

Documentation Générale

Sommaire

Documentation Générale

Sommaire

[Comment fonctionne le système de probabilité ?](#)

[Comment créer un ennemi ?](#)

[Comment configurer le spawn d'ennemis ?](#)

[Comment l'UI peut-elle accéder aux statistiques et upgrades des armes ?](#)

[Comment upgrade les armes depuis l'UI ?](#)

[Comment créer un nouveau pick-up d'Upgrade ?](#)

[Consultez la documentation sur les scripts pour voir comment les configurer](#)

Comment fonctionne le système de probabilité ?

Certaines fonctionnalités comme le Spawn d'un ennemi ou le loot d'un power up nécessitent une variable de probabilité et voici comment les utiliser. Pour `EnemySpawn` par exemple, vous avez une liste d'ennemis potentiels et la probabilité de spawn de chaque ennemi qui est définie dans le script `Enemy`, la variable de probabilité est un simple float et la chance d'un ennemi de spawn sera égale à : la variable probabilité divisée par les sommes des variables probabilité de chaque ennemi de la liste, donc si dans la liste de `EnemySpawn` nous avons 2 ennemis avec {(Ennemi1 probabilité = 1), (Ennemi2 probabilité = 2)}, la chance de l'Ennemi1 de spawn sera de 1/3 et la chance de l'Ennemi2 sera de 2/3.

Comment créer un ennemi ?

Sur votre Prefab, ajoutez simplement le script `Enemy` et `Health`, et éventuellement le script `LootOnDie` si vous voulez que votre ennemi puisse lâcher quelque chose quand il meurt.

Comment configurer le spawn d'ennemis ?

Pour permettre l'apparition d'ennemis, vous aurez besoin de deux scripts sur la scène, en premier lieu le script `BoundariesArea` pour définir une zone de jeu et ensuite le script `EnemySpawn`.

Après cela, il suffit d'appeler la méthode `EnemySpawn.StartSpawn()` dans votre Level Manager.

Comment l'UI peut-elle accéder aux statistiques et upgrades des armes ?

À partir de votre script, récupérez le script `Weapons` sur le joueur, puis enregistrez-vous sur l'événement `Weapons.onStatsChange` et utilisez les méthodes : `Weapons.GetCurrentStat`, `Weapons.GetTemporaryUpgrades` et `Weapons.GetPermanentUpgrades` pour récupérer les données et mettre à jour votre interface utilisateur.

Comment upgrade les armes depuis l'UI ?

A partir de votre script, récupérez le script Weapons sur le joueur, puis appelez la méthode `weapons.Upgradeweapons` et passez en paramètre une liste d'Upgrade que vous souhaitez ajouter.

Comment créer un nouveau pick-up d'Upgrade ?

Sur votre Prefab, il suffit d'ajouter les scripts `Pickup` et `UpgradePickup` et de les configurer.

Consultez la documentation sur les scripts pour voir comment les configurer