INB383 AI for Games
Assignment 2 - Predator vs Prey

Prepared for: Dr Rune Rasmussen

Prepared by: Thanat Chokwijitkul n9234900

Queensland University of Technology (QUT)

3 June 2016

STATEMENT OF COMPLETENESS

This statement is to confirm that the assignment 2, *Predator versus Prey*, has been completed with all the components required according to the assignment specification and additional components to enhance the simulation. The simulator comprises of the following features:

- **Fifteen fish** represented by small spheroids
    - These fish boids are physical objects.
    - These fish objects have a buoyancy so that they can sit suspended in space.
    - The maximum swimming radius of each fish is 20 meters.
    - These fish boids have food value used by sharks in satisfying their hunger. In this case, three fish getting eaten by a shark will put that shark in its *full* state for a period of time (5 seconds).
    - The simulator utilises Separation, Alignment and Cohesion forces for steering so that all the fish swim in school. However, fish will be swimming in the schooling fashion when they are not being threatened by sharks.
    - When a fish gets eaten, it will lose its boid property and turn semitransparent. At this state, that dead (inactive) fish cannot be detected by any shark.
    - A dead (inactive) fish becomes alive and turns opaque after a period of time (10 seconds).
    - All reliving fish recover their living boid state instantly after becoming alive again.
    - Fish will flee when they are threatened by sharks.
    - The normal colour of fish is sky blue (the state of being safe and calm). Fish will turn yellow (the state of being frightened) when a shark gets near and become semitransparent after getting eaten.
    - Fish move with lower speed compared with sharks. However, fish have better turning speed when fleeing from sharks.

- **Five sharks** represented by larger spheroids
    - These shark boids are physical objects.
    - These shark objects have a buoyancy so that they can sit suspended in space.
    - The maximum swimming radius of each shark is also 20 meters.
    - Each shark has its maximum appetite value of 3. Therefore, if a shark eats three fish, it will be in the *full* state for a period of time (5 seconds).
    - The simulator utilises Separation, Alignment and Cohesion forces for steering so that all the sharks swim in school. However, the shark school is less cohesive than the fish school.
    - Each shark object utilises the Blumberg's top-up and decay model in determining its appetite value.
    - Sharks hunts and attacks fish.
    - The normal colour of sharks is shark-skin grey (the state of being full and calm). Sharks will turn red (the state of being hungry and aggressive) when they are hungry.
    - Sharks move with higher speed compared with fish.
    - Sharks will always chase after the nearest fish. If another fish gets closer and the currently targeted fish becomes further away. That fish will become a new target.

ISSUES ENCOUNTERED

Object Collisions

When a shark gets near its targeted fish, it will keep chasing indefinitely since the collision between the shark and the fish objects cannot occur. Even though the OnCollisionEnter method has been properly implemented, but the problem is that collision detection does not work with the fish objects since they are not triggers. The issue can be solved by turning all the fish into triggers.

Object Speeds

When a shark is trying to hunt its targeted fish, it will keep chasing that fish forever. This is because the shark and its target have the same movement speeds. In addition to that, fish also have better turning speed, which makes them more difficult to be caught. However, this issue can be fixed by increasing the speed of the shark object (this can be done by adding more force to the object's rigidbody).

The Closest Object

In this case, a shark cannot decide which fish is the closest object to attack. This causes the movement of that shark to be terminated. This issue can be fixed by writing a method that iterates through the list of fish boids and computes the distance between the shark and each fish in order to determine the closest fish's distance and its exact position.

Swarm System Formation

When forming a school of fish, sometimes it is difficult to guarantee that fish will swim in an obvious cohesive fashion. This is because the swarm systems are sensitive to initial conditions. Therefore, this issue can be moderated by ensuring that the Separation, Alignment and Cohesion forces are properly configured as well as all the values in the deployer class.

Swarm System Limitations

Apart from the issues mentioned above, there are other apparent limitations of the swarm system. In the case of fish schooling, it is obvious that the school of fish is not fully under control or even uncontrollable sometimes. The swarm system cannot be controlled from the outside. It can only be slightly adjusted or tweaked from the inside, which makes it unpredictable and difficult to control. However, in order to form a school of fish, the only object that needs to be implemented is the fish boid, associated with object customisation in the game engine and a script that determines its behaviours. It is necessary to ensure that these simple behavioural rules at the individual level are well-designed since they are practically adequate to emerge a sophisticated group behaviour.