**Document Technique pour MoveSystem.cs**

**Nom du script** : MoveSystem

**Namespace** : GamePlay.Sys

**Vue d'ensemble :**

Le script MoveSystem gère le déplacement des unités sur une grille en utilisant un système de pathfinding. Il calcule et affiche les plages de mouvement disponibles pour une unité et déplace l'unité vers une cible spécifiée.

**Composants clés :**

* **Références privées** :
  + gridScale : Taille d'une cellule dans la grille.
  + tileSystem : Instance de la classe TileSystem pour la gestion des tuiles sur la grille.
  + pathfindingAStar : Instance de la classe PathfindingAStar pour effectuer des recherches de chemin.
  + activeTilesPosition : Liste des positions des tuiles actuellement actives pour la plage de mouvement.

**Méthodes principales :**

* **Méthodes publiques** :
  + void Activate(Grid \_grid) :
    - Initialise les références nécessaires pour le système de déplacement.
    - Configure le pathfindingAStar et la liste des tuiles actives.
  + void ClearActiveTiles() :
    - Efface toutes les tuiles actuellement affichées comme actives en les supprimant du système de tuiles.
  + void MoveUnit(Unit \_unit, Vector3 \_targetPosition) :
    - Déplace l'unité vers une position cible en utilisant le pathfinding pour calculer le chemin.
    - Convertit les positions de cellule en positions du monde et ordonne à l'unité de se déplacer le long du chemin trouvé.
  + void DisplayMoveRange(int \_unitMobility, Vector3 \_unitGroundCellPosition) :
    - Affiche la plage de mouvement possible pour une unité en fonction de sa mobilité.
    - Vérifie la marche des tuiles et les marque comme actives si elles sont accessibles.
* **Méthodes privées** :
  + (Aucune méthode privée n'est définie dans ce script, mais il peut y avoir des méthodes internes utilisées dans le code qui ne sont pas explicitement listées ici.)

**Remarques :**

* **Optimisation** :
  + La méthode DisplayMoveRange calcule la plage de mouvement en utilisant une approche de grille, ce qui peut être optimisé pour éviter les vérifications redondantes ou inutiles des tuiles. La vérification de la marche (Ground.VerifieWalkability) pourrait également être optimisée si elle est appelée fréquemment.
* **Gestion des exceptions** :
  + Le script ne contient pas de gestion des exceptions explicite. Il pourrait être utile d'ajouter des vérifications supplémentaires pour éviter des erreurs en cas de données invalides ou de conditions inattendues.
* **Considérations supplémentaires** :
  + Assurez-vous que les appels à Ground.VerifieWalkability dans DisplayMoveRange sont efficaces, surtout si la méthode implique des opérations coûteuses comme les raycasts.
  + La gestion des positions de cellule (Vector3Int) et des positions du monde (Vector3) doit être cohérente pour éviter des erreurs de conversion.

Ce document fournit une vue d'ensemble et des détails sur le fonctionnement du script MoveSystem, expliquant comment il gère les déplacements des unités, la visualisation des plages de mouvement et l'interaction avec le système de tuiles.