CS22510 Assignment Report

# Overview

I tried very hard during this assignment. Whilst it may have been entertaining (as learning a new language always is), I have spent longer than any other attempting to do what I wanted. The final product may not look like much, however the code included includes methods and algorithms that, when used properly, are efficient and intelligent.   
The main point I’d like to bring up here is the problem (and believe me, I’ve been working for days to fix this one problem and still I cannot find a solution);

The program freezes after a couple of loops round the program. The program reproduces the animals, it makes the animals fight and die (if that’s what the result of the chance was) and it makes them move around the board. However after this it just stops working- no error, no warning, nothing. The program sometimes doesn’t even end- it just stays running and doing nothing. The keyword there is “sometimes”; occasionally the program will do another loop and freeze, but other times it will do only one loop and end. This issue arose after the implementation of the aphid and ladybug fighting.

As you will see from the screencast, if I were to remove the fighting capability (and, in turn, the reproduction ability as otherwise there will be no balance and there will be too many of each animal being created, leading to a large demand in memory) the program would be stripped to the moving around the grid. This algorithm works perfectly.

The program creates, moves, kills and reproduces the animals as it should, the only issue is a bug on the display afterwards. I understand this is a serious flaw, however if the code behind it can just be seen then maybe the reader can understand that I didn’t rush this and that I’ve spent a large amount of time on it.

# Reasoning behind program design

The program consists of an object of animal that could be either an aphid or a ladybug. The design that I have created allows for it to be very easy to implement new aspects. For example, I have created an object of “Configuration”. This means that during the program, if it was to be implemented, the configuration details for the animal or program in general can be changed very easily and it is all highly customisable. I also chose the path of creating cell objects that would hold vectors of its contents. This makes it very easy to give every single value that it holds extra details and data; where other people would just simply have a string representing data type or something.

A mathematics object was also created as I found that calculations could be made in an object and retrieved much easier separately, in case of use somewhere else in the program.

A few places I also used a ‘pair’ as a return type. I found this a clever approach because I needed to return an object and details about whether that object matched criteria, and returning two values from a function has not been a possible ability for me until now.