

# Modelling the flight of birds in groups

Année 2017 – 2018

Anda-Monica LĂZĂRESCU<sup>1</sup>, Dan-Vasile ARION<sup>1</sup>

Teacher : Ariana Văcărețu

Colegiul Național “Emil Racoviță” Cluj-Napoca, Romania

Researcher: Lorand Parajdi, Universitatea “Babeș Bolyai”

## The Research Topic

We will try to model the flight of birds in groups. We will limit ourselves initially to a plane model (2D). The following rules are considered concerning the movement of the birds:

- ✦ One bird is characterized by a position in  $x$  and  $y$ , and a speed also in  $x$  and  $y$ .
- ✦ Every bird has a sight of radius  $R$ . It can't see the birds outside her visual field.
- ✦ It has a visual field smaller than 180 degrees and it cannot see beyond it.
- ✦ It is attracted inversely proportional by the gravity centre of the birds it can see.
- ✦ It is moving away from the birds it sees proportionally with the reverse of the square of the distance between them.
- ✦ Every bird changes its direction step by step (speed in  $x$  and  $y$ ) with the neighbouring birds.

The attraction and the remoteness are forces that you should consider as accelerations (with constant mass), which you should calculate at each step and add them at the speed.



Figure 1. Birds flying in group

In order to model the so-called flocking behaviours, we used the JavaScript programming language. The script file was included in a minimal HTML web page, in order to simulate the animation. The algorithms developed in this paper are based on the three rules developed by Craig Reynolds, the first person that has modeled the flocking behaviours in 1986: alignment, cohesion, and separation [1, 2]. The developed script allows changing the radius that defines the field of visibility of each bird, the initial speed, the initial number of birds, the dimensions of the

birds etc. Additionally, we have added some functionalities: - the birds can be added by clicking on the canvas surface; - using the +/- keys one can increase or decrease the birds' speed.

### Features of a bird

In this paper, each bird is graphically represented by an isosceles triangle, drawn on the canvas surface, shown in Figure 2. The tip of the triangle always shows the direction the bird is moving. The dot denotes the center of the bird, while the arrow is the velocity vector. A bird is characterized by a field of view. This is defined by a distance measured from the center of the bird, denoting the radius of the field of view ( $R$ ), and the angle of view ( $\alpha=180^\circ$  or 3.14 radians, in this study). Each bird sees only inside of its field of view and has the following features:

- 1 – *Location*: the x and y coordinates of the current position of the bird;
- 2 – *Direction*: the displacement vector or the angle of the current trajectory;
- 3 – *Velocity*: the speed at which a bird flies.

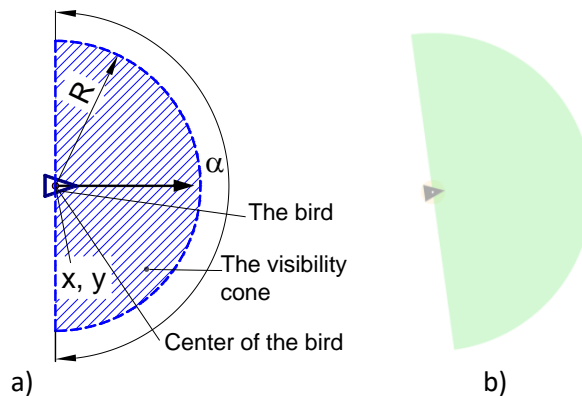


Figure 2. Schematic representation of a bird a) and a bird drawn using the JavaScript programming language b)

### The main pseudo-code for the simulation of flocking behaviour

The main pseudo-code for the simulation of flocking behaviour with animation contains the following procedures, shown in Figure 3:

- The "Initialisation\_of\_the\_birds" procedure draws and puts the birds on the canvas surface at the starting location. This procedure, also include the parameters that define the field of view of a bird. Each bird starts from a random position in a random direction at a random speed.
- The alignment (), cohesion () and separation () are algorithms that describe the behaviour of a bird and these involve vector operations. The result of these algorithms are the direction and the speed towards a target position, for example towards the center of the mass. These algorithms will be explained in the next sections.
- The procedure called "For\_each\_bird\_update\_the\_position", computes the new position of each bird for one step (or frame) of simulation. A bird moves towards the target position in several steps (frames).
- The "draw\_bird" procedure draw one frame of the animation on the canvas surface and repeat.

```

Initialisation_of_the_birds
For_each_step
  For_each_bird
    alignment ()
    cohesion ()
    separation ()
  end
  For_each_bird_update_the_position
    this.x=this.x + cos(this.direction) * this.speed
    this.y=this.y + sin(this.direction) * this.speed
    draw_bird
  end
end

```

Figure 3. The main pseudo-code for the simulation of flocking behaviour

## The alignment algorithm

Figure 4a graphically shows the alignment behaviour. This rule tries to make the birds mimic each other's trajectory and speed. Therefore, the alignment algorithm involves the updating of direction and velocity of a bird. The main steps of the alignment algorithm are:

- Check if a neighbour is inside the vision field;
- Compute the average speed of neighbours, which means summarizing the neighbours' speeds and dividing the computed vector by the neighbour count.
- Adjust the speed and direction of the bird.

The normalization of the speed is done in order to keep all the birds moving at the same pace, but we could have different birds with different maximum speeds.

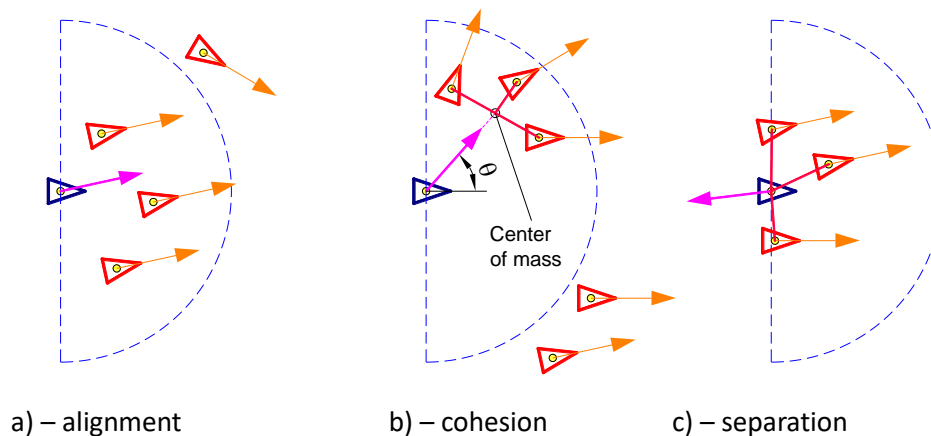


Figure 4. Graphically description of the bird behaviour

Checking if a neighbour is inside the vision field implies the computation of the distance between the reference bird and the neighbour as well as the angle measured from the reference bird's direction of flight and the straight line that joins the center of the two birds. A neighbour is inside the visual field if the distance is smaller than the visual radius and the angle is smaller than 180 degrees.

## The cohesion algorithm

Figure 4b graphically depicts the cohesion behaviour of the birds. The cohesion algorithm models the tendency of a bird to move towards the average position (center of the mass) of the local group, called target position. The result of the cohesion algorithm is the direction and sense towards the center of the mass. The main steps are listed below:

- The first step consists in checking if a neighbour is within the visual field of the reference bird and their counting.
- The next step implies the computation of coordinates of center of mass (center\_x and center\_y) (Fig. 4b). This is simply the average location of all neighbours. For example, assuming we have n neighbours, the center\_x is given by  $center\_x = (x_1 + x_2 + \dots + x_n) / n$  and the center\_y =  $(y_1 + y_2 + \dots + y_n) / n$ , where the  $x_i$  and  $y_i$  are the position coordinates of the  $i$ -neighbour.
- In the following the direction toward the center, the  $\theta$  angle, is computed using the inverse trigonometric functions:  $\theta = \text{atan}\{(center\_y - y) / (center\_x - x)\}$ , where x and y are the Cartesian coordinates of the bird in question.
- Partial straightening to the mass center.

## The separation algorithm

Figure 4c schematically shows the separation rule. The purpose of this algorithm is to keep a minimum distance between the birds to avoid collision.

The distance between the current bird and the neighbour is computed:  $distance = \sqrt{(x - x_i)^2 + (y - y_i)^2}$

If this distance is within a minimum defined distance ( $d$ ), the bird will move with  $1/d^2$  in the opposite direction. As in the case of cohesion, the distance from the reference bird to the neighbour is added to a computation vector. The computation vector is divided by the corresponding neighbour count. The computed vector needs to be negated in order for the reference bird to steer away from its neighbours properly.

## Result and Discussion

In order to check the functionality of the program, we started by considering only two birds (Fig. 5). After a few seconds, the two birds met and formed a group (Fig. 6).

The second scenario, shown in Figure 7, consider a number of 100 birds. As one can observe from this figure, the birds start in random directions. Figure 8 shows this case after the formation of some groups of birds.



Figure 5. A single bird meet a single bird – after the initialisation

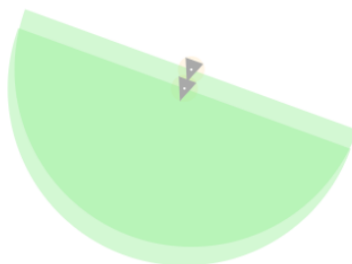


Figure 6. A single bird meet a single bird – after the flocking

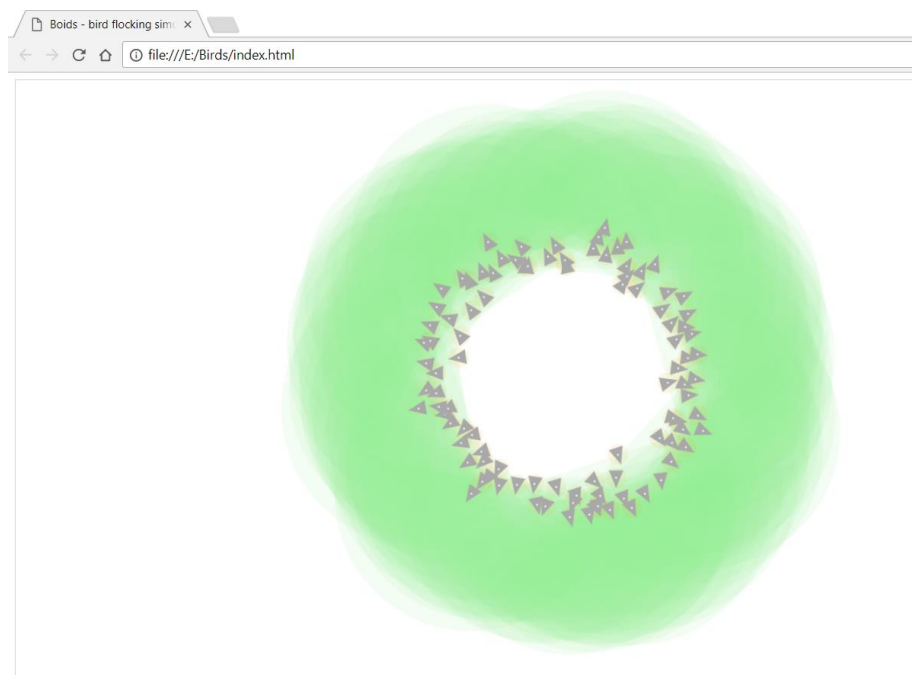


Figure 7. Starting with 100 birds – after the initialisation

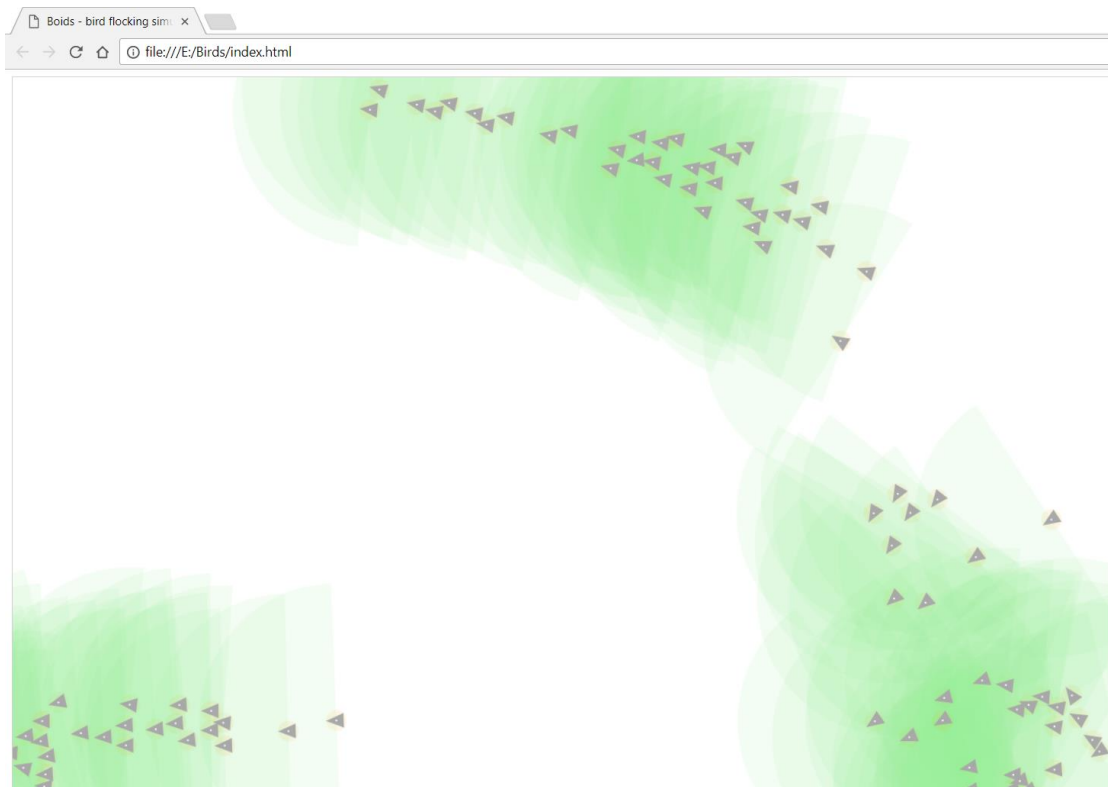


Figure 8. Starting with 100 birds – after the flocking

## Bibliography

- [1] Craig W. Reynolds, Flocks, Herds, and Schools: A Distributed Behavioral Model, Computer Graphics, Volume 21, Number 4, July 1987.
- [2] Craig W. Reynolds. Boids, Background and Update. <http://www.red3d.com/cwr/boids/>, July 30, 2007. [Online accessed: June 10, 2018].