# Assignment № 1
# Flocking Phenomena

Embodied AI, Vrije Universiteit 02/06/2020

## Problem Description

Flocking is the phenomena observed when a flock of birds fly together forming different simple to complex, often irregular, patterns. The collective behaviour involves no collisions while the entire flock is constantly moving, and most importantly, in a self-organizing fashion, thus, highlighting the complexity of the collective (Figure 1). Consequently, modelling this complex behavior has been a research focus for many swarm intelligence and swarm robotics projects throughout the end of 20th century and even nowadays. In this assignment we will examine in more depth the seminal implementation by Craig Reynolds [1].

Figure 1: Flocking

### Problem Definition According to Reynolds

In his work Reynolds introduced an individual-based model where each agent acts autonomously only based on information from its neighbors and without centralized control. An agent, known as 'boid', could execute some simple actions, and by having many of these boids acting together in a synchronous manner he would be able to create something similar to a bird's flocking behavior. The possible actions were defined as follows:

- **alignment** : steer towards the average direction of local boids

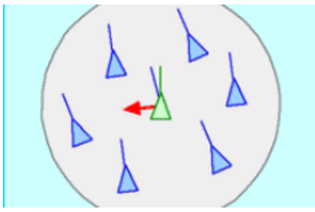

Figure 2: Alignment [2]

- **separation** : steer away to avoid being in too close proximity to local boids

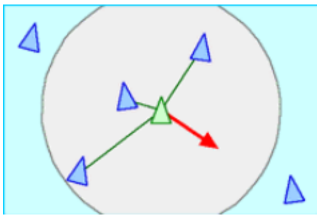

Figure 3: Separation [2]

- **cohesion** : steer towards the average position (center of mass) of local boids

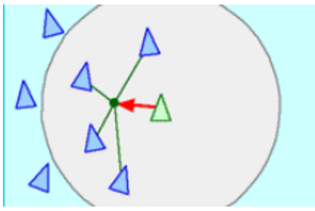

Figure 4: Cohesion [2]

**Implementation of the Forces**

As the figures already allude, the actions can be implemented by simple adjustments to the boid's velocity vector. In particular, the three actions can be expressed by the following vector updates (see below), where each action creates a steering force that updates a given boid's velocity, which subsequently updates the boid's position via 1st-order kinematics equation:

$$\mathbf{z}_{total} = \mathbf{z}_{alignment} + \mathbf{z}_{separation} + \mathbf{z}_{cohesion}$$
$$\mathbf{v}_{boid} = \mathbf{v}_{boid} + \mathbf{z}_{total}$$
$$\Delta x = \mathbf{v}_{boid} \times \Delta t$$

where $\mathbf{z}_{*}$ are the individual steering forces, which all are combined in one steering vector $\mathbf{z}_{total}$, $\mathbf{v}_{boid}$ is the velocity vector, $\Delta t$ is the duration of the movement, and $\Delta x$ is the resulting displacement.

- **alignment** (Figure 2):

$$\mathbf{v}_{neighbors} = \frac{\sum \mathbf{v}_{neighbor}}{N}$$
$$\mathbf{z}_{alignment} = \mathbf{v}_{neighbors} - \mathbf{v}_{boid}$$

  where $\mathbf{v}_{neighbors}$ is the average velocity of the neighboring boids (indicated by the blue line in the figure), $\mathbf{v}_{neighbor}$ is each individual neighbor velocity, and $N$ is the number of neighbors

- **separation** (Figure 3):

$$p_{diff} = p_{boid} - p_{neighbor}$$
$$p_{average} = \frac{\sum p_{diff}}{N}$$
$$\mathbf{z}_{separation} = p_{average}$$

  where $p_{boid}$ is boid's position coordinates, $p_{neighbor}$ is neighbor boid's position coordiantes, $p_{diff}$ is the difference between the boid and its neighbor (green line in the figure), $p_{average}$ is the average distance to all neighboring boids.

- **cohesion** (Figure 4):

$$p_{center} = \frac{\sum p_{neighbor}}{N}$$
$$f_{cohesion} = p_{center} - p_{boid}$$
$$\mathbf{z}_{cohesion} = f_{cohesion} - \mathbf{v}_{boid}$$

  where $p_{center}$ is the average position of neighboring boids (green dot in the figure), $f_{cohesion}$ is the cohesion force

In this assignment you will be provided with a python code for all of these forces. In the code you might notice some minor differences from the very simple description you have above, this is due to the fact, that most of these vectors are normalized, and the final forces are truncated in order to obtain stability.

## Assignment Description

In the present assignment you will be asked to experiment with a simulation of a flocking behavior. In particular, you will need to modify the file parameters.py. In this file you can find settings that define how the boid acts within the environment. Do not change the settings under the caption 'General settings', these are just meant for the general functionality of the code. What you can play around are the settings under the caption **'Flock class parameter'** and **'Boid class parameters'**.

In table 1 you can find a brief description of each parameter, while the general structure of the code you can find in the supplementary document 'Code tutorial'.

By the end of the week you are expected to present a video of the simulation with the best settings you found for each one of the tree possible environment settings: no object, boids inside a hallow shape, boids outside a hallow shape (so 3 videos in total). As a group you can choose which object type to use: convex or circular (you can play around with both, but you need to present only one of them). An example video can be found on canvas.

| | |
|---|---|
| ***Flock class parameters*** | |
| N AGENTS | number of boids in the environment |
| OBSTACLES | *True* if an obstacle is present in the environment, otherwise *False* |
| OUTSIDE | *True* if boids are positioned outside the obstacle, otherwise set to *False* |
| CONVEX | *True* if initialize a convex object, otherwise set to *False* |
| ***Boid class parameters*** | |
| MASS | boid 'mass' influences how easily a neighbor force affects a boid |
| MAX SPEED | maximum velocity a boid can have |
| MIN SPEED | miminum velocity a boid can have |
| RADIUS VIEW | how many neighbors a boid considers/'sees' |
| MAX FORCE | maximum strength of the neighbor forces applied |
| WANDER RADIUS | how big of a wander 'loop' to perform |
| WANDER DIST | how far to travel with the wandering action |
| WANDER ANGLE | updates the boid's wandering angle |
| *WEIGHT | all parameters with *weight influence how strongly a given force influences the boid's movement |

Table 1: Parameter settings for Flocking

## Mock Presentation

As mentioned in the section before you will have to have short presentation of your work by the end of the first week. Important to note that this presentation is not graded, but is rather a practice round so you can learn from it for the next ones. The presentation should last max 30 min (including showcasing the created simulation), and the structure should be as follows:

| | |
|---|---|
| **Introduction** | Briefly introduce the report: purpose of the simulation, type of analysis being done, questions that will be answered with the simulation. |
| **Methodology** | Present how agents are modeled. How is their interaction implemented? Try to explain the main idea behind the code without showing the actual code (e.g. use finite state machines, formulas, or pseudo-code if everything else fails). |
| **Results** | Present the result, in the form of images and videos of the simulations, trajectories of the agents, and simple plots. It is important to show here how different parameters affect the collective behavior, especially parameters that have a connection on the idea of embodiment and physical interactions (between agents and between agents and the environment). |
| **Conclusion** | Conclude stating what you have learned. How do different choices concerning the embodiment affect the collective behavior/intelligence of the population? |

Table 2: Presentation Sections

## References

[1] C. Reynolds, "Boids, 1986," *Homepage: https://www. red3d. com/cwr/boids*, vol. 2, 2018.

[2] R. Zandie, "Simulating bird flock behavior in python using boids," May 2019.