

○ ○ ○

```
private void TrySpawnSlime()
{
    // Create a list of available slime prefabs based on enabled types
    List<GameObject> availableSlimes = new List<GameObject>();

    if (spawnGreen && slimeGreenPrefab != null) availableSlimes.Add(slimeGreenPrefab);
    if (spawnFire && slimeFirePrefab != null) availableSlimes.Add(slimeFirePrefab);
    if (spawnIce && slimeIcePrefab != null) availableSlimes.Add(slimeIcePrefab);

    // If no slimes are available to spawn, exit the method
    if (availableSlimes.Count == 0) return;

    // Select a random slime prefab from the available list
    GameObject prefab = availableSlimes[Random.Range(0, availableSlimes.Count)];

    // Select a random spawn location
    Transform spawnPoint = spawnLocations[Random.Range(0, spawnLocations.Length)];

    // Instantiate the slime at the chosen location and rotation
    GameObject newSlime = Instantiate(prefab, spawnPoint.position, spawnPoint.rotation);

    // Add and initialize the SlimeTracker component to the new slime
    SlimeTracker tracker = newSlime.AddComponent<SlimeTracker>();
    tracker.Initialize(this);

    // Increment the current mob count and reset the spawn cooldown
    currentMobs++;
    ResetCooldown();
}

public void DecrementMobCount()
{
    // Decrease the current mob count but keep it from going below zero
    currentMobs = Mathf.Max(0, currentMobs - 1);
}

private void ResetCooldown()
{
    // Reset the cooldown timer with a random value between 60% and 100% of spawnTimer
    cooldown = Random.Range(spawnTimer * 0.6f, spawnTimer);
}
```