

Création d'une simulation (acc1)

Étape préliminaire pour la création de la simulation

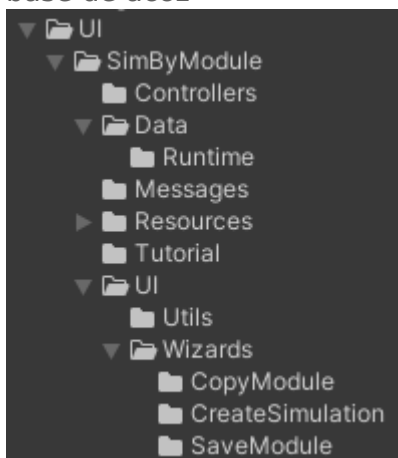
Mettre à jour les taxes

- Avant de commencer, faire une mise à jour manuel des taxes si c'est pertinent pour l'année actuel:

<https://wiki.uniksim.com/books/programmeur/page/proc%C3%A9dure-annuel-pour-updater-les-taxes>

Rendre la simulation général

- Si nécessaire, rendre la simulation original (copié) plus général afin de simplifier la nouvelle simulation.
 - Si c'est le cas, le dossier **{allsims}/UI/SimByModule** est créé pour la base de acc1



- Pour se faire, les fichiers modifiés doit avoir un namespace **namespace UnikSim.SimByModule** au lieu de **namespace UnikSim.SimByModule.acc1**

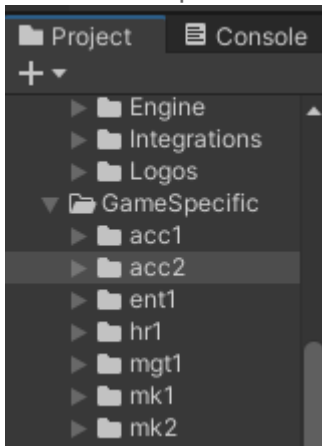
- Si le fichier n'est pas dans gameSpecific, il n'est nécessaire le déplacer
- Le nom du fichier doit aussi faire référence à "ByModule" (Ex: StaticDataAcc1 => StaticDataByModule) puisqu'il sera généralisé
 - Si le fichier n'a pas de acc1 dans son nom, il n'est pas nécessaire de le rename
- Les fichiers doivent aussi se faire déplacer/transférer à **SimByModule** dans le dossier correspondant

Dupliquer une simulation intérieure

- Avant de commencer, créer une branche au nom de la nouvelle simulation sur Git Extensions
 - N'oublie pas de voir les informations supplémentaires à la fin de ce document pour certaines précisions
-

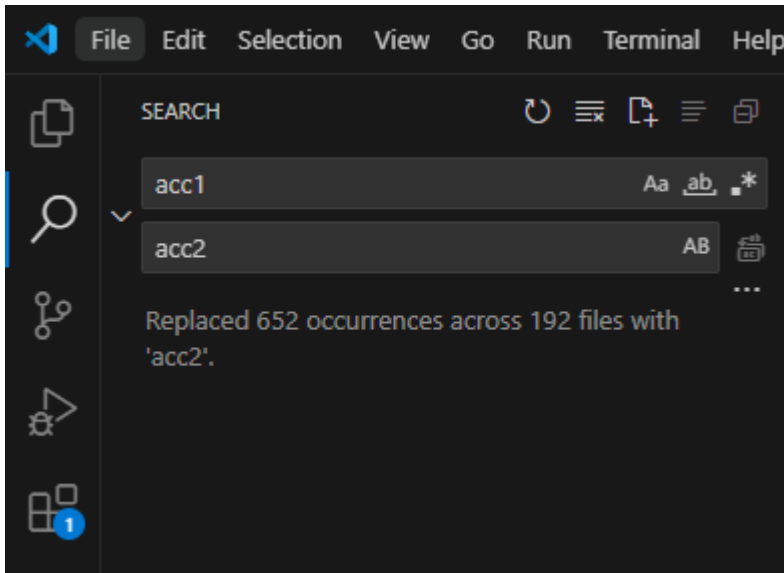
- Sur unity:

1. Faire une copie de **GameSpecific/acc1** (Ex: acc2)



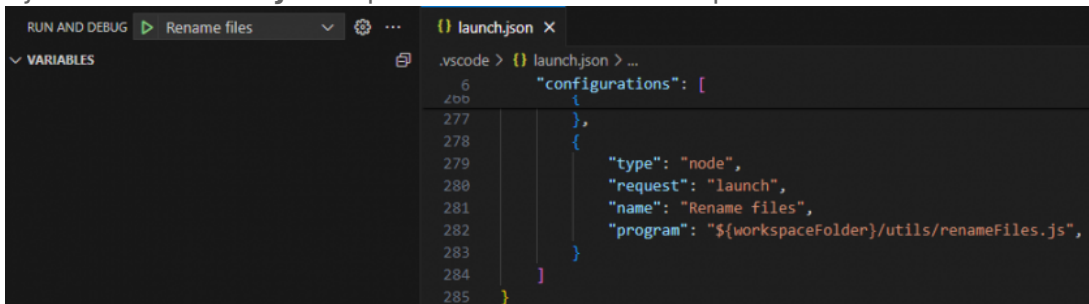
2. Renommer toutes les instances de code lié à l'ancienne simulation (Incluant les noms de fichiers et les codes).
 - **ATTENTION:** Vérifier si les changements n'affecteront pas l'original, **avant** et **après**

modifications



Un script nommé "renameFiles" est disponible pour renommer les noms de fichiers. {allsims}/node.js/utils/renameFiles.js

- Pour utiliser **renameFiles**, il faudra créer une copie temporaire dans le nouveau dossier
- Ajouter un **launch.js** temporaire avec les mêmes paramètres



- Puis modifier les paramètres suivant:

```
let startingDirectory = './'; // First directory to search
let find = "acc1"; // Value to search
let replace = "acc2"; // Replace value
let onlyRenameDebug = true; // Display only the rename files
let allCase = false; // All case includes (upperCase, lowerCase)
```

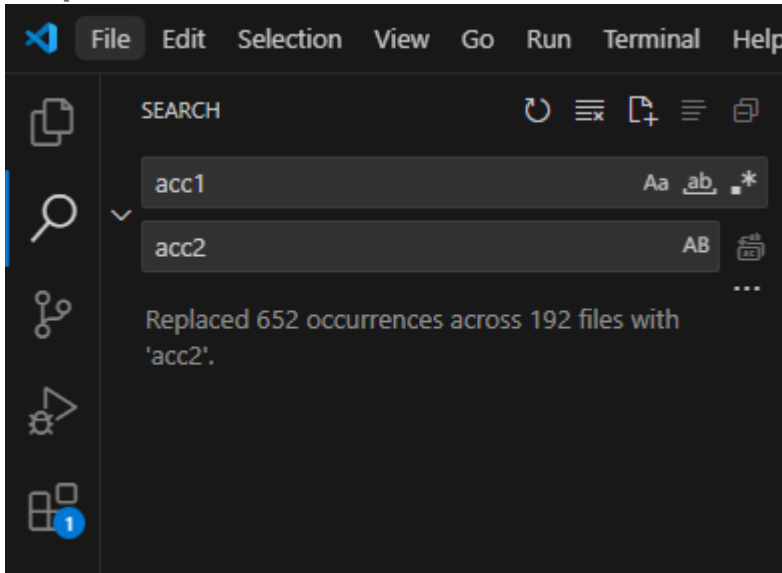
- Note 1: Le **renameFiles** est case sensitive pour **find** et **replace**, il se pourrait qu'il faudrait le faire plus d'une fois
- Note 2: Une fois les refactor des noms faits, il se peut que Unity ne pourra pas compiler. (Certains fichiers possèdent les mêmes noms)
- Pour régler ce problème, il faudra refactor avec **Visual Studio** chacun de ces fichiers ou refaire l'étape Rendre la simulation général.
 - **ATTENTION:** Si vous voulez le faire manuellement, assurer vous de ne pas impacter le code originel (copié)

- Sur node.js:

1. Faire une copie de **{allsims}/node.js/backend/braincloud/server_scripts/acc1** (Ex: acc2)

```
> acc1
> acc2
```

2. Comme à l'étape précédente, renommer toutes les instances de code lié à l'ancien script (Incluant les noms de fichiers et les codes).
 - **ATTENTION:** Vérifier si les changements n'affecteront pas l'original, **avant** et **après** modifications



Un script nommé "renameFiles" est disponible pour renommer les noms de fichiers. {allsims}/node.js/utils/renameFiles.js

- Pour utiliser **renameFiles**, il faudra modifier les paramètres suivant:

```
let startingDirectory = './'; // First directory to search
let find = "acc1"; // Value to search
let replace = "acc2"; // Replace value
let onlyRenameDebug = true; // Display only the rename files
let allCase = false; // All case includes (upperCase, lowerCase)
```

- Note 1: Le **renameFiles** est case sensitive pour **find** et **replace**, il se pourrait qu'il faudrait le faire plus d'une fois

Modification et ajout pour la nouvelle simulation

- Ajouter tous les switch case:

- Faites un **Ctrl + Shift + F** ensuite "acc1" pour trouver toutes les instances de acc1
- Faites une copie de chaque case avec l'acronyme de votre nouvelle simulation.
-

Par exemple :

```
case SharedConstants.GameUid.eGameUid.eAcc1:
    return new TeamByModuleController();
case SharedConstants.GameUid.eGameUid.eAcc2:
    return new TeamByModuleController();
```

- Note : Il ne sera pas nécessaire de faire de nouvelle fichier de traduction. En acc_X, le même fichier sera utilisé

- Modifier la class static pour StaticDataByModule:

- Le fichier "StaticDataAccX" (Renommer par l'étape précédent), devra être modifier afin d'identifier le fichier

```
using UnikSim;
using UnikSim.SimByModule;
using UnityEngine;

namespace UnikSim.SimByModule.Acc2
{
    1 reference
    public class StaticDataAcc2 : StaticDataByModule
    {
        9 references
        public override string GameUid { get { return SharedConstants.GameUid.ACC2; } }
    }
}
```

- Ajouter la class static à **CreateStaticData** dans Factory

```
10     1 reference
11     public static StaticDataBase CreateStaticData()
12     {
13         switch (Game.Instance.CurrentGameUidEnum)
14         {
15             case SharedConstants.GameUid.eGameUid.eEnt1:
16                 return new SimByYear.Ent1.StaticDataEnt1();
17             case SharedConstants.GameUid.eGameUid.eMk1:
18                 return new SimByYear.Mk1.StaticDataMk1();
19             case SharedConstants.GameUid.eGameUid.eHR1:
20                 return new SimByYear.Hr1.StaticDataHr1();
21             case SharedConstants.GameUid.eGameUid.eMk2:
22                 return new SimByYear.Mk2.StaticDataMk2();
23             case SharedConstants.GameUid.eGameUid.eMgt1:
24                 return new SimByYear.Mgt1.StaticDataMgt1();
25             case SharedConstants.GameUid.eGameUid.eAcc1:
26                 return new SimByModule.Acc1.StaticDataAcc1();
27             case SharedConstants.GameUid.eGameUid.eAcc2:
28                 return new SimByModule.Acc2.StaticDataAcc2();
29             default:
30                 return null;
31         }
    }
```

- Ajouter une nouvelle variante de MainControlStudentByModule et MainControlTeacherByModule:

- Si la copie de votre simulation ne contient pas de MainControlStudentAccX ou MainControlTeacherAccX, veuillez suivre les étapes suivantes, sinon passez cette étape
 - Méthode 1: Faire une copie de **MainControlStudentAcc1** et **MainControlTeacherAcc1** dans la nouvelle simulation et le renommer en conséquent

Situé à {allsims}/GameSpecific/acc1/Resources/Prefabs/UI

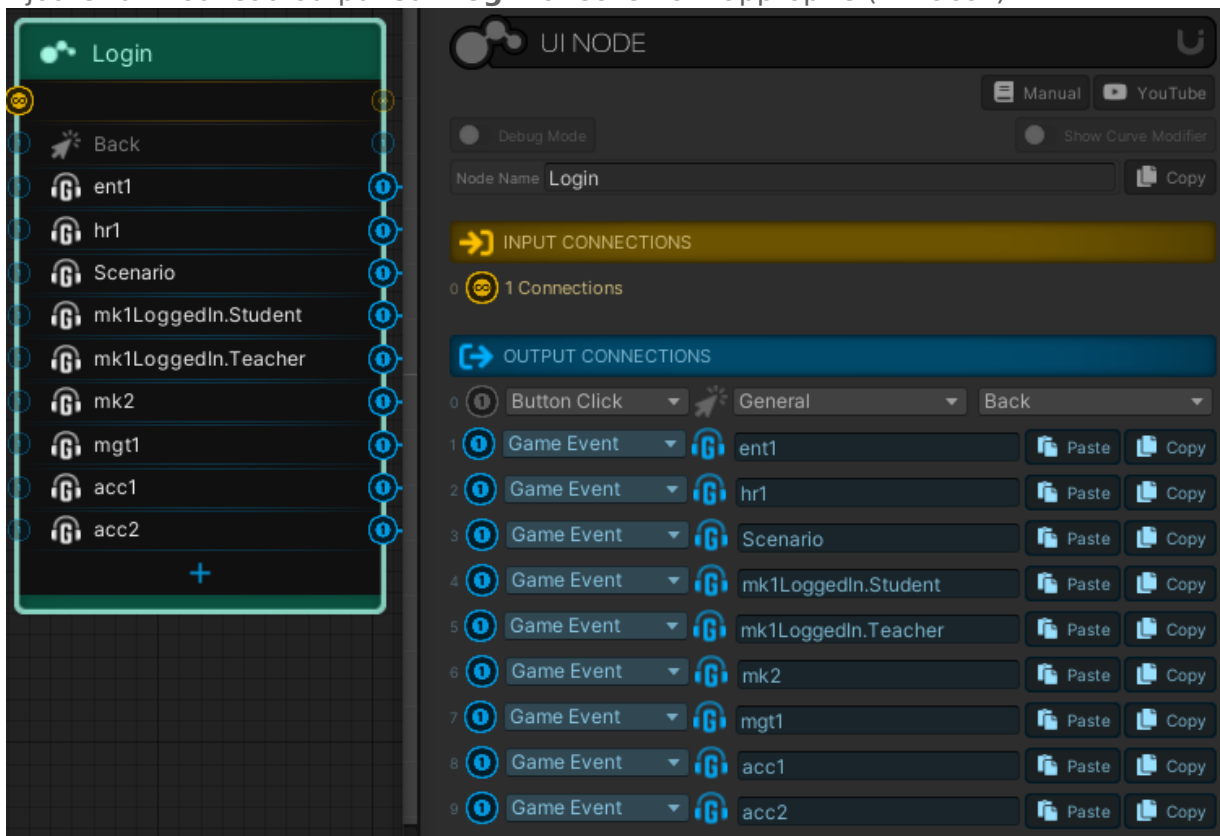
- Méthode 2: Créer une variante avec **MainControlStudentByModule** et **MainControlTeacherByModule**

Situé à {allsims}/UI/SimByModule/acc1/Resources/Prefabs/UI

- Cliquez droite, puis **Create>Prefab Variant**
- Déplacer le fichier au bons endroit et le renommer en conséquent
- Partie Doozy:

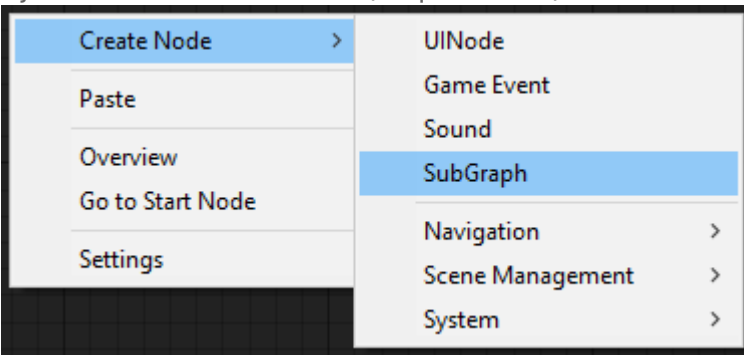
Situé à Resources>MainGraph

- Ajouter un nouveau output sur **Login** avec le nom approprié (Ex: acc2)

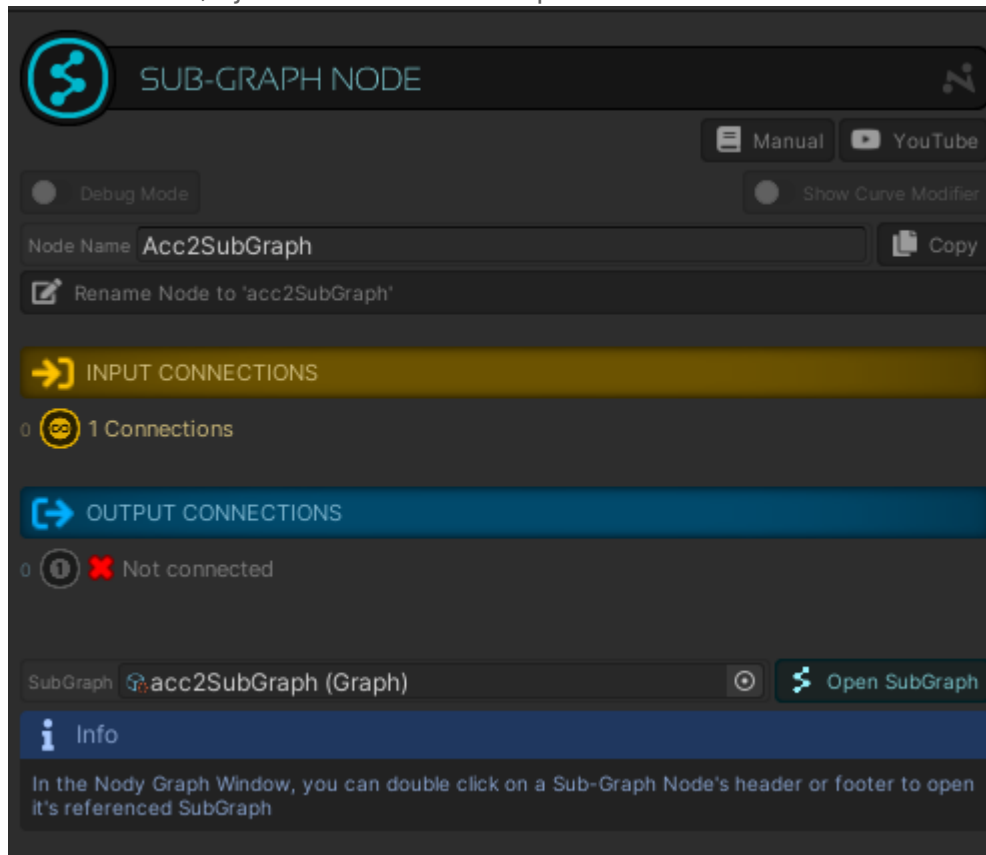


◦

Ajouter un nouveau node (clique droite) avec un nom approprié



- Dans ce node, ajoutez le AccXSubGraph

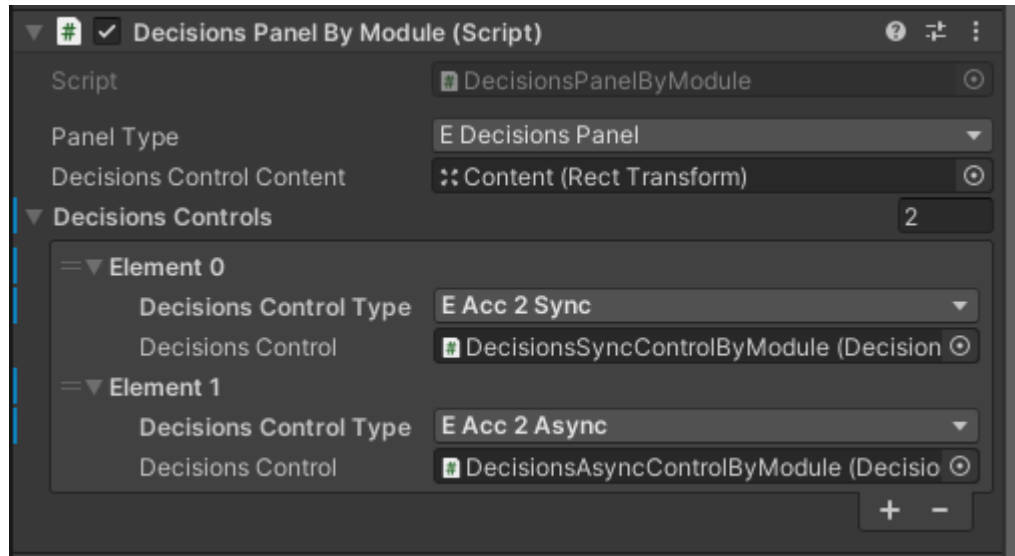


- Connecter le output avec le nouveau node
- Si la duplication c'est bien terminé, il ne sera pas nécessaire de modifier AccX SubGraph
- Modifier les scènes **accXTeacher** et **accXStudent** :

Situé à [GameSpecific/accX/Resources/Scenes](#)

- Dans la section **DecisionsPanel** (Hierarchy), modifier le "Decisions Controls" pour

votre simulation



Modification des thèmes de la simulation

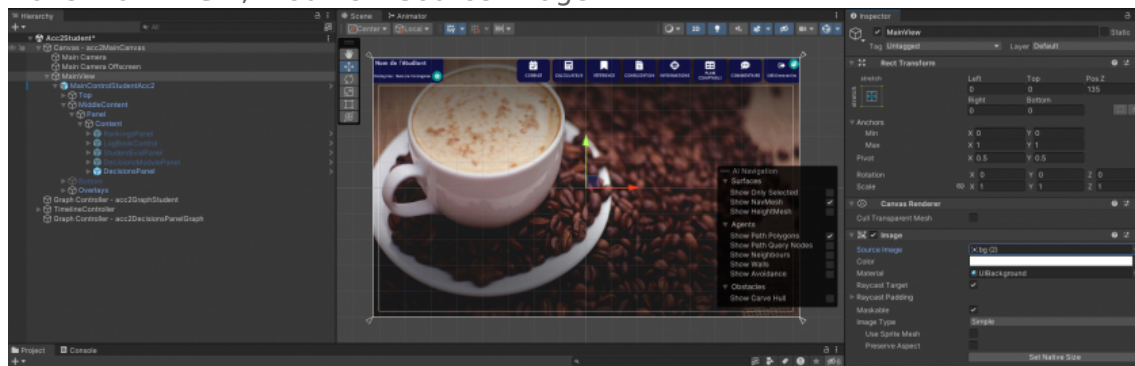
- Remplacer les images de la nouvelle simulation avec le thème choisi :
 - Écran titre : **accX**
 - Écran de jeu : **bg**
 - Note : Il serait peut-être intéressant de demander les images à l'équipe pour éviter des problèmes de choix esthétiques ou de design

Situé à GameSpecific/accX/Resources/images

- Pour modifier **bg**, il faudra aller sur la scène de la nouvelle simulation (**accXTeacher** et **accXStudent**, vous devriez le faire sur les deux scènes)

Situé à GameSpecific/accX/Resources/Scenes

- Dans **MainView**, modifier "Source Image"



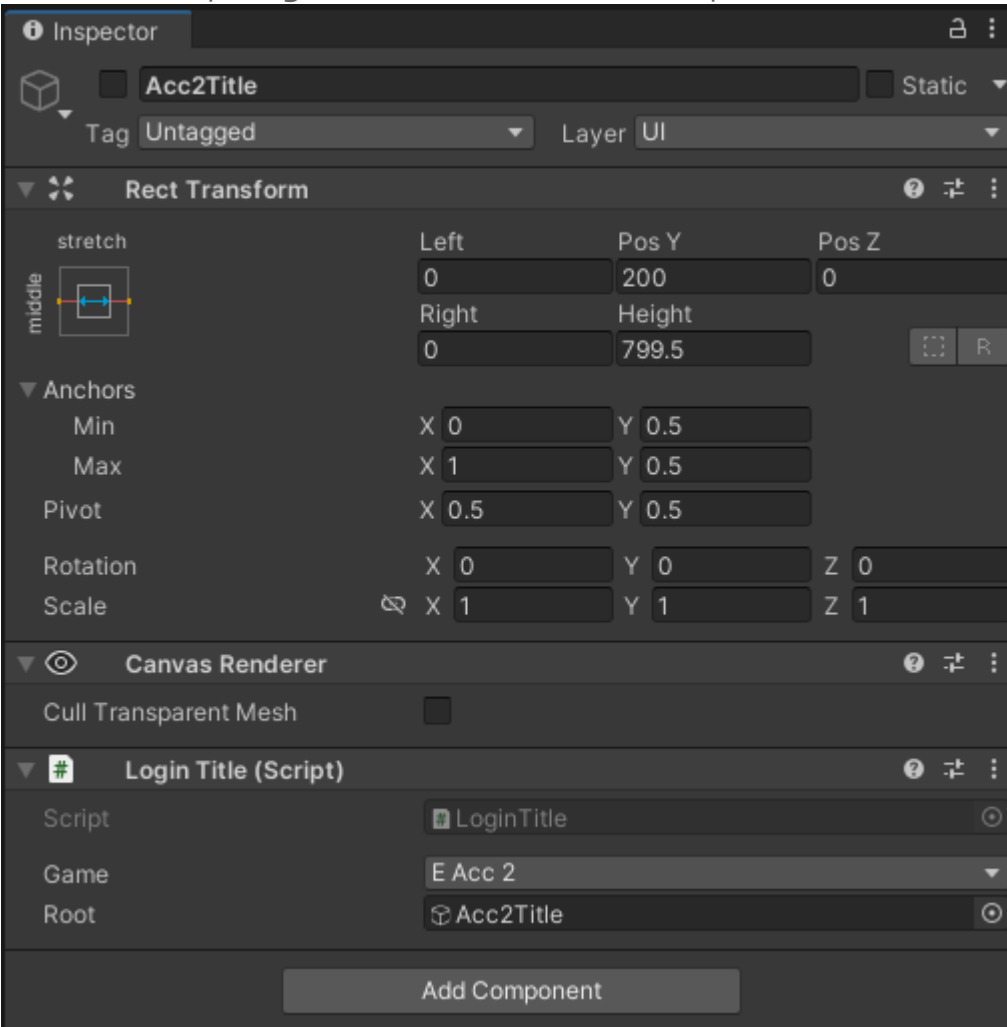
- Ajouter le titre dans **LoginControl**:

◦

Faire une copie de la simulation original (copié) et le renommer



- Modifier le script **LoginTitle** dans le menu de l'inspecteur



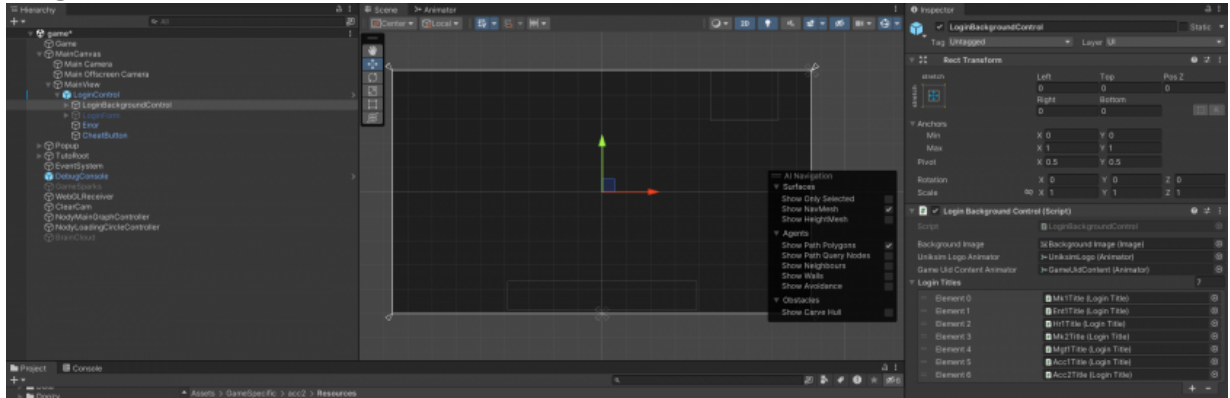
- Ajouter dans le google sheet **shared_text**
 - **accX_theme_label** (la couleur)
 - **accX_login_title** (le titre principal, dans ce cas "Comptabilité X")
 - **accX_login_subtitle** (le nom de la nouvelle simulation)

| | | | |
|-----|---------------------|---------------------|-----------------|
| 984 | acc2_theme_label | Vert | Green |
| 985 | acc2_login_title | Comptabilité 2 | Accounting 2 |
| 986 | acc2_login_subtitle | Les machines à café | Coffee machines |

- Dans **AccXTitle>Title2**
 - Modifier le Text Uid pour **accX_login_title**
- Dans **AccXTitle>Title3**
 - Modifier le Text Uid pour **accX_login_subtitle**
- Faire un **importTextShared** pour voir les changements

Situé à {allsims}/node.js/exports/importTextShared.bat

- Modifier le thème de couleur de la simulation :
 - Sur la scène de **game**, dans **LoginBackgroundControl**, ajouter le "Login Title" dans "Login Titles"



Le login title à ajouter devrait se trouver à GameSpecific/accX/Resources/AccXSkin

- Dans **AccXSkin** :
 - Modifier le gameId
 - Modifier le thème de couleur

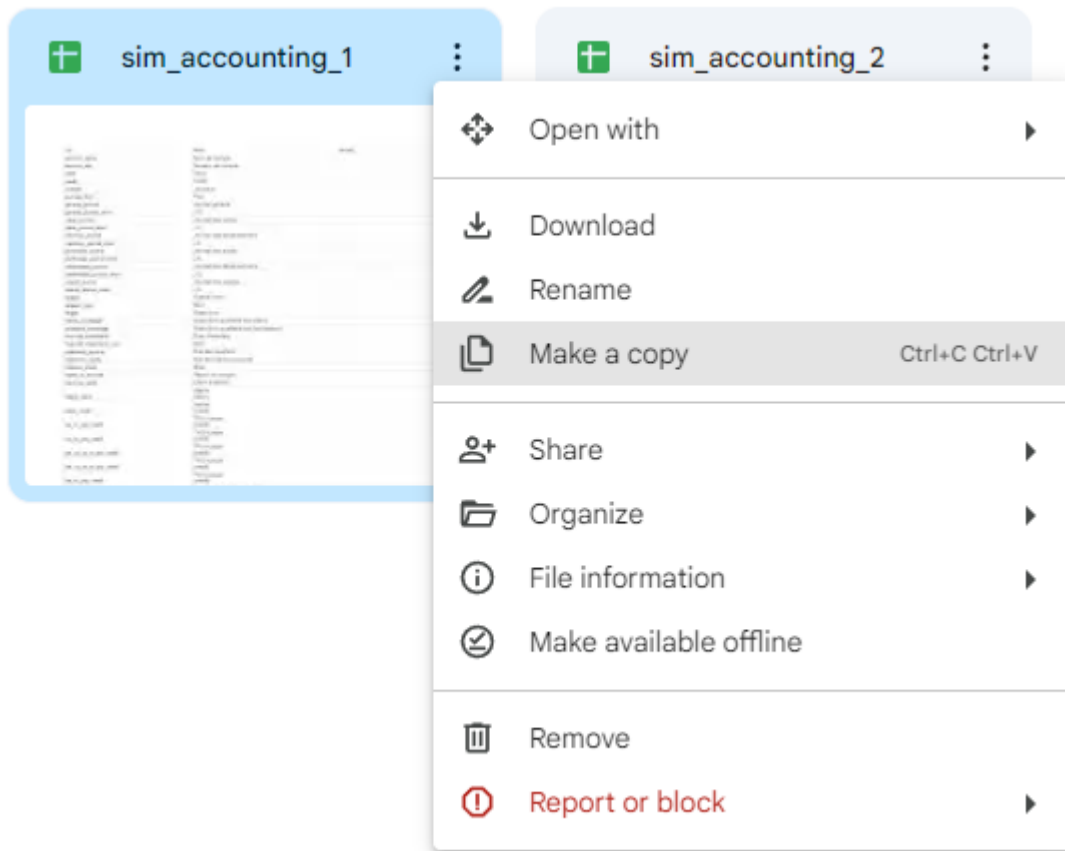
Google Sheet et StaticData

- Faire une copie du googleSheet sim_accounting_1 :

Situé à "Shared With Me/Dev/Games/acc1 - compta_cegep/sim_accounting_1"

◦

Faire une copie du fichier



- Renommer le fichier (modifier le numéro)
- Modifier le fichier si nécessaire
 - Note : Un fichier de base existe pour tous les simulations accounting

sim_byModule

- Modifier les imports de fichiers :
 - Il faudra alors updater **{AllSimsRepository}/node.js/exports/exportAllData.js**

afin de pouvoir exporter les nouveaux document créer avec la structure suivante:

```
149 //region acc2
150 games['acc2'] = {};
151 games['acc2'].sheet = '1otxrFdakMFsjRQU4mzPJCoMD-I1CdS_1P34PukNovwc';
152
153 games.acc2.tabs = {};
154
155 games.acc2.tabs['partners'] = 280929805;
156 games.acc2.tabs['employees'] = 629824938;
157 games.acc2.tabs['companies'] = 726577385;
158 games.acc2.tabs['conditions'] = 1100005961;
159 games.acc2.tabs['products'] = 1478064019;|
160 games.acc2.tabs['services'] = 88761296;
161 games.acc2.tabs['salaries'] = 2068898836;
162
163 games.acc2.additionalData = ['accounts'];
164 //endregion
```

- Si nécessaire, modifier **importAllDataGoogleSheet** avec "withBaseV = true" pour import aussi la base (sim_byModule).

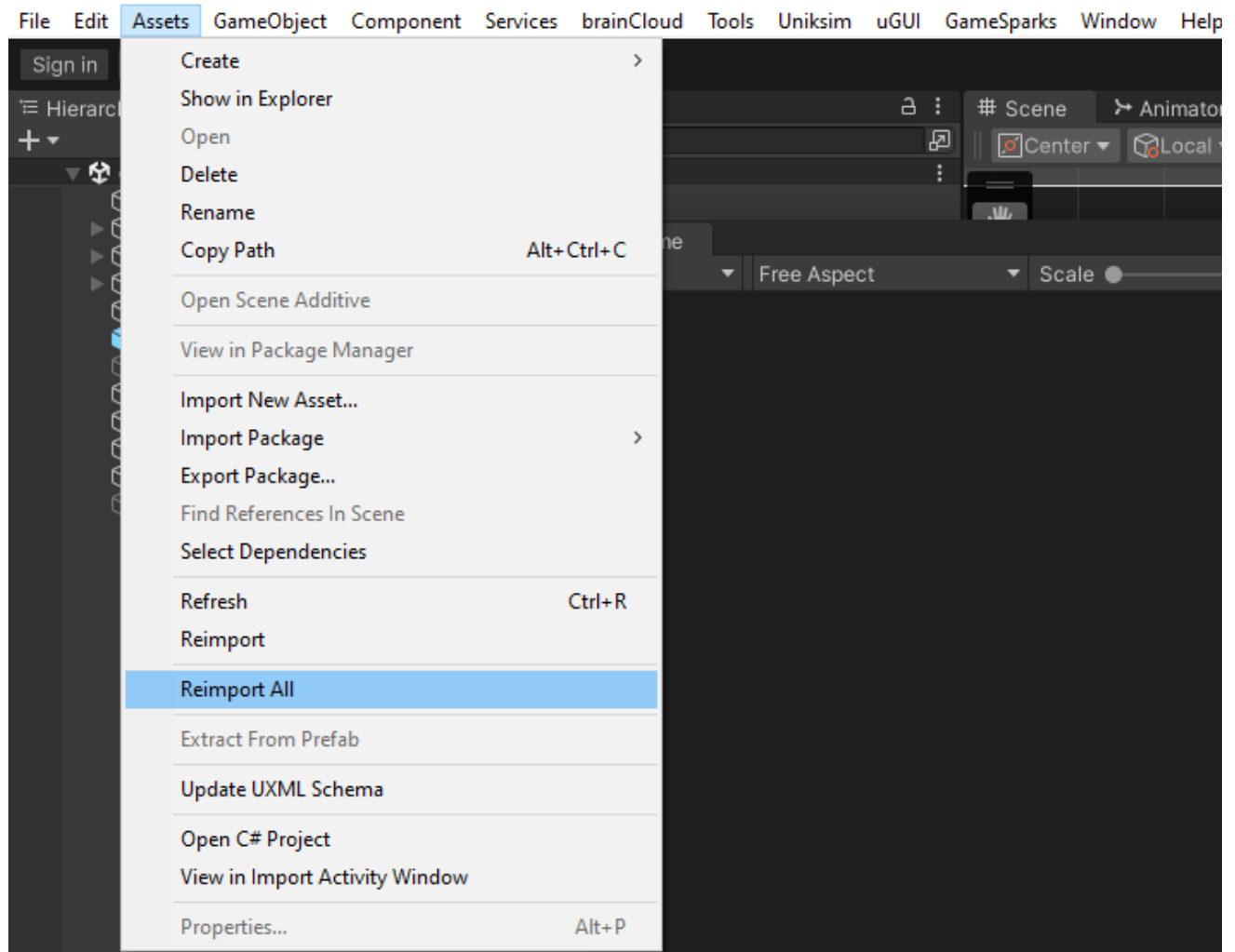
```
407 function importAllDataGoogleSheet(gameUid) {
408     var gameIdPath = games[gameUid].gameId || gameUid;
409     var withBaseV = false;
410     dataPath = '../game/Assets/GameSpecific/' + gameIdPath + '/Resources/data_' + gameIdPath + '/';
411
412     importDataFromGS(gameUid, withBaseV, dataPath, onExportFinished);
413 }
```

- Mettre la première partie de l'url (après "d/") du document **sim_accounting_X**
- Dans les autres plus spécifiques (Ex : partners): Mettre le "gid" de l'url
 - games.accX.tabs = obtenir les valeurs sur googleSheet
 - games.accX.additionalData = les valeurs sont à l'extérieurs du googleSheet (se trouve directement sur Unity)
- Importer les fichiers **sim_accounting_X** pour la nouvelle simulation :
 - À l'aide de la commande "**Google Sheet - Import All**" de node.js avec les valeurs appropriés
- Exporter les fichiers sur Braincloud :
 - À l'aide de la commande "**Google Sheet - Export All Data**" de node.js avec les valeurs appropriés (StaticData)
 - À l'aide de la commande "**Google Sheet - Export All Data**" de node.js avec les valeurs appropriés (StaticDataChunk)

Information supplémentaire

- Note 1 : Les simulations en "accounting" ou "accX" sont tous mis dans une base nommé **byModule** (ex: braincloud>staticData, node.js>...>byModule_utils, UI>SimByModule, etc.)
- Note 2 : Certains/Plusieurs changements risquent de briser Unity. Il faudra alors faire un

"Reimport All"



Revision #54

Created Fri, May 10, 2024 1:37 PM by [Larry](#)

Updated Tue, May 28, 2024 2:40 PM by [Larry](#)