### Document Technique pour Unit.cs

**Namespace** : Units

#### Classe Unit

La classe Unit représente une unité ou un personnage dans le jeu. Elle gère les mouvements, les attaques, et l'état général de l'unité.

#### Champs :

* **string title** :
  + **Description** : Le nom de l'unité.
* **float health** :
  + **Description** : La santé actuelle de l'unité.
* **int attack** :
  + **Description** : La valeur d'attaque de l'unité.
* **int defense** :
  + **Description** : La valeur de défense de l'unité.
* **int spirituality** :
  + **Description** : La valeur de spiritualité de l'unité.
* **int mobility** :
  + **Description** : La mobilité de l'unité, définissant combien elle peut se déplacer.
* **Weapon weapon** :
  + **Description** : L'arme équipée par l'unité.
* **Commander commander** :
  + **Description** : Le commandant auquel l'unité appartient (PLAYER\_1, PLAYER\_2, ou NULL).
* **float moveSpeed** :
  + **Description** : La vitesse de déplacement de l'unité.
* **bool isWasMoved** :
  + **Description** : Indique si l'unité a déjà été déplacée pendant ce tour.
* **bool isCanMove** :
  + **Description** : Indique si l'unité peut encore se déplacer.
* **bool isCanAttack** :
  + **Description** : Indique si l'unité peut attaquer.
* **bool isMoving** :
  + **Description** : Indique si l'unité est actuellement en mouvement.
* **List<Vector3> movePath** :
  + **Description** : La liste des positions que l'unité doit atteindre pour se déplacer.
* **const float ATTACK\_MULTIPLIER** :
  + **Description** : Multiplicateur utilisé pour les coups critiques (valeur fixe de 1.5).

#### Propriétés :

* **bool IsMoving** :
  + **Description** : Indique si l'unité est en train de se déplacer.
* **Weapon Weapon** :
  + **Description** : Accède à l'arme de l'unité.
* **int Mobility** :
  + **Description** : Accède à la mobilité de l'unité.
* **bool IsWasMoved** :
  + **Description** : Indique si l'unité a été déplacée.
* **bool IsCanAttack** :
  + **Description** : Indique si l'unité peut attaquer.
* **Commander Commander** :
  + **Description** : Accède au commandant de l'unité.

#### Méthodes :

* **void Start()** :
  + **Description** : Initialise isCanMove à true.
* **void FixedUpdate()** :
  + **Description** : Vérifie si l'unité doit se déplacer et commence le mouvement si nécessaire.
* **public void ResetWasMovedState()** :
  + **Description** : Réinitialise l'état de déplacement de l'unité.
* **public void Move(List<Vector3> \_path)** :
  + **Description** : Définit le chemin de déplacement de l'unité si elle n'a pas encore été déplacée.
* **IEnumerator PerformMove()** :
  + **Description** : Exécute le déplacement de l'unité le long du chemin spécifié.
* **public void Wait()** :
  + **Description** : Marque l'unité comme ayant terminé son tour.
* **public float CalculateHit()** :
  + **Description** : Calcule le taux de réussite de l'attaque en fonction de l'arme et des attributs de l'unité. Inclut un potentiel coup critique.
* **public void Attack()** :
  + **Description** : Marque l'unité comme ayant attaqué.
* **public void TakeDamages(float \_inflictedDamages)** :
  + **Description** : Applique des dégâts à l'unité, réduisant sa santé en conséquence.

#### Événements :

* **public static event System.Action OnWasMoved** :
  + **Description** : Événement déclenché lorsque l'unité a été déplacée.

### Notes :

* **Gestion des Déplacements** :
  + Le déplacement de l'unité est géré par une coroutine qui interpole la position de l'unité le long du chemin spécifié.
* **Calcul des Attaques** :
  + Le calcul du taux de réussite de l'attaque est basé sur l'arme équipée et les attributs de l'unité, avec la possibilité d'un coup critique.