**Document Technique pour CombatSystem.cs**

**Nom du script** : CombatSystem

**Namespace** : Aucun spécifié

**Vue d'ensemble :**

Le script CombatSystem est responsable de la gestion des combats dans le jeu. Il calcule les ennemis attaquables en utilisant un système de pathfinding pour déterminer la portée d'attaque d'une unité, en ignorant les obstacles lors de la recherche de chemins.

**Composants clés :**

* **Références sérialisées** :
  + pathfindingAStar : Instance de la classe PathfindingAStar pour effectuer des recherches de chemin.
  + tileSystem : Instance de la classe TileSystem pour convertir les positions du monde en positions de cellules.
  + unit : L'unité dont les ennemis attaquables sont calculés.
  + enemies : Liste des ennemis à vérifier pour la portée d'attaque.
  + maxRange : Plage maximale d'attaque de l'unité.
  + minRange : Plage minimale d'attaque de l'unité.

**Méthodes principales :**

* **Méthodes Unity** :
  + Update() :
    - Appelle la méthode GetAttackableEnemiesWithoutConsideringObstacles pour vérifier les ennemis attaquables à chaque mise à jour du cadre.
* **Méthodes personnalisées** :
  + GetAttackableEnemiesWithoutConsideringObstacles(GameObject unit, List<GameObject> enemies, int maxRange, int minRange) :
    - Calcule les ennemis attaquables en ignorant les obstacles en utilisant le pathfinding.
    - Convertit les positions des unités et des ennemis en coordonnées de cellules à l'aide du tileSystem.
    - Utilise pathfindingAStar.FindPath pour obtenir un chemin vers chaque ennemi.
    - Vérifie si l'ennemi est dans la portée d'attaque spécifiée par maxRange et minRange.
    - Retourne une liste des coordonnées des ennemis attaquables.
  + NotVerifieWalkability(Vector3 position) :
    - Vérifie si une position donnée est walkable (praticable) en utilisant un raycast vers le bas.
    - Retourne true si la position est considérée comme praticable ("Ground" ou "Tree"), sinon false.

**Remarques :**

* **Optimisation** :
  + La méthode GetAttackableEnemiesWithoutConsideringObstacles effectue des calculs intensifs et peut être optimisée en réduisant les appels à FindPath si nécessaire.
  + Le raycast dans NotVerifieWalkability est effectué en utilisant une position fixe (hauteur de 4 unités), ce qui peut être ajusté en fonction des besoins du jeu.
* **Débogage** :
  + Le script utilise des appels à Debug.Log pour afficher les chemins trouvés et les ennemis ajoutés à la liste des attaquables. Ces lignes peuvent être supprimées ou désactivées en production.

Ce document fournit une vue d'ensemble et des détails sur le fonctionnement du script CombatSystem, expliquant comment il calcule les ennemis attaquables en utilisant des méthodes de pathfinding et des vérifications de praticabilité des positions.