Investigating the Impact of a Genetic Algorithm on Inter-Flock Dynamics

Matthew James Hunter

BSc(Hons) Computer Games Technology
2019

School of Design and Informatics
University of Abertay Dundee

Contents

List of Figures

List of Tables

Abstract

Here is my abstract and where i shall be placing the content of soon.

1 Introduction

1.1 General Introduction and Background Information

Flocking is a behaviour in which all social organisms engage; it is the common movement of organisms guided by both social and environmental pressures. These flocks of organisms can be found interacting with other flocks in nature, and the way they interact is as varied as it is interesting.

Producing flocking behavior that recreates those found in nature is an endeavor already undertaken and in constant update. The field, taking off in 1987 with Craig Reynolds' influential paper (Reynolds, 1987), shows how realistic flocking behaviour can already be achieved by applying 3 simple rules to each boid ('boid' as coined in the prior paper, this is essentially an agent) in regard to its neighbours: Cohesion, movement toward the average position; Alignment, movement toward the average direction; and Separation, movement to avoid collision.

Since then, the original flocking algorithm has been extended in various ways, with communication techniques, mathematical models for how leadership arises in the flock, as well as models for how consensus is made in a flock. These expansions allow for more complex behaviours and reactions to the enviornment and surroundings.

Learning behaviours have also been added. The behaviour and strategies flocks produce are patterns. This means if a flock can learn and understand those patterns it has a significant advantage over that other flock – an interesting example would be a group of honey hunters and smoke; bees flee the nest if they think there is a fire, the first warning sign to this is smoke, as a group (or flock) they take advantage of this by releasing smoke into the hive. The bees evacuate the nest; they get honey.

This dissemination of new knowledge, either through behaviour or communication is interesting because it can increase the complexity of the reactions the flock has to a given situation. This added complexity may lead to new behaviours that may not have been easily predicted or thought of as something a flock could produce.

1.2 Aim, Objectives and Research Question

Aim To investigate the impact of a genetic algorithm on the dynamic interaction of flocks with each other to see if this has a beneficial effect in comparison to regular flocking algorithms.

Objectives

- To research and evaluate AI techniques, studying their relevance and potential for further development in applying them to flocking algorithms, with particular focus on artificial life techniques.

- To produce an application that models flocking behaviour, and allows the observation and comparison of AI flocking strategies to regular flocking algorithms. This will be developed using the prior identified AI techniques most likely to produce viable flocking behaviours.
- To analyse the effectiveness of strategies that the AI come up with in their interactions with other flocks, comparing and contrasting that to the behavior of standard flocking algorithms.

Research Question What impact can a genetic algorithm have on the dynamic interaction of flocks with each other in comparison to that of regular flocking techniques?

2 Literature Review

2.1 Flocking in Nature - Swarm Behaviour

Flocking algorithms draw a lot of their inspiration from the behaviour of flocks in the natural world (Flake, 1998). As there are many examples of this behaviour in organisms across the planet, there is a lot of information and insight that can be gleaned on how to design these algorithms. Flocks are included in the field of Swarm Behaviour, that is, the study of swarms in the natural world

Decision Making The way decisions are made in a flock emerges is varied. The two main ways are via consensus and leadership. An interesting look at this can be found in the behaviour of pigeons. A study conducted into the behaviour of these birds in a flock (Jorge and Marques, 2012) found that leadership initially emerged from younger pigeons in the flock, but as the flight went on older members of the flock led the group, the paper then goes on to discuss how social versus personal information affects the behaviour of the flock. What this displays, is how the extra experience of older members of the flock is taken advantage of in determining leadership and therefore the actions they take as a flock, in this way they build consensus on their leadership through the choices individual flock members make in their group. This is further confirmed in *'Misinformed leaders lose influence over pigeon flocks'* Watts et al. (2016). What this displays is the interaction between consensus and leadership in making decisions, and demonstrates that they can both be present in the process.

Eusocial Behaviour While not specifically flocking as it pertains to biology, this falls under the bracket of swarm behaviour and has interesting relationships which can inform us on how to expand flocking algorithms in beneficial ways. For example, communication can occur in a variety of ways; ants use pheromones to find shortest paths (Dorigo and Gambardella, 1997); bees use a waggle dance to inform others in the hive of food sources and potential new nest sites (Al Toufailia et al., 2013), and

Dynamic Adaptation to Environmental Pressures

Learning and Curiosity

Flocking and Emergence

The Advantages of Flocking to Species

Why this is relevant This is relevant as it inspires the work done in flocking algorithms and the work of similar fields, which has influence on the design of said algorithms.

2.2 Flocking Algorithm

Flocking algorithms draw inspiration from the natural world, however the design of these algorithms involves mathematical approximations of the behaviours involved, the interactions that take place and the systems overall.

Reynolds Boids In the often cited paper when it comes to discussions of flocks: *'Flocks, herds and schools: A distributed behavioral model'* Reynolds (1987), we see that we have the three main forces that make up a basic flock: Separation, Cohesion and Alignment. These are approximative forces that represent the aggregate motion of a collection of boids (representing flock members, which in turn can represent different species in nature).

Expanded Boids There are many variations of the expanded boids model [insert citations here] dependent upon what is necessary for the research conducted. The expanded model presented by Kawabayashi and Chen (2008) is very useful as their implementation is very clear and displays useful expansions to the model for this research. It adds functionality for running away from predators, attraction to food sources, relations as towards other neutral flocks, the ability to search its surroundings, boundary conditions, and obstacle avoidance. All of which are useful in designing more natural flocking behaviour.

Decision Making In parallel to the way decisions are made in natural flocks, flocking algorithms can also make decisions. An interesting way of producing leadership in an artificial flock can be found in the paper textit'Autonomous Boids' Hartman and Benes (2006). Here they propose the use of an 'Eccentricity' variable to determine leadership in the group in order to make decisions based on the boids proximity to the front of the flock, the closer it is, the higher a chance of leading the flock. This mimics the way that some species of bird, such as starlings, decide on leadership within the group. This leadership can then be used to influence decisions by the rest of the flock.

Learning and Curiosity

2.3 The Interaction of Flocking Algorithms

In the paper *'Simulating Species Interactions and Complex Emergence in Multiple Flocks of Boids with GPUs'* Husselmann and Hawick (2011), multiple flocks, each a different type of boid, are run in a closed environment to see what aspects of species interaction could be reproduced. The

2.4 Genetic Algorithm

The literature on genetic algorithms also matters as this is what will be used to expand upon the capabilities of the flock. There is

2.5 The Potential Effect of AI on Flocking Algorithms

A paper discussing the use of reinforcement learning in a multi-agent system by Jaderberg et al. (2018) displays the potential of AI applied to a group of agents.

2.6 The effect of Genetic Algorithms on Boids

Genetic Algorithms used to improve the boids model is nothing new and provides useful information into how to go about this research. For example in *'Genetic Algorithms for Optimization of Boids Model'* Chen et al. (2006)

2.7 The Interaction of an AI Flock with another Flock

The interaction of an artificially intelligent flock with other flocks is where the research becomes thinner and where the author feels based upon their own background research there is a room for expansion.

Figure 1: A UML Class Diagram of the Application

3 Methodology

An application was produced to test the hypothesis. This application used the Games Education Framework (GEF) [reference] as the framework, which was chosen based on its lightweight overhead, its portability and recommendations from several university lecturers. It was developed using Visual Studio 2017 in C++, chosen due to the author's familiarity with them and the flexibility afforded by both tools.

GitHub Desktop was used as the tool for source control, ensuring risks to the project were kept to a minimum. Further to this, a GitFlow tool was developed in order to stem the risk of pushing and pulling work from incorrect branches, and prevent any conflicts when committing and pushing new work. The tool essentially acted like a smart checklist, by starting a programming session with it, it meant everything was set up correctly before opening individual programs which it did for you as the correct items were ticked off.

3.1 Application Design

The application was designed through an iterative process, and required many rethinks in its design,

Fig.1 displays the class diagram

3.2 Test Environment Design

The application simulated an environment, in which two flocks interact (one of which is improved via a genetic algorithm) within a limited space of scarce resources, with the aim of finding out if a flocking algorithm improved by a genetic algorithm can out-compete one without.

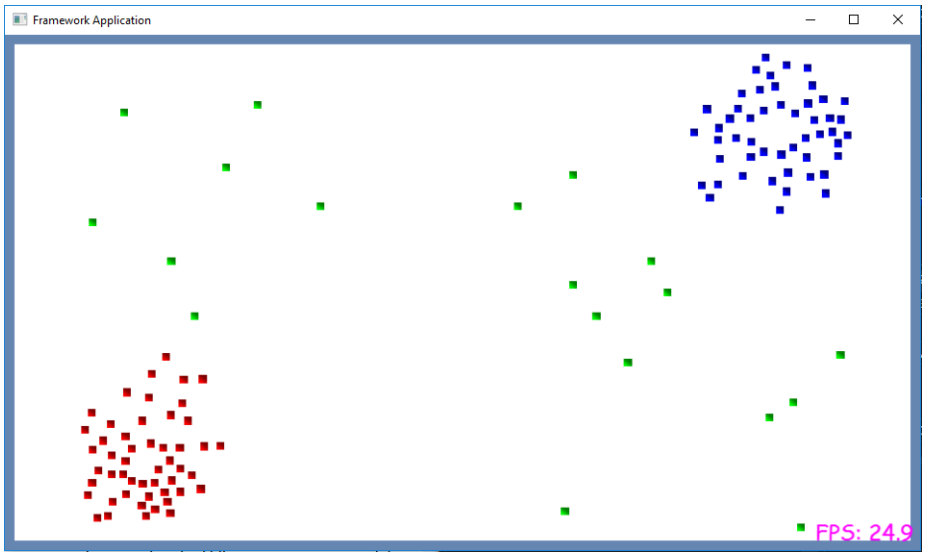

Figure 2: A screenshot of the test environment in action

To answer the research question required removing as many base assumptions as possible - reducing the problem to its core necessities. To do that, three core decisions were made: The environment should be limited so as to keep the flocks in competition with each other, this means the flocks have to interact; there should be a limited amount of resources in the environment, less than that required to support two full populations of flocks, this should drive the flocks to compete over resources spawned in the environment; finally there should only be two flocks, one improved by a genetic algorithm over simulated generations and one produced through a tested expanded flocking algorithm design. This ensures a competitive environment in which the effects of decisions the genetic algorithm makes can be singled out, and the data produced analysed. Fig.2 displays a screenshot of this environment in action: the red entities representing the regular flock; the blue entities representing the genetic algorithm modified flock; the green entities representing resources in the scene; and the white area displaying the space in which the boids can move about.

3.3 Flocking Algorithm Design

The flocking algorithm was originally designed primarily by adapting the example from Shiffman (n.d.). As this was a very clear representation of how to design a basic flocking algorithm from a code perspective. There were also clear areas for improving the efficiency of the algorithm presented too. The most major way was combining the calculations for Alignment, Separation and Cohesion. This was so each boid was not checking against all others for each force calculation, and only needed to check against all other boids once instead

of three times; doing that was simple and reduced the requirement for N-Body problem solutions like the Barnes-Hut algorithm later on. The other main way was combining the calculations required for each force as they use shared variables, so calculating them once instead of multiple times for each force also increased efficiency.

This worked ok for the original three forces based on Reynolds (1987). However, issues came up when expanding the application to include extra forces. Especially since in the aforementioned implementation it wasn't strictly clear how the forces would graph out due to the lack of separation between direction and weighting. This caused issues to pile up and it was clear a change in the flocking algorithm design was necessary to expand and edit the program appropriately.

The boids' forces as they are in the final program were modelled primarily off Kawabayashi and Chen (2008), which has a very clear mathematical representation of each force. The version of the expanded boids algorithm found in this paper was modified by optimising the algorithm for the environment in which it was placed. The forces each boid experienced were: Cohesion, Alignment, Separation, Food Attraction and Flock Avoidance. Each of these forces were the result of multiplication of a unit vector and their respective weight, mathematically represented in Eq.1

$$\boldsymbol{ForceVector} = \boldsymbol{v} \cdot \boldsymbol{w} \tag{1}$$

Where $\boldsymbol{v}$ is the unit vector which describes the direction of the force, and $\boldsymbol{w}$ is the weight that magnifies the force dependent on its relation to its environment.

The relationship of the weight to the unit vector will become clear as the boid forces are described further on. This separation of direction vector and weight, meant it was far easier to mathematically model and graph out each force to predict their behaviour; this caused a reduction to the amount of work necessary to produce a functioning flocking algorithm. The resulting accelerative force then is the sum of these forces. That sum is then used in the Semi-Implicit Euler method to produce the updated position of each boid every frame.

3.3.1 Cohesion

This is the first of the boid forces, cohesion pulls the boid in the direction of flock members, in this case towards its local flock centre. The boids local flock centre is determined by its neighbours in communicable range and is the average position of those flock members. The further the distance away from its neighbours, the stronger the force. Below is Eq.2 describing the direction vector and weighting:

$$\begin{aligned} \hat{\boldsymbol{v}}_{\boldsymbol{coh}} &= \frac{LFCVector}{|LFCVector|} \\ \boldsymbol{w}_{\boldsymbol{coh}} &= \frac{(|LFCPos - BoidPos|)^2}{30 \cdot FlockSize} \end{aligned} \tag{2}$$

Where $LFCVector$ represents the vector from the boid to the local flock centre, $LFCPos$ is the position of the local flock centre, $BoidPos$ is the position of the boid and $FlockSize$ is the number of flock members as a whole.

3.3.2 Alignment

Alignment is a vector that accelerates the boid in the average direction of its neighbouring boids. This produces the common direction the flock as the boids individually move about. By taking the average velocity of neighbouring boids as a direction vector and multiplying it one over the distance between the local flock centre and itself a suitable alignment vector is produced. Below is Eq.3 describing the two components of the alignment vector:

$$\begin{aligned} \hat{\boldsymbol{v}}_{\boldsymbol{ali}} &= \frac{LFVelVector}{|LFVelVector|} \\ \boldsymbol{w}_{\boldsymbol{ali}} &= \frac{1}{10 \cdot |(LFCPos - BoidPos)|} \end{aligned} \tag{3}$$

Where $LFVelVector$ is the average velocity of the local flock members, and $LFCPos$ and $BoidPos$ are the same as in Eq.2.

3.3.3 Separation

Separation is the force that stops flocks getting too tightly knit. By having enough space between flock members it grants maneuverability where there would otherwise be collisions that slow movement down; that could be from obstacles, other moving entities or other boids in the flock. The unit vector that represents the direction for the force to be applied is calculated by taking the negative of the vector to the nearest neighbouring boid; multiplying that by the weighted multiple of the number of neighbours divided by the distance to the closest neighbour produces the separation vector described below in Eq.4:

$$\begin{aligned} \hat{\boldsymbol{v}}_{\boldsymbol{sep}} &= -\frac{ClosestNeighbourVector}{|ClosestNeighbourVector|} \\ \boldsymbol{w}_{\boldsymbol{sep}} &= 0.025 \cdot \Big(\frac{NeighbourCount}{|ClosestNeighbourVector|}\Big)^2 \end{aligned} \tag{4}$$

Where $ClosestNeighbourVector$ is the vector from the boid to the closest neighbour, and $NeighbourCount$ is the number of neighbouring boids.

3.3.4 Food Attraction

This provides an accelerative force toward the nearest resource in the environment. This attraction to nearby food sources is the basis for the boids survival in the scene. It is the unit vector to the closest resource that gives the direction, and it is the weight which gives its relationship to food in terms of distance

away. The equations used are described in Eq.5:

$$\begin{aligned} \hat{\boldsymbol{v}}_{\boldsymbol{fda}} &= \frac{ClosestResourceVector}{|ClosestResourceVector|} \\ \boldsymbol{w}_{\boldsymbol{fda}} &= 0.0025 \cdot |ClosestResource|^2 + \frac{36}{|ClosestResource|^2} \end{aligned} \tag{5}$$

Where $ClosestResourceVector$ is the vector pointing from the void to the closest resource to it, and $|ClosestResource|$ is the distance to the closest resource.

3.3.5 Flock Avoidance

This is a vector that produces a repelling force away from another flock, and increases at the boid gets closer to the other flock; this is what stops the boids from ignoring each other and keeps them in a competitive environment as they will both be competing for those same resources. The mathematical relationship of the two factors producing can be seen in Eq.6:

$$\begin{aligned} \hat{\boldsymbol{v}}_{\boldsymbol{fla}} &= -\frac{OtherFlockVector}{|OtherFlockVector|} \\ \boldsymbol{w}_{\boldsymbol{fla}} &= \frac{300}{|AvgOtherFlockPos|} \end{aligned} \tag{6}$$

Where $OtherFlockVector$ is the vector pointing from the boid to the average position of another flock within interactable range, and $AvgOtherFlockPos$ is the position vector of the average position of another flock within interactable range.

3.4 Genetic Algorithm Design

The functionality for the genetic algorithm (GA) was spread over two classes - genetic_algorithm and DNA - where the genetic_algorithm class interacted with the flock and boid classes, and DNA interacted with the genetic_algorithm and boid classes. The GA had a standard approach to its design which can be seen in Fig.3.

Five populations were generated each time the algorithm was initialised; it could be setup to be either random or heuristic, where each population had a random set of data generated for its alleles, and the same but one population received the heuristic used in the regular flocking algorithm respectively. These populations of boids then made up the genetically enabled flock.

Each population produced contained chromosomes unique to that population, which is represented by an array in the DNA class, each GA boid had its own DNA object which is modified by the genetic_algorithm class. Collectively they made up their respective genotype. Each gene in a chromosome is used in the weight calculations of its respective boid, this can be seen represented below

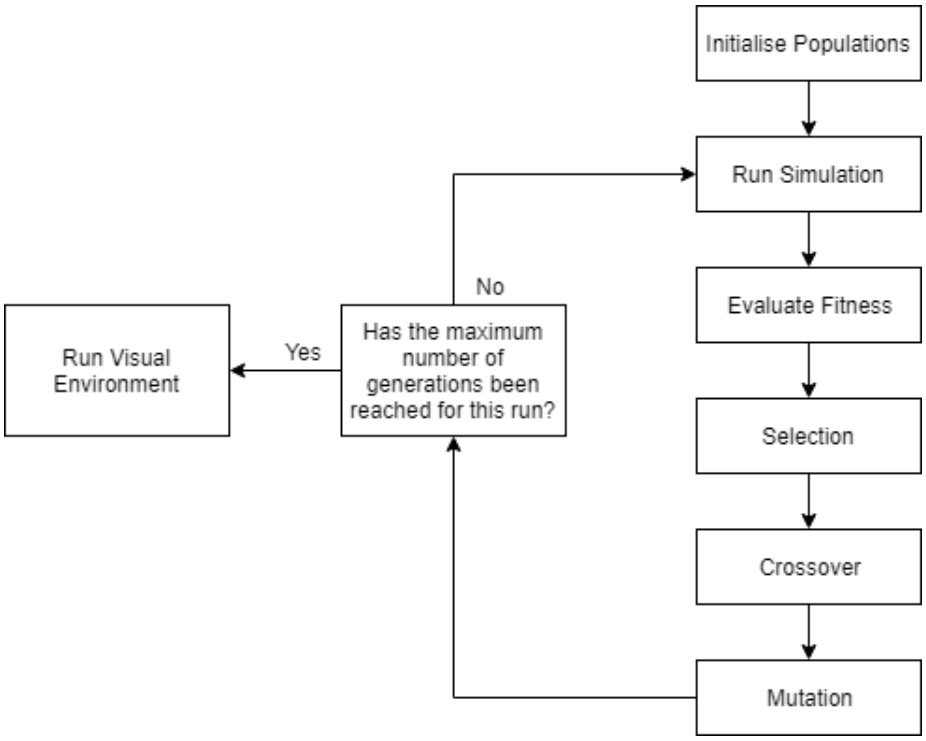

Figure 3: Flowchart displaying the process the GA takes to improve over time

in Eq.7:

$$
\begin{aligned}
\boldsymbol{w_{coh}} &= \frac{(chromo[0] \cdot |LFCPos - BoidPos|)^2}{chromo[1] \cdot FlockSize} \\
\boldsymbol{w_{ali}} &= \frac{chromo[2]}{chromo[3] \cdot |(LFCPos - BoidPos)|} \\
\boldsymbol{w_{sep}} &= chromo[4] \cdot \Big(\frac{chromo[5] \cdot NeighbourCount}{chromo[6] \cdot |ClosestNeighbourVector|}\Big)^2 \\
\boldsymbol{w_{fda}} &= chromo[7] \cdot |ClosestResource|^2 + \frac{chromo[8]}{chromo[9] \cdot |ClosestResource|^2} \\
\boldsymbol{w_{fla}} &= \frac{chromo[10]}{chromo[11] \cdot |AvgOtherFlockPos|}
\end{aligned}
\tag{7}
$$

Where $chromo[x]$ represents each gene in the chromosome for that boid, changes of the allele of each gene affect the weights in the scene. These weight calculations made up the phenotype representation in the environment space, and determined the behaviour of a boid.

3.4.1 Evaluation

At the end of each simulation each boid was evaluated against the fitness function, which can be seen in Eq.8, the evaluation function then sorts the boids into population types for Selection

$$FitnessFunction = BoidHealth + FlockHealth - OtherFlockHealth \tag{8}$$

Where $BoidHealth$ is the health of the boid, $FlockHealth$ is the combined health values of all boids in the flock, and $OtherFlockHealth$ is the combined health of all boids in the competing flock.

3.4.2 Selection

The application contained two different kinds of selection algorithms - Roulette Wheel Selection (RWS) and Stochastic Universal Sampling (SUS) - which have both been used in answering the research question. In both selection algorithms bigger probabilities are given

3.4.3 Generations

Part of running the GA was simulating the environment in sped-up runs for the genetic algorithm to improve itself over successive generations. This was done by cutting off unnecessary calculations, that meant all generations occurred in the update function in a loop until the maximum number of generations had been reached. This cut out the requirements for rendering or indeed any

calculations not related to the simulation of boid movement and running the genetic algorithm.

The calculations for the simulation physics were also cut down to speed up simulation time in each generational loop. By increasing the time per physics calculation you effectively speed up the time passed each simulation. For example: If one second of simulation time happens in one frame, then if the application runs at sixty frames per second, you can get sixty second of simulation time per real time second.

Each generation has a simulation time limit before

3.5 Testing Criteria and Metrics

Maybe write this one once youve actually done the tests youre wanting to talk about tbh because you have zero test design rn, and have given it thought, but not enough to write on, whereas if you just do the tests then you've got a whole thing to setup and the tests can be designed by trial and error.

Test for if the solutions that the GA produce converge at all because thats a sign of potentially finding the global or a significant local maxima in terms of a solution to the problem.

References

Al Toufailia, H., Couvillon, M. J., Ratnieks, F. L. W. and Grüter, C. (2013), 'Honey bee waggle dance communication: signal meaning and signal noise affect dance follower behaviour', *Behavioral Ecology and Sociobiology* **67**(4), 549–556.
URL: *https://doi.org/10.1007/s00265-012-1474-5*

Chen, Y.-W., Kobayashi, K., Huang, X. and Nakao, Z. (2006), Genetic algorithms for optimization of boids model, Vol. 4252, Springer Verlag, pp. 55–62.

Dorigo, M. and Gambardella, L. M. (1997), 'Ant colonies for the travelling salesman problem', *Biosystems* **43**(2), 73 – 81.
URL: *http://www.sciencedirect.com/science/article/pii/S0303264797017085*

Flake, G. (1998), *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*, A Bradford book, Cambridge, Massachusetts.
URL: *https://books.google.co.uk/books?id=0aUhuv7fjxMC*

Hartman, C. and Benes, B. (2006), 'Autonomous boids', *Computer Animation And Virtual Worlds* **17**(3-4), 199–206.

Husselmann, A. and Hawick, K. (2011), Simulating species interactions and complex emergence in multiple flocks of boids with gpus, *in* 'Proc. IASTED International Conference on Parallel and Distributed', IASTED, pp. 100–107.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., Sonnerat, N., Green, T., Deason, L., Leibo, J. Z., Silver, D., Hassabis, D., Kavukcuoglu, K. and Graepel, T. (2018), 'Human-level performance in first-person multiplayer games with population-based deep reinforcement learning'.

Jorge, P. E. and Marques, P. A. M. (2012), 'Decision-making in pigeon flocks: a democratic view of leadership', *Journal of Experimental Biology* **215**(14), 2414–2417.
URL: *http://jeb.biologists.org/content/215/14/2414*

Kawabayashi, H. and Chen, Y. (2008), Interactive system of artificial fish school based on the extended boid model, *in* '2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing', pp. 721–724.

Reynolds, C. W. (1987), 'Flocks, herds and schools: A distributed behavioral model', *SIGGRAPH Comput. Graph.* **21**(4), 25–34.
URL: *http://doi.acm.org/10.1145/37402.37406*

Shiffman, D. (n.d.), 'Flocking'.
URL: *https://processing.org/examples/flocking.html*

Watts, I., Nagy, M., de Perera, T. B. and Biro, D. (2016), 'Misinformed leaders lose influence over pigeon flocks', *Biology Letters* **12**(9), 20160544.
URL: *https://royalsocietypublishing.org/doi/abs/10.1098/rsbl.2016.0544*