

UI, prefabs et composants récurrents

Frontière

Cette page décrit uniquement les briques UI réellement communes. Les écrans Data-Driven Accounting vont dans By Module; les écrans de cycle annuel vont dans By Year; les prefabs propres à un jeu restent dans sa page game-specific.

Dossiers

- Assets/Src/UI/ : base commune des contrôles;
- Assets/Src/UI/SimByModule/ : spécialisation By Module;
- Assets/Src/UI/SimByYear/ : spécialisation By Year;
- Assets/GameSpecific/<gameUid>/Resources/Prefabs/UI/ : prefabs propres au jeu.

Composants communs importants

Famille	Classes représentatives
Navigation / panels	MainControlStudent, MainControlTeacher, MainContentPanel, SimPanel, SubPanelBase
Popups / wizards	WizardPopupControlBase, WizardPageControlBase, YesNoPopupControl, popups d'édition
Inputs	FormattedTMPInputFieldBase, inputs nombres/percentages/positifs, date picker, autocomplete
Sélecteurs	SelectorControl, grouped/non-grouped, dropdown overlays, toggles
Profil / équipes	ProfilePopupControl, StudentCardControl, GroupStudentsControl, GroupTeamsControl

Famille	Classes représentatives
Communication	<code>ChatPopupControl</code> , <code>NotificationsListPopupControl</code> , <code>toasters</code>
Scénario	<code>ScenarioPanelControlBase</code> , <code>ScenarioPopupControl</code> , <code>ScenarioItemControl</code>
Tooltips / tables	Tooltip base, table rows/cells, scroll helpers et layout helpers

Règle de prefab

Un prefab commun doit rester générique. S'il commence à contenir des références spécifiques à Accounting, MK1, HR1, etc., créer/maintenir une variante dans le dossier approprié plutôt que d'ajouter des branches `GameUid` au prefab commun.

Diagnostic UI

1. Identifier le prefab instancié.
2. Repérer le contrôle C# attaché.
3. Remonter vers sa classe Base.
4. Identifier le controller/model qui fournit les données.
5. Vérifier `TextLoader` si le souci est textuel.
6. Vérifier le message RTT si l'UI est seulement désynchronisée.

Créer un composant UI commun de A à Z

1. Vérifier qu'il est réellement commun

Si le contrôle connaît un `modulePageUid`, un segment MK, un employee HR ou une règle ENT, il n'est probablement pas commun. Le mettre dans la famille ou le `GameSpecific` concerné.

2. Créer le script de contrôle

Placer le C# sous `Assets/Src/UI/` dans le dossier logique existant. Exposer les références Unity avec `[SerializeField]` lorsque possible plutôt que des `Find()` runtime.

3. Créer/modifier le prefab

Le prefab réutilisable doit vivre dans les Resources/prefabs communs adaptés. Ne pas modifier une instance de scène sans propager les changements au prefab.

4. Brancher les textes

Utiliser `TextLoader` ou `Game.Texts` ; ne pas hardcoder un texte utilisateur dans le prefab. Ajouter FR/EN via le pipeline Textes.

5. Ajouter les events/listeners

Les listeners de boutons doivent être ajoutés dans `Start/OnEnable` selon le lifecycle du contrôle et retirés dans `OnDestroy/OnDisable` lorsqu'ils ciblent des events statiques/globaux.

6. Si le composant doit être ciblable par un tutoriel

Ajouter `TutorialUIObject` ou `TutorialUIButton` au prefab avec un UID stable. Voir la page **Tutoriels — création de A à Z et architecture**.

7. Si la donnée est live

L'UI ne devrait pas écouter BrainCloud directement. Le controller reçoit le RTT, modifie le modèle, puis l'UI observe l'event/controller.

Checklist prefab

- références sérialisées non null;
- anchors/Canvas adaptés aux résolutions WebGL;
- TextLoader/UIDs présents;
- aucune dépendance game-specific cachée;
- listeners nettoyés;

- état initial correct lors d'une réouverture du popup/panel;
- test dans prefab variant et scene réelle;
- build si prefab/asset modifié.

Debug UI

Si l'UI affiche une mauvaise valeur : inspecter d'abord le modèle/controller. Si le modèle est correct, inspecter le binding/control. Si le modèle est faux seulement en live, suivre le RTT. Si le contrôle fonctionne en Editor mais pas WebGL, vérifier les références sérialisées et les exceptions dans la console navigateur.

Revision #2

Created 2026-09-18 19:10:41 UTC by Mathis

Updated 2026-09-18 20:10:21 UTC by Mathis