To ensure efficient and maintainable code for enemy behavior, the following patterns are utilized:

1. Enemy State Management with Enums

An enum is used to define and manage the enemy's current state. The states include Idle, Patrol, and Chase. This provides a centralized and easily modifiable way to track and switch between enemy behaviors.

2. Enemy Behavior Interfaces

Interfaces are implemented to promote flexibility and ease of extension for different enemy types.

- **IDealDamage**: For enemies that can deal damage.
- **IAffectSurroundings**: For enemies that can affect surroundings
- **IDestructible**: For enemies that can be destroyed.
- **IMeleeEnemy or IRangeEnemy**: For enemies with specific attack methods (melee or ranged).

These interfaces provide a common structure for all enemies while enabling easy customization for unique enemy types. For example, the explosive barrel wouldn't take the same things as the melee enemy, but they may have similarities. They may both take IDealDamage but utilize it in different ways.

3. Scriptable Objects for Enemy Stats

Scriptable Objects store enemy-specific data, such as name, health, damage, speed, FOV, and view distance. This data-driven approach allows for easy adjustments to enemy attributes directly in the Unity Inspector, providing flexibility without modifying core game code. Each enemy type can have its own Scriptable Object to manage its stats and behaviors.