

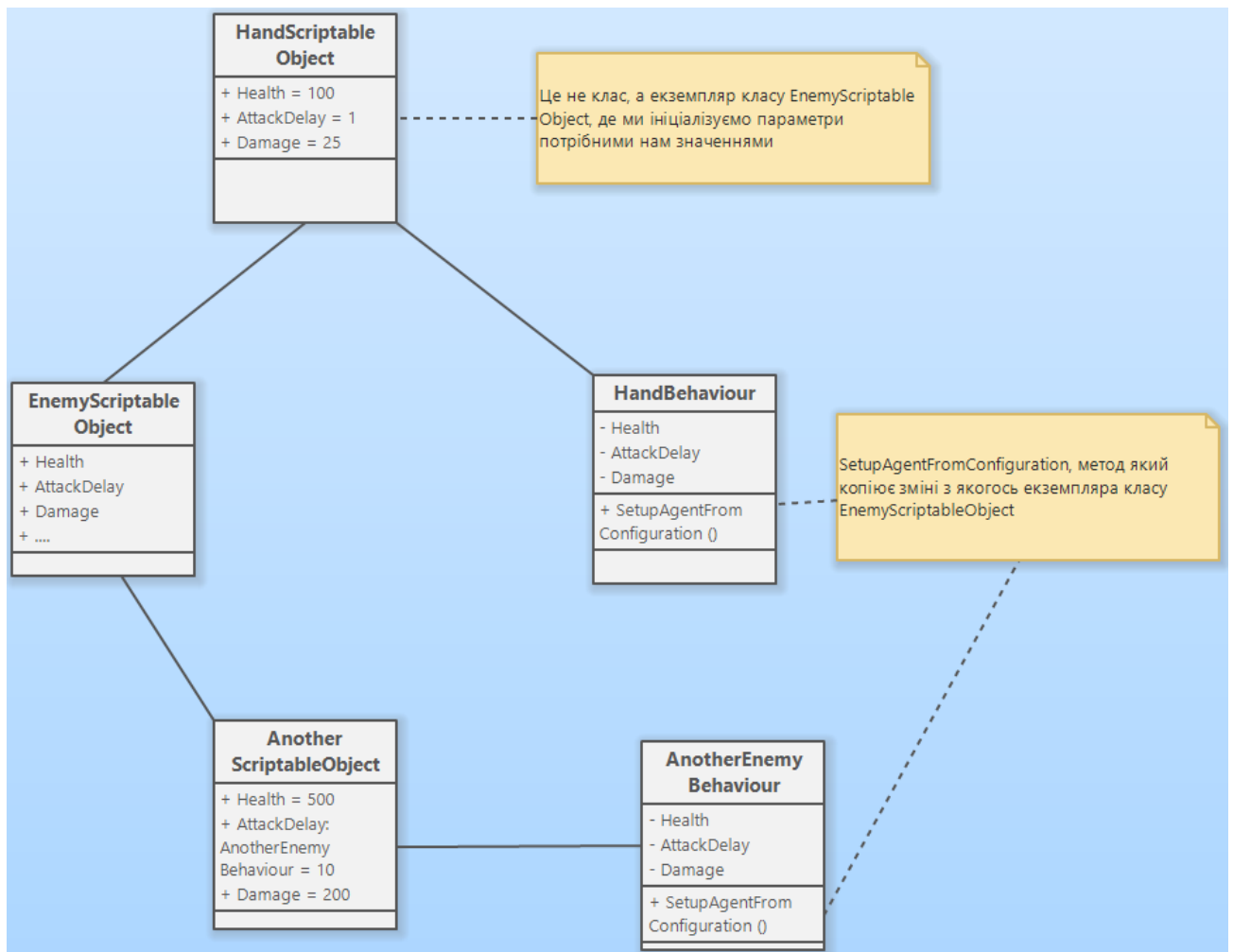
Легковаговик — це структурний патерн проектування, що дає змогу вмістити більшу кількість об'єктів у відведеній оперативній пам'яті. Легковаговик заощаджує пам'ять, розподіляючи спільний стан об'єктів між собою, замість зберігання однакових даних у кожному об'єкті.

В моєму проекті цей патерн використовується багато де, але в основному для об'єктів, які можуть або будуть повторюватися. Для прикладу можна взяти ворогів.

Зрозуміло, що ворог одного типу буде мати одні й ті ж самі параметри, і тому замість того, щоб дублювати одне і теж саме, простіше буде винести весь такий код в один екземпляр, звідки всі копії ворогів будуть брати дані.

Для цього я використовую клас `EnemyScriptableObject`, цей клас успадковується від класа `ScriptableObject`, тобто це значить, що він не буде використовуватися, як ігровий об'єкт. А буде “блоком даних” звідки об'єкти будуть брати різні параметри.

Цей клас є універсальним і підходить для всіх типів ворогів, тобто в ньому задані деякі змінні, які будуть ініціалізовані пізніше при створенні екземпляра. В створеному екземплярі за цим класом ми і прописуємо параметри всіх змінних для конкретного типу ворога. Така схема допомагає уникнути дублюванню кода.



Об'єктів з класом **HandBehaviour** в мене може бути дуже багато, але велика кількість змінних, які використовуються в цьому класі, будуть одні й ті ж самі, тому що всі вони знаходяться в одному екземплярі класу **EnemyScriptableObject** (для **HandBehaviour**, це **HandScriptableObject**, або в проекті просто **Hand**).

```

namespace EnemyBaseNS
{
    /// <summary>
    /// ScriptableObject that holds the BASE STATS for an enemy. These can then be modified at object creation time to buff up enemies
    /// and to reset their stats if they died or were modified at runtime.
    /// </summary>
    [CreateAssetMenu(fileName = "Enemy Configuration", menuName = "ScriptableObject/Enemy Configuration")]
    [Unity Script|1 reference]
    public class EnemyScriptableObject : ScriptableObject
    {
        public SkillScriptableObject[] Skills;
        // Enemy Stats
        public int Health = 100;
        public float AttackDelay = 1f;
        public int Damage = 5;
        public float AttackRadius = 1.5f;
        public float AttackDistance = 1.5f;

        // NavMeshAgent Configs
        public float AIUpdateInterval = 0.1f;

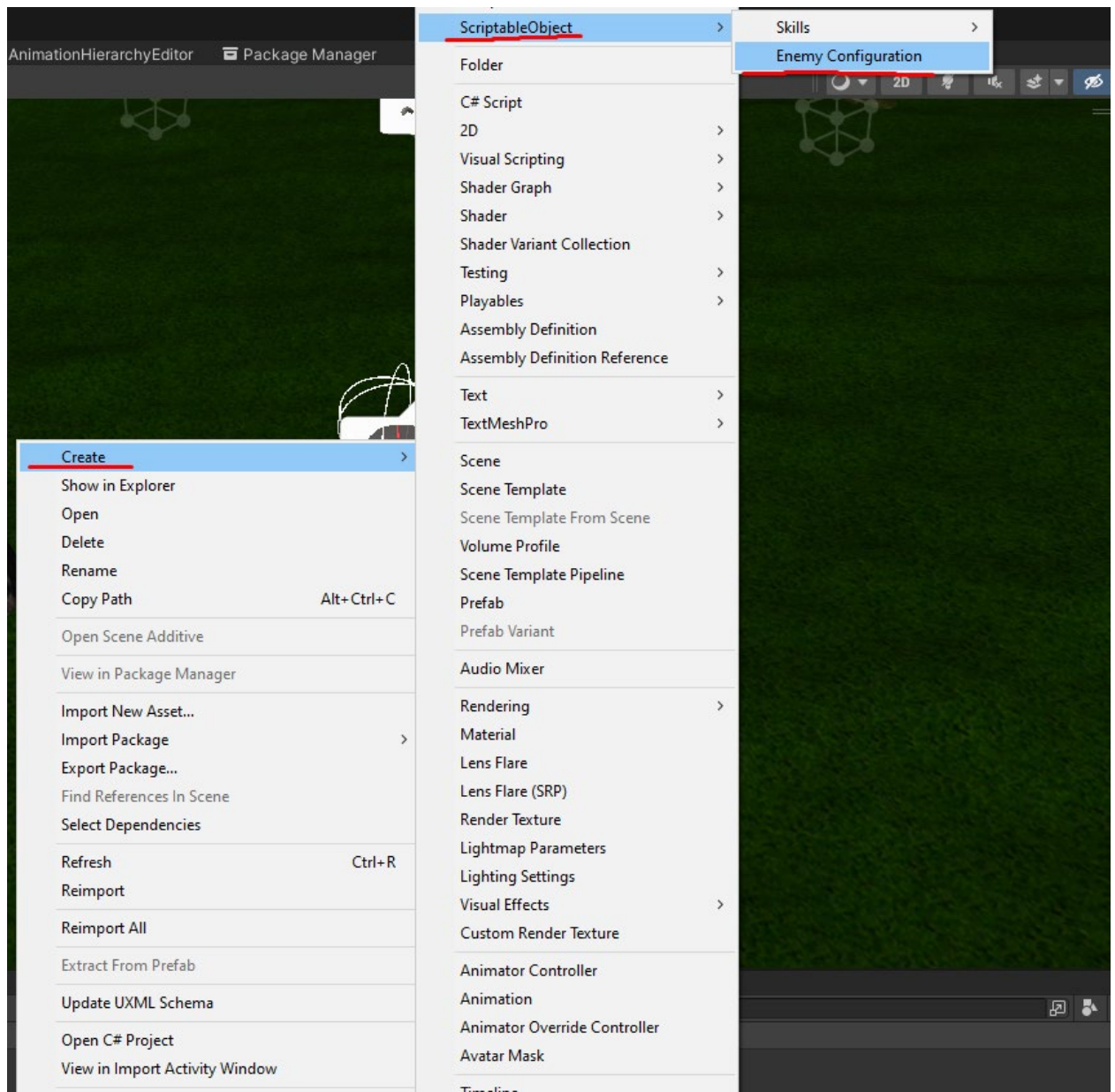
        public float Acceleration = 8;
        public float AngularSpeed = 120;
        // -1 means everything
        public int AreaMask = -1;
        public int AvoidancePriority = 50;
        public float BaseOffset = 0;
        public float Height = 2f;
        public ObstacleAvoidanceType ObstacleAvoidanceType = ObstacleAvoidanceType.LowQualityObstacleAvoidance;
        public float Radius = 0.5f;
        public float Speed = 3f;
        public float StoppingDistance = 0.5f;
    }
}

```

Клас EnemyScriptableObject з великою кількістю змінних.

Наслідування від ScriptableObject дозволяє нам створити екземпляр класу в редакторі Unity.

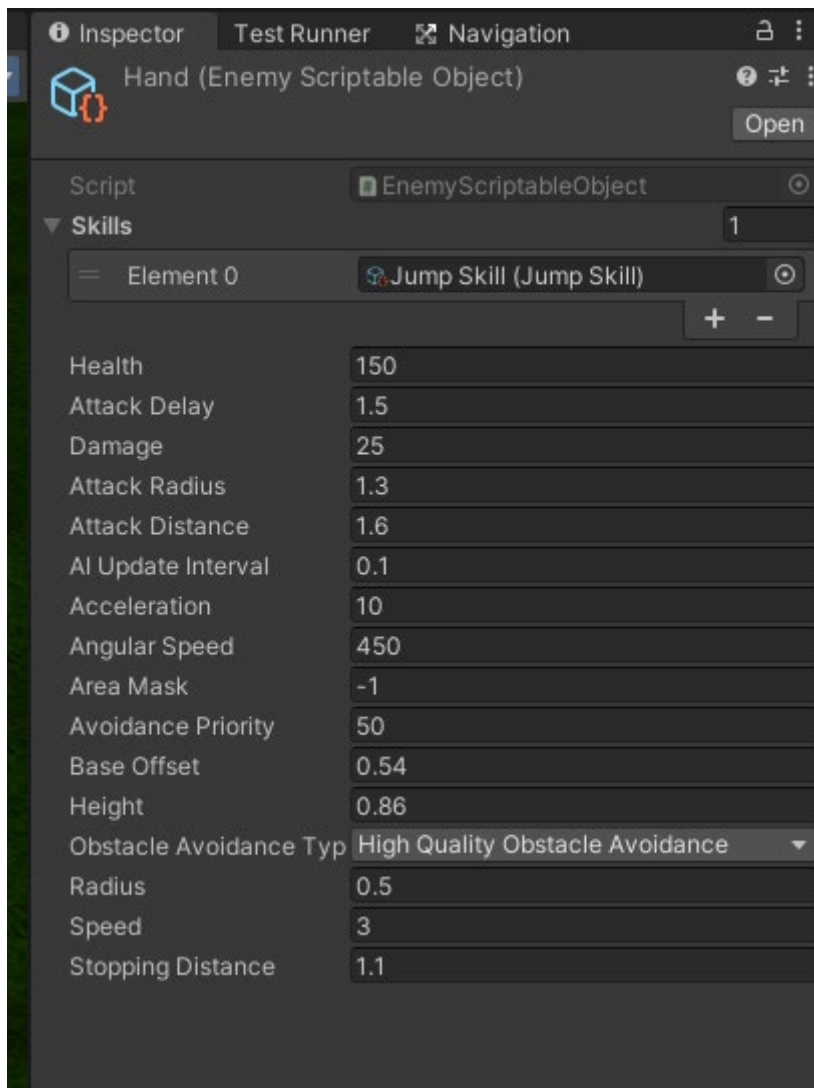
Тобто в редакторі Unity з'явиться окрема кнопка для створення екземплярів класу EnemyScriptableObject в які ми можемо задати потрібні нам параметри.



Після чого створиться екземпляр. (Hand це і є HandScriptableObject)



Де ми можемо задати всі параметри.



Тепер для типу противника Hand в нас є всі потрібні змінні, які містяться в одному екземплярі та не дублюються.

В класі HandBehaviour є метод (який успадковується від класу Enemy) SetupAgentFromConfiguration, який встановлюємо параметри з HandScriptableObject в кожен об'єкт класу.

```

public virtual void SetupAgentFromConfiguration()
{
    Agent.acceleration = EnemyScriptableObject.Acceleration;
    Agent.angularSpeed = EnemyScriptableObject.AngularSpeed;
    Agent.areaMask = EnemyScriptableObject.AreaMask;
    Agent.avoidancePriority = EnemyScriptableObject.AvoidancePriority;
    Agent.baseOffset = EnemyScriptableObject.BaseOffset;
    Agent.height = EnemyScriptableObject.Height;
    Agent.obstacleAvoidanceType = EnemyScriptableObject.ObstacleAvoidanceType;
    Agent.radius = EnemyScriptableObject.Radius;
    Agent.speed = EnemyScriptableObject.Speed;
    Agent.stoppingDistance = EnemyScriptableObject.StoppingDistance;
    Health = EnemyScriptableObject.Health;
    EnemyAttack.AttackRadius = EnemyScriptableObject.AttackRadius;
    EnemyAttack.AttackDistance = EnemyScriptableObject.AttackDistance;
    EnemyAttack.AttackDelay = EnemyScriptableObject.AttackDelay;
    EnemyAttack.Damage = EnemyScriptableObject.Damage;
    Skills = EnemyScriptableObject.Skills;
}

```

За допомогою цього патерна уникається зайве дублювання кода, а також він дозволяє, якщо потрібно, поміняти деякі параметри для конкретного типу противника і при цьому не лізти в класи та код.

Скрипти можете знайти за шляхом:

Assets/Scripts/Enemies/Base/EnemyScriptableObject.cs

Assets/Scripts/Enemies/Base/Enemy.cs

Assets/Scripts/Enemies/EnemiesType/Hand/HandBehaviour.cs.

Assets/Scripts/Enemies/EnemiesType/Hand/Hand.asset