⚠ Ant Simulation ⚠

Programming Tasks (Extension)

Extension 1

In the standard simulation, ants will continue to follow a pheromone trail towards a food source, even when that food source has been exhausted. Introduce a second pheromone, “food source exhausted”, to the **Ant** class. When an ant finds that the food source has less than 10% remaining, that ant should lay down the new pheromone rather than the standard pheromone as it returns to the nest. The new “food source exhausted” pheromone should quadruple the decay level of the standard pheromone to help erase the path more quickly.

Extension 2

In the standard simulation it is possible for the **Queen** ant to be culled at random. Introduce a new method into the **Nest** class which detects if a **Queen** has not been present for three stages and promotes a normal **WorkerAnt** to a new **Queen**.

Extension 3

Introduce the concept of a soldier ant which can spray formic acid. Create a new **SoldierAnt** class which moves using the normal random movement around the simulation world. At each stage the soldier ant should review all of its neighbouring cells. If the soldier ant finds an ant or ants that belong(s) to a different nest, it should spray formic acid towards that cell, killing all of the ants in that cell.

Extension 4

Introduce a “message-passing” system whereby ants can leave messages for each other by bumping heads as they move past each other. Modify the **Cell** class to include the concept of “food quality” on a scale of 1–5 (1 being poor, 5 being highly nutritious). As ants pass each other taking food back to the nest, the returning ant, carrying food, “bumps heads” with other ants to inform them of the quality of the food source. Modify the **ChooseCellToMoveTo** method to use this information as additional weighting for deciding if an ant will follow the pheromone trail.

Extension 5

Introduce the concept that a nest can have a maximum of 30 ants belonging to it. Once that capacity has been reached, create a new **Queen** ant which leaves the nest and moves at random through the simulation to create a new nest. Two nests cannot be in neighbouring cells.

Extension 6

Introduce two new options, 6 and 7, to the main menu to give the user the ability to save and reload a simulation using a CSV file.

Extension 7

Implement a dictionary data structure into the **WorkerAnt** class which stores the location of any food source found and how much food there is. At each stage, use the strength of the pheromone trail leading to the food source to make a guess at how many ants have visited the food source and therefore how much food is being consumed per stage, and update the dictionary. Use this within the **ChooseCellToMoveTo** method as additional weighting for selecting whether to move along a pheromone trail back to a food source.

Extension 8

Implement the A* algorithm for a worker ant to find the shortest path back to its nest when carrying food. Use Manhattan distance as the heuristic.

Extension 9

Modify the **Cell** class to allow different types of food – for example, food and water. Introduce the same concept to the **Nest** class to monitor how much has been collected by the worker ants. Modify the **WorkerAnt** class so that an ant won't collect a particular food source if there is already plenty in the nest.

Extension 10

Introduce polydomous colonies which can have multiple nests. Track shared resources to allow ants to deposit food at nests within their colony rather than just at one specific nest. If an ant gets to a nest and there is already plenty of food, then it should go to another nest.

Extension 11

Modify the **Nest** class to create a new method called **ShareFoodWithNearbyNests** which allows ants in one nest to take food to the ants in another nest if it is running low.

Extension 12

Implement Tandem Running by implementing new **LeadAnt** and **FollowerAnt** classes. Once food has been found, instantiate two ants in the nest: one "Lead" and one "Follow". The "Follow" doesn't follow the pheromone, instead moving randomly, but has double the food-carrying capacity. The "Lead" follows the pheromone trail, moving slowly (one cell per two stages). At each stage it should check to see if the follower is within one cell distance. If not, it waits. The follower ant has a 90% chance of moving towards the leader's current position. It stops if already adjacent. When both ants reach the food cell, the follower ant picks up the food and takes it back to the nest.

Extension 13

Introduce the concept of "most food per unit of time" (stage). Modify the **Cell** class to restrict a cell to allowing only three ants present at any time. Introduce a new **Bridge** class which can be placed into a cell which has a capacity of 10 ants present at any time. Add a new option 6 to the main menu to allow the user to place bridges into the simulation at any location except the nest. Add a new option 7 to the main menu to allow the user to place food into cells. Demonstrate ants ignoring a large amount of food being placed in a cell at a location which can't be navigated to using bridges in favour of a smaller amount of food that is easier to get to because overall the nest gains a greater amount of food per unit of time.

Extension 14

Introduce the concept of "unsociality" whereby a worker ant defects to a different nest if that nest has more food in it and therefore offers the ant a greater chance of survival.

Extension 15

Introduce the concept of "nutritional value" in a food source. Once a unit of food has been brought back to the nest, it is assessed by the ants in the nest. If it is lower than other food sources they have looked at, a **ReviewerAnt** follows the pheromone trail out to the food source, removing it.