2017 Paper 1 Prep Pack

# Analysis

List the current classes and state location where they are instantiated:

|  |  |
| --- | --- |
| Class | Location of instantiation |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

List examples of:

1. Composition
2. Aggregation
3. Inheritance
4. Polymorphism

What controls the reproduction of Foxes? How can the program CreateNewFox() during the simulation?

What is the purpose of the variable/property/parameter “Variance”?

# Extensions

This is a fluid list at the moment! It will be honed into a single list of the most likely extensions BUT it is worth you trying them all as we go… this is complex code and only by experimenting will you get a better understanding of how it works…

## Validation

There is very little current validation…. The following are in no particular order

### Coordinate validation

The method simulation.InputCoordinate could house all the required validation (range, type, presence)

### Custom Settings Values

### Landscape size (range (3->30?), type, presence)

### Number of Warrens (range 1-> < #locations, type, presence)

### Number of foxes (range 1-> < #locations, type, presence)

### Variability (range 1->100, type, presence)

### Inspect Fox/Warren

Actually just uses InputCoordinate BUT could list the locations of the animals, maybe a lookup validation/a message

### Various menus (not really needed!)

### Format check???

## New Classes

### Animal that is VERY similar to a rabbit (?mouse?) with a number of mice in a hole

 Would also need the hole class creating..

### Animal that is VERY similar to fox (?owl?) that eats mice/ rabbits

### Animal that eats Foxes (?Gruffalo?) Quite similar to a fox but key differences (tba)

## Changes to existing classes (there are hundreds of possible trivial changes)

### Move Fox

### Rabbits Transfer when Warren is Full

* Create an additional instantiation method for warren (sub New)
* It should accept an array of rabbits that will come from the full warrant
* It should change Rabbit count to reflect the number of rabbits
* Change (or overload) CreateNewWarren to create a new warrant using the new Sub NEW
* Create a method in Warren called GetTransferedRabbits
* It should select 30 rabbits
* Return them as an array of rabbits
* It should destroy their instance in the current warren
* It should change rabbitcount
* CreateWarren should now use GetTransferedRabbits
* Then pass them to the New warren

### Adding variability to death age

All rabbits die at 4. Use variability to allow a percentage of rabbit to live an extra year