GENERATEURS D’ENNEMIS

Le but est de créer des générateurs aléatoires d’ennemis, qui soient à la fois rythmé, varié, et qui s’adapte à la performance du joueur.

Le système coordonne plusieurs « clock » d’ennemis : on aura un générateur par classe d’ennemi présent dans la scène. On peut représenter ces Clock par des jauges qui se remplissent avec une valeur max correspondant au temps avant spawn d’une unité de cette classe.

# DESCRIPTION DU SYSTEME

Les règles sont les suivantes :

* Une jauge par classe d’ennemi, avec une valeur max propre à chaque classe : **clockCost**.
* Chaque jauge a sa propre sf ::Clock propre qui l’incrémente : **currentClock**.
* Chaque ennemi tué augmente la currentClock de toutes les jauges en fonction de la classe de l’ennemi tué, et avec une pondération (**clockCost** \* **killMultiplier**)**.**
* Lorsqu’une jauge est pleine, elle spawn un ennemi de sa classe. Il y a une dose d’aléa en fonction d’une proba « **criticalSpawnChance** » et d’une proba « **missSpawnChance** » :
  + Sur un jet aléatoire, si **criticalSpawnChance** est vrai, alors la jauge de la classe concernée par le spawn ne consomme que (1 – X%) de la totalité de la jauge (autrement dit la jauge redémarre à X% au lieu de 0%). La valeur X s’appelle **criticalSpawnCost**
  + Sur ce même jet aléaoire, si **missSpawnChance** est vrai, alors la jauge se vide entièrement sans que l’ennemi ne soit spawné.
  + Sur les probabilités restantes (qui ont vocation à être le cas majoritaire), l’ennemi est spawné et la jauge est vidée entièrement.
* Si un ennemi est spawné, cela décrémente toutes les autres jauges de la valeur **clockCost** de la jauge qui spawn l’ennemi, pondérée par un facteur **collateralCostMultiplier**. Les autres jauges doivent pouvoir aller dans des valeurs négatives dans ce cas-là.