Projet S1
L3 Magistère Economiste Statisticien
(updated)

Kevin Godin-Dubois

November 23, 2021

Contents

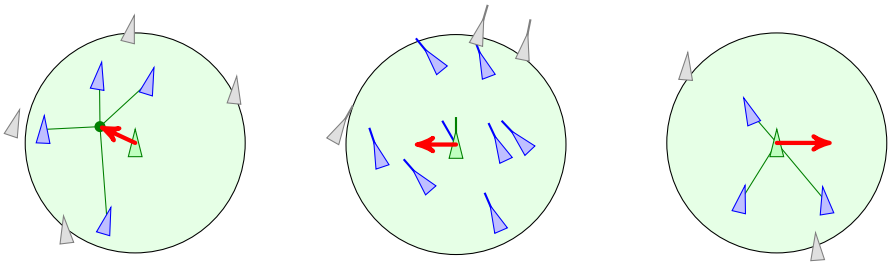
Change log

23/11

Parameters modifiable by GUI and built-in graph capabilities are now considered optional (you can still do them for additional points)

1 Generalities

This project is designed to test your programming skills in the context of a complete complex system simulation. In teams of three, you will plan out every element of your application, from the definition of individual agents to the production of exploitable behavioral data. In the following sections, I start by presenting the context of this experiment before detailing the various deliverable:



(a) Cohesion (b) Alignment (c) Separation

Figure 1: Three simple rules for flocking behavior. Images (reproduced) from `http://www.red3d.com/cwr/boids/`

- the application itself (source code)
- the *individual* summary (pdf)
- the oral presentation (pdf)

The use of english is encouraged, but by no means necessary, and, depending on the quality, will result in a bonus point (see table 1).

2 Boids

An emergent behavior is a seemingly complex high-level behavior (e.g. bird flocking) that arises from the interaction of numerous individuals following a simple rule set. As such the Boids[1], created by Craig Reynolds in 1986, are one of its simplest illustration: only 3 rules are needed to emulate the flocking behavior of potentially very large groups of individuals. The minimal set of rules is illustrated in figure 1.

All three rely solely on local information (the colored area) to derive an appropriate reaction. Moreover, each rule is of an elegant simplicity both in natural and programmation languages:

1a Cohesion Maintain group integrity by moving towards the local center

1b Alignment Attempt to move in the same direction as the group

1c Separation Prevent collision by avoiding excessive proximity

This local information is assimilated to the visual capabilites of an individual boid and is limited both in range r and angle α. Indeed, based on the morphologies of extent birds it would make little sense for a 360° vision. The resulting field of vision is as depicted by figure 2: a partial disk.

[1] https://doi.org/10.1145/37401.37406

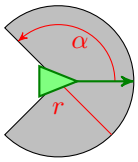


Figure 2: Boids have limited vision both in range and span. (reproduced from same source)

A typical boids simulation thus proceeds along the following lines: create a reasonable number of individuals, have each of them detect neighbors, take appropriate gregarious/evasion measures, modify the positions, repeat. Furthermore, numerous extensions (outside the scope of the project) have been developed to explore more complex scenarios (3D, obstacles ...) and types of interaction (predators, leadership ...).

3 Code specifications

For this assignment, you are expected to develop a complete complex systems simulation, from the ground up. To guide you in this process, multiple key areas are identified in the following sections corresponding to the different levels seen in TP (physical simulation, graphical dressing, parametrization and outputs). The different components will be developed in independent modules with non-overlapping scopes (e.g. boids, physics, gui and main).

3.1 Implementation of the emerging flocking behavior

Each boid will be an instance of a dedicated class with appropriate attributes. Its behavior will only depend on local information as defined by figure 2 and the resulting "decision" will correspond to an acceleration vector, to be aggregated with the boid's velocity and, ultimately, position. Upon creation of a simulation, boids are randomly placed in the environment with a random velocity. Boundary conditions must be implemented preferentially as toroidal (exiting by the left side warps back to the right side).

3.2 Graphical rendering of the simulation

Proper rendering of a simulation allows not only to quickly integrate its dynamics but is also of tremendous help to disseminate potential results. For this project, the use of PyQT5[2] is mandatory. The key graphical element of this experiment is the boid which can be displayed as a simple (oriented) triangle. Additional debug visualizations (e.g. velocities, neighbors, ...) are encouraged if only for properly testing the behavior of your system. Similarly providing

[2] as presented in TP3-4

play/pause and single step functionality is also desired, as it will help in locating and correcting bugs.

3.3 Modifiable global parameters

Although this model seems straightforward at first glance, you will quickly realize that a number of parameters have a crucial impact on the (lack of) emergence of a flocking behavior. Thus you are expected to explicitly implement some control mechanism to modify them. As stated in the notation grid, some groups of parameters are essential (vision and behavioral weights) and require particular attention.

Bonus: You can also design a dynamically configurable simulation where parameters can be changed on-the-fly to better assert their impact. This may be extended to toggling/controlling eventual (custom) debugging tools.

3.4 Plotting relevant data

One of the key characteristics of this kind of complex system is that they are constantly treading the line between structure and chaos. To extract meaningful observations from your application you will implement logging facilities. Every step, you will write to a file a set of essential metrics extracted from your system (e.g. number of boids, neighbors, time ...). Additionally, you will use such extracted data to plot various key dynamics. A sufficient approach is to use an external program to do the actual plotting.

Bonus: Instead, you may use the matplotlib Python library[3] which can also be integrated into a PyQT application for a unified output. Furthermore, you are also encouraged to dynamically update said graph while your application is running.

4 Summary

Following the development of the system, you are expected to produce an *individual* summary of the experience. It shall contain the distribution of roles inside your team and should focus on your personal contribution to the whole project. To better capitalize of the experiment, you will also detail its most salient positive and negative aspects so that you can better tackle future projects. Finally, you will conclude by drawing links between the concepts practiced during this course and your global formation, especially in terms of transversal knowledge.

5 Presentation

This course will be concluded by a 15 minutes presentation of realized work during which you will perform a demonstration of your application and highlight

[3] `https://matplotlib.org/`

how it fits the specifications. This presentation will also include a numerical analysis of a few dynamics of your system. You are expected to investigate the relationship between the number of boids and the evaluation time of a single simulation step. For this, your are required to provide an appropriate regression model allowing the prediction of said time for an arbitrary number of boids. Furthermore, you are strongly encouraged to perform up to two additional analysis on noteworthy dynamics.

The presentation shall be concluded by a clear description of the responsibilities of each team member and an objective analysis of how well the initial objectives were achieved.

Annex: Notation guidelines

Tasks	Subtasks	Weight	Crucial elements
Code	Simulation	5.5	Global update Boid class Decision Motion Distance-based vision Blind side Random init Boundary conditions
	Interface	3	Boid visualisation Main window Update loop Debug drawing Play/pause/step
	Parameters	2.5	Vision (range, angle) Behavior (cohesion, alignment, separation) Miscellaneous (number of boids, base speed, seed) Debug options *Dynamic modification (via GUI)
	Outputs	1	Data logging Graph generation *Built-in graph panel *Dynamic graph *Selectable x/y data
Summary		4	Distribution of roles Individual contribution Positive/negative aspects Link with your formation
Presentation		4	Demonstration Numerical analysis Distribution of roles Objectives analysis
English		1	Documents, comments, code ...

Table 1: Indicative notation guidelines for the evaluation of the project. Items denoted with a star are now considered optional.