

### NOTE

Join the Astra RPG Discord server!

There is now a dedicated **Discord server** for Astra RPG Framework and its extensions. Join to **receive notifications** about new extension releases and important updates, to **ask for new features, report bugs, share ideas**, and **showcase your Astra creations** with other developers.

 Join the Discord Server: <https://discord.gg/nJVRMkGrZg>

## Introduction

Astra RPG Health estende il framework base, [(Astra RPG Framework)], aggiungendo funzionalità per la manipolazione della vita e del calcolo dei danni per le entità. Il package è pensato con la stessa filosofia di design dell'asset base: architettura basata su Scriptable Objects per incentivare flessibilità, modularità, e testabilità. Se avete già familiarità con il package base, vi sentirete a casa con l'utilizzo delle sue funzionalità.

Puoi definire i tuoi tipi di danno, designare le statistiche difensive usate per ridurre ciascun tipo di danno, configurare la pipeline per calcolare i danni con gli step desiderati, configurare il lifesteal per certi tipi di danni, le strategie da attuare all morte delle entità, e molto altro.

## Vocabolario di Astra RPG Health

### Damage Type

I damage type rappresentano le diverse categorie di danno possono essere inflitte alle entità. Esempi comuni includono "Fisico", "Magico", "Fuoco", "Ghiaccio", ecc. Puoi creare tipi di danno personalizzati per adattarli alle esigenze del tuo gioco.

### Damage Source

Le damage source rappresentano l'origine del danno inflitto. Questo è altamente specifico al contesto del gioco. Ad esempio, un gioco potrebbe avere damage source come "Entity", "Environment", "Potion", ecc. Un altro, potrebbe voler definire damage source più specifiche come "Attack", "Spell", "Equipment", "Trap", "Environment", "Damage Over Time", ecc. La differenza principale tra damage source e damage type è che i damage type categorizzano il danno in base alla sua natura, mentre le damage source identificano da dove proviene il danno. La differenza può essere sottile, ma è importante per la logica di gioco e le meccaniche. Torneremo su questi concetti più avanti, dove vedremo le differenze pratiche tra i due.

### Heal Source

Le heal source rappresentano l'origine della guarigione. Simile alle damage source, le heal source sono specifiche al contesto del gioco. Esempi comuni includono "Potion", "Spell", "Lifesteal", "Ability", "Environment", ecc. Talvolta, in certi giochi, una classica definizione di Heal Source può comprendere "Self" e "Ally", in quanto permettono di implementare meccaniche di aumento della guarigione fornita/ricevuta basate sul caster e sul target.

## Barrier

Il concetto di HP temporanei può assumere svariati nomi nei vari giochi, sebbene la meccanica di fondo sia sempre la stessa: fornire un ammontare di hit points extra ed effimeri che vengono detratti al posto della vita quando si subiscono danni. In Astra, questi punti vita temporanei vengono chiamati "Barrier".

## Raw and Net Damage

Con Raw Damage si intende il danno che un certo attacco o skill intende infliggere. Questo danno non tiene conto di resistenze, critici, modificatori, ecc. Il Net Damage è il risultato dell'elaborazione del Raw Damage tenendo conto di modificatori di danno, resistenze, barrier, colpi critici, ecc.

## Damage Modifiers

I damage modifiers sono componenti che possono alterare il danno calcolato in vari modi. Possono essere usati per implementare meccaniche come riduzione del danno, aumento del danno, resistenze, vulnerabilità, e altro ancora. I damage modifiers sono generalmente utilizzati dalla pipeline del calcolo del danno effettivo (che vedremo a breve).

## How is Astra RPG Health organized and how does it work?



### EntityHealth

# Namespace ElectricDrill.AstraRpgHealth

## Classes

### [EntityHealth](#)

Manages the health, damage, and healing mechanics for an entity. Handles damage calculation, health regeneration, barriers (temporary HP), and death events.

## Enums

### [EntityHealth.HpBehaviourOnMaxHpDecrease](#)

### [EntityHealth.HpBehaviourOnMaxHpIncrease](#)