**Ant Farm Simulation System**

**Overview**

The **Ant Farm Simulation System** is a computer-based simulation program designed to model and visualize the behavior of an ant colony. Using object-oriented programming (OOP) and design patterns, the system simulates the interactions between ants and their environment, including tasks such as food search, reproduction, and colony expansion. This system mimics a real-world ant colony, where ants exhibit behaviors like foraging, defense, and communication through pheromones, based on various environmental factors and internal mechanisms.

The primary objective of the simulation is to offer insights into the dynamics of ant colonies, providing a platform for experimenting with different scenarios, behaviors, and environmental conditions. The system uses various design patterns (Creational, Structural, and Behavioral) to model and organize the components of the simulation in a modular and extensible way. The simulation is built using **C++** and integrates several design principles to ensure flexibility, scalability, and efficiency.

**Features and Functionalities**

The Ant Farm Simulation, based on the functional and technical requirements, is designed to simulate the ecosystem of ant colonies, their interactions, and various life processes. The following features and functionalities are implemented in the code, aligned with the specified requirements:

**1. Meadow Class (Singleton Pattern)**

* The **Meadow** class represents the environment where the ant farms exist. It is implemented as a **singleton**, ensuring only one meadow instance is present in the simulation at any time.
* The meadow serves as the central point for all ant farm interactions and manages the creation of new ant farms (colonies).

**2. Ant Farm (Builder Pattern)**

* Each **AntFarm** is created by building a network of rooms using the **Builder Pattern**. These rooms serve as locations for spawning ants and managing various ant activities.
* The ant farm can be dynamically expanded with new rooms, which may include worker rooms, resting rooms, and combat areas.
* The **AntFarm** class is templated to handle different types of ants (worker, warrior, queen, etc.).

**3. Ants and Colony Management (Factory and Decorator Patterns)**

* The creation of ants, specifically **Drones**, **Warriors**, and **Queens**, is done through the **Factory Pattern**, ensuring that each ant type is created with the proper attributes.
* The **Decorator Pattern** is used to manage the ants' attributes dynamically, allowing them to gain bonuses or enhancements based on their species and experiences (e.g., harvest food faster or increase combat effectiveness).

**4. Species Variability**

* The simulation accommodates multiple species of ants, and the number of species is determined by the formula: (Student\_RollNumber % 6) + 10.
* Each species has its own unique bonuses, which can include increased efficiency in food collection, better combat abilities, or a combination of both.

**5. Battle Mechanics and Ant Death**

* **Ants battle other ants** from rival colonies. When an ant loses a battle, it dies, and the winner gains the defeated ant's attributes (e.g., strength, health).
* If an ant colony's queen is killed, the killing colony’s queen absorbs the dead queen’s population and gains additional attributes from the fallen colony.

**6. Resting and Food Consumption**

* Ants need to **rest periodically**. Resting ants consume food, and an ant farm has a limit on how many ants can rest at the same time based on the available rooms.
* The number of resting ants is limited by the number of rooms dedicated to resting in the **AntFarm**.

**7. Tick-Based Simulation (Mediator Pattern)**

* The simulation operates on a **tick-based system**, where each tick represents a unit of time. During each tick, ants perform various actions, such as:
  + **Drones** look for food.
  + **Warriors** search for enemies to battle, or return to the colony for food.
  + **Queens** spawn new eggs to create more ants.
* The **Mediator Pattern** is used to manage the interactions between ants, rooms, and the overall simulation environment.

**8. Colony Death and Simulation End**

* The simulation ends when there is **1 or fewer active colonies** at the end of a tick. This occurs when all but one colony has been defeated or merged into a single colony.

**9. Room Construction and Worker-Ticks**

* **Rooms** in the ant farm require worker ants to dig and build them. The time to build a room is proportional to the number of workers and their efficiency. For example, 100 worker ants may be able to dig a room in 1 tick, while 50 workers would take 2 ticks.
* Constructing rooms is a costly process, requiring resources and time, but it is essential for expanding the capacity of the ant farm and allowing more ants to rest.

**10. Command Line Interface (CLI)**

The interface allows for interaction with the simulation via a command-line interface, supporting several commands to control and observe the simulation:

* **spawn X Y T**: Create a new ant colony at the specified coordinates (X, Y) with species type T.
* **give I R A**: Provide colony I with resource R (such as food or ants) of amount A.
* **tick [T]**: Perform T ticks (if T is not specified, perform one tick).
* **summary I**: Display a summary of colony I, including species, number of ants, kills, and status.

Example command interactions:

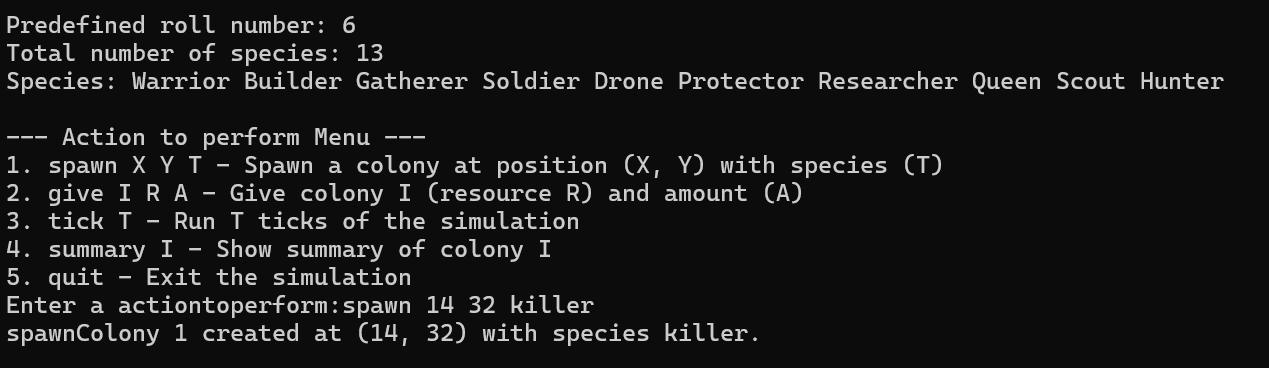
* spawn 14 -32 Killer – Creates a colony of Killer ants at position (14, -32).
* give 1 food 50 – Gives colony 1, 50 units of food.
* tick 10 – Performs 10 ticks in the simulation.
* summary 1 – Provides a summary of colony 1, including species, workers, warriors, kills, and more.

**11. End of Simulation**

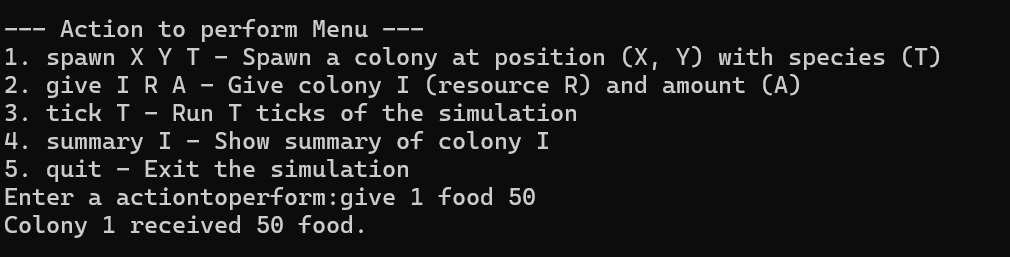
* The simulation concludes when, at the end of a tick, there is **1 or fewer active colonies/queens**.
* The game also features the possibility of **ant species merging** if one queen kills another, leading to stronger colonies as they absorb the attributes of the other species.

**Output:**

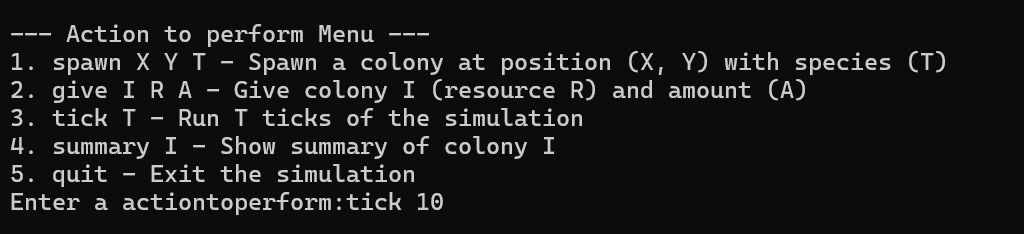
**Spawning a colony:**

****

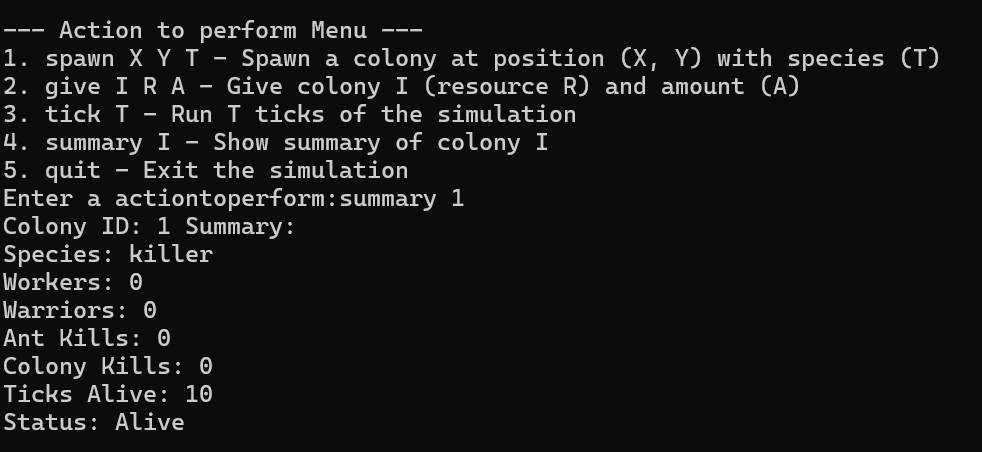
**Giving food:**



**Tick a specie:**

****

**Summary:**

****

**Conclusion:**

The Ant Farm Simulation incorporates numerous patterns like Singleton, Builder, Factory, Decorator, and Mediator, and features a detailed, interactive command-line interface to control and observe the colony dynamics. The primary aim is to simulate the life cycle, behaviors, and interactions of ants in a colony, including their battles, reproduction, and resource management. This simulation also includes unique mechanics such as merging species, building rooms, and handling the life and death of colonies, creating a complex ecosystem for the user to interact with.