**Document Technique pour GameManager.cs**

**Nom du script** : GameManager

**Namespace** : GamePlay

**Vue d'ensemble :**

Le script GameManager centralise la gestion du jeu, coordonnant les systèmes d'entrée, de caméra, d'interface utilisateur, de mouvement, de combat, et de gestion des tours. Il gère les événements principaux liés aux unités, aux phases de jeu, et aux interactions de l'utilisateur.

**Composants clés :**

* **Événements statiques** :
  + OnUnitAttackButtonWasClicked : Déclenché lorsque le bouton d'attaque est cliqué.
  + OnUnitWaitButtonWasClicked : Déclenché lorsque le bouton d'attente est cliqué.
  + OnDisplayUnitActionAborded : Déclenché lorsque l'affichage des actions d'unité est annulé.
* **Références sérialisées** :
  + board : Instance de la classe Board pour gérer la génération et le placement des unités.
  + inputHandler : Instance de la classe InputHandler pour gérer les entrées utilisateur.
  + uIController : Instance de la classe UIController pour gérer l'affichage de l'interface utilisateur.
  + cameraManager : Instance de la classe CameraManager pour gérer le basculement entre les caméras.
  + moveSystem : Instance de la classe MoveSystem pour gérer les déplacements des unités.
  + tileSystem : Instance de la classe TileSystem pour gérer les tuiles sur la carte.
  + fightSystem : Instance de la classe FightSystem pour gérer les combats entre les unités.
  + turnBaseSystem : Instance de la classe TurnBaseSystem pour gérer les phases de jeu au tour par tour.
* **Variables privées** :
  + activeUnit : Référence à l'unité actuellement active.
  + currentCameraController : Référence au contrôleur de caméra actuellement actif.

**Méthodes principales :**

* **Méthodes Unity** :
  + Awake() :
    - S'abonne aux événements des systèmes d'unités, de gestion des tours, et d'entrées utilisateur.
  + OnDestroy() :
    - Se désabonne des événements pour éviter les références nulles après la destruction du script.
  + Start() :
    - Initialise et active les différents systèmes du jeu (génération de la carte, placement des unités, systèmes de mouvement et de combat, interface utilisateur, gestion des tours).
* **Gestion des phases de jeu** :
  + OnTurnPhaseUpdated(Commander \_phase) :
    - Change la caméra active en fonction de la phase de jeu.
    - Actualise les références dans inputHandler et met à jour l'affichage de l'interface utilisateur pour indiquer la phase actuelle.
* **Gestion des actions d'unités** :
  + DisplayUnitMoveRange(Unit \_unit, Vector3 \_unitGroundCellPosition) :
    - Affiche la plage de mouvement de l'unité sélectionnée.
    - Effectue un zoom sur l'unité et affiche des conseils sur l'interface utilisateur.
  + MoveUnit(Vector3 \_targetPosition) :
    - Déplace l'unité active à la position cible si l'unité appartient au joueur en cours.
  + OnEscapeKeyForCancelPressed() :
    - Annule l'action en cours, réinitialise l'affichage de la plage de mouvement, effectue un dézoom et affiche les conseils principaux.
  + DisplayUnitActions(Unit \_unit) :
    - Affiche les actions disponibles pour l'unité active si elle appartient au joueur en cours.
  + OnUnitWasMoved() :
    - Affiche les actions disponibles après le déplacement de l'unité et détecte les ennemis à proximité.
  + HadleDisplayUnitActionMenu(Unit unit) :
    - Gère l'affichage du menu d'actions pour l'unité, et lie les boutons d'action aux événements correspondants.