

Project 1: Flocking and Avoidance with Boids

Per Magnus Veierland
permve@stud.ntnu.no

February 4, 2016

PROGRAM IMPLEMENTATION

The project has been implemented using the *Python* language with the *Qt* Graphical User Interface (GUI) framework through the *PyQt* bindings library.

With a naive implementation, the update function of a boid program would have an $O(N^2)$ time complexity which significantly limits the scalability of the program. For the average case this can be improved by using a spatial data structure to store boids. The implemented structure consists of a 2D-grid of square cells, where each cell has a width and height equal to the boid neighbor radius. This allows the neighbors of a boid to be found by examining only four cells; the cell which the boid is currently in, plus the three cells cornering the quadrant of the current cell which is occupied by the boid in question. This approach scales well as long as not too many boids congest within the same neighborhood.

After beginning an implementation using a class hierarchy it was found that the memory layout of the boid values could be made more efficient by using plain arrays to represent each boid. Each boid, both prey and predators, is represented as an array of 17 floats – and each obstacle is represented by an array of 3 floats. Due to overheads experienced with using *numpy* data structures, plain trigonometric math was implemented directly.

FORCES

The separation force (equation 1) is intended to prevent boids from establishing singular flocks through forcing them apart. For a given boid b , the separation force is calculated as the average normalized difference between the position of the boid, \mathbf{p}_b , and the position of each neighbor, \mathbf{p}_n .

$$\mathbf{F}_{separation}(b) = \frac{1}{|N_b|} \sum_{n \in N_b} \frac{\mathbf{p}_b - \mathbf{p}_n}{|\mathbf{p}_b - \mathbf{p}_n|} \quad (1)$$

The alignment force (equation 2) is intended to align the heading and speed of a boid with its neighbors. For a given boid, b , the alignment force is calculated as the average velocity vector, \mathbf{v} , for the boid's neighborhood N_b .

$$\mathbf{F}_{alignment}(b) = \frac{1}{|N_b|} \sum_{n \in N_b} \mathbf{v}_n \quad (2)$$

The cohesion force (equation 3) is intended to force boids towards their neighbors. For a given boid, b , the cohesion force is calculated as the average position of its neighbors, N_b , minus its position; \mathbf{p}_b .

$$\mathbf{F}_{cohesion}(b) = \left[\frac{1}{|N_b|} \sum_{n \in N_b} \mathbf{p}_n \right] - \mathbf{p}_b \quad (3)$$

The obstacle avoidance force is intended to keep boids from colliding with objects. Its calculation can be described in three steps, see equation 4-6. First the difference of the position of the obstacle closest and in front of the boid, \mathbf{p}_o , and the position of the boid, \mathbf{p}_b , is projected in the velocity direction of the boid to create the vector ξ_b (equation 4). If the magnitude of ξ_b

is positive, then the obstacle is in front of the boid. Further, δ_b is calculated as the difference between the projected vector ξ_b and the center of the obstacle (equation 5). If the magnitude of δ_b is less than the radius of the obstacle, R_o , plus the radius of the boid, R_b , then a corrective force, $F_{object}(b)$, is calculated (equation 6).

$$\xi_b = (\mathbf{p}_o - \mathbf{p}_b) \cdot \frac{\mathbf{v}_b}{|\mathbf{v}_b|} \quad (4)$$

$$\delta_b = \xi_b - (\mathbf{p}_o - \mathbf{p}_b) \quad (5)$$

$$\mathbf{F}_{object}(b) = \frac{\delta_b}{|\delta_b|} \cdot (R_o + R_b - |\delta_b|) \quad (6)$$

The predator avoidance force is used by prey boids to avoid predator boids (equation 7). It is calculated as the sum of the normalized differences between the position of the boid, \mathbf{p}_b , and the position of each predator, \mathbf{p}_p , which are within a given predator distance of the boid; denoted as the set P_b .

$$\mathbf{F}_{predator}(b) = \sum_{p \in P_b} \frac{\mathbf{p}_b - \mathbf{p}_p}{|\mathbf{p}_b - \mathbf{p}_p|} \quad (7)$$

EMERGENT BEHAVIOR

1. **Low separation, low alignment, high cohesion:** Existing flocks of boids gather into tightly formed packs. Flocks keep merging until there is only a single tight flock consisting of every boid.
2. **Low separation, high alignment, low cohesion:** Flocks grow into large structures as the boids all agree on their direction. Over time all flocks assemble into a single monolithic structure.
3. **High separation, low alignment, low cohesion:** Boids spread out across the world and keep away from each other; resulting in evenly distributed boids and no flocking. Boid behavior is erratic as they continuously attempt to keep separated.
4. **Low separation, high alignment, high cohesion:** Boids gather into tight packs with a shared heading. Due to the increased alignment weight, the resulting single tight flock gathers quicker and is better maintained compared to scenario #1.
5. **High separation, low alignment, high cohesion:** With large and equal separation and cohesion weights, erratic normal-sized flocks emerge with boids rapidly switching between gathering and separation within their flocks.
6. **High separation, high alignment, low cohesion:** Using large weights for alignment and separation, broadly spaced large fields of boids moving in a single direction can be created. No distinct flocks occur, but there is a clear global pattern of uniformly moving and evenly spaced boids.