STARLING MURMURATION

A Mathematical and Programmable Model for the simulation of Flocking of boids

Vinayak Rastogi (2016CS10345)

Rahul Bansal (2016CS10344)

COP290-Spring(18)

INTRODUCTION

This project is to model and simulate a the fascinating phenomenon of "*Starling Murmuration*". We will computationally simulate the phenomenon by modelling each bird as an independent agent communicating and cooperating with other neighbouring agents. Our objective will be to measure from a realistic simulation the average energy spend by each bird, the angular momentum and the force that each bird has to withstand in a typical flight ritual.

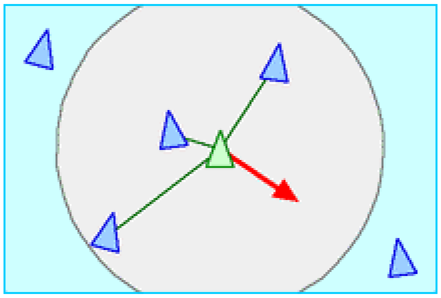
We will study and simulate *Boids*, which is basically an artificial life program, which simulates the *Flocking* behaviour of *Starlings*.

BOIDS

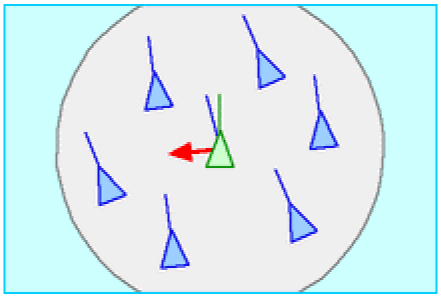
The name "boid" corresponds to a shortened version of "bird-oid object", which refers to a bird-like object.

Boids is an example of emergent behavior; that is, the complexity of Boids arises from the interaction of individual agents (the boids, in this case) adhering to a set of simple rules. The rules applied in the simplest Boids world are as follows:

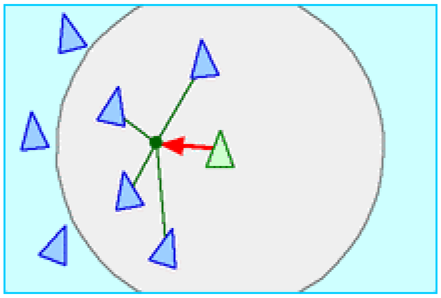
- **Separation**:- steer to avoid crowding local flockmates



- **Alignment**:- steer towards the average heading of local flockmates

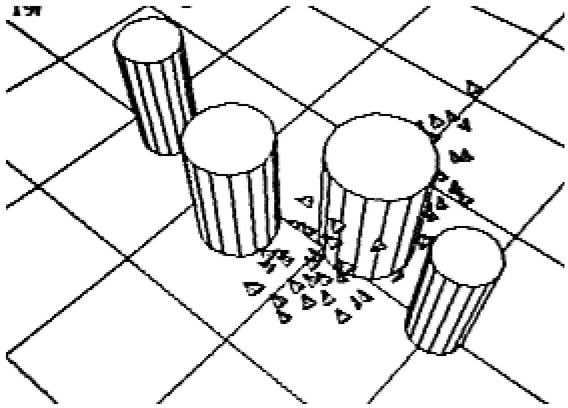


- **Cohesion**:- steer to move toward the average position (center of mass) of local flockmates



Following are some of the more complex and advanced features:

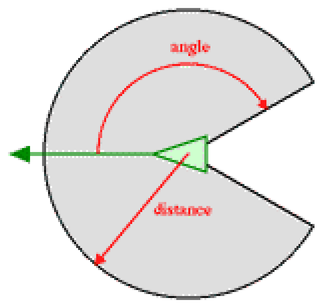
- Obstacle avoidance



(Simulated Boid flock avoiding cylindrical objects)

- Goal Seeking
- Fear and its effects
- Transmission of emotion between animals (through pheromones modelled as particles in a free expansion gas)
- A complementary force to the alignment called the change of leadership (defines the chance of the boid to become a leader and try to escape)

Each boid has direct access to the whole scene's geometric description, but flocking requires that it reacts only to flockmates within a certain small neighborhood around itself. The neighborhood is characterized by a distance (measured from the center of the boid) and an angle, measured from the boid's direction of flight. Flockmates outside this local neighborhood are ignored. The neighborhood could be considered a model of limited perception (as by fish in murky water) but it is probably more correct to think of it as defining the region in which flockmates influence a boids steering.



A Boid's Neighborhood

Movement of Boids can be characterized as either:

1) Chaotic (splitting groups and wild behaviour) or
2) Orderly.

Unexpected behaviours, such as splitting flocks and reuniting after avoiding obstacles, can be considered "*Emergent*"

Applications of Boids:

- Computer graphics (providing realistic-looking representations of flocks of birds and other creatures)
- For direct control and stabilization of teams of simple *Unmanned Ground Vehicles* or *Micro Aerial Vehicles* in Swarm Robotics
- To automatically program Internet multi-channel radio stations
- Visualizing information
- For optimization tasks
- Cellular Automation
- Data Mining
- Multi-Agent Reinforcement Learning
- Screensavers :P

FLOCKING (BEHAVIOUR MODELING)

Flocking behaviour is the behaviour exhibited when a group of birds, called a flock, are foraging or in flight. There are parallels with the shoaling behaviour of fish, the swarming behaviour of insects, and herd behaviour of land animals.

We're talking about Modeling the Behaviour…

From the perspective of the mathematical modeller, "flocking" is the collective motion of a large number of self-propelled entities and is a collective animal behaviour exhibited by many living beings such as birds, fish, bacteria, and insects. It is considered an emergent behaviour arising from simple rules that are followed by individuals and does not involve any central coordination.

With the three simple rules of Separation, Alignment and Cohesion, the flock moves in an extremely realistic way, creating complex motion and interaction that would be extremely hard to create otherwise.

Measurements are done using Computer Analysis of High-Speed Cameras

ALGORITHMIC COMPLEXITY

A basic implementation of a flocking algorithm has complexity $O(n^2)$ - each bird searches through all other birds to find those which fall into its environment

This is because as there is no central control and each bird behaves autonomously, each bird has to decide for itself which flocks to consider as its environment. Usually environment is defined as a circle (2D) or sphere (3D) with a certain radius (representing reach)

IMPROVEMENT:

One possible improvement is $O(nk)$, where we use the “Bin-Lattice Spatial Subdivision Approach”, in which the spatial data structure allows entire area where the flock can move in, to be divided into a large number of bins where each bin stores which birds it contains. This allows the boids to be kept sorted by their location. Each time a bird moves from one bin to another, the lattice has to be updated. Here, k would be the number of surrounding bins to consider, and bin access time as O(1)

THE PSEUDOCODE

Birds collective flocking behaviour is taking into account by separation, cohesion and alignment. These three act as forces to accomplish it.

Firstly we create a function that takes an agent and returns the force on it. This function internally calls the individual functions and saves its values in variables

ali = alignment(bl);

coh = cohesion(bl);

sep = seperation(bl);

Finally, after each frame, the variables scaled by the corresponding factor is added to acceleration vector of a boid.

acc += ali * f1

acc += coz * f2

acc += sep * f3

COHESION works on following guidelines **:**

```
Function {
    Vector AvgPos; Vector force;
    for (Boid b: boids) {
        d = pos – b.pos);
        If  (0<d<=neighborhoodRadius ) {
            posSum . add ( b . pos ) ;
            count++;
        }
    }
    If ( count >0 ) {
            PosSum.div((float)count) ;
    }
    steer = PVector.sub(posSum, pos);
    steer . limit (maxSteerForce );
    return posSum ;
```

ALIGNMENT AND SEPARATION WORK SIMILARLY

KINETIC ENERGY

The Kinetic Energy is ½*m*(v^2)

Also, the system possesses some potential energy due to the 3 essential forces

A boid when moving has a definite velocity at any Instant, which imparts it a definite contribution to systems total kinetic energy. A variable needs to be maintained.

Int energy;

The call of energy method begins by initialising energy to 0. Then through an iterator add contribution of each to energy

```
for (Boid b: boids) {
        energy += 1/2∗m∗v∗v
        }
```

For angular momentum similarly begin with mom declaration Vector mom;

In the procedure,

```
for(Boid b: boids) {
        momentum += massxb.posx(b.vel));
}
```

Lastly call the functions energy() and momentum() in small time intervals.

MOUSE STEERING:

Mouse steering functionality requires tracking mouse position in small time frames. When on call add acceleration in direction of the mouse to our boid.

acc.add((mouseX, mouseY, pos.z));

REVERSE:

Reverses is implemented inside a method which when called iterates and reverses velocity direction of each boid.

```
for (Boid b: boids) {
    boid . reverse ();
}
void reverse (){
    vel .mult(–1);
    acc . mult ( –1);
}
```

PREY:

Getting hold of preys Another force is added, for each boid

```
Preyforce {
    for (prey: preys) {
        d = pos – prey.pos
        If (0<d<=neighborhoodRadius) {
            repulse = pos – b.pos();
            repulse.normalize();
            repulse . div(d);
            posSum.add(repulse );
        }
    }
}
```

CONSTRAINTS

The constraint would be that albeit our simulation runs successfully mimicking the flocking of boids in 3 dimensions, we were not able to calculate the real amount of energy because the velocity in the 3d sim. will be proportional to the real velocity but not equal. Hence the we can get an estimate of how much the energy changes during flocking but we can not get the real amount of energy.

RESULTS

We were successfully able to simulate the phenomenon of starling murmuration in 3D environment with the following features:

1. 3D view
2. Modifiable values of Cohesion, Seperation and Alignment
3. Multithreading
4. Prey/Obstacles
5. Direction reverse
6. Mouse following
7. Display of Kinetic energy and the Momentum and the Total number of boids
8. Renders 1000 boids w/o any hassle
9. A wind functionality
10. Toggle between flaps and normal boid

REFERENCES

1. https://radnorshirebirds.files.wordpress.com/2014/01/starlings-2.jpg
2. https://en.wikipedia.org/wiki/Flocking_(behavior)
3. https://en.wikipedia.org/wiki/Starling
4. https://en.wikipedia.org/wiki/Boids
5. http://www.red3d.com/cwr/boids/
6. http://cp3-origins.dk/a/14187