**This lab is worth 5% of your final AI4Games grade.**

**Due date October 6th 5pm**

**Lab 4**

On blackboard you will find a VS project called Flock.

The project implements a flocking algorithm as covered in the lecture notes.

Download and review this project to become familiar with the main components:

* Game
* Flock
* Boid

When you run the program, it draws 100 characters (triangles) on the screen and starts the flocking algorithm.

To do:

1. Modify the code so that only 10 characters are produced and see how the flock moves.

**Answer**: At the very beginning it might split into 2 or 3 groups or sometimes one of the boid is by itself, when it’s in a Flock they go slightly faster than if the “Boid” is by itself.

1. Modify the code so that 150, 200, 250 characters are produced and see how the program reacts.

**Answer**: doesn’t matter how many of them are 150, 200 or 250. They all split into groups and after a few seconds that group splits into group(s) or sometimes 2 groups merge into 1.

1. Pressing the right-hand mouse introduces a predator. Observe how the flock respond. Check to see how this is implemented in the code.  
   **Answer**: It seems that the “Boids” try to get away from the predator by going into a different direction.
2. Can you improve the code so that the flocking runs more fluidly at higher numbers of boids?  
   **Answer**:   
   1) There a few ways of doing this, like using the grid and check if they are inside the grid specifically or make a radius/bounding box and check if the boids are inside.  
   2) Or make the distance less (neighbourDistance = 20 instead of 50).  
   3) Add a clock something like if (clock.getElapsedTime().asSeconds() > 0.05) inside Boid.cpp < void Boid::flock(vector<Boid>& v) function.
3. I have written the shell of a swarm() method as part of the boid class, called via the flock.swarming() method. When the user presses the Space Bar the flock behaviour toggles between **flocking** and **swarming**. Fill in the rest of the boid.swarm() method to implement the Lenard-Jones potential function.
4. Show me when it is done.

