COMP3180 – Final Project Report

Name: Katrina Jillian Tagle
Student ID: 46728066
GitHub Username: KitKat5100

Table of Contents

Note: I could not set the original project repository to public since it's just a fork. So, I made a copy of everything into another repository that I could make public and link here.

1 Project Deliverables

The aim of this project was to use the BOIDs AI algorithm to accurately simulate the movement and flocking behaviours of a school of fish. I constructed the following deliverables:

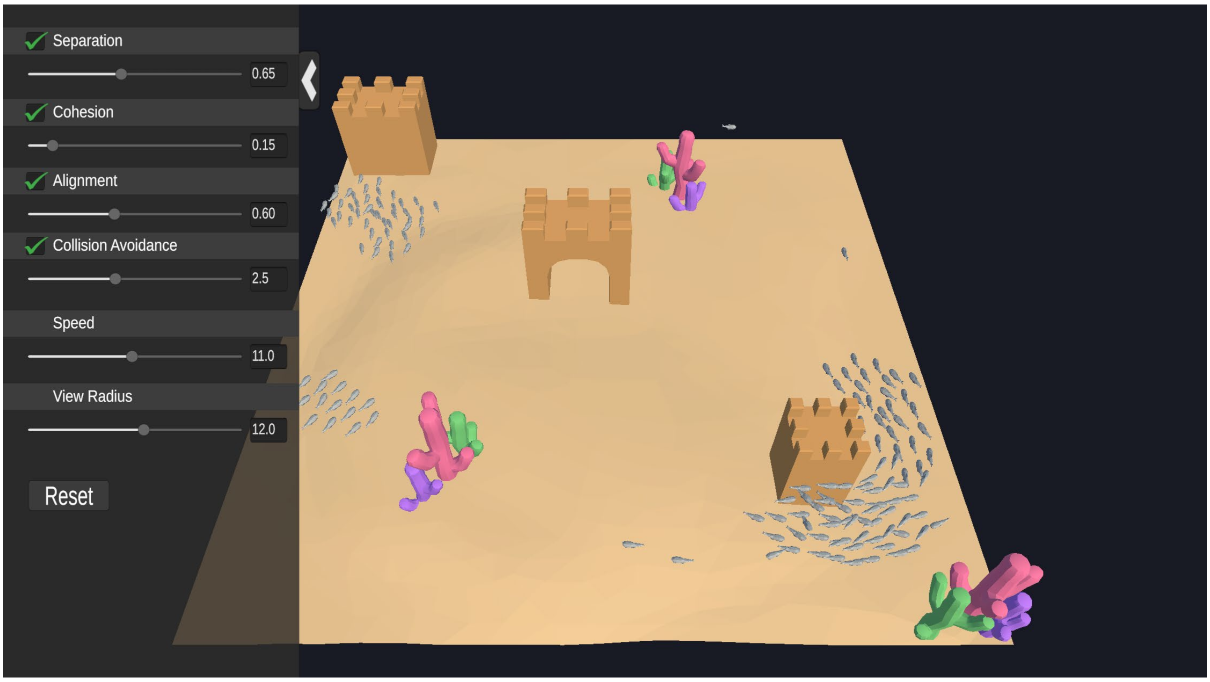
1.1 Unity Project (Portfolio Piece)

https://github.com/KitKat5100/Final-Project-Copy/tree/d44f4003c6a90ed34da8675736fa556f7183f77c/Final%20Project%20-%20BOIDS

A Unity project featuring 150 BOIDs moving around a simple scene. Obstacles in the scene can be moved around, and clicking on the left arrow will open a parameters panel where users can edit some of the values used for the BOIDs movement.

The overview of the algorithm is as follows:

1. The BOIDs algorithm was implemented as specified in Craig Reynolds' (1987) paper, using some of the math from Millington's (2019) book. Each BOID has a limited view range and moves according to the concepts of alignment, cohesion, and separation. The vectors corresponding to each concept are combined using a weighted average
2. The BOIDs also have raycast-based collision detection for obstacles and the edge of the scene.
3. Some tweaks to the algorithm have been made according to Hemelrijk and Hildenbrandt's (2008) research paper. Namely, dynamic view radius based on neighbourhood density, cruise speed, as well as changes to their view angles.



The scene in play with the parameters panel open

There's also a folder with a .exe file here: https://github.com/KitKat5100/Final-Project-Copy/tree/d44f4003c6a90ed34da8675736fa556f7183f77c/BOIDS

1.2 Demonstration Video (Portfolio Piece 2)

Youtube Link: https://youtu.be/FjnkfPxM0uA
GitHub Link: https://github.com/KitKat5100/Final-Project-Copy/blob/main/Boids%20Demo%20Video.mp4

The video showcases the BOIDs in motion, with an explanation of the algorithm and tweaks made to it.

1.3 Playtesting Data

https://github.com/KitKat5100/Final-Project-Copy/blob/d44f4003c6a90ed34da8675736fa556f7183f77c/BOIDs%20Evaluation%20Spreadsheet.xlsx

Playtesting data for 13 participants. The survey asked participants to rate their satisfaction with different aspects of the BOIDs movement, and rate how realistic the simulation was overall.

1.4 Playtesting Report

https://github.com/KitKat5100/Final-Project-Copy/blob/d44f4003c6a90ed34da8675736fa556f7183f77c/3180%20Playtest%20Report.pdf

The report contains my methodology for playtesting, as well as an analysis of the results, which will be summarised in Section 3 below.

2 Milestones

2.1 Up to Week 7

The first 4 weeks were spent researching about the topic. I quickly landed on the BOIDs algorithm and went from there. By week 7 I had the basic algorithm implemented with cohesion, separation, and alignment. I had experimented with two ways of combining the vectors: weighted average and a priority system. Math isn't my strongest suit, so I spent a lot of time just trying to understand it.

By this point, I had scoped out one of the planned deliverables: a third person underwater scene with a player character that could move around to look at the BOIDs

2.2 Week 8

I implemented collision avoidance (after some failed attempts) and combined it with the other flocking behaviours.

After spending some hours last week getting the priority system working, I couldn't nicely combine it with collision avoidance and ultimately removed it in favor of the weighted average method.

2.3 Mid-session break

Now that the project was taking more shape I rewrote most of the code to make it more efficient and removed the remnants of past failed implementations of things. I also spent some hours watching the simulation and fine tuning the parameters until they looked right.

2.4 Week 9

This week I reevaluated my progress and cut a few thing out: optimizing the algorithm and movement along the y-axis.

I also implemented most of the UI elements needed for my evaluation: the parameters panel and the ability to move obstacles + corresponding UI. Obstacle movement was trickier than I thought.

2.5 Week 10

This week I completed the parameters panel and made some decorations to the scene. I tweaked the BOIDs code to add cruise speed and a dynamic ViewRadius, and made a spawner for the BOIDs.

2.6 Week 11

This week I made the google forms to be used for the evaluation, and got some people to answer it. I also fine-tuned the BOIDs parameters a bit.

2.7 Week 12

This week I got more people to answer the google forms for my evaluations, and started on the Final Project Report and the Demonstration Video

2.8 Week 13

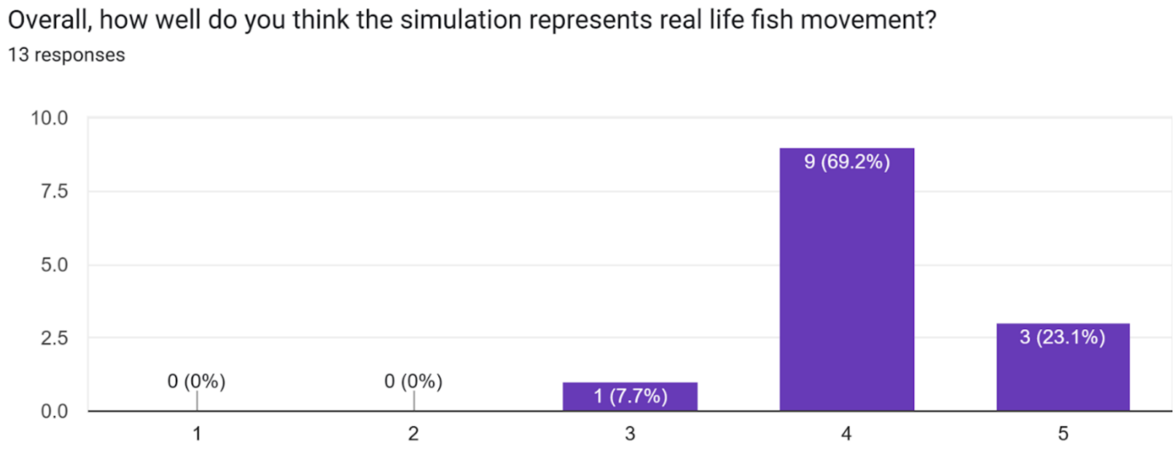
This week I finished evaluation, and worked on all necessary documentation for the submission of this project (final report, playtest report, demo video, etc).

3 Evaluation

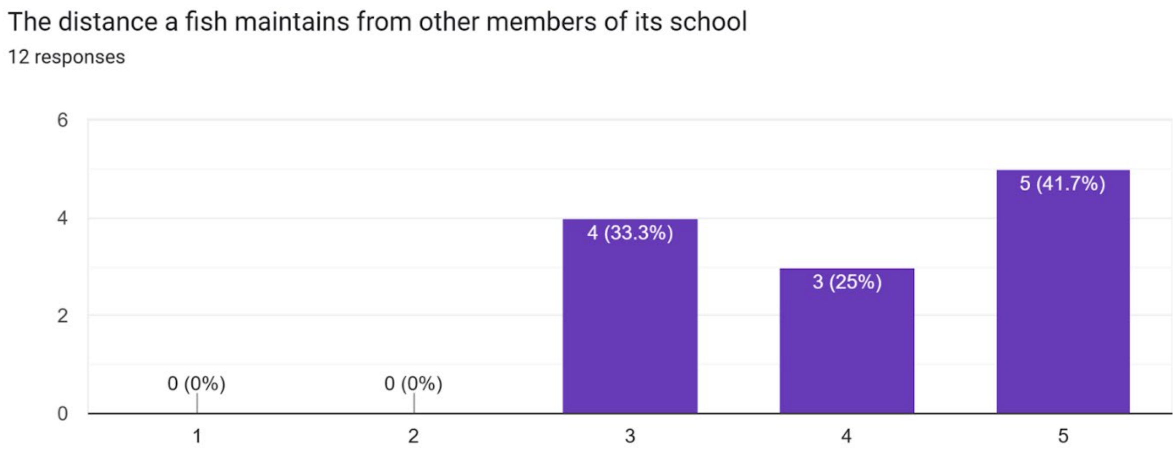
The goal of the project was to simulate real life flocking behaviors of fish, and used a survey to evaluate the success of the project.

The survey asked participants to observe the simulation and rate their satisfaction with different aspects of the BOIDs' movement in a sieries of Likert Scale questions. Afterwards, they could tweak some of the BOIDs' parameters to their liking. As the application of BOIDs in games isn't confined to certain genres, and the simulation was made to cater to a general sense of what people think is accurate, it did not seem necessary to target certain demographics.

The simulation was received very positively, with none of the respondents giving a 1 or 2 (unsatisfactory response) on any of the likert-scale questions, with the scores of each question average out to a 4 or 5. Respondents agreed that the simulation was overall fairly realistic.



The part of the algorithm that needs the most work is their separation code, which was the lowest rated aspect of the BOIDs.



This mirrors some of the respondents qualitative answers describing the fish as too "orderly" and requesting less distance between them:

> " I think that they could stand to be closer to each other at times, or be more okay with proximity and overlapping."
>
> "I also feel like it could be a little more disorderly or free."

However, some complained the opposite: that the fish overlap each other too much. Additionally, when given the chance to edit the BOIDs values, respondents on average increased separation.

Lastly, a number of respondents commented on the "jitteriness" of the individual BOIDs' movement when in a group, which is also caused by their separation code:

> "When fish are together, there is an awkward stiff wobble on the Y-axis."

“Fish movement a little bit jerky, could be a little smoother”

In the future, perhaps separation calculations could be further tweaked to have the fish stay closer together yet decrease their chances of overlapping, and also remove their jittery movement.

Another aspect that respondents seemed divided on was the movement speed of the fish. Although the likert scale responses were positive, when given the ability to tweak the values 3 out of 7 significantly decreased their speed while 2 out of 7 significantly increased it.

However, respondents seem very satisfied with their flocking abilities overall, as the BOIDs’ ability to form and maintain schools, as well as the shape of the schools that they formed, were the highest rated aspects of their movement (figures 1-3). This, along with the general positive reception of the simulation, is a good indicator that the main goal of this project: simulating flocking behaviors, was succesfully achieved.

4 Learning Reflection

For this section I am not using the exact wording used in my research report, as, upon revisiting it, what I wrote was rather unsatisfactory. I have reworded it to convey the same thing in a better way.

1. Create a 3D underwater scene that uses the BOIDs algorithm to simulate fish flocking behaviours.

   I mostly completed this goal, making a scene showcasing the BOIDs algorithm fully working. Although, the fish only move along x and z axes, as I had to scope out full 3D movement due to a lack of time. Math is not one of my strong suits, but through this project I was able to practice and improve my skills with vector maths, although I underestimated how much time it would take me. I had to work a lot with vectors, velocities, and quaternions, and such practice will surely be valuable to me in the future, as that field of math is often used in video games for movement, projectile calculations, etc.; even if the implementation of the BOIDs algorithm itself is not something I’ll use often in industry.

   Additionally, I ran into a roadblock during my research as I could not understand the details of Craig’s original paper on BOIDs. Instead of continuing to struggle with it I looked for other resources online that could teach me the concept in a way that I would understand. This is an important lesson for self-directed learning: don’t give up when faced with an unfamiliar topic, look around, go on forums, and maybe there’s someone out there that can explain it in a better way.

2. Experiment with different tweaks and implementations of the algorithm for a more realistic simulation

   This part of the project took more time than I thought, and, regrettably, ended with a lot of hours spend on code that never made it into the final product, but I did experiment with the algorithm, with varying results. It was hard to find applicable ways to tweak the algorithm that were well-documented, as most changed the algorithm in oddly specific ways tailored to their project, but it worked out in the end. Comparing videos of the project before and after said tweaks were made, I believe that it was able to add to the accuracy of the simulation. I would've liked to do more, if it weren't for the performance limitations of my algorithm.

3. Optimize the algorithm thorough the implementation of parallel programming or spatial subdivision.

I was completely unable to satisfy this learning goal. I had over scoped, spent longer than I though implementing and tweaking the algorithm, and found that I didn't have the time to optimize it. I overestimated my math skills and underestimated the math that went into the algorithm. I shall be careful to not make the same mistake in the future. This topic is something I can tackle if I ever revisit the project.

5 Industry Relevance and Future Work

Although the BOIDs algorithm isn't useful across all games, I still learned and developed skills that are broadly applicable across the industry. Vector maths in particular are used often in video games for coding movement, projectiles, rotations, etc. It was also an exercise in general coding practices of keeping code organized and well commented. Even if I never got the point of optimizing the algorithm on a large scale, I had refactored the code multiple times, removing smaller instances of dead code, repeated lines that could be put in a function, values that could be stored instead of recalculated, etc. Working on this project improved my general coding skills, and the project itself is a proof of my ability to deliver on these skills.

Throughout my research I found many resources that, while utilizing BOIDS in some way, offered no information to me that was useful or within scope. For self-directed learning, it was important to skim a paper to quickly determine if it was relevant to me to not waste too much time on research and not get sidetracked. Additionally, if I did not understand a concept, instead of continuing to struggle with it, I needed to search for other resources that could explain it to me better. I believe that these concepts will continue to be important to me for self-directed learning in industry contexts.

6 Appendices

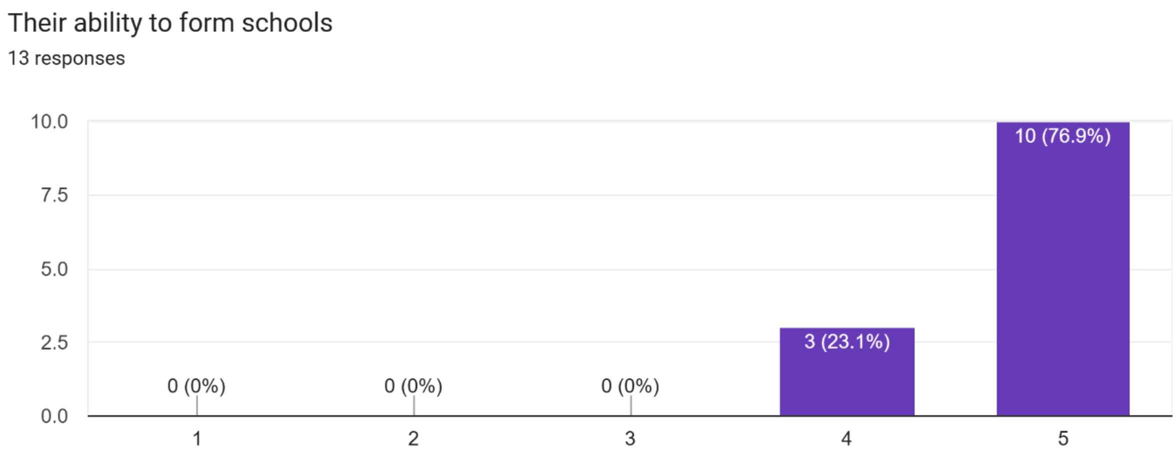


Figure 1: excerpt from playtest results

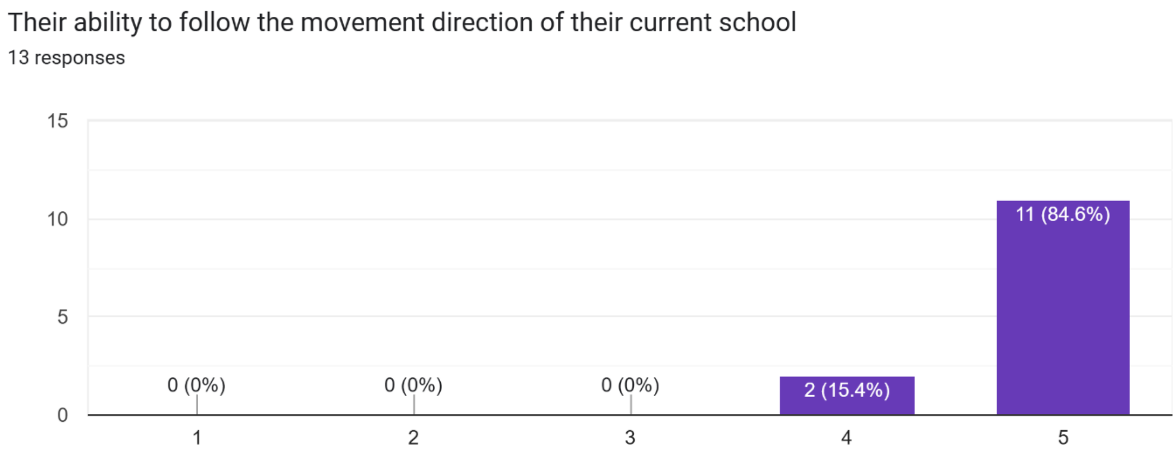


Figure 2: excerpt from playtest results

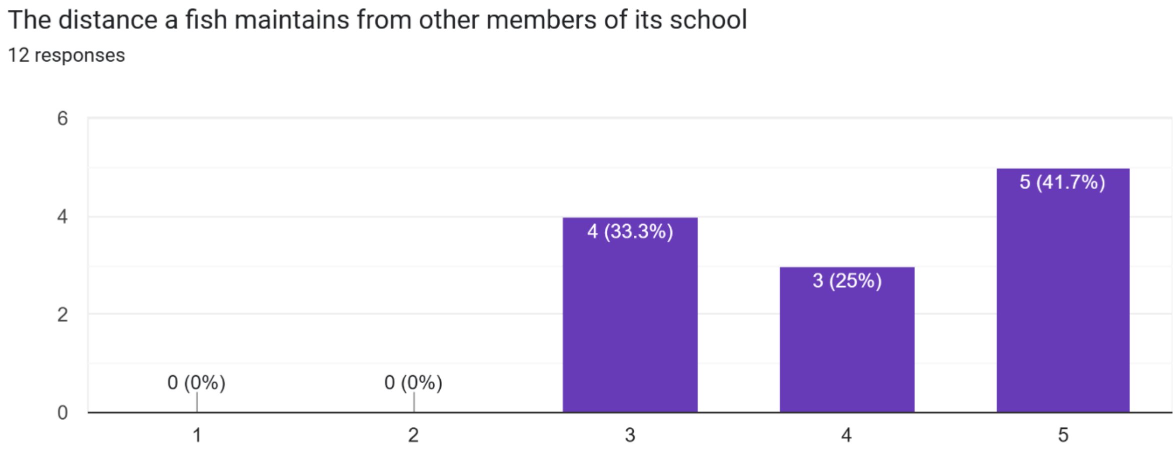


Figure 3: excerpt from playtest results

7 References

Hemelrijk, C. K., & Hildenbrandt, H. (2008). Self-Organized shape and frontal density of fish schools. *Ethology, 114(3), 245-254.* https://doi.org/10.1111/j.1439-0310.2007.01459.x

Millington, I. (2019). AI for Games, Third Edition. *CRC Press*. https://www.oreilly.com/library/view/ai-for-games/9781351053280/

Reynolds, C. (1987). Flocks, Herds and Schools: A Distributed Behavioural Model. *Computer Graphics, 21(4)*. https://dl.acm.org/doi/abs/10.1145/37401.37406

Third-Party Assets:

Fish Model:
Alstra Infinite. "Fish – PolyPack."
https://assetstore.unity.com/packages/3d/characters/animals/fish/fish-polypack-202232