**Document Technique pour UIController.cs**

**Nom du script** : UIController

**Namespace** : GamePlay.UIs

**Vue d'ensemble :**

Le script UIController gère l'affichage des éléments de l'interface utilisateur dans le jeu, y compris les menus de sélection d'unités, les notifications de phase, et les conseils à afficher à l'utilisateur. Il utilise des éléments d'interface tels que les boutons et les panneaux pour fournir des informations et des options aux joueurs.

**Composants clés :**

* **Variables publiques et sérialisées** :
  + GameObject textPrefab : Préfabriqué pour les textes affichés dans l'UI.
  + GameObject panelPrefab : Préfabriqué pour les panneaux de notification.
  + GameObject buttonPrefab : Préfabriqué pour les boutons d'action.
  + Transform unitActionMenuContainer : Conteneur pour les boutons d'action des unités.
  + Transform topLeftContainer : Conteneur pour le texte d'aide en haut à gauche.
  + Transform bottomContainer : Conteneur pour les textes d'aide en bas.
  + Transform phaseNoticeContainer : Conteneur pour les notifications de phase.
* **Variables privées** :
  + Text topLeftHelpText : Référence au texte d'aide en haut à gauche.
  + Text bottomLeftHelpText : Référence au texte d'aide en bas.

**Méthodes principales :**

* **Méthodes publiques** :
  + public void Activate() :
    - Initialise l'UI avec les conseils principaux.
  + public void DisplayPhaseNotice(Commander \_phase) :
    - Affiche une notification de phase et met à jour le texte d'aide en haut à gauche.
  + public void DisplayUnZoomTips() :
    - Affiche les conseils pour les actions de dézoom.
  + public void DisplayMainTips() :
    - Affiche les conseils principaux pour les actions disponibles.
  + public Dictionary<UnitAction, Button> DisplayUnitActions() :
    - Crée et affiche les boutons pour les actions des unités et retourne un dictionnaire des actions avec leurs boutons correspondants.
* **Méthodes privées** :
  + IEnumerator DisplayPhaseNotice(Commander \_phase, float time) :
    - Affiche un panneau de notification de phase pendant un temps donné avant de le supprimer.
  + void SetTopLeftText(string text) :
    - Met à jour le texte d'aide en haut à gauche.
  + void SetBottomHelpTexts(string[] texts) :
    - Affiche les textes d'aide en bas.
  + void ClearContainer(Transform container) :
    - Efface tous les éléments enfants du conteneur spécifié.

**Remarques :**

* **Instanciation et Détruit** :
  + Les objets textPrefab, panelPrefab, et buttonPrefab sont instanciés dynamiquement pour créer des éléments de l'interface utilisateur. Assurez-vous que ces préfabriqués sont correctement configurés dans l'éditeur Unity avec les composants requis.
* **Gestion des Textes** :
  + SetTopLeftText et SetBottomHelpTexts mettent à jour les textes d'aide affichés. Ces méthodes utilisent le composant Text des préfabriqués pour afficher les messages.
* **Affichage Dynamique** :
  + DisplayPhaseNotice utilise une coroutine pour afficher une notification pendant une durée spécifique avant de la retirer. Cela permet de créer des animations ou des messages temporaires à l'écran.
* **Actions des Unités** :
  + DisplayUnitActions crée des boutons pour les actions disponibles des unités, tels que "ATTACK" et "WAIT". Ces boutons sont ajoutés au conteneur des actions des unités et associés à leurs actions respectives via un dictionnaire.
* **Performance et Nettoyage** :
  + ClearContainer est utilisé pour effacer les éléments d'un conteneur avant d'en ajouter de nouveaux, ce qui aide à gérer l'affichage dynamique et à éviter les accumulations d'éléments UI.

Ce document fournit un aperçu détaillé des fonctionnalités et de l'organisation du script UIController, expliquant comment il gère l'affichage des éléments d'interface utilisateur dans le jeu. Assurez-vous que tous les préfabriqués sont correctement configurés et que les conteneurs sont correctement assignés dans l'éditeur Unity pour garantir le bon fonctionnement du script.