**Insect Colony Simulator**

**Overview:**

The Insect Colony Simulator is a command-line program that allows users to create and manage insect colonies in a simulated environment. Users can perform various actions, such as setting up new colonies, running simulation cycles, allocating resources and inspecting colony statuses.

**Program Components:**

Classes:

Insect: Represents individual insects with basic behaviors such as resting, consuming food, and constructing.

Methods:

engage\_combat(): Placeholder for insect combat logic.

take\_rest(): Displays a message indicating that the insect is resting.

eat\_food(): Displays a message indicating that the insect is eating.

begin\_building(): Displays a message indicating that the insect is constructing.

InsectColony: Represents a group of insects belonging to a specific type.

Attributes:

breed: Specifies the type of insect in the colony.

insect\_group: A vector containing Insect objects.

Methods:

insertInsect(Insect &ins): Adds an insect to the colony.

executeCycle(): Placeholder for executing a simulation cycle for the colony.

Habitat: Manages multiple insect colonies within the simulation.

Attributes:

colonies: A vector containing InsectColony objects.

Methods:

addColony(InsectColony colony): Adds a new colony to the habitat.

User Interface (Main Function):

The program interacts with the user through a command-line interface. The following commands are supported:

setup [X Y Type]

Description: Creates a new insect colony of a specified type at a given location.

Example Usage: setup 10 20 Beetles

Program Output: “Creating a new colony of insects...”

run [N]

Description: Simulates N cycles for all colonies in the habitat.

Example Usage: run 5

Program Output: “Executing the requested cycles...”

allocate [ColonyID Resources Insects]

Description: Allocates resources and/or insects to a specified colony by its ID.

Example Usage: allocate 1 100 20

Program Output: “Allocating resources and insects to the colony...”

inspect [ColonyID]

Description: Displays the current status and details of a specified colony.

Example Usage: inspect 2

Program Output: “Retrieving colony information...”

end

Description: Terminates the simulation.

Example Usage: end

Program Output: “Simulation complete. Goodbye!”

**Execution Flow:**

The program starts by displaying a welcome message and the list of available commands.

Users input commands to interact with the simulation.

The program processes the commands and performs the corresponding actions:

Creating colonies.

Simulating time cycles.

Allocating resources.

Inspecting colony details.

The simulation continues until the user inputs the end command.

Upon termination, the program displays a goodbye message.

**Example Execution:**

Program Start:

Welcome to the Insect Colony Simulator!

Please choose from the following actions:

1. setup [X Y Type] - Establish a new insect colony.

2. run [N] - Perform N time cycles.

3. allocate [ColonyID Resources Insects] - Distribute supplies or assign insects to a colony.

4. inspect [ColonyID] - View the status of a specific colony.

5. end - Exit the simulator.

User Input: setup 10 20 Ants Output: Creating a new colony of insects...

User Input: run 5 Output: Executing the requested cycles...

User Input: inspect 1 Output: Retrieving colony information...

User Input: end Output: Simulation complete. Goodbye!

**Future Improvements:**

Implement detailed logic for colony actions such as building, resource allocation and combat.

Add persistence to save and load simulations.

Enhance error handling for invalid commands.

Introduce graphical user interface (GUI) for improved user experience.

**Conclusion:**

The Insect Colony Simulator provides a framework for simulating and managing insect colonies. It allows users to create colonies, simulate their activities, and manage resources through a user-friendly command-line interface