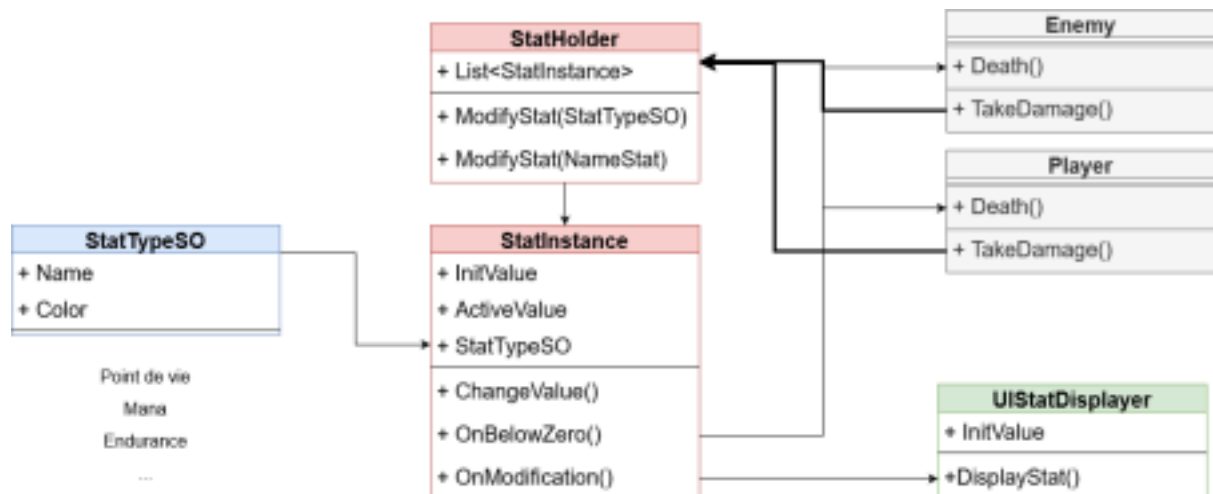


Acte 2 : Game Structure

Système de caractéristiques

Objectif:

- Mettre en place un système d'attribut générique



Construisons un type général pour générer l'ensemble des "stats" de notre jeu, ce système s'occupera des points de vie, du mana, de l'endurance et tout autre ressources que nous souhaiterions ajouter...

Le "scriptable objet" **StatTypeSO** représente un type de statistique. Une statistique doit avoir un nom et une couleur. Comme pour la séance précédente, utiliser la mention `[CreateAssetMenu]` au-dessus de la définition de classe pour pouvoir en créer des instances dans votre projet (clic droit dans les Assets). Les instances de ScriptableObject peuvent être comparées efficacement avec l'opérateur `==`.

Commencer par instancier un **StatTypeSO** qui représentera la ressource **Life** (Clic droit dans le projet > Create > StatTypeSO).

La classe **StatsInstance** représente l'instance d'une statistique, c'est-à-dire ces valeurs par défaut et sa valeur à un instant donné. Cette classe est une classe classique C# (pas un ScriptableObject ni un MonoBehaviour). Elle devra avoir une instance de **StatTypeSO** ainsi qu'une valeur d'initialisation et une valeur active. De plus, ajouter un attribut public **UnityEvent** ou (un `delegate` C#, voir syntaxe en fin de sujet) permettant d'enregistrer des méthodes qui seront invoquées lorsque la valeur est modifiée (`OnModification`) et lorsqu'elle tombe à 0 (`OnBelowZero`).

Construire ensuite le script **StatsHolder**, un MonoBehaviour qui portera et recensera les différentes **StatsInstance** d'un objet. Ce script doit permettre :

- D'initialiser les statistiques

- D'y accéder à travers un **StatTypeSO** ou un string (le nom de la statistique)

Attachez au joueur et à l'ennemi le composant **StatsHolder**, définissez leur une **StatsInstance** de type *Life* et adaptez le code existant pour que le système de dommage utilise le StatsHolder.

Notion : [ScriptableObject](#), [CreateAssetMenu](#), [UnityEvent](#)

Effet additionnel:

- Les GameObjects du joueur et du monstre contiennent une barre de vie. Ajouter dans le GameObject Slider un script permettant à la barre de vie de se mettre à jour automatiquement.

Pour aller plus loin :

- Ajoutez la statistique *Stamina*, qui se régénère automatiquement et est nécessaire pour sauter ou sprinter à votre convenance.
- Rendez le Prefab du monstre paramétrique en utilisant des ScriptableObject, par exemple modifier la taille et la vitesse du monstre ainsi que ces statistiques.
- Adapter l'interface IDamageable pour pouvoir interagir avec les différents types de statistiques.

Delegates : C# propose une fonctionnalité similaire au UnityEvent

```
public delegate void voidDelegate(); // signature of delegate
public voidDelegate onModify;       // actual delegate
```

De nouvelles méthodes peuvent être enregistrées dans le "delegate" en utilisant l'opérateur '+=', l'appel des méthodes s'effectue avec la méthode Invoke.