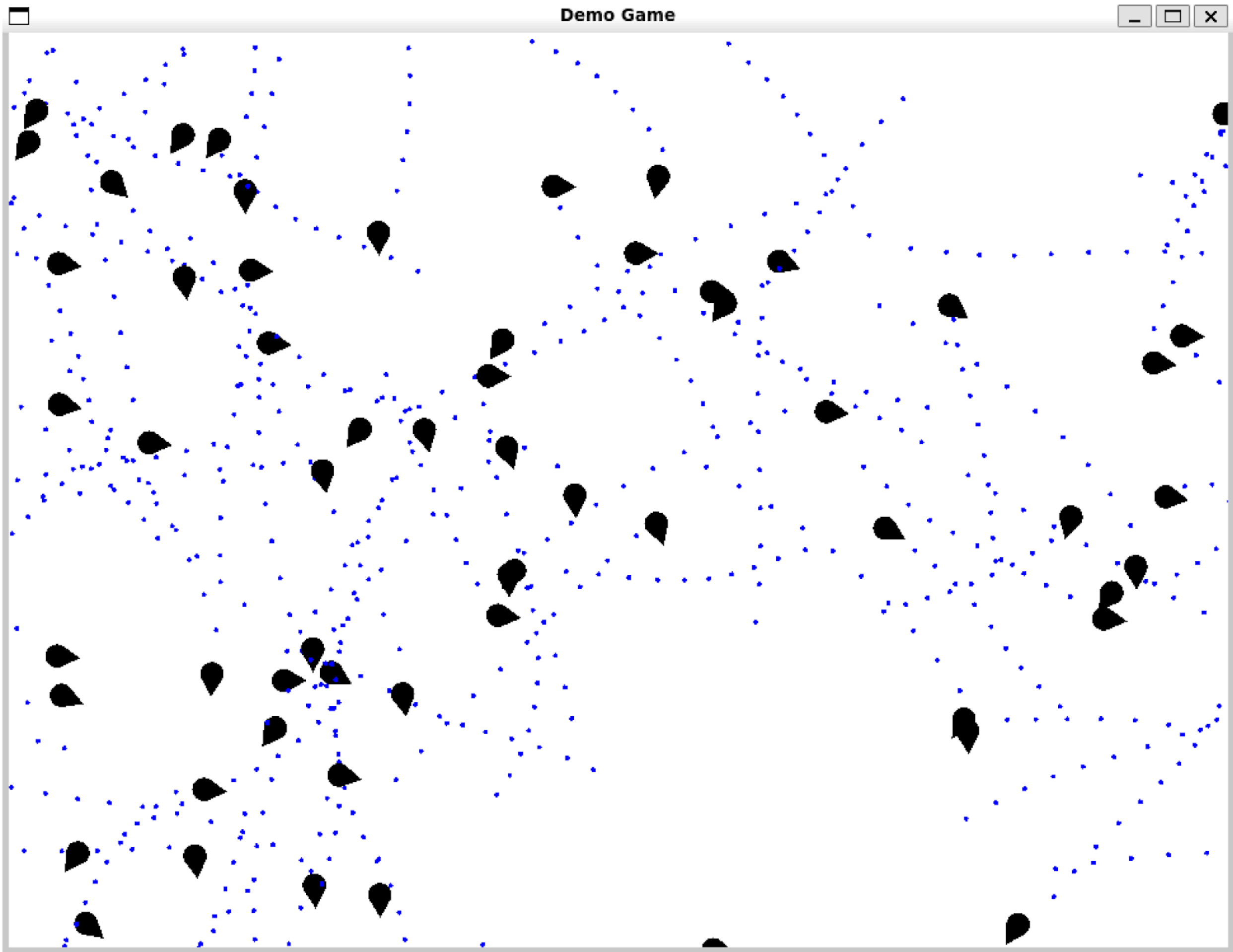
Initial flock explosion after starting part4



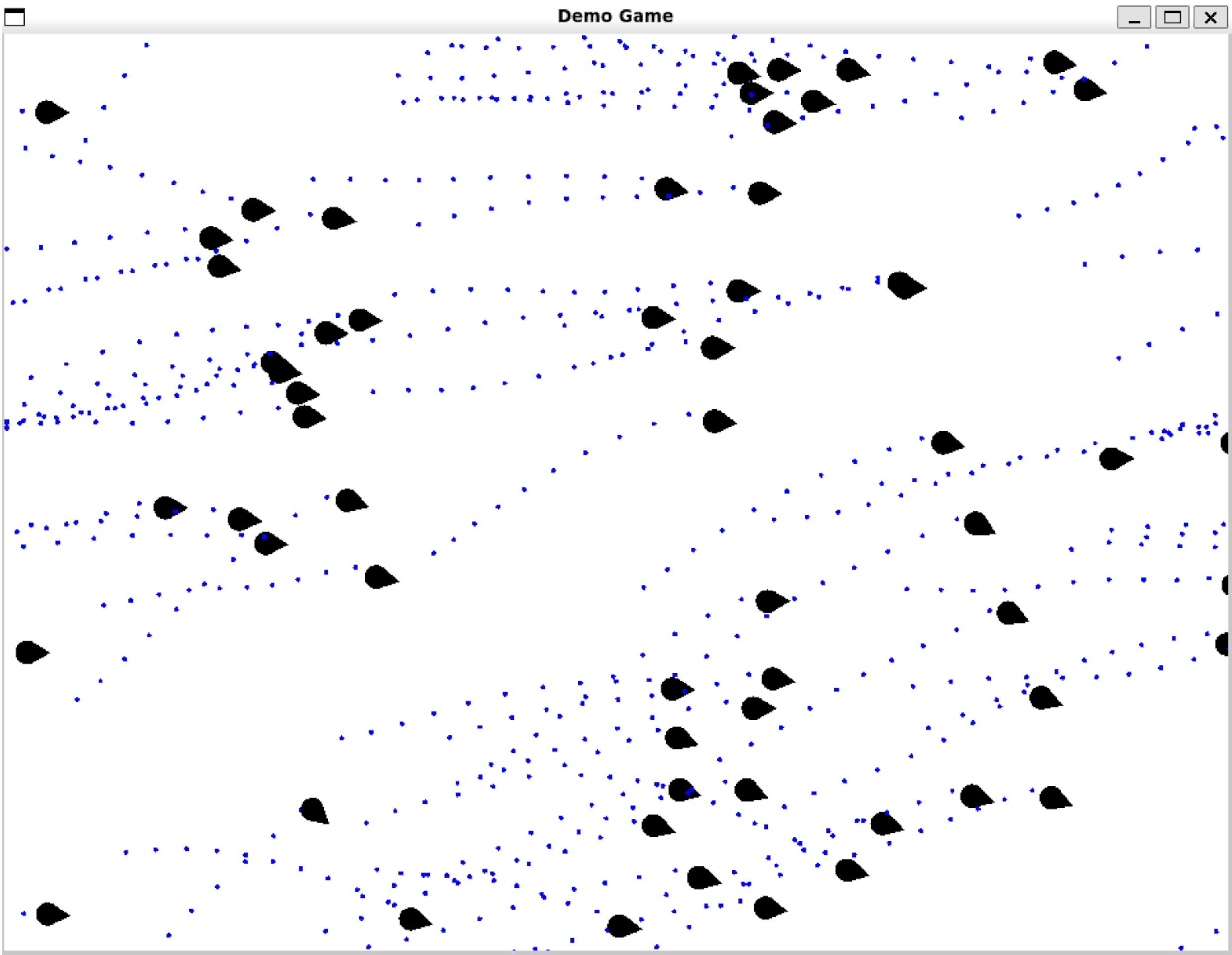
Some flocking shown in part4



Part 4 can be somewhat cohesive but tends to be repetitive.

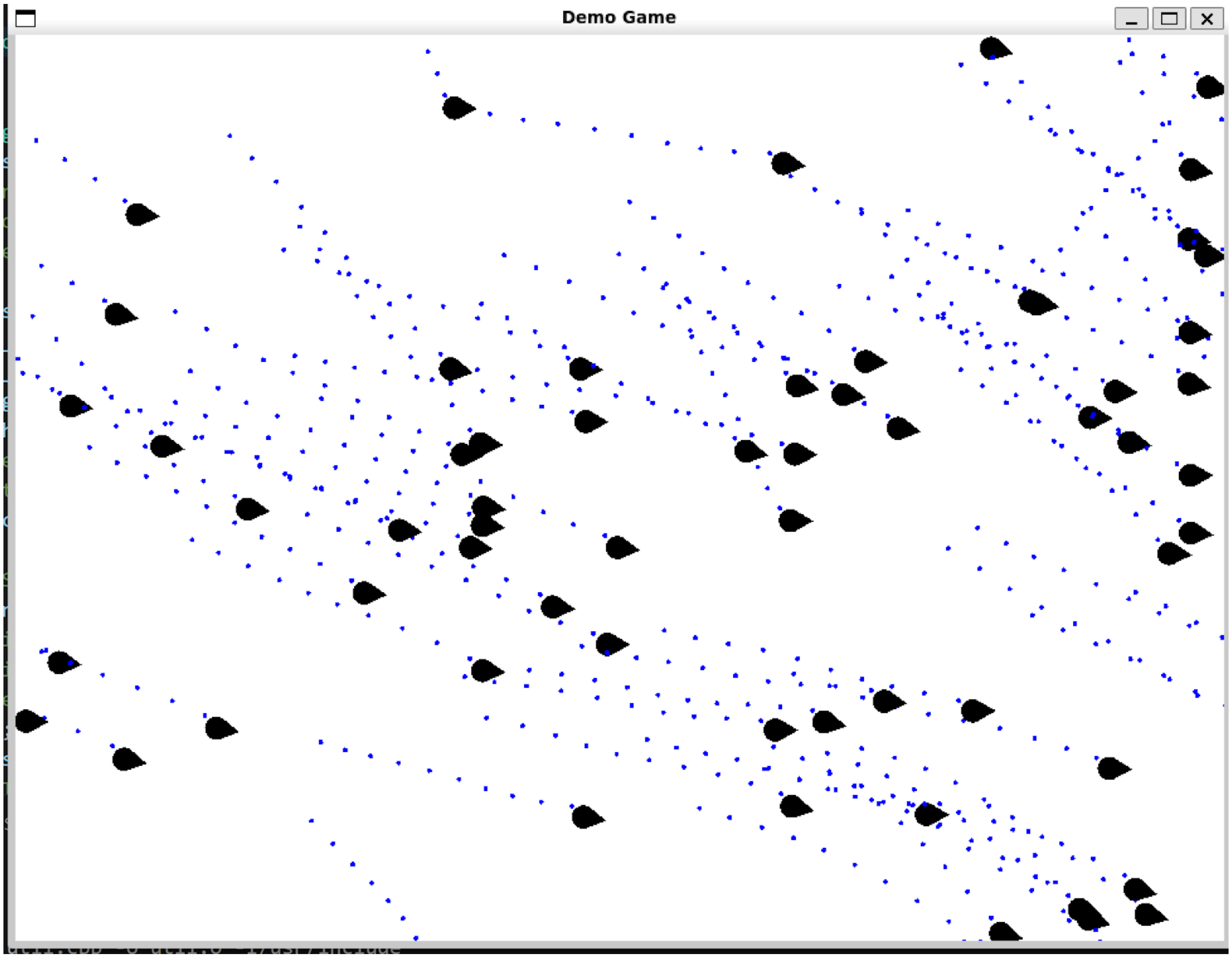


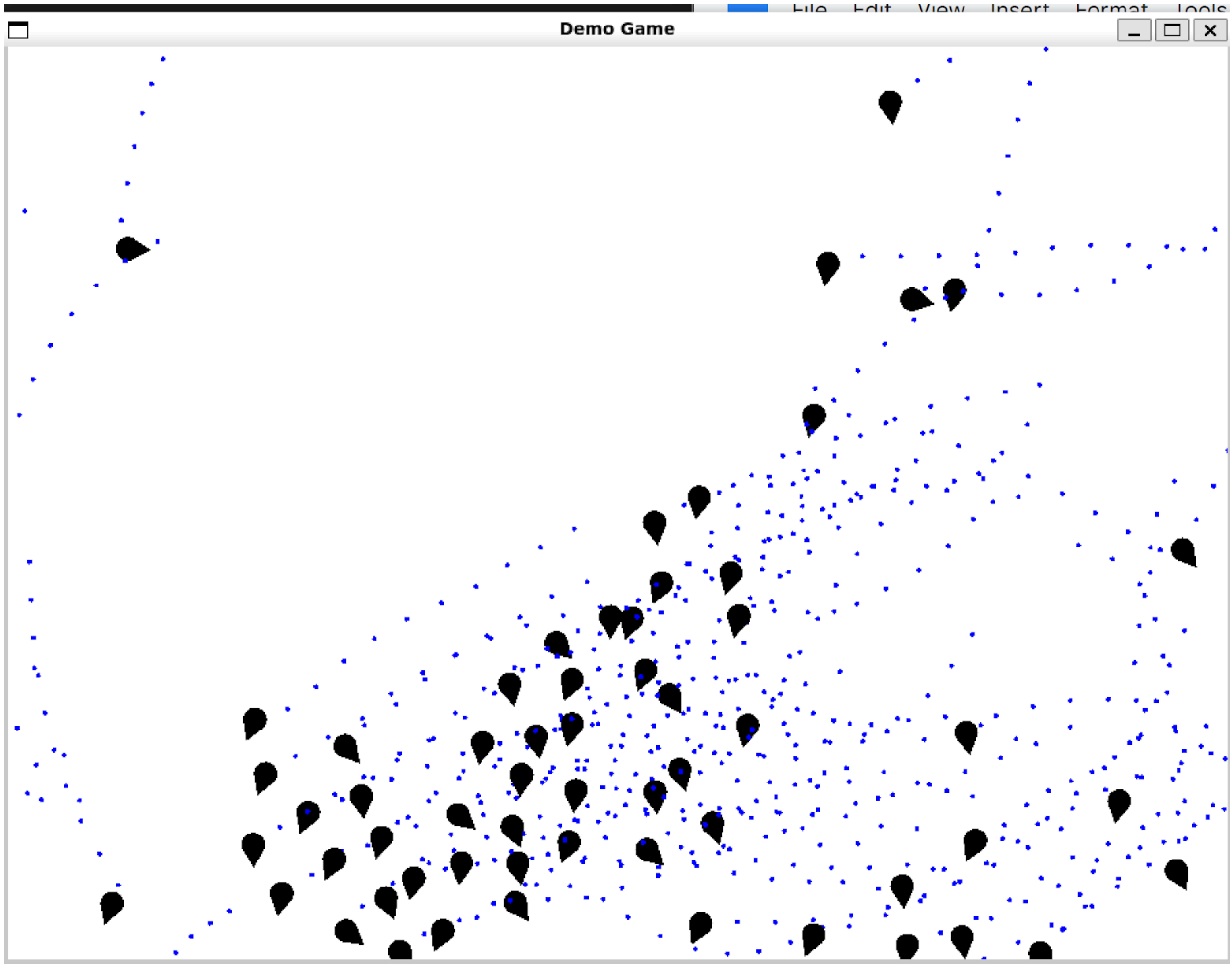
Part4copy right after launch



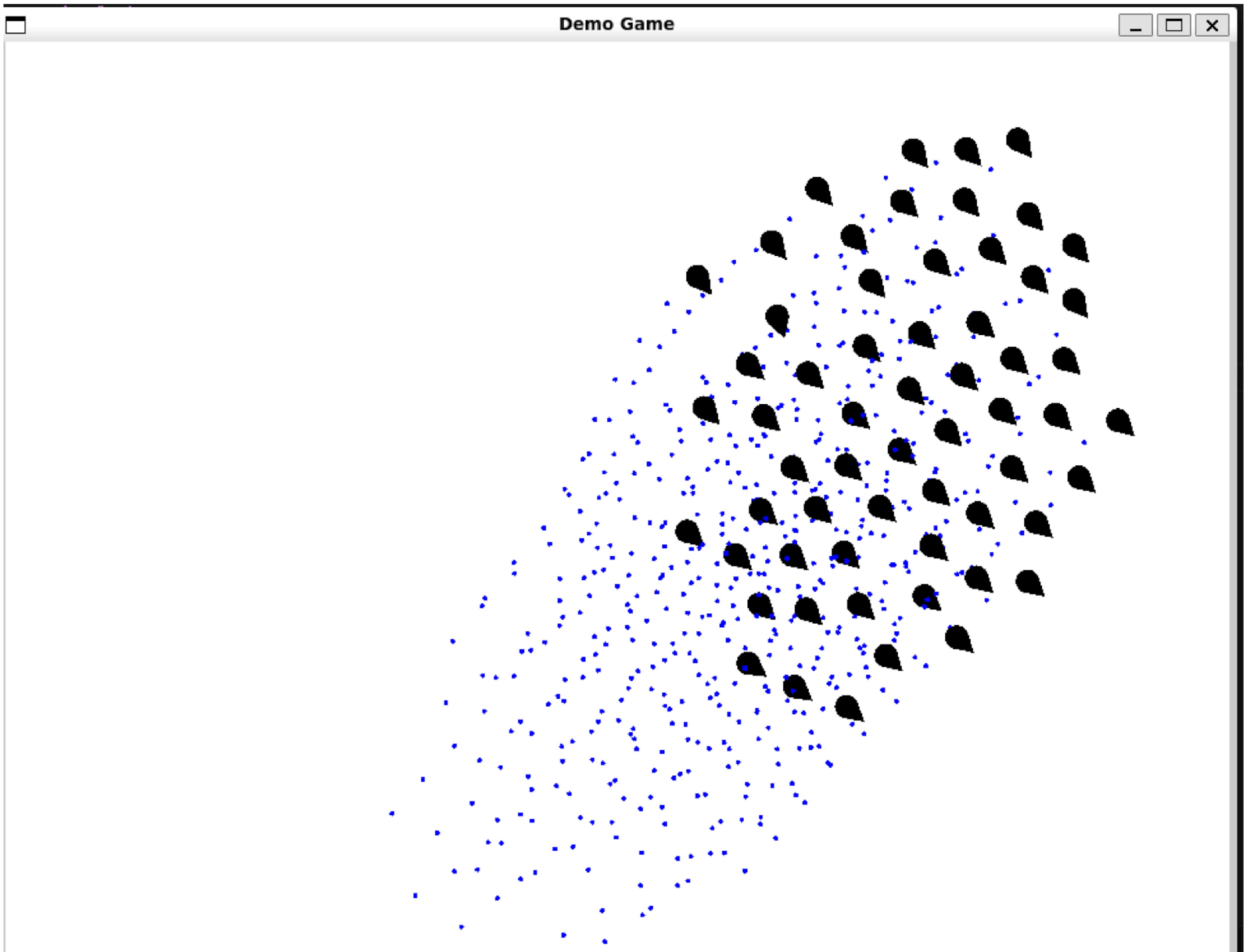
Part4copy with parameters mostly evenly weighted

Several minutes later becomes cohesive

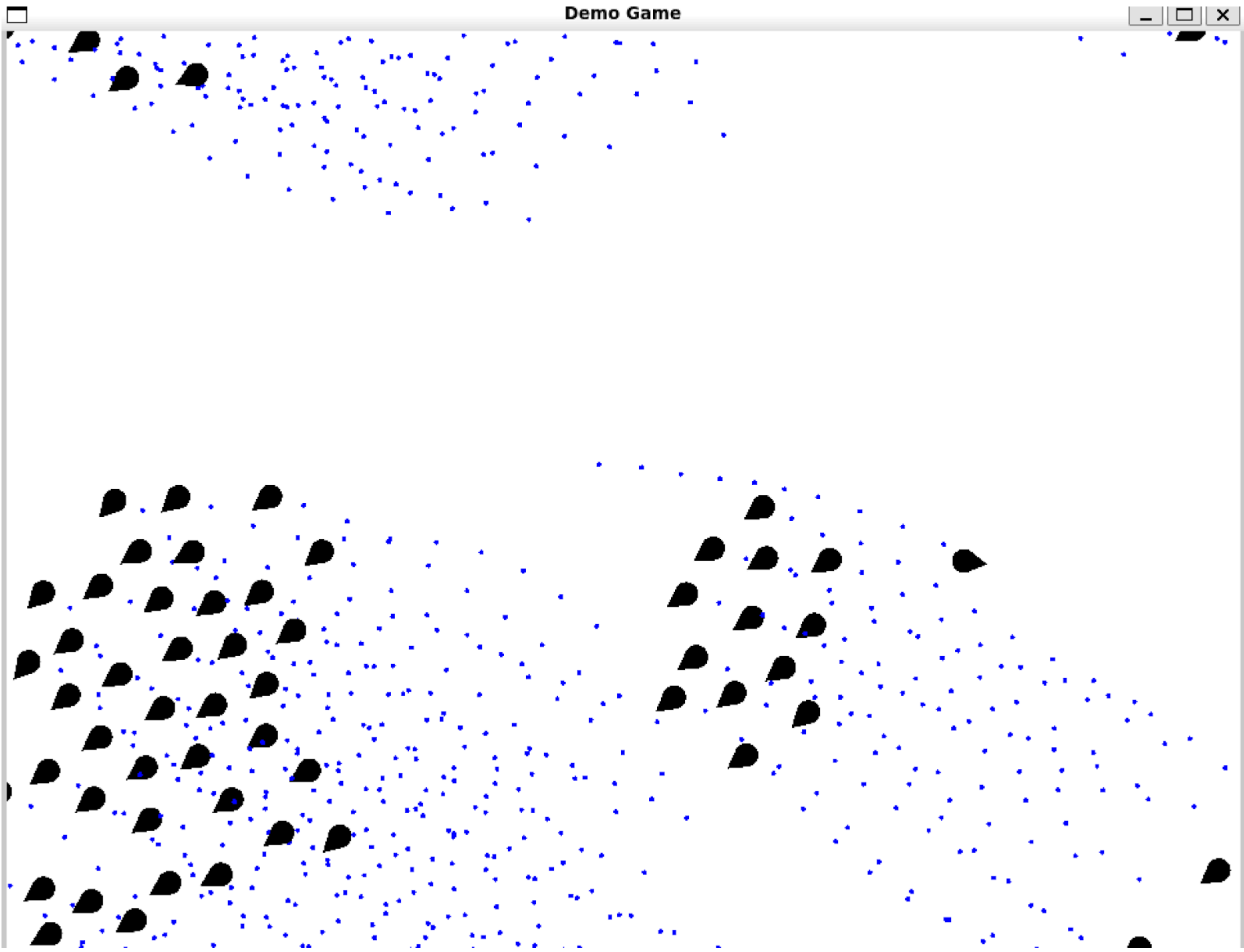

Demo Game



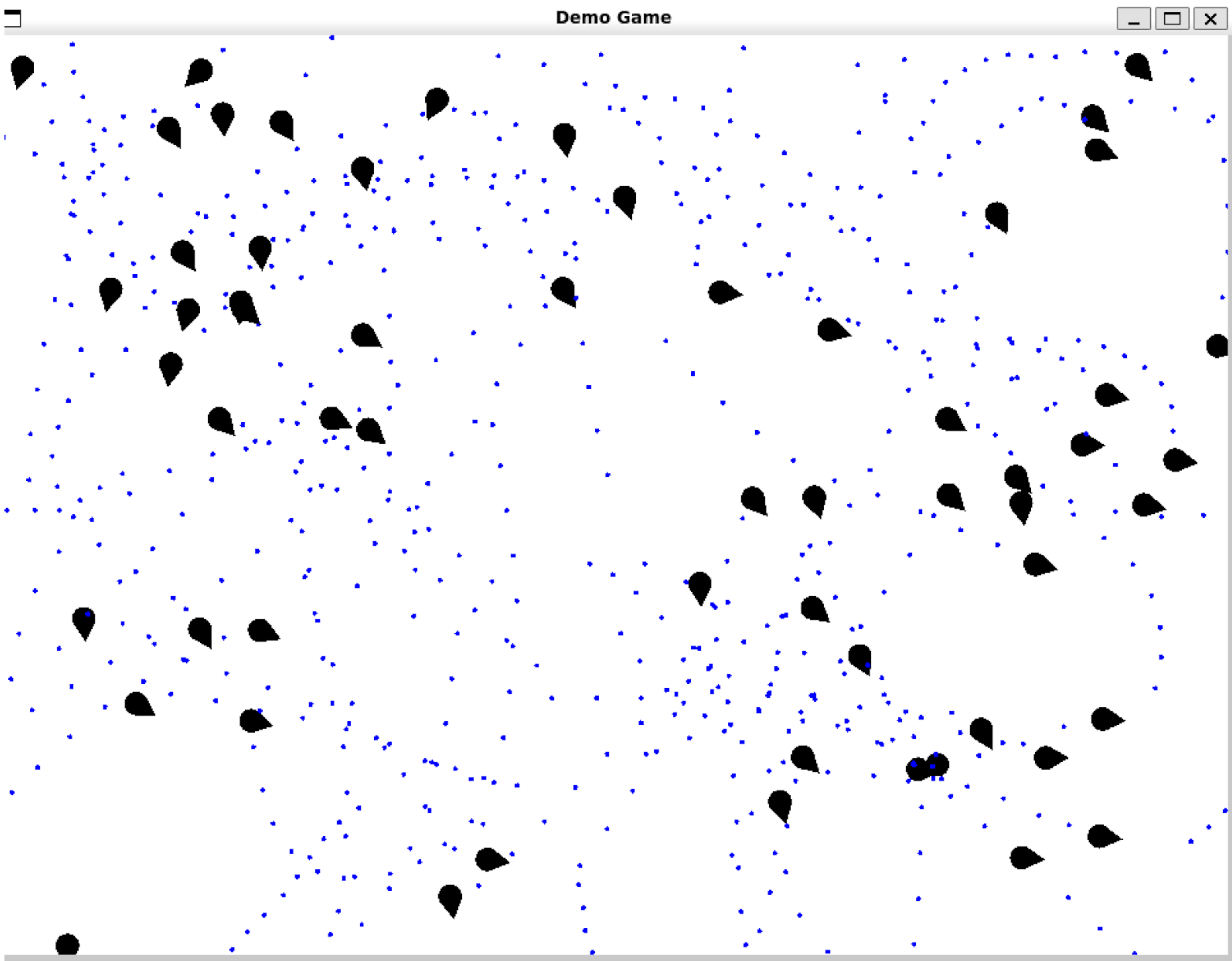
Part4 flocking becomes cohesive much more quickly comparatively.



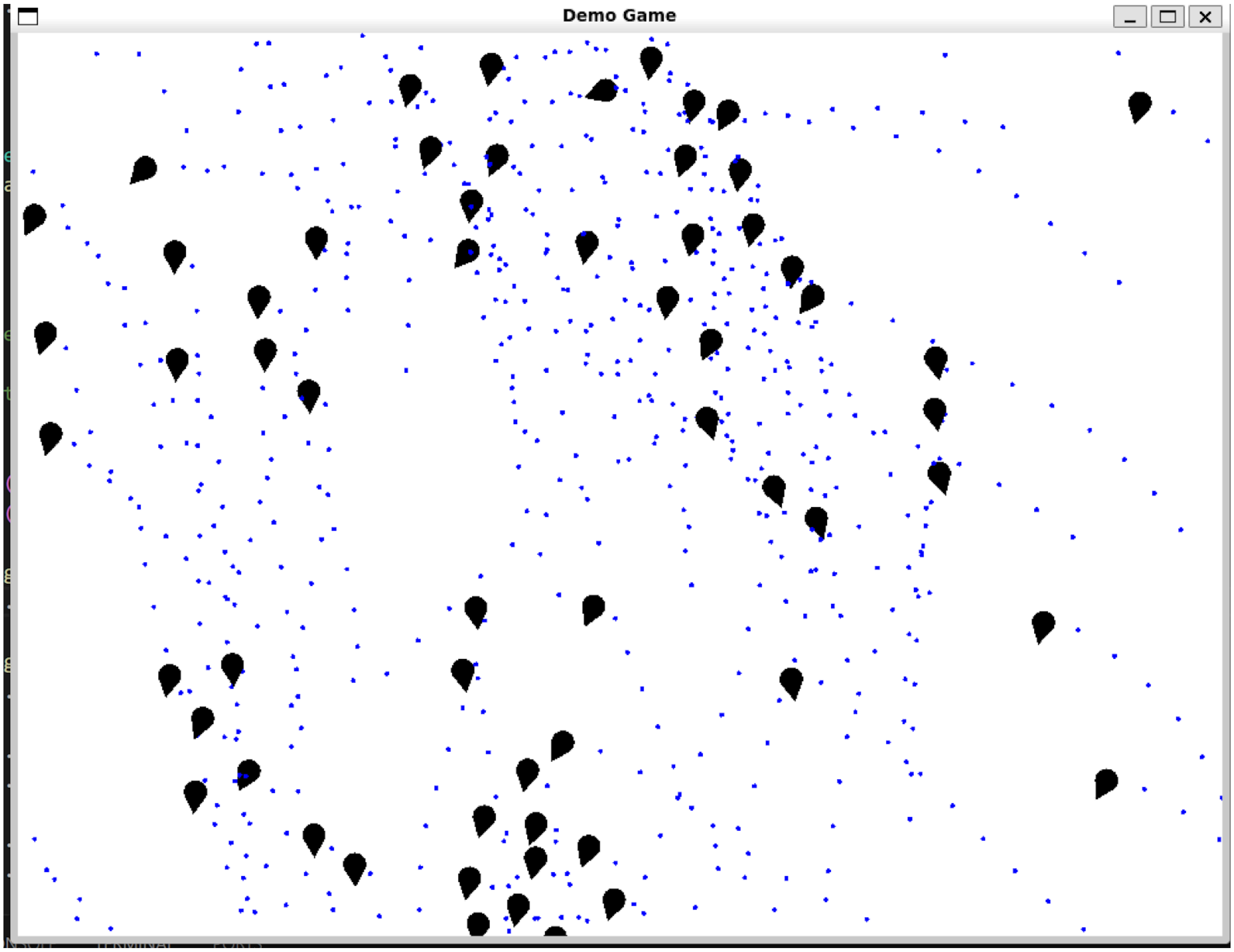
Part4 manages to create a sense of fluidity.



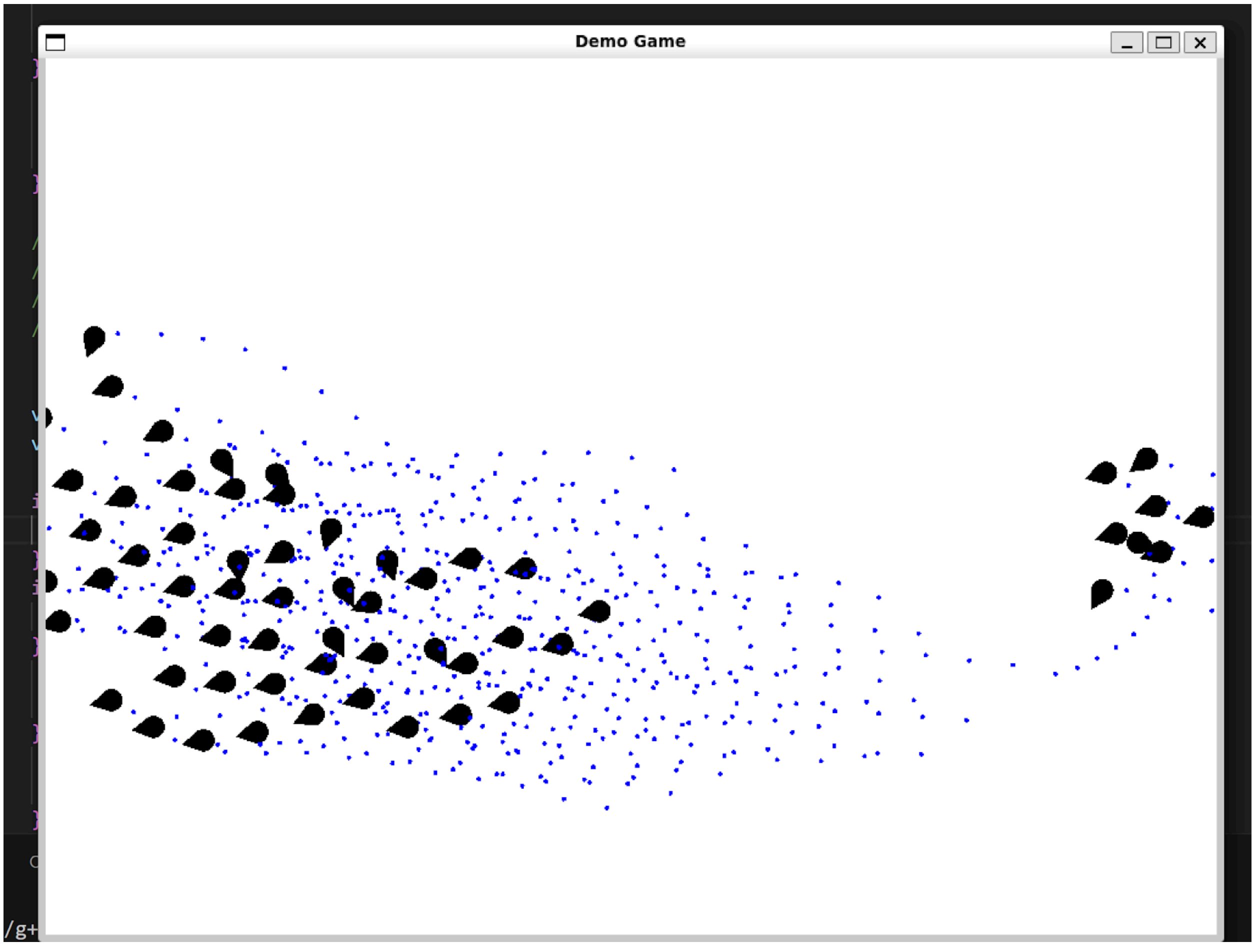
There’s usually at least one rouge boid. This is generally caused by their velocity falling low enough to be affected by the algorithm's attempt to encourage edge avoidance. It is largely ignored when there are sufficiently many boids flocking (the turn adjustment here is .2)



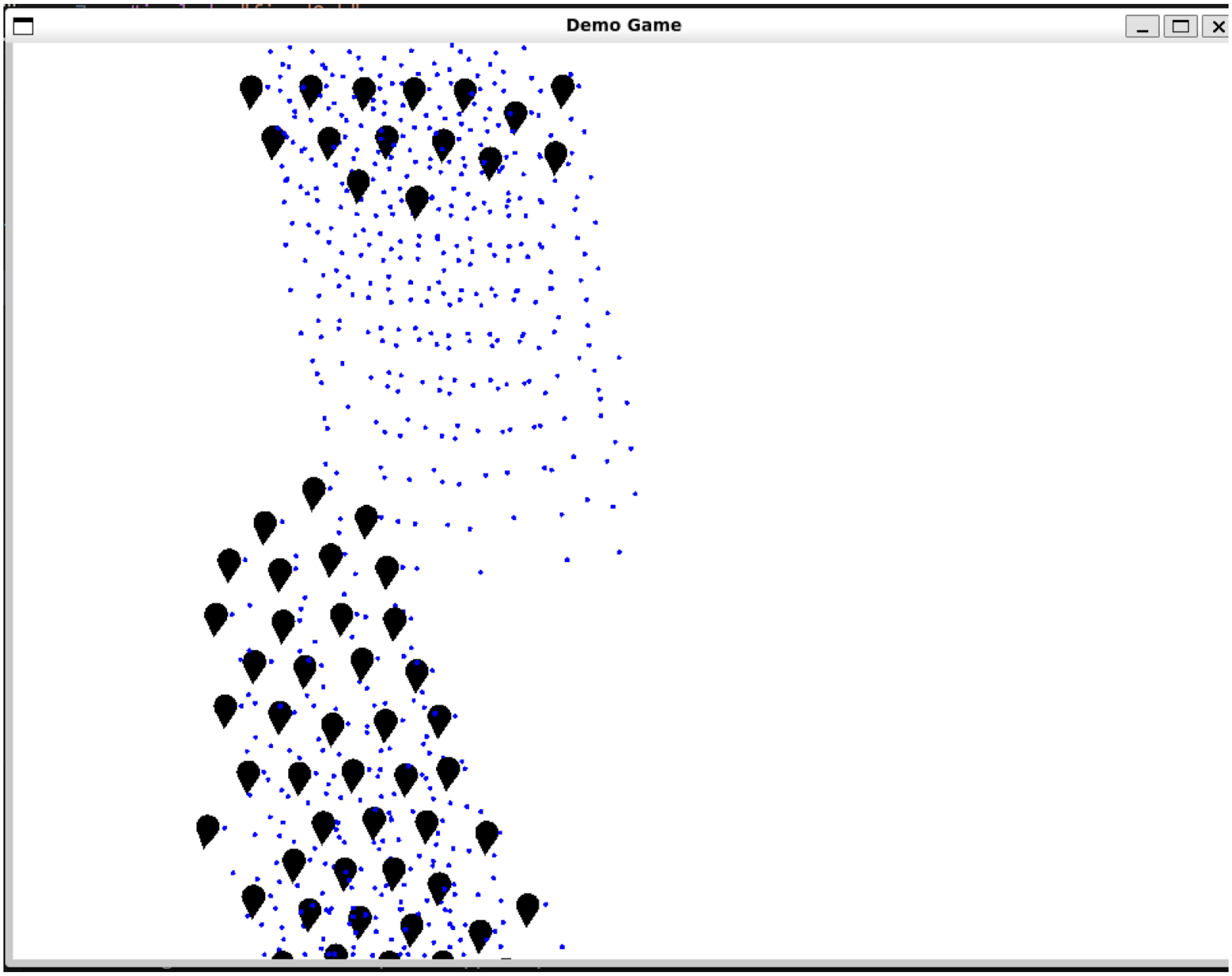
With a turn adjustment of 5 they finally pay attention to the turn modifier but become unable to flock



Multiplying the affected velocity component by -1 seems to break down flocking significantly as well



At .8 turn supplied they still flock well however they are encouraged to flock fairly tightly it seems



They can still change direction however, though not frequently.