Crowd Simulation: Boids++

Boids Framework — A Modern take on Craig Reynolds' Boids.

DI ZHAO, University of Califonia, Davis
MUHAMMAD OSAMA, University of Califonia, Davis

Additional Key Words and Phrases: Crowd Simulation, Boids, Computer Animation

1 INTRODUCTION

Craig Reynolds program, Boids[?], illustrated how flocks, herds and schools can be animated as artificial life using three simple rules. The word 'boid' originated from 'bird-oid object', which means 'bird-like object'. The simulation that Craig proposed follow the following three simple rules:

- **separation**: as the name suggest, this simply means to avoid other local objects within the simulation.
- **alignment**: is aligning the direction of the local objects to the average heading of surrounding objects.
- **cohesion**: is movement towards the average position, which in a crowd is often the center of mass of the local objects.

These three rules, very simply and beautifully modeled the flock, herds and schools animation. The contribution of our project is to extend the Boids model, effectively called **Boids++**, to include more complex rules and to provide a framework and abstraction to define other custom rules not accounted for.

2 RELATED WORKS

Craig Reynolds' work has been extended in several ways since 1986. The basic model had rules added on top of that, which mapped things like emotions between objects [?], these emotions could be an attraction factor or fear. An introduction of "change of leadership" was presented by Hartman and Benes [?].

3 PROPOSAL

Using the very intuitive Unity game engine, we intend to design the first three simple rules of Boids simulation for starters. This is work already in progress by Di Zhao. We then would like to write an abstraction that can be called to represent rules for two major systems within the Boids simulation:

- **Local Animation and Rules**: describes the local movement of a boid within its own model. This can easily be mapped, animated through Blender and imported into the Unity scene we intend to design. And can encapsulate a wide range of animations and models, including simple models such as birds, fish, but also some complicated ones, such as four-legged animals like dogs or wolves, simulated as a "pack" within Boids.
- **Global Rules**: these are additional rules we intend to add within our systems. Some of the ones we have considered include the idea of a "first follower" [?] who may have the ability to increase the search radius of a crowd within Boids and "attract" other objects to the crowd, we have also considered a "coup", which depicts splitting of the Boids animation and introducing repulsive force between the splits based on some random factor. And we would also like to write a clean abstraction to add future rules if we find them interesting.

On the technical end, we are interested in looking at fast k-Nearest Neighbor (kNN) searches on the GPU and including that as our shader program to speed-up our computation. The kNN here is used for several different scenarios, including but not limited to finding other Boids to attract or avoid.

4 IMPLEMENTATION

The core idea of boids is put forward by Craig W. Reynolds in 1986 [?]. There is no need to implement a real commander which sends commands to each boid for maintaining the flock. Each individual only need to follow the 3 principles: alignment, cohesion, and separation.

To simulate the aggregate motion, for each boid, we iterate and collect both position and velocity(both speed and direction) information of all other flockmates to calculate average velocity, average distance to other flockmates, and central position of this flock.

The average velocity is used for alignment. Each boid adjusts its heading towards the average heading of the whole flock. It also checks the speed with average, so if the current boid is too fast, slowing down to wait for flockmates catching up, otherwise speed up. The more important part of alignment is heading match process. It directly results whether it's jerky or smooth when the flock makes a turn. To achieve this following equation is implemented:

$v_j = (\frac{1}{n}\sum_{i=1,i\neq j}^{n} v_i) \cdot S + v_j \cdot (1 - S)$

Where v_j is velocity of current boid, n is the number of boids in flock. S is a parameter set to control the boids motion. Its value varies from 0 to 1. A larger S means the changing vector takes more advantage than the current status. The default setting is 0.5. In this case the changing vector is $average_velocity - v_j$.

Flock center position is used for cohesion. A force to the center of the flock is always added to each boids no matter what situation. Note that when flock cross an obstacle, it is possible that the boids are going to divide into 2 or more parts then concentrate together again after then. In this case there is no need to consider the center of each cluster when flock is temporarily divided. Otherwise the process may look like one flock initiated divide into two or more instead of an obstacle come across a single flock. The cohesion equation is : $v_j = ((\frac{1}{n}\sum_{i=1,i\neq j}^{n} pos_i) - pos_j) \cdot S + v_j \cdot (1 - S)$

Pos is the position of a boid. The whole idea is very similar to alignment part so no further detailed description here.

Average boid distance is used for separation. The equation is like this:

Authors' addresses: Di Zhao, University of Califonia, Davis, devzhao@ucdavis.edu; Muhammad Osama, University of Califonia, Davis, mosama@ucdavis.edu.

$$v_j = (\frac{1}{n}\sum_{i=1,i\neq j}^{n}\frac{v_i}{dist(pos_i,pos_j)})\cdot S + v_j\cdot(1-S)$$

$dist(pos_i, pos_j)$ is distance vector from boid[i] to boid[j]. Here both position and velocity of other flockmates are considered. The further boid[i] away from boid[j] the less influence will be added for separation. In our script only the boids with close enough distance will implement separation force in order to avoid unnecessary calculation. If only the position of two boids is considered without velocity, the flock could not be separated out well in some situations (e.g. When boids are too close to each other at the beginning).

The S parameter from 3 equations have the same function but does not necessary to be the same. This script is attached to each individual without further control of all boids following strictly with the condition of the real nature world.

5 OPTIMIZATION

Based on the 3 rules we further implement two other optimization approaches of our own.

First, iterating all flockmates for each boid per frame is too expensive and may cause tones of lagging or even crashing. We took Sebastian Lague's project as reference and implemented a GPU based multi-thread iteration for calculating the 3 key parameters mentioned above. This time each boid has one thread on gpu and iterates simultaneously.

We also add a random force to each boid every 20 frames. In some situation(e.g. When the flock move straight forward for a time), the boids situation keeps no change and make the whole flock look too ordered and becomes even sterile. So an uneven force with random direction is added to make more change to the flock motion. This force is not designed to completely break the flock order and reform again. It's order to downgrade the precision of computer calculation and make the motion more realistic.

5.1 Avoid Obstacle

The interaction with obstacle object is the most interesting part of the whole simulation. It makes the flock motion more random and unpredictable when it comes across an object with complex shape. Sometimes the boids could be divided into several smaller groups but the whole motion looks still smooth and you could feel that the small clusters still forming one single flock.

The strategy to achieve this is complicated. Craig W. Reynolds mentioned two different attempts: force field model and steer-to-avoid, and we implemented the latter one. We designed several rays sending from the front of the boids to detect if there are obstacles ahead. But directly detecting the collision of rays on the surface of object could be very expensive if the object is polehydural. We simplify this question by casting the obstacle into a sphere. In this case for each ray sent from boid, the obstacle could be casted into a circle on a 2D panel with this ray. With this cast we solve 2 problems together, the complexity of directions in 3D and arbitrary shape of obstacles.

When an obstacle is detected, a normalized force will be added to the boid on the same panel of the detection ray. This force is designed to turn the heading of the current boid and the angle it turns will be based on golden ratio. In some simplified cases, for example if a boid heading perpendicular to a flat surface, the boid will steer a golden spiral to avoid collision. This is designed to avoid shape turns and make the whole process smooth and natural. Since the flock is still maintaining during the avoid process, the avoid force is added on the 3 rules and it is not necessary to worry the force from the 3 rules may affect the performance of avoiding.

5.2 Prey and Predator

To test the durability of our system, we introduced a prey and predator into our system to simulate a real natural scene.

The prey randomly pick a point in the scene and move toward that point. To make the motion of prey smooth and real, we implement a force from the position of prey to the target point on prey each frame. When the prey reaches the target, another point will be randomly picked. The flock chases the prey by a similar way. A force heading the prey is added on boid each frame together with 3 rules.

We also added a predator to chase the flock. The predator is always heading the center of flock and the strategy is exactly the same as boid chasing prey. Each boid get an updated predator position each frame. If the distance of predator is less than an avoid radius an escape force will be added toward an opposite direction of predator on boid. The difference between avoid predator and obstacles is the boids do not follow 3 rules when escaping predators. We notice that 3 rules should not be implemented to simulate this process like what it usually happen in nature.

6 CONCLUSIONS

Nullam vulputate enim ut tortor mollis pharetra. Cras pellentesque sem a accumsan malesuada. Donec at massa nisl. Sed malesuada felis id nisl maximus efficitur. In pretium metus non faucibus pulvinar. Sed pulvinar elit ultrices mauris vehicula, id ultricies purus finibus. Fusce tempus elit molestie, consequat ipsum eget, iaculis nibh. Cras tincidunt, orci in lacinia tempus, mauris leo finibus orci, vitae dignissim dui risus et odio. Sed commodo ultricies nulla, et varius velit aliquam quis. Sed efficitur, ex non facilisis dignissim, lacus orci accumsan massa, dictum facilisis arcu lacus ac leo. Sed quis tellus dictum massa egestas dapibus vel et justo. Nulla euismod lectus ut purus hendrerit porttitor. Suspendisse quis dui ligula. Proin non porta libero. Maecenas vel feugiat urna.