Boids on NAC
Group movement over Computer Assisted Narrative

Estela Barros Barroco
DEI \ IST
Av. Rovisco Pais
1049-001 Lisboa - Portugal
(+351) 218 417 000
estela.barros.barroco@ist.utl.pt

ABSTRACT
Flocks and Herds movement has always been subject of enormous interest from humans. Although it might seem quite simple, computer simulating it turned out a great challenge, subject of various studies all around the globe. This paper summarizes the work done in this area, while searching for the best method to support a flock in NAC, Narrativa Assistida por Computador\(^1\), a program developed aiming towards aiding ordinary people creating stories. There was a successful integration between the program and the modulation of such synchronized animal group.

Keywords
NAC, Algorithms, Flocks/Herds/Schools, Models, Imagination, Choice.

1. INTRODUCTION
Through out the years, flock and other herds movement has always been able to capture human's interest and imagination; nowadays is no exception. Although it might seen fairly easy to reproduce, computationally it turned out a great challenge, subject of various studies all around the globe.

Nonetheless it is a wide area, involving many works related to flocks, herds among other animal group movements as well as humans. Due to humans’ great complexity, works related to us are generally focused on a single activity; the most recent ones exploit evacuation simulations and extras for games.

This work proposes a simulation of a flock/herd movement. Aiming towards the visual component of the research and developed made, it was applied over an existing application, NAC, Narrativa Assistida por Computador\(^1\), developed by the author and Leonora Henriques during the academic year of 2006/2007. This application aims to assist users in their stories creation, supporting the definition of scenes/characters and actions selection, which have both visual and textual descriptions associated. NAC distinguish itself for presenting no boundaries to the user’s imagination; the story teller is free to make whatever he desires with all elements of his story.

2. RELATED WORK
Considering the existence of a base application, NAC, great part of the representation problems have already been solved by it. Therefore the following works study the representation of group behavior. Although the objective proposed is on animals, many of the mechanisms are similar between species, thus the study of both animal and human flocks. The next paragraph summarizes some research made in this area.

Reynolds describes a revolutionary behavior model for bird flocking, where he believes a complex behavior can be achieved throw basic local rules over each element. Yu and Terzopoulos created a natural ecosystem simulation framework that valorizes realistic movements and complex individual behavior, carrying out a virtual fish world. Bouvier et Al defined a human urban simulator using a Reynolds based particle system, where each element is guide by attractive and repulsive forces between elements. Brogan and Hodgins produced a successful flock simulator (later applied to cycling) revealing great dynamics throw the use of a herding algorithm that added physical properties to the movement, making it much more realistic. Musse and Thalmann developed a real time crowd simulator with hierarchical models, where each human receives commands from its group and each group, of 3 possible kinds accordantly to independence, is defined by the general mob. Later, Niederberger and Gross exploit deepener this model, presenting a heterogeneous hierarchy for real time applications.

All these works are really good in their specific areas but none quite solves the proposed goal, being, therefore, a question of adapting a model to this proposed agenda.

3. SYSTEM ANATOMY
This project could be just theoretical however it is always more interesting, even from the designer’s point of view, to have some visualization of the research and development done. For this reason, it was applied over an existing application, NAC.

In this chapter, the application’s architecture is presented. There has been two major areas in this project; linking to NAC and algorithms development/implementation. The algorithms define the group’s behavior and will be detailed in their own section; in movement terms, one must be able to observe the group as well as access its controls, which shall be the subject of this chapter.

3.1 Access
In order for the user to interact with this new functionality, it must, somehow, be available to him, thus making vital the interface definition for this contact.

\(^1\) Computer Assisted Narrative
Since it is supposed to be part of the scenery, the groups are presented in this menu, scenario definition. To chose the animal displayed, the user has to select the proper 3D model in the main menu, always available. Nonetheless, once the 3D group models are loaded to the display, the only way to change them is creating a new group, destroying the existing one.

To introduce a group, the user has a textbox with a number on it which indicates how many elements take part on the group, by default 0. To add one group to the scenario, the user has to simply introduce the desirable number of elements, which shall be automatically displayed. Although appearing on the screen, the group won’t move while the intended algorithm is not specified by the user. There are 3 available ones; Algorithm1, Reynolds and Merged.

Figure 1 – Users interface on NAC
The great thing about this modularity is the possibility to modify variables in any given moment, such as the number of elements on the group, and to see the different results immediately, as a direct consequence of the alteration introduced.

3.2 Integration
The new module and its functionalities implied some modifications in NAC’s code. The main requisite of the new functionality is to obey NAC’s architecture; therefore its information must be contained somehow. Aiming towards minimizing and concentrating the code changes in a specific area, a new group manager, the BoidManager, was added. This new class contains everything necessary for the movement and definition of the group. A single class was chosen over an entire folder because this new functionality does not deal with great amounts of information.

BoidManager defines the main functions necessary for the welfare of the project. Firstly, it calculates the original position of the whole group; then it sets the graphical engine to take care of the correct representation of the elements accordingly with 3D models, position and dimensions. Finally, every one second that passes by, the class determines the elements new position, considering both time and chosen algorithm. After that, and once more, sets the graphical engine to display, only this time the move of the elements from its current place to the calculated one. This flow is responsible for the sense of motion and animation of the group.

3.2.1 Graphical Models
There are thousands of 3D models for both objects and animal over the Internet. However, frequently the quality and format of those models does not quite correspond to the expectations, which is extremely relevant when looking for specific 3D models; in this case, birds or fishes. Since the graphic engine already support a great deal of functionality, the quality of the representation is a direct consequence of the 3D model detail. In contrast to what one might think, the majority of the available models has a great deal of detail; nonetheless, in this particular case, this detail becomes a burden since it implies much slower processing, compromising the fluidity of the program, without adding substantial improvements over a simpler model complemented by a good texture. Therefore, after some treatment and acknowledgment of the important issues above, two reasonable 3D models where selected to the program, presented in the figure below.

Figure 2 – 3D models available for the new functionality

4. ALGORITHMS
The algorithms implemented are described in this section, introducing the general idea of each one as well as its basic work.

There are 3 algorithms, all referenced in NAC’s menu and, therefore, always available. The user must choose one of them that will condition the group’s behavior; nonetheless, at any given moment one can change the algorithm and observe the differences immediately. Notice that a cycle of calculations is corresponds to the call of the MovimentaBoid function, implying having passed at least 1 second of the previous iteration.

4.1 Algorithm 1
The name of this algorithm is a direct consequence of being the first one created/implemented.

Its basic idea is to have an imaginary defined route, set by some points in the space. In each interaction cycle, the system searches the next control point on the route. It compares this point with the position of the key element of the group, the leader, stored in position number 0. From the subtraction of the positions in question, a displacement is calculated, which will be applied to each element of the group; this way, the elements of the group maintain their relatively position towards each other within the flock. Thus, adding the displacement to each element position, the system calculates everyone’s new position and shall dislocate the elements during the next drawing cycle of the Irrlicht, NAC’s graphical engine.

Although quite simple, the same displacement for every element at the same time produces one movement alone, since all elements maintain their exactly relative position, like the following image illustrates. This brought up the need for modifications on the algorithm, in order to make the movement less uniform, thus adding more realism.
first, the last one may seem easier to control; after all, it is enough not to move all elements at the same time for the effect to wear off; the problem with this approach is that the MovimentaBoid is a time related calculus, since the new positions are not determined in each iteration of the program. Moreover, for a group of reasonable size, many of the elements will be moved at the same moment which implies that, although smaller groups, some elements of the flock will maintain the undesirable behaviour. So, the vector approach was used instead; if all the components of a vector are normalized by one scalar, the movement’s direction is retained, even though its module is different, which implies differences in the distance covered by the movement.

![Figure 4 – Distinct displacements applied to the elements.](image)

With this correction, the group elements staying close to each other although with different relative positions, thus obtain a much more realist movement.

4.2 Reynolds Algorithm

Craig W. Reynolds introduced the revolutionary idea that one could simulate a flock/herd of animals using only simple rules for local control of the elements. Despite later self-assessment reviews added other elements, such as states machine, the basic idea of its work was implemented in this algorithm. Notice that to Reynolds the elements of the group are referred as boids.

In each cycle, each boid position must be recalculated. In this algorithm, the new position of one boid is the result of the addition of its current position with its calculated speed at that moment. This speed is obtained throw the addition of the current speed with the local rules Rule1, Rule2 and Rule3, followed by some constrain limitation, in order to ensure that the boids stay within visible area of the screen (v4). Next, the algorithm pseudo-code is presented, followed by a brief explanation of each rules.

```plaintext
foreach boid
    v1 = Rule1
    v2 = Rule2
    v3 = Rule3
    v4 = SetVisibleArea
    ElemSpeed = ElemSpeedNow + v1 + v2 + v3 + v4
    newPosition = ElemPosition + ElemSpeed
```

**Rule 1 – Boids try to fly towards the centre of mass of neighboring boids.** Reynolds calculates the position’s average for all boids (mass center); although this calculation reduces computational effort, the boid’s own position is interfering with the calculus. Therefore, the perception center is used instead.

**Rule 2 – Boids try to keep a small distance away from other objects, including other boids.** This rule aims towards preventing collisions, whether between boids or other objects in the virtual world. To do so, one checks the distance between the boid and every world object available; if it is smaller than a particular scalar, then the boid should pull way from it, by adding a contrary direction of the object. With this movement, a small repulsion is obtained, without renouncing a flow motion.

**Rule 3 – Boids try to match velocity with near boids.** With the exception of its one, all the other boid’s speeds are used in the calculus of the global arithmetic mean of the speed.

4.3 Merged Algorithm

As its proper name suggests, this algorithm is the result of the junction of the previous two. The underlying idea is a Reynolds algorithm with some guidance on the group’s movement. This away, instead of simply applying Reynolds local rules to each element, a sense of route is also added, which is taken into account when determining the element’s new position.

With that in mind, the new position is determined by the formula

```
newPosition = Position_A1 + ElementSpeed
```

where Position_A1 is obtain through Algorithm 1 and the ElementSpeed is a direct consequence of Reynolds rules apply to the element under review.

This way one can observe a certain displacement on each element despite its clear sense of flock.

5. RESULTS AND PERFORMANCE

With realism in sight, a reasonable number of participants, respect for the physics laws and real time was established as goals for the project. The minimization of the computational effort and the secondary plan of the functionality on NAC implied the elimination of characteristics not absolutely necessary, as intelligence, hierarchy and heterogeneity. All these expectations where achieved, resulting in at least 50 elements groups in every algorithm without any extra cost.

For a better algorithm comparison/evaluation, two tests were used; cargo test and scatter measures. The results are presented in the two following subsections.

5.1 Cargo Tests

These tests allow users and developers to measure the performance of applications. In this particular case, the main interest is to determine the moment when processing information starts to interfering with NAC’s functioning. With this objective in mind, each algorithm was tested 10 times where, every time, 50 more elements where added to the group. The partial results, in 100 multiple groups, are presented in the table below.

| Table 1 – Cargo test results in FPS (Frames per Second) |
|---|---|---|---|
| Number Elements | Algorithm 1 | Reynolds | Merged |
| 100 | 90 | 75 | 142 |
| 200 | 48 | 20 | 29 |
| 300 | 33 | 0 | 0 |
| 400 | 25 | 0 | 0 |
| 500 | 20 | 0 | 0 |

2 Simulated bird-like objects, "bird-oid".
By the results obtain, one can easily verify that Algorithm 1 has the best performance, processing up to four hundred elements, slowing down only at 450 elements.

The values obtain throw this test are a direct consequence of the algorithms complexity. Algorithm 1, with the best performance, has a N complexity, which means that in each iteration a calculation is made for each one of the N group elements, whereas both Reynolds and Merged Algorithms present $N^2$ complexities. Nonetheless, these last two algorithms have transparent rules applications, which imply one iteration for each rule on each element. The rules can easily be condensed in a single iteration, therefore reducing the complexity to N, being this effort obtained throws code reading and memory during the processing.

5.2 Scatter Measures

This study objective is to determine the relative positioning and distance between the elements of the group, so that it can be comparable to animal groups in nature.

The sample data used, a set of 3D vectors, was retrieved throw two observations for each algorithm, one with 10 elements and the other with 20. The data analysis appealed to Inter-Quartile amplitude, average and shunting line-standard. This last one gives the variability or dispersion in the same unit as the sample data, being as big as the dispersion of the data. The Inter-Quartile amplitude, on contrary of the amplitude, is not extremely sensible to the existence of a small number of very small or great observations in the sample. This measure is defined as being the difference between 1st and 3rd Quantiles, which implies that 50% of the observations made are contained by an interval with that amplitude.

These concepts applied to the sample evidenced that the elements are relatively next to each others, verifying the unit/flock sensation that it intends to transmit. Although not so evident, there is a trend for the second observations to get bigger values then the first ones. This happens because with more elements becomes more complicated to, simultaneously, concentrate the group elements around themselves and prevent the collisions. In the algorithms with attractive/reclusive rules, as Reynolds and Merged Algorithms, at any given moment, the elements are to move away or come closer to each other, being difficult to foresee which will be the type of movement at the moment of the sample harvest. This is the reason why Reynolds Algorithm presents great intervals on the second observation (20 elements).

6. CONCLUSIONS

With the intention of understanding the involved complexity in the project proposed to replicate nature’s beauty, the author studied innumerable developed works in the area. From its analysis it was showed that there isn’t a better model; the great question here was how to adapt the existing ones to best solve the proposed problem, adding a set of animals (livened up 3D models) that dislocate autonomous and synchronously over NAC. In order for this to happen, it is essential to define an entity, a set of animals, with a minimum number of participants to resemble itself to a real group, reacting to the virtual world in real time, preventing collisions and respecting the laws of physics, excluding the none essential characteristics such as hierarchy and global control, aiming towards low processing and time spending costs.

When implementing, 3 distinct algorithms where created, all using a Reynolds similar approach (Flocking System), where the whole group is controlled only by local rules. The reason of these approaches proximity is lingered with the simplicity of functioning to the fact that the project is an enrichment of an existing program, not a central subject of whole new application.

After some analysis, Algorithm 1 was considered the best one, obtaining lesser dispersion values, better performances in the cargo tests and where the image of the movement is closer to what was intended to reproduce. However, all the algorithms accomplish the proposed objectives and, despite the distinct movements, all attended with success to simulate a group of animals, being up to the user to select the algorithm to apply, depending on the intended effect for the history being create.

Having achieved the intended effect, a nature similar simulation, fulfilling the proposed metrics as well as implemented the most relevant characteristics identified, it is considered that the objectives had been reached and, therefore, the project was finished successfully.

6.1 Future Work

Animals have more elaborated and abstract behaviors then simply follow the group throw a certain route and prevent painful collisions. Nowadays, there are behavior models that take into account other factors, such as hunger, search of food, escape the predators, sleeping between many others. Despite the fact that these motivations have not been considered, they might result in an interesting new approach to the subject. Equally, in the intent of obtaining more complex behaviors, hierarchy of the elements could be applied, passing the motivations throughout the power chain.

The developed movement is available for only one species. However, many other animals withhold similar behavior and therefore they can also be shaped by these algorithms, pending on the obtaining of the physical 3D models necessary. Also, the animals should move their members in accordance with its own displacement; in order to do so, instead of the current static models, dynamic ones should be used, although these are normally not available for free over the Internet. This topic is excellent and pertinent for the credibility of simulated groups.

In processing terms, its simplification should pass by the use of smart-objects, a more-value for project when improving the iteration between agent and object and, therefore, the proper animation.

7. ACKNOWLEDGMENTS

Thank my family, friends and colleagues whom stimulated me in the most difficult times during this project.

My special thanks to Pedro for his unconditional support and Leonora for her total availability.

8. REFERENCES


