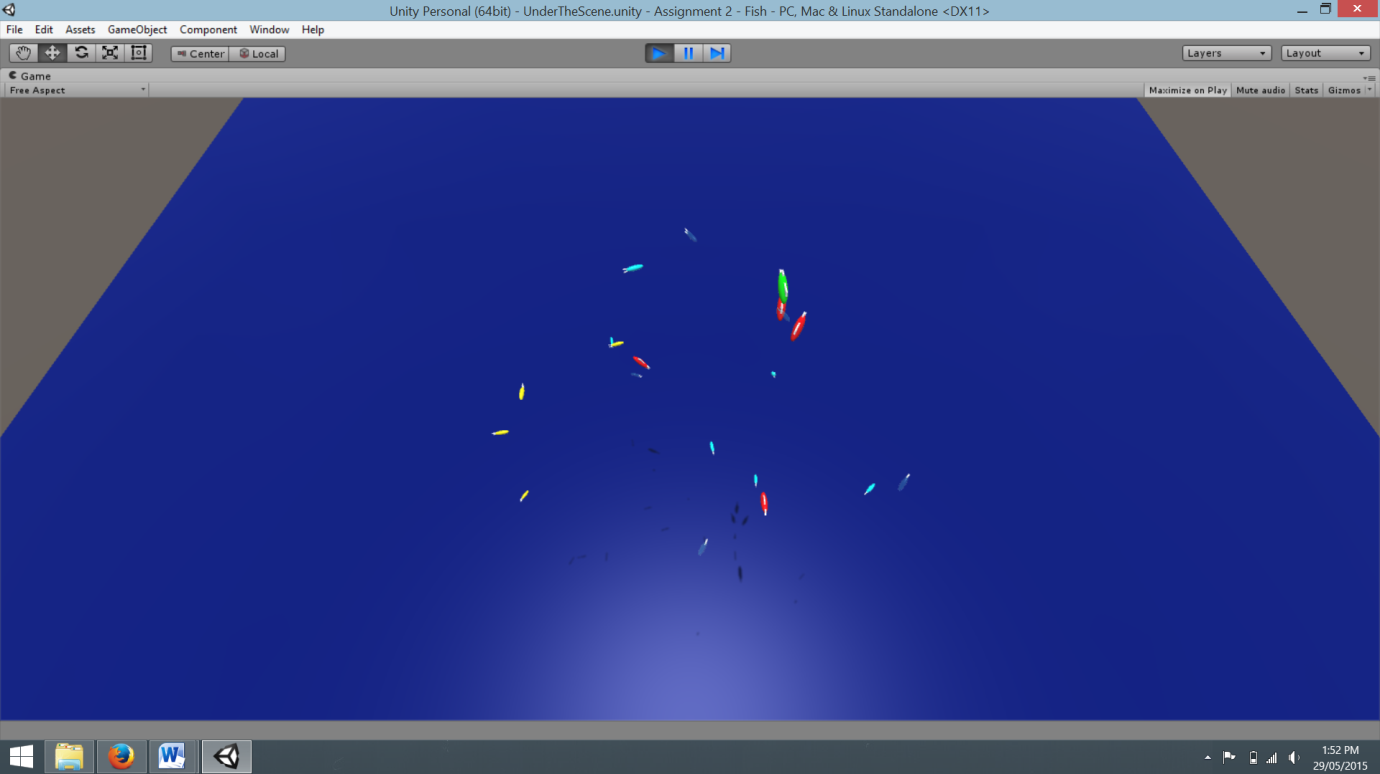
AI for Games – Assignment 2 Fish and Sharks – Statement of Completeness



Unthreatened Fish

Scared Fish

Dead Fish (Transparent)

Hungry Shark

Full Shark

It is my belief that this assignment is complete to its fullest extent outlined in the Assignment CRA with no errors.

The deployer spawns 15 fish which:

* are physical objects
* Have buoyancy
* Have a maximum swim radius of 20 meters
* Have a food value used by sharks (this value is handled in the shark script)
* Use separation, alignment and cohesion forces for steering so as to swim in schools (when not being threatened by nearby sharks and near other fish)
* Lose their boid properties, become transparent and die on collision with a hungry shark.
* Dead fish become living opaque fish with all previous boid properties 10 seconds after dying
* Living fish will flee from sharks using the formula outlined in the lecture

Fish Added Features/Additional Info:

* Fish will turn cyan (light blue) when in the fleeing state and yellow when not fleeing (I considered making colour changes for each state but this looked very cluttered).
* Fish will spawn in a random xyz position within the simulation radius
* Fish will evade the closest shark at any one frame, changing which to flee from (if any are within the fish’s’ comfort radius) whenever another is closer. When two sharks are the same distance from the fish it will evade the one it was already fleeing from (this means if multiple sharks prey on the same fish from two very different directions they will sometimes have a hard time evading them)
* The advantage fish have over sharks in this simulation is that they have better turning speeds due to the way the formula works

The deployer spawns 5 sharks which:

* Are physical objects
* Have buoyancy
* Have a maximum swim radius of 20 meters
* Have a maximum appetite value (will stop eating after 3 fish)
* Use separation, alignment and cohesion forces for steering so as to swim in schools (when not hunting fish)
* Each shark uses Blumberg’s top-up and decay model. Sharks start at empty, every 10 seconds a sharks appetite increases enough that it needs to eat an additional fish. If the shark eats fast enough to accrue 3 points of “hunger” it is no longer hungry for 5 seconds or until the next time hunger decays
* Sharks will hunt and eat fish

Sharks Additional Features/Additional Info:

* If a shark eats fast enough to acquire 3 hunger in less than or equal to 5 seconds of its previous hungry decay, its hunger will decay faster for the next decay time. This means sharks become hungry faster when they’re full (I did this so that successful sharks wouldn’t just keep topping up on 1 fish every 10 seconds, this way they are fairly active even after the first full feed).
* Sharks will turn red when hunting a fish and green when full
* A full shark will find it easier to cohere with other sharks as they aren’t being separated by forces applied to hunt fish (sharks will still cohere reasonable when going for the same fish)
* To compensate for the fish having better turning speed than the sharks, sharks move with almost 3 times as much force (translating to higher speeds)
* Sharks will always attack the closest living fish decided every frame; if they are chasing a fish which becomes further away than another fish, they will change targets
* Sharks will spawn in an area

Difficulties Encountered

* Sharks would not get near fish
  + Turned out the simulation centres for sharks and fish were not properly aligned; leaving areas where one could be but not the other, fixed by making simulation area the same
* Sharks would not affect fish at all on collision
  + Collisions weren’t working because I hadn’t set either object to be a trigger, fixed by making the fish triggers
* Sharks would not correctly estimate fish predicted path
  + Sharks were trying to chase all fish simultaneously, this was identified by debugging by associating behaviours to material colours; rapid changes in colour indicated that the sharks were looking at and acting on all fish very quickly, fixed by changing method style to calculate closest fish and only chase that one
* All fish would revive simultaneously
  + Fish revival was being done is waves, a revival happening whenever the number of frames since start could be divided into 300 with no remainder, fixed by storing individual times of death and waiting for current time (Time.time) to become at least 10 seconds greater than this value.
* Sharks would become hungry simultaneously
  + Same problem and solution as fish revival. Although because all sharks hunger decay starts at the same time, if more than one shark has a full stomach at relatively close timings and their next hunger decay is less than 5 seconds away, they will become hungry simultaneously (this is intentional; “in real life” two identical sharks born at the same time would become hungry at the same time assuming same max appetite and hunger decay rate)