Computing for Animation

Design and Research

Flocking system
Part 1

Ioanna Ivanova

Computer Visualization and Animation
Bournemouth University

Project introduction:

What is flocking system?

Flocking system – The original flocking algorithm was developed by Craig Reynolds in 1986. As Reynold says 'It was based on three dimensional computational geometry of the sort normally used in computer animation or computer aided design'. This whole system definitely has some really good real-world applications.

Applications:

In computer animation, filmmaking and VFX, simulations and optimizations.

- 1987 the first novel using algorithmically-generated flocking and schooling behavior was "Stanley and Stella in Breaking the Ice". (1987)
- Batman Returns (1992) - bat swarms which were procedurally generated using algorithms similar to these.

Approach:

Based on the rules of Craig Reynolds presented in 'Flocks, herds and schools: A distributed behavioral model' flocking system is created.

The simulated flock can be seen as a particle system, with simulated 'birds' being the particles, and the motion of which is created by distributed behavioral model – as in a real flock.

Birds – able to choose their own path, following some simple rules.
They navigate in this application following the three behaviors that describe how an individual agent maneuvers based on the positions and velocities its nearby flockmates.

- Separation – collision avoidance
- Alignment – velocity matching /attempt to match velocity of the nearby flockmates /

- cohesion – flock centering /stay close to nearby flockmates/

How it will work?

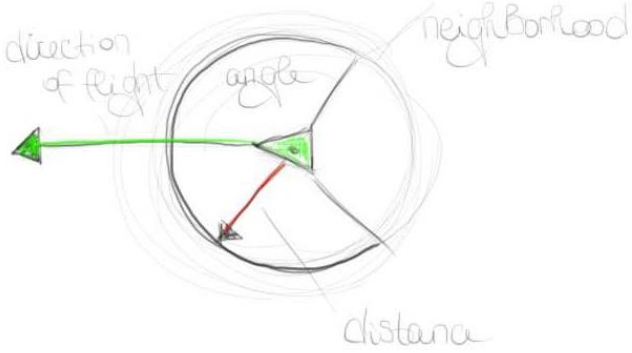
Each of the 'birds – boids' moves in the box, while is ruled by the rules given above.
Each of the boids has its own location velocity, which changes for each update (frame) – a new velocity is calculating using the flocking algorithm. Location and velocity – represented as vector objects.
This algorithm uses:

- The boid's current velocity
- Its 'neighbours' velocities
- The position relative to its neighbours

Different components of the implementation

Neighbourhood:

Each boid react to two or three of the closest boids /neighbors. The neighborhood is characterized by: Distance (measured from the center of the boid) and Angle, measured from the boid's direction of flight (Reynolds, 1986). The neighborhood is the area where the flockmates influence boids navigation.

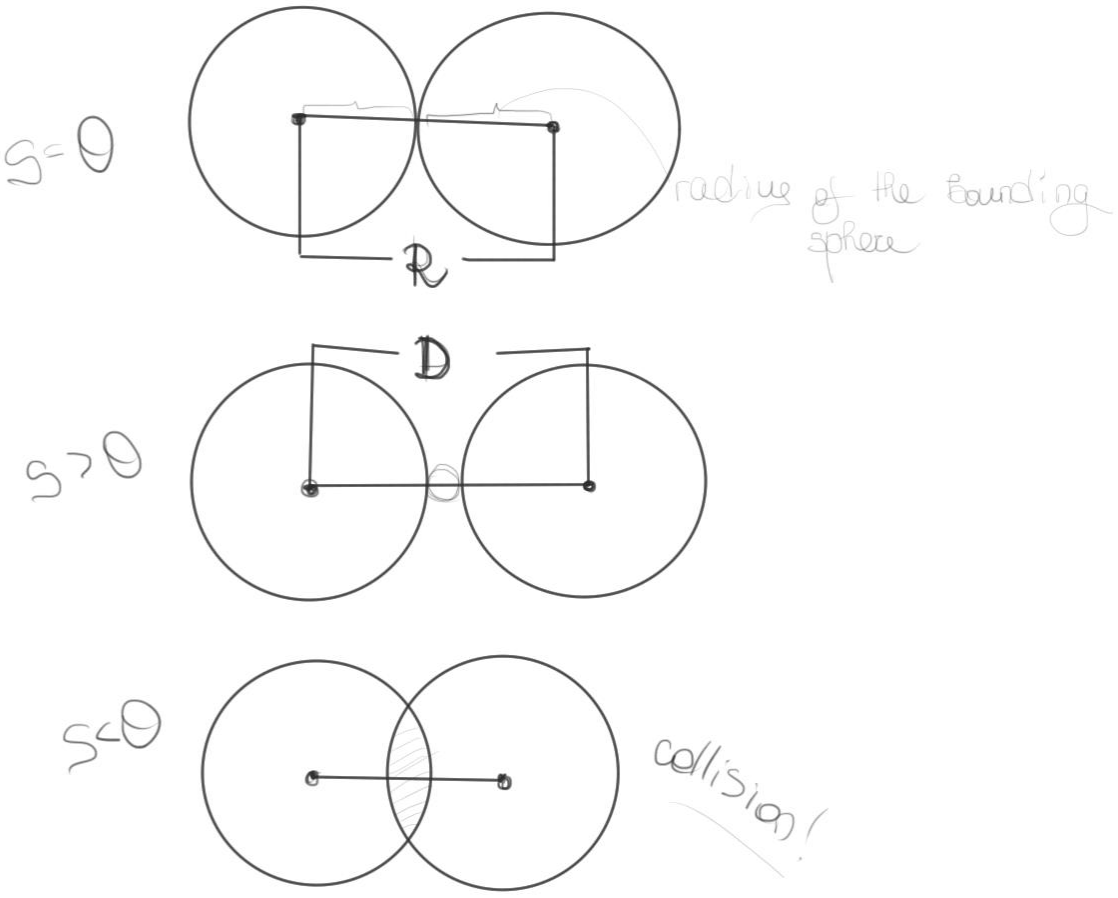


Goal:

Each of the boids in the flock has the same goal.for each update (frame) the position of the leader is updated and re-writen. Every other boid creates a goal vector based on the information of the leader's position. It is done by subtracting the leader p1osition from the boids positions.

Separation/Collision Avoidance with the agents:

Collision must be avoided, in order to simulate a natural look of a flock.
Every boid will have a bounding sphere.
In order to avoid collision, first we find the distance (vector 'S') between the two centers of the bounding sphere. The value of this vector is calculated by subtracting the second bounding box radius from the first.
If the value of S > 0, then there is no collision.
If the value of S = 0, then the collision occurs.
If the value of S < 0, then there is intersection.



In case of collision a collision response (algorithm, that still I am not aware of) is used to keep away the boid – the distance between the two centres of the boinds. A loop should be used to check constantly if there is s flockmate that is too close to the boid.
Calculating the velocity to see if boids are moving toward each other and if there will be a collision between them, it can be prevented.

Cohesion:

A flock is staying close to each other – all the flock is staying together. This is because of the cohesion part of the algorithm. Every update (frame), each boid will look at the position of each other boid to see if it is in a neighbour radius. The position of the boids in the neighborhood is average, so every boids try to move to that position. The

locations of the boids that are inside of a neighbourhood are summed up in order to find the center of the flock.

Alignment – velocity matching:

Each boid in a flock tries to reach the same direction as the others. Each boid checks the heading of its neighbours and tries to match their heading. The velocity of each boid in the neighbourhood is average to find the average heading of the central boid. Their velocitiy can reach maximum and minimum, and once they are reached, the velocity is adjusted to the average movement?
The velocity of every boid change when it gets in the flock.

Objects in the environment?

Avoiding the obstacles that are put in the environment. This is really needed if we want to achieve a believable flocking system.
The steer-to-avoid is one of the two types of shapes of environmental collision avoidance, which according to Reynolds paper, is more robust and seems closer in spirit to the natural mechanism.
The boids consider every obstacle in front of it. It finds the silhouette edge of the object closest to the point of impact and calculate its positin. If there is a chance of a collision path, then a radial vector is computed so any intersections can be prevented. If there is a collision the boid must calculate a new path to avoid the collision.

Predator:

Predator – begins at a random position with a random vector. The flock class calculate its locations and a direction is applied to it.
I still have to figure out how to involve it.

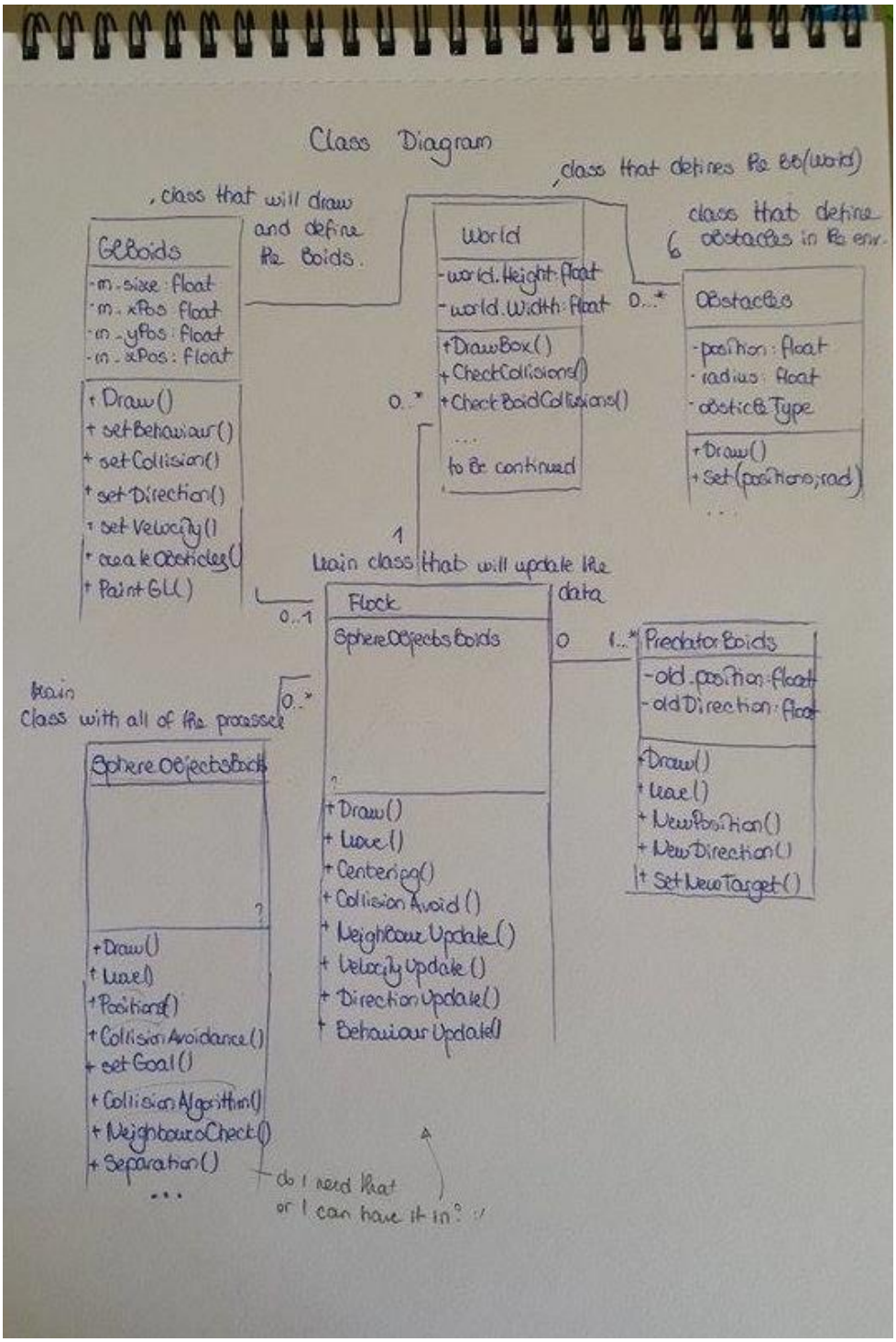
Leador’s position:

A bounding box for the leader boid should be specialized. The moment it reaches a heigh and width maximum it changes its directions.

Bounding box/ World:

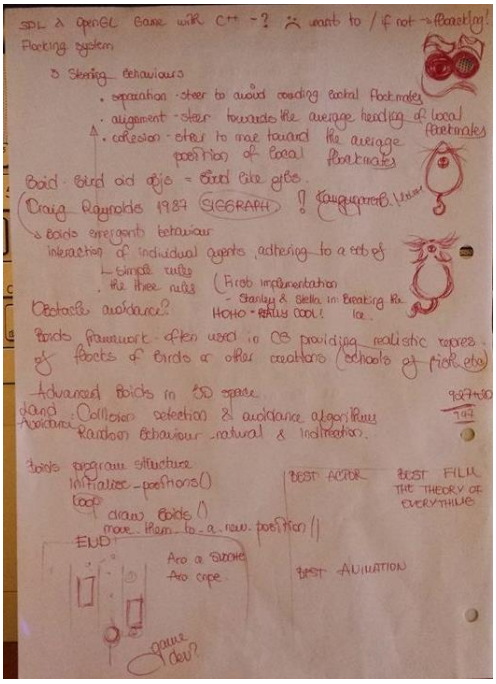
Creating a world – bounding box in OpenGl, whit specified coordinates.

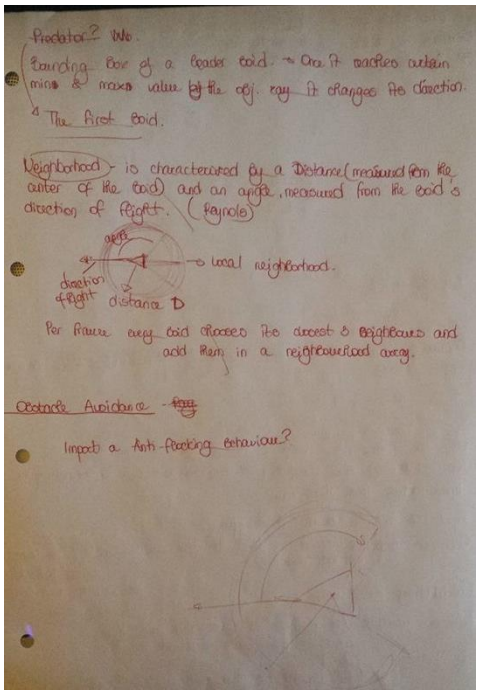
Class Diagram (Not absolutely specified yet!)

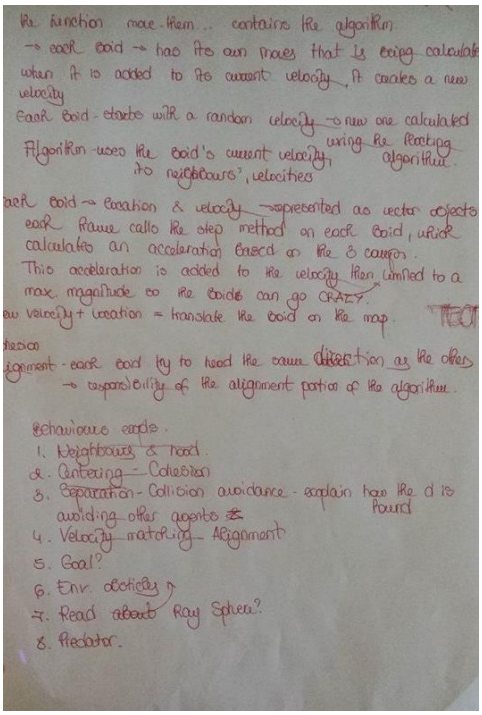


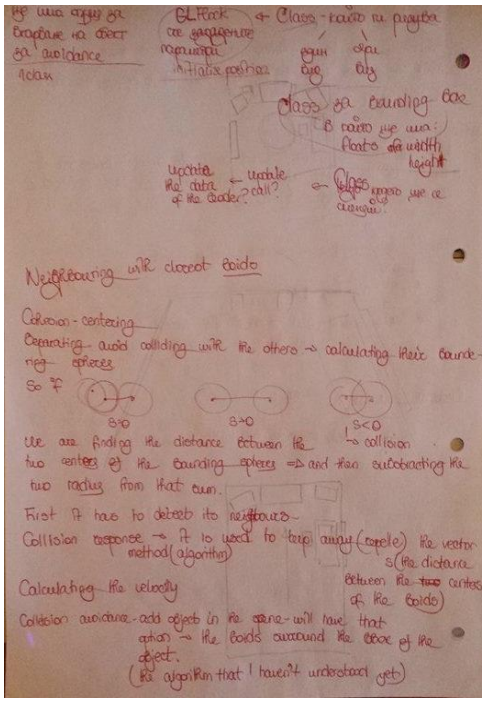
*The diagram will change in the process of working, understanding and developing the program (system).

Beginning of the research:









References

Article:

Craig W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*, ACM SIGGRAPH Computer raphics, v.21 n.4, p.25-34, July 1987

Websites:

Aherne, S. (2015). *Fleck: An OpenGL Flocking System*. [online] ResearchGate. Available at: http://www.researchgate.net/publication/237545092_Fleck_An_OpenGL_Flocking_System [Accessed 19 Jan. 2015].

Brundage, H. (2011). *Neat Algorithms - Flocking - Will You Harry Me*. [online] Harry.me. Available at: http://harry.me/blog/2011/02/17/neat-algorithms-flocking/ [Accessed 19 Jan. 2015].

Cs.toronto.edu, (2015). *Craig Reynolds: Flocks, Herds, and Schools: A Distributed Behavioral Model*. [online] Available at: http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/ [Accessed 19 Jan. 2015].

Haptic-data.com, (2015). *Shiffman Flocking - Toxiclibs.js*. [online] Available at: http://haptic-data.com/toxiclibsjs/examples/shiffman-flocking-pjs [Accessed 18 Jan. 2015].

Kfish.org, (2015). *Boids Pseudocode*. [online] Available at: http://www.kfish.org/boids/pseudocode.html [Accessed 17 Jan. 2015].

Macey, J. and profile, V. (2013). *Jon's Teapot: CA 1 Initial Design General Feedback*. [online] Jonmacey.blogspot.co.uk. Available at: http://jonmacey.blogspot.co.uk/2013/01/ca-1-initial-design-general-feedback.html [Accessed 19 Jan. 2015].

Navgen.com, (2015). *3D Boids Simulation | About*. [online] Available at: http://www.navgen.com/3d_boids/index.html [Accessed 19 Jan. 2015].

Nccastaff.bournemouth.ac.uk, (2015). *NCCA Coding Standard*. [online] Available at: http://nccastaff.bournemouth.ac.uk/jmacey/NCCACodingStandard/ExampleClassHeader.html [Accessed 19 Jan. 2015].

Red3d.com, (2015). *Boids (Flocks, Herds, and Schools: a Distributed BehavioralModel)*. [online] Available at: http://www.red3d.com/cwr/boids/ [Accessed 19 Jan. 2015].

Red3d.com, (2015). *Not Bumping Into Things*. [online] Available at: http://www.red3d.com/cwr/nobump/nobump.html [Accessed 19 Jan. 2015].

Wikipedia, (2015). *Flocking (behavior)*. [online] Available at: http://en.wikipedia.org/wiki/Flocking_(behavior) [Accessed 19 Jan. 2015].