



Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Immersive Serious Game Factory

Manuel d'utilisation

Version du 01.11.2022

Superviseures :

Houda Chabbi Drissi

Sandy Ingram

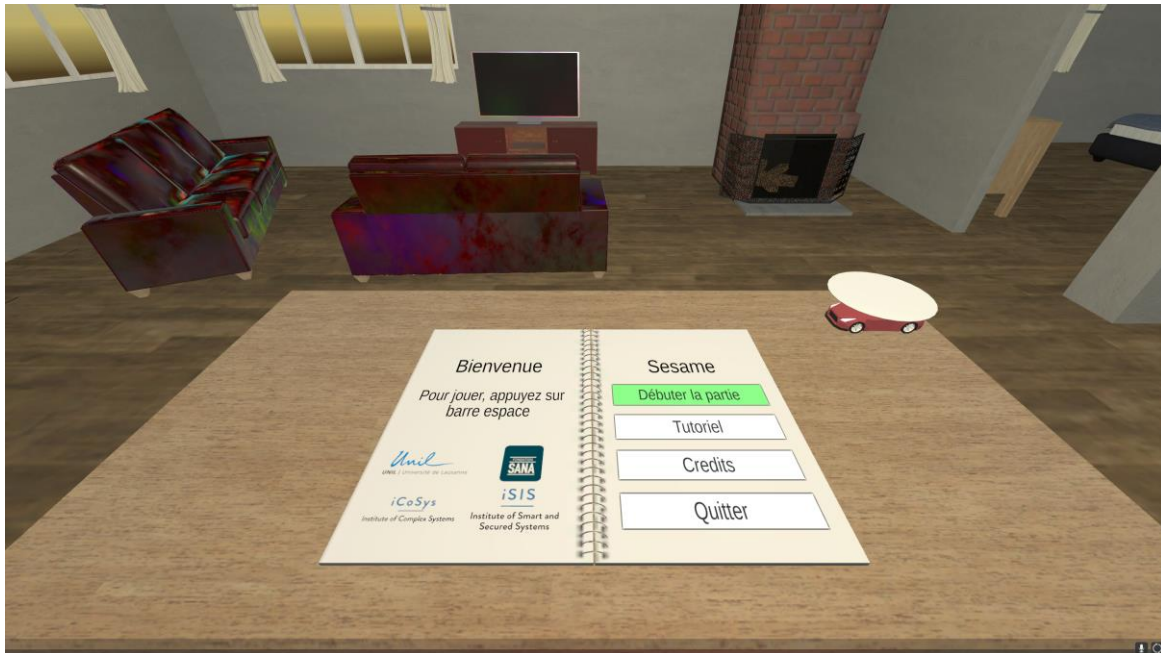
Robin Cherix, Nicolas Baeriswyl

Table des matières

1	Déroulement du jeu.....	3
2	Préparer le jeu.....	6
2.1	Ajouter une nouvelle ressource.....	6
2.1.1	Générer des ressources pour interférences animées ou éléments de décor... 6	6
2.1.2	Ajouter une ressource audio.....	7
2.1.3	Ajouter une image.....	8
2.1.4	Ajouter des fichiers sur Quest.....	8
2.1.5	Ajouter des fichiers sur Hololens.....	8
2.2	Générer les exécutables.....	9
2.2.1	Installation Windows.....	9
2.2.2	Installation Hololens.....	9
2.2.3	Installation Quest.....	10
2.2.4	Quand générer un nouvel exécutable ?.....	10
2.3	Structure d'un fichier d'expérience (JSON).....	10
2.4	Charger un fichier d'expérience.....	11
2.5	Chargement de l'expérience.....	11
2.6	Récupérer un fichier de log.....	12
3	Modifications des touches / boutons.....	13
3.1	Ajouter un nouveau bouton.....	13
3.2	Modifier un bouton.....	14
4	Créer un nouvelle Interférence animée (AssetBundle).....	15
4.1	Interférence partiellement animée.....	15
4.2	Interférence complètement animée.....	18
5	Créer un nouveau niveau.....	23
5.1	Créer une nouvelle variante du Level1.....	23
5.2	Créer un tout nouveau niveau.....	25
6	Créer un nouveau stimuli.....	26
7	Ajouter un nouveau point d'encrage.....	27

1 Déroulement du jeu

Le jeu commence par un menu qui indique au joueur comment jouer ainsi que l'option de jouer au jeu, faire le tutoriel ou quitter l'application.



Lors du jeu, des feuilles blanches à tamponner vont arriver du tas de feuilles présents sur la gauche.

Le joueur doit appuyer sur le bouton indiquant dans le menu (ici barre espace) pour tamponner la feuille.

Le but est de tamponner la feuille le plus vite possible une fois qu'elle s'est arrêtée au milieu de la table.



Mais attention ! Il ne faut pas tamponner la feuille si elle possède un signal Stop ! Ce signal Stop est représenté par un rectangle rouge sur la feuille qui apparaît après un certain temps, appelé Stop Signal Delay (SSD).

Le joueur doit donc réagir le plus vite possible, mais pas trop vite non plus.

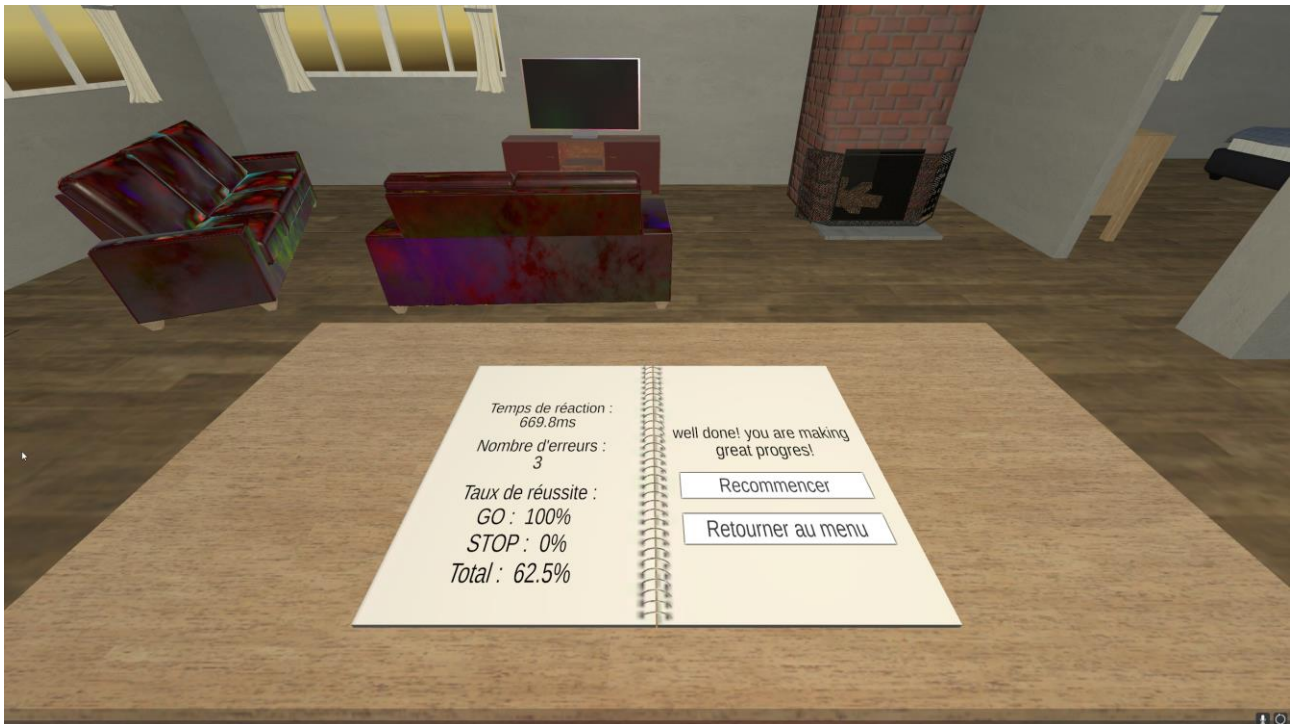
Au cours du jeu, des interférences de différentes natures peuvent venir perturber le joueur afin de le pousser à commettre des erreurs.



Au cours du jeu, des pauses vont survenir pour laisser le joueur se reposer. Ainsi, le joueur peut rester le plus concentré possible tout au long du jeu.

Une fois que le jeu est terminé, le score du joueur s'affiche.

Le score est composé du temps de réaction moyen, en millisecondes, ainsi que le nombre d'erreurs que le joueur a fait, c'est-à-dire le nombre de fois qu'il a tamponné une feuille qui était un signal Stop.



Depuis cet écran, le joueur a la possibilité de rejouer ou retourner au menu.

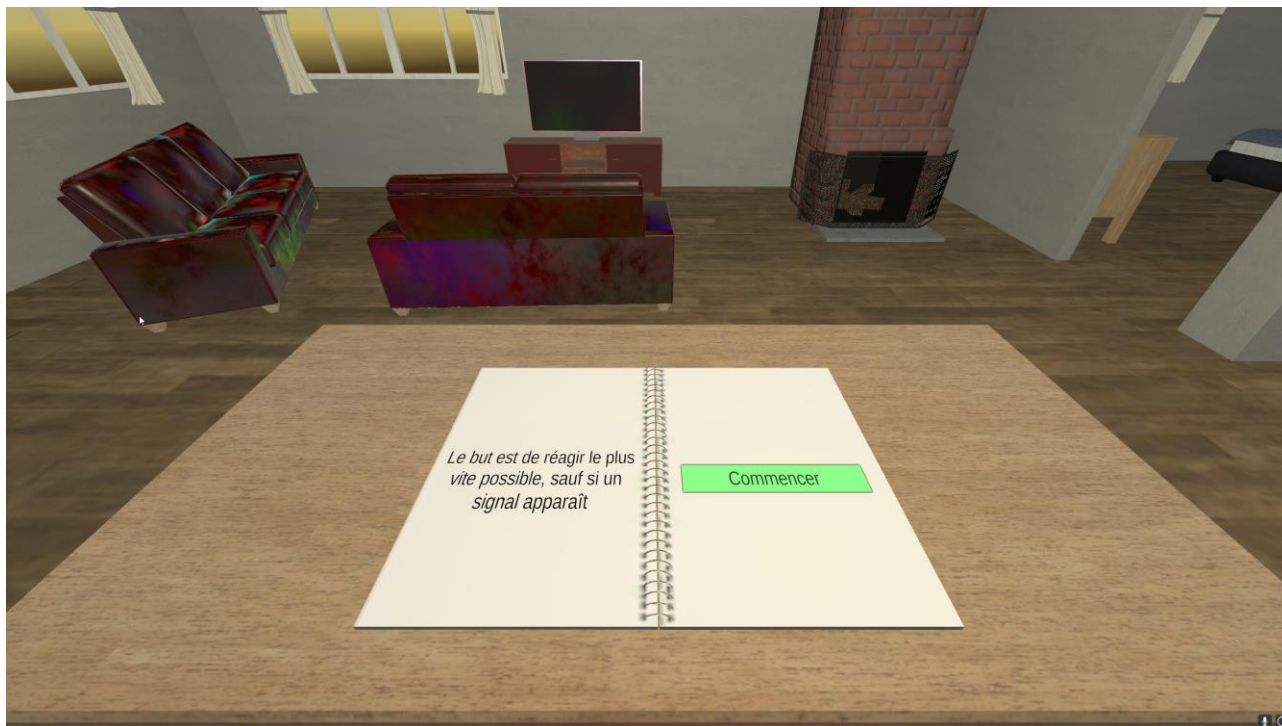
Si le joueur le souhaite, il peut faire le tutoriel qui lui montrera comment jouer.

Cette option est sélectionnable depuis le menu.

Quand le bouton « Tutoriel » est sélectionné par le joueur, un nouveau texte s'affiche pour expliquer au joueur les principes du jeu.

Un bouton « Commencer » permet de démarrer une session de jeu courte et prédéfinie afin que le joueur puisse se familiariser avec le jeu.

A la fin du tutoriel, le joueur revient au menu.



Ce jeu est disponible dans trois versions :

- 2D sur Windows
- Réalité virtuelle sur Oculus Quest 2
- Réalité augmentée sur Microsoft Hololens 2

Si le jeu se déroule en réalité virtuelle ou augmentée, alors le joueur a une manette en main et le bouton sur lequel il doit appuyer est le bouton A.

Des vidéos de gameplay sont disponibles dans le dossier Videos du repository.

2 Préparer le jeu

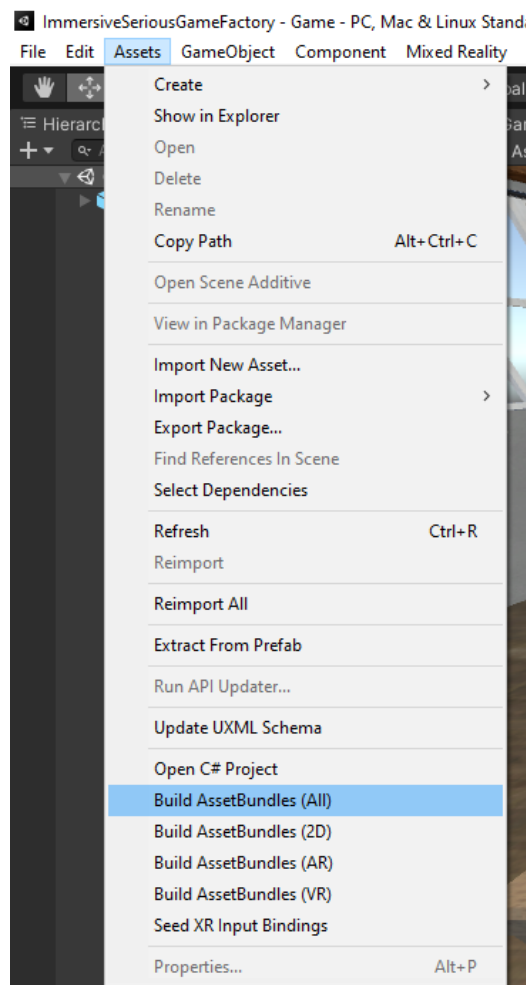
2.1 Ajouter une nouvelle ressource

2.1.1 Générer des ressources pour interférences animées ou éléments de décor

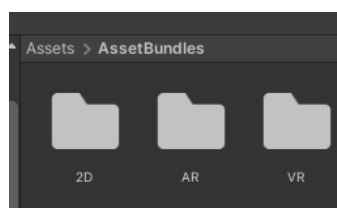
Pour générer une nouvelle ressource, il faut commencer par ouvrir le projet Unity (voir guide d'installation pour l'installer) dans l'éditeur Unity.

Une fois l'éditeur Unity ouvert, créer un nouvel Asset Bundle contenant la nouvelle ressource à intégrer dans le jeu. Pour cela, voir section 2.10 Intégration de ressources post-build du rapport.

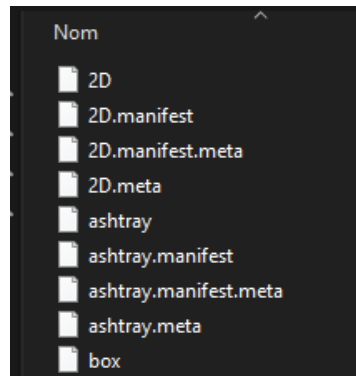
Une fois l'Asset Bundle créé, aller dans le menu Assets et sélectionner Build AssetBundles (All) pour générer le nouvelle Asset Bundle pour toutes les plateformes, ou sélectionner une plateforme spécifique.



L'Asset Bundle est généré dans le dossier AssetBundles du projet Unity. Chaque plateforme dispose d'un dossier où ses Asset Bundles sont placés.



Chaque Asset Bundle génère deux fichiers par plateforme : un fichier sans extension et un fichier .manifest.



C'est le fichier sans extension qui doit être ajouté.

Ce fichier doit être ajouté dans un dossier spécifique selon la plateforme. Le tableau ci-dessous montre où copier le fichier de quel dossier vers quel dossier.

Plateforme	Dossier source	Dossier destination
Windows	AssetBundles/2D	%userprofile%/AppData/LocalLow/HEIAFR/ImmersiveSeriousGameFactory/AssetBundles
Oculus Quest	AssetBundles/VR	storage/emulated/0/Android/data/com.heiafr.immersiveseriousgamefactory/files/AssetBundles
Microsoft Hololens	AssetBundles/AR	%userprofile%/AppData/Local/Packages/ImmersiveSeriousGameFactory/LocalState/AssetBundles

Comme montré dans une vidéo le dossier Videos du repository, il est possible de copier la ressource à l'aide de l'outil de transfert de fichiers.

2.1.2 Ajouter une ressource audio

Pour ajouter une ressource audio, utilisée pour la musique du jeu ou les interférences sonores, il faut copier le fichier audio dans un dossier spécifique selon la plateforme.

Le tableau ci-dessous montre où copier le fichier.

Plateforme	Dossier destination
Windows	%userprofile%/AppData/LocalLow/HEIAFR/ImmersiveSeriousGameFactory/Sounds
Oculus Quest	storage/emulated/0/Android/data/com.heiafr.immersiveseriousgamefactory/files/Sounds
Microsoft Hololens	%userprofile%/AppData/Local/Packages/ImmersiveSeriousGameFactory/LocalState/Sounds

Les fichiers mp3 et wav sont supportés.

2.1.3 Ajouter une image

Pour ajouter une image, utilisée pour les interférences visuelles, il faut copier l'image dans un dossier spécifique selon la plateforme.

Le tableau ci-dessous montre où copier le fichier.

Plateforme	Dossier destination
Windows	%userprofile%/AppData/LocalLow/HEIAFR/ImmersiveSeriousGameFactory/Visuals
Oculus Quest	storage/emulated/0/Android/data/com.heiafr.immersiveseriousgamefactory/files/Visuals
Microsoft Hololens	%userprofile%/AppData/Local/Packages/ImmersiveSeriousGameFactory/LocalState/Visuals

Les fichiers jpg, png, bmp, tif sont supportés.

2.1.4 Ajouter des fichiers sur Quest

Pour ajouter une ressource sur Oculus Quest, il faut brancher le casque à l'ordinateur, mettre le casque sur la tête et accepter la connexion avec l'ordinateur.

<https://store.facebook.com/en-gb/help/quest/articles/headsets-and-accessories/using-your-headset/transfer-files-from-computer-to-headset/>

Puis accéder au système de fichiers du casque via l'explorateur de fichier Windows.

La vidéo ISGF-upload-config.mp4 dans le dossier Videos du repository montre la procédure de charger une configuration. La procédure d'ajouter des autres fichiers est identique, il faut simplement choisir quel type de ressource on aimerait ajouter.

2.1.5 Ajouter des fichiers sur Hololens

Pour ajouter une ressource sur Microsoft Hololens, il faut connecter le casque à l'ordinateur, mettre le casque et passer par le Windows Device Portal.

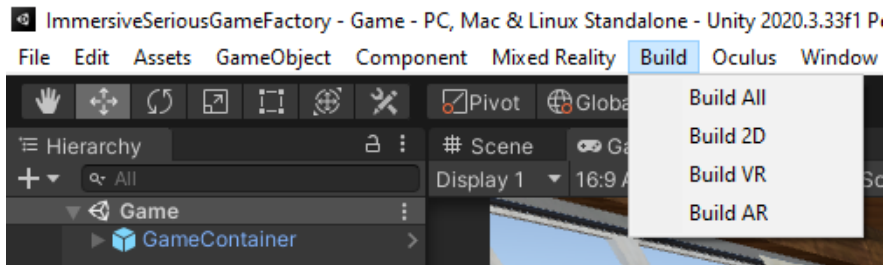
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/using-the-windows-device-portal>

Les vidéos 0-Connect-Hololens.mp4 et 1-Hololens-upload-resources.mkv dans le dossier Videos/AR du repository montre la procédure.

2.2 Générer les exécutables

Attention : Les AssetBundles (Niveau, Stimuli et Interférence Animée) et les exécutables doivent être générés sous la même version (mineur) de Unity.

Pour générer des nouveaux exécutables, il faut ouvrir le projet dans l'éditeur Unity et sélectionner Build > Build All, ou sélectionner une plateforme spécifique



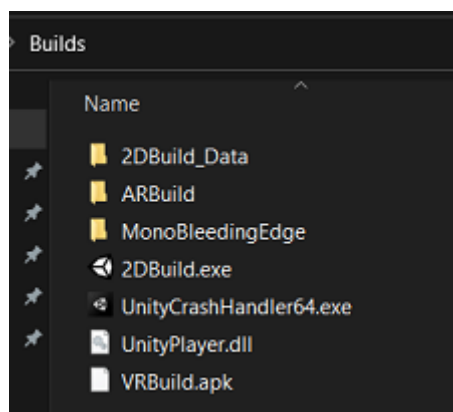
Il faut ensuite donner un dossier de destination où stocker les exécutables.

La vidéo ISGF-generate-builds.mkv dans le dossier Videos du repository montre la procédure.

2.2.1 Installation Windows

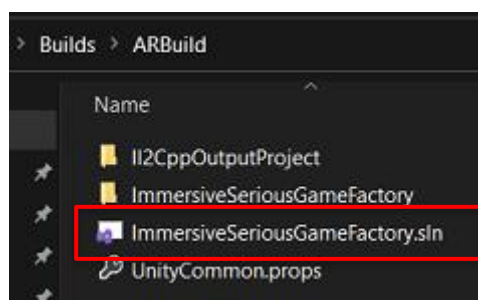
La version Windows ne demande pas d'installation particulière.

Il faut juste ouvrir le dossier de destination et double-cliquer sur 2DBuild.exe



2.2.2 Installation Hololens

Pour déployer l'exécutable sur Hololens, il faut connecter le casque à l'ordinateur par USB, puis ouvrir la solution (.sln) qui se trouve dans le dossier ARBuild en double-cliquant dessus.



A partir de là, suivre la documentation de Microsoft : <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/using-visual-studio?tabs=hl2>

La vidéo 0-Hololens-Install.mkv dans le dossier Videos/AR du repository montre la procédure.

2.2.3 Installation Quest

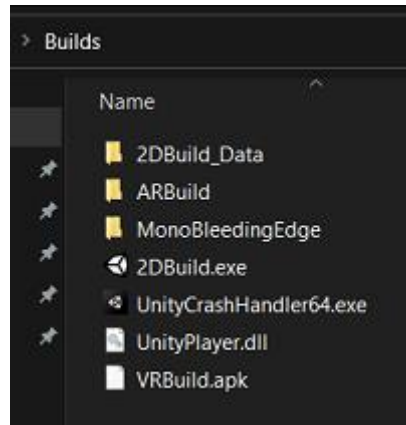
Pour installer l'application sur le Quest, suivre la documentation de Oculus :

<https://developer.oculus.com/documentation/native/android/mobile-device-setup/>

et

<https://developer.oculus.com/documentation/unity/unity-enable-device/>

Et installer le fichier VRBuild.apk sur le Quest.



La vidéo 0-Quest-Install.mkv dans le dossier Videos/VR du repository montre la procédure.

2.2.4 Quand générer un nouvel exécutable ?

Il est nécessaire de générer un nouvel exécutable lorsque :

- Un nouveau binding de bouton est ajouté à l'action du joueur
- Le comportement d'un module doit être modifié
- La scène par défaut doit être modifiée

Pour les cas suivants, il est plus facile de créer un nouveau niveau en copiant le niveau par défaut (suivre les instructions du chapitre 4).

- Un ou plusieurs éléments du décor doivent être supprimés ou déplacés
- Un ou plusieurs points d'ancrages doivent être ajoutés ou déplacés
- Un ou plusieurs distracteurs doivent être déplacés, ajoutés ou supprimés
- Le stimulus ou le plateau de jeu doit être modifié
- La lumière doit être modifiée

2.3 Structure d'un fichier d'expérience (JSON)

Voir la section 3.2.3 Structure du JSON du rapport technique. Ce fichier de configuration pourra être généré à l'aide du configurateur.

2.4 Charger un fichier d'expérience

Le fichier de configuration d'une expérience doit porter le nom config.json.

Il doit suivre la structure décrite dans le chapitre précédent et se trouver dans un dossier spécifique selon la plateforme.

Le tableau ci-dessous montre où copier le fichier.

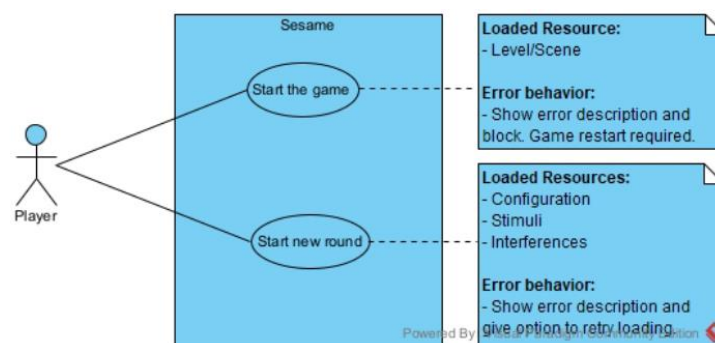
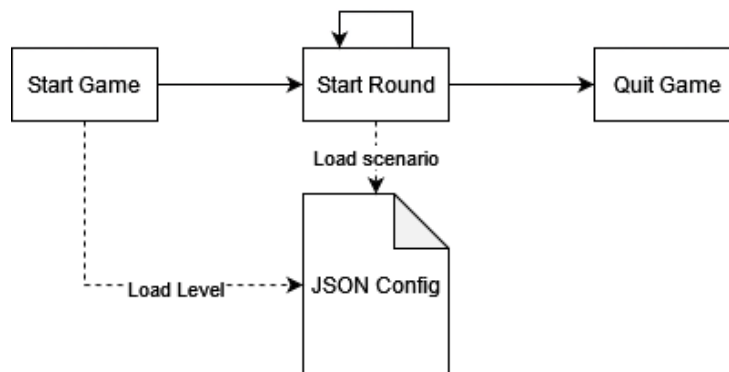
Plateforme	Dossier destination
Windows	%userprofile%/AppData/LocalLow/HEIAFR/ImmersiveSeriousGameFactory
Oculus Quest	storage/emulated/0/Android/data/com.heiafr.immersiveseriousgamefactory/files
Microsoft Hololens	%userprofile%/AppData/Local/Packages/ImmersiveSeriousGameFactory/LocalState

La video ISGF-upload-config.mp4 dans le dossier Videos montre comment charger une configuration en utilisant l'outil de transfert de fichiers.

2.5 Chargement de l'expérience

Le diagramme suivant montre comment le fichier de configuration/scénario est chargé :

- Le niveau (environnement) est uniquement chargé au moment de démarrer le jeu. Il est important de noter que tout autre niveau dans la scène sera supprimé et uniquement celui configuré dans le scénario sera chargé.
- Le scénario est chargé à chaque fois qu'un on clique sur "Débuter la partie" dans le menu principale



2.6 Récupérer un fichier de log

Les fichiers de log des expériences sont stockés dans un dossier spécifique selon la plateforme.

Le tableau ci-dessous montre où copier le fichier.

Plateforme	Dossier destination
Windows	%userprofile%/AppData/LocalLow/HEIAFR/ImmersiveSeriousGameFactory/ExperimentLogs
Oculus Quest	storage/emulated/0/Android/data/com.heiafr.immersiveseriousgamefactory/files/ExperimentLogs
Microsoft Hololens	%userprofile%/AppData/Local/Packages/ImmersiveSeriousGameFactory/LocalState/ExperimentLogs

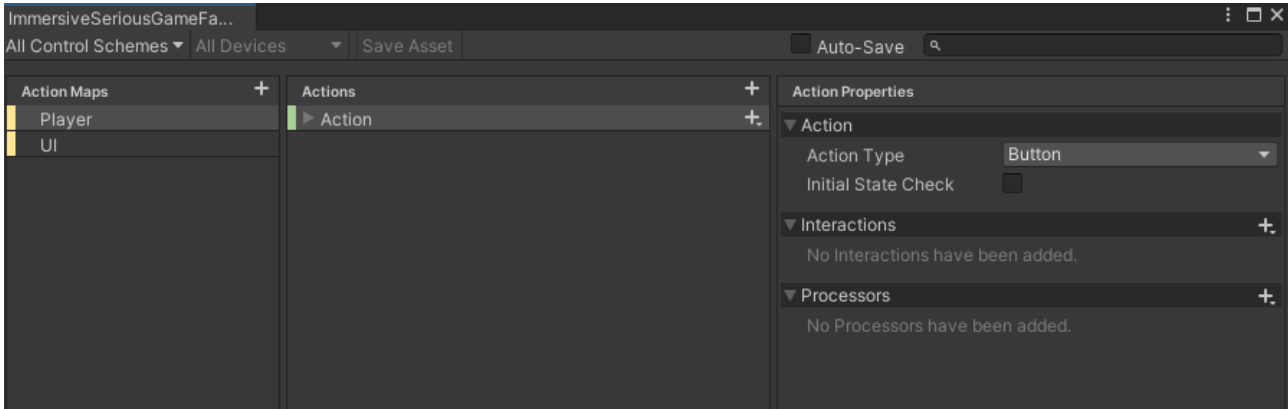
Pour récupérer les fichiers stockés sur Quest et Hololens, voir les chapitre 2.4 [Ajouter des fichiers sur Quest](#) et 2.5 [Ajouter des fichiers sur Hololens](#) de ce document.

Ces fichiers peuvent également être récupérés à l'aide de l'outil de transfert de fichiers comme décrit dans la vidéo ISGF-download-logs.mp4 du dossier Videos.

3 Modifications des touches / boutons

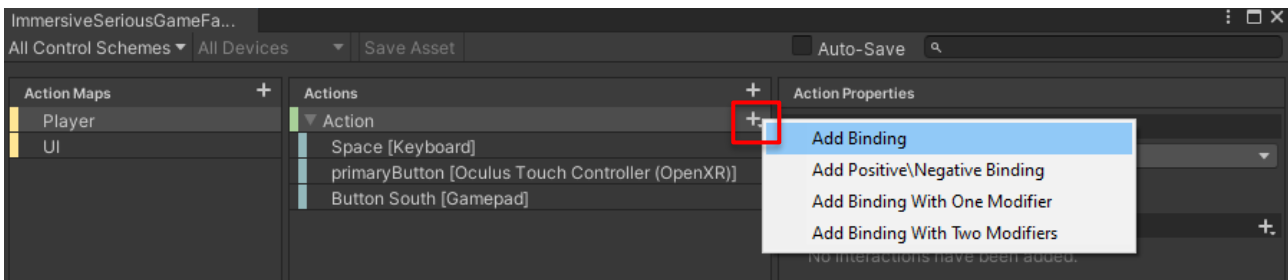
Pour modifier les touches / boutons utilisables dans ce projet, il faut ouvrir le projet dans l'éditeur Unity et ouvrir le fichier ImmersiveSeriousGameFactory.inputactions en double-cliquant dessus.

La fenêtre ci-dessous doit alors s'ouvrir.



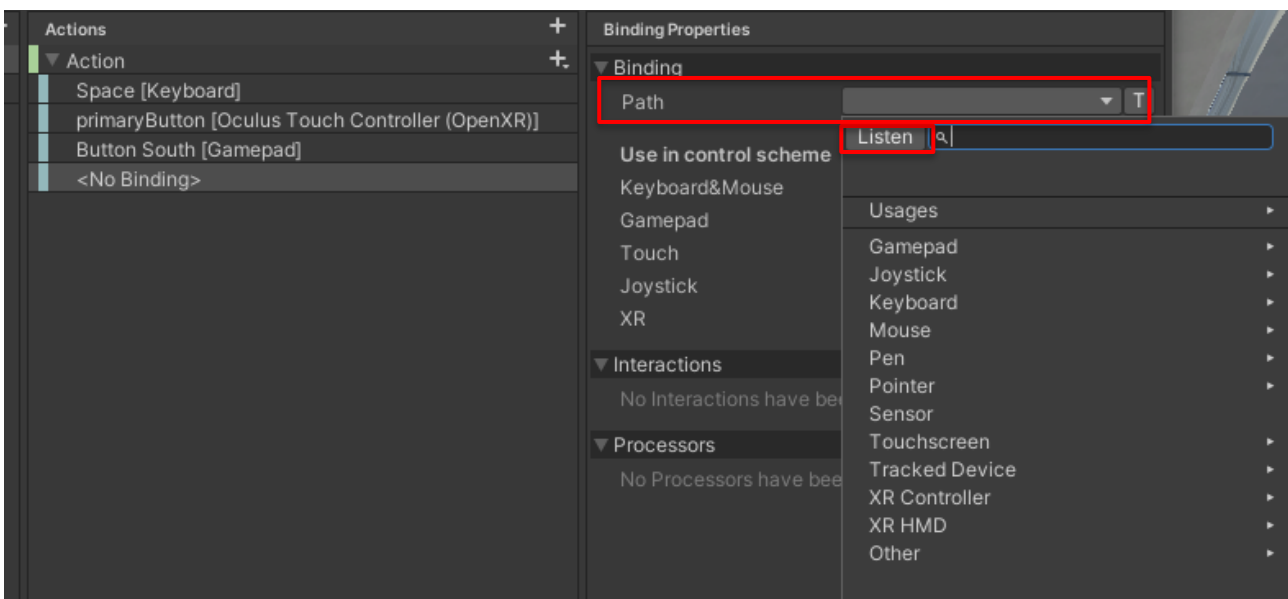
3.1 Ajouter un nouveau bouton

Pour ajouter un nouveau bouton lié à l'action du joueur, il faut appuyer sur le symbole « + » qui se trouve à droite de Action et cliquer sur Add Binding.



Il est ensuite possible de sélectionner à quelle touche ou bouton associer cette action en cliquant sur le dropdown à droite de Path.

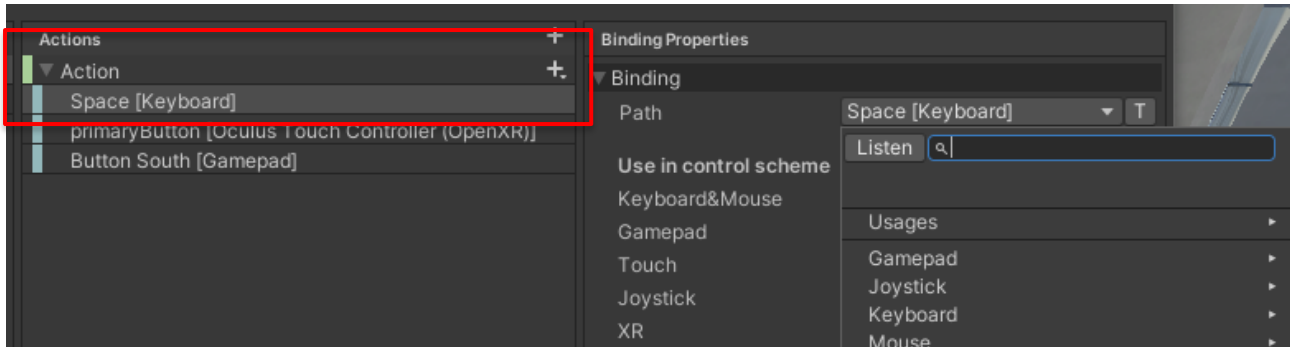
Il est aussi possible d'activer la fonction d'écoute en cliquant sur le bouton « Listen ». La prochaine touche ou bouton pressé sera alors attribué à l'action.



3.2 Modifier un bouton

Pour modifier un bouton attribué à l'action du joueur, il faut sélectionner le bouton / touche que l'on veut modifier sous Action (Space dans l'image ci-dessous), puis cliquer sur le dropdown à droite de Path et sélectionner un bouton ou une touche à attribuer.

Il est aussi possible d'activer la fonction d'écoute en cliquant sur le bouton « Listen ». La prochaine touche ou bouton pressé sera alors attribué à l'action.



4 Créer un nouvelle Interférence animée (AssetBundle)

Ce chapitre explique comment créer une nouvelle interférence animée et comment créer l'Asset-Bundle qui pourra être chargé dans le jeu.

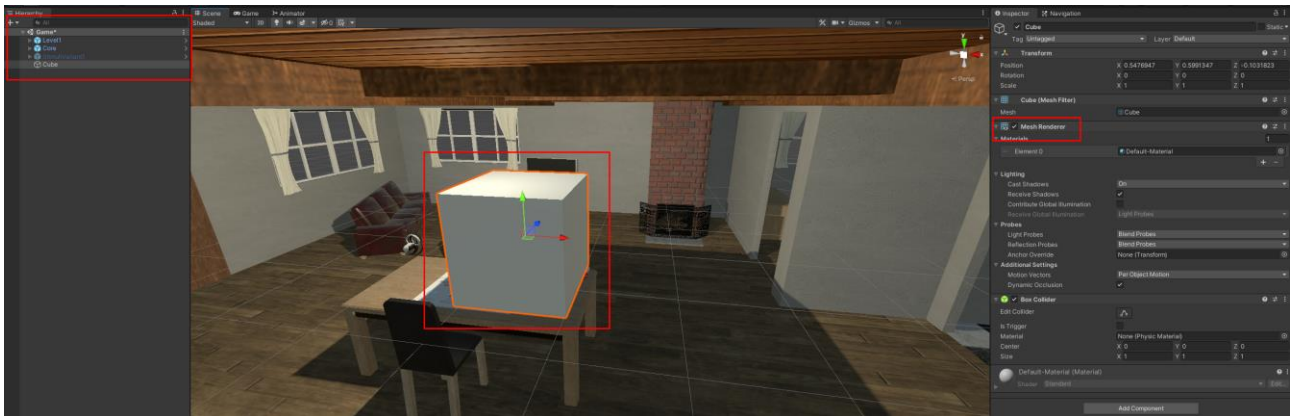
Attention : Les AssetBundles (Niveau, Stimuli et Interférence Animée) et les exécutables doivent être générés sous la même version (mineur) de Unity.

4.1 Interférence partiellement animée

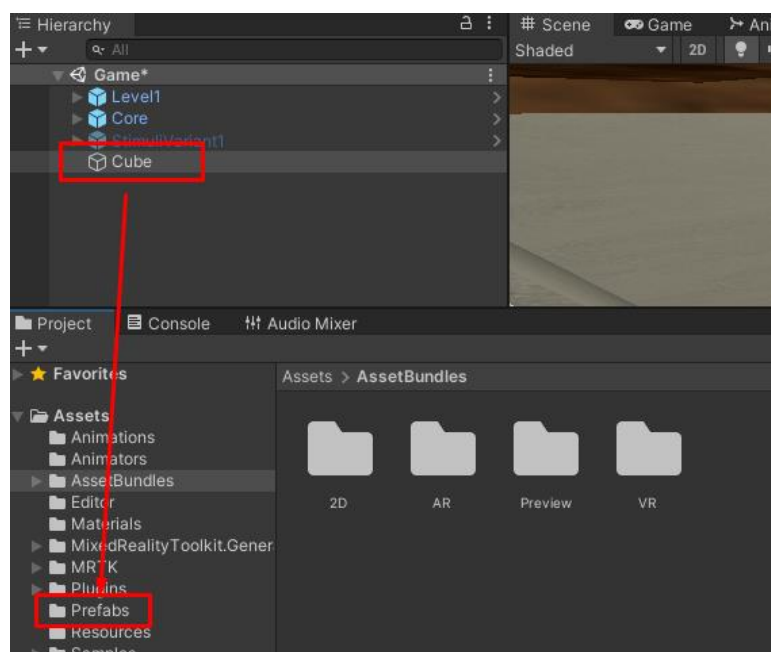
Contrainte : L'asset qui sera utilisé comme interférence partiellement animée doit être construit d'un seul **GameObject** (sans enfants) qui possède un composant **MeshRenderer**. Des objets de ce type sont souvent trouvable dans l'AssetStore de Untiy.

Dans l'exemple suivant nous allons créer une nouvelle interférence tout simple qui consiste d'un simple cube. Il peut s'appliquer sur tout objet qui respect la contrainte mentionnée.

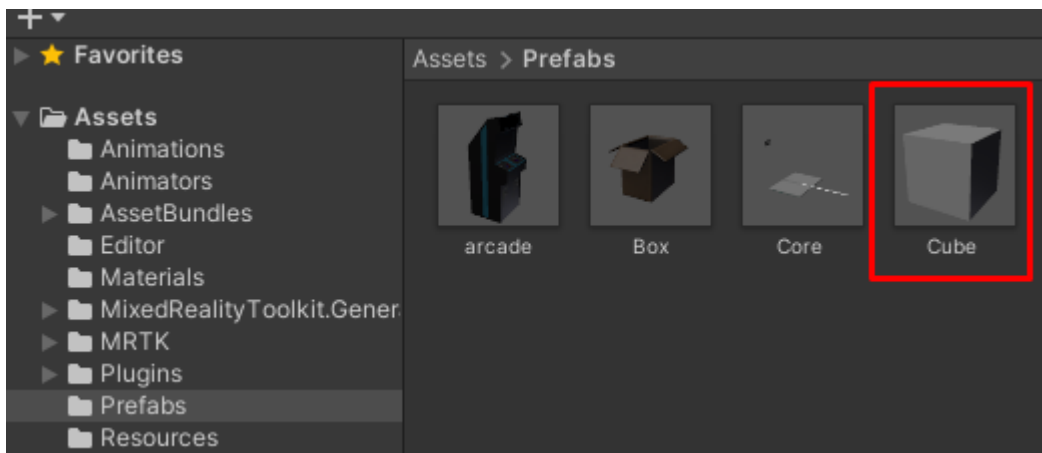
Pour commencer, on va mettre l'objet dans la scène (dans notre cas, on crée un nouveau GameObject Cube). Sa position, taille et emplacement dans l'arborescence n'est pas important. Voici le cube qui était ajouté dans la scène et qui possède un MeshRenderer.



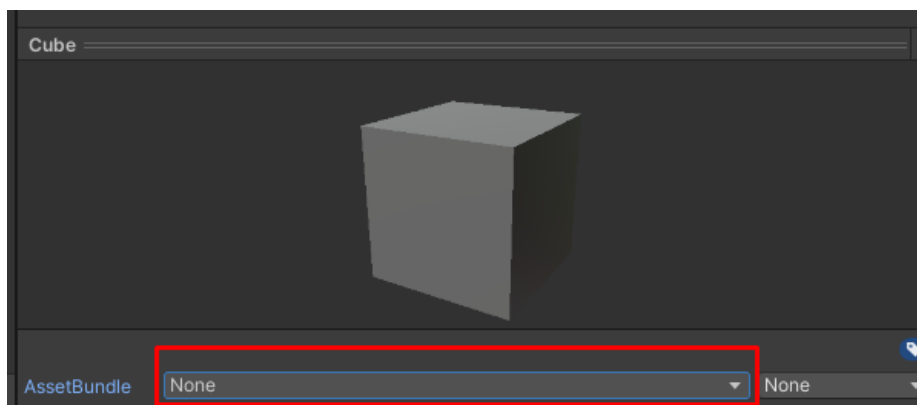
A l'aide d'un drag and drop, cet objet est maintenant copié dans le dossier **Prefabs**.



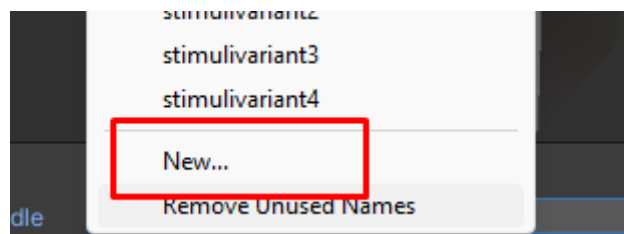
L'objet est maintenant disponible dans le dossier Prefabs. On sélectionne le nouvel objet pour que ses détails s'affichent dans l'inspecteur du côté droit. **Le nom qui est visible ici est important au moment du chargement de la ressource. Dans le fichier de configuration il sera mis dans le champ "prefab".**



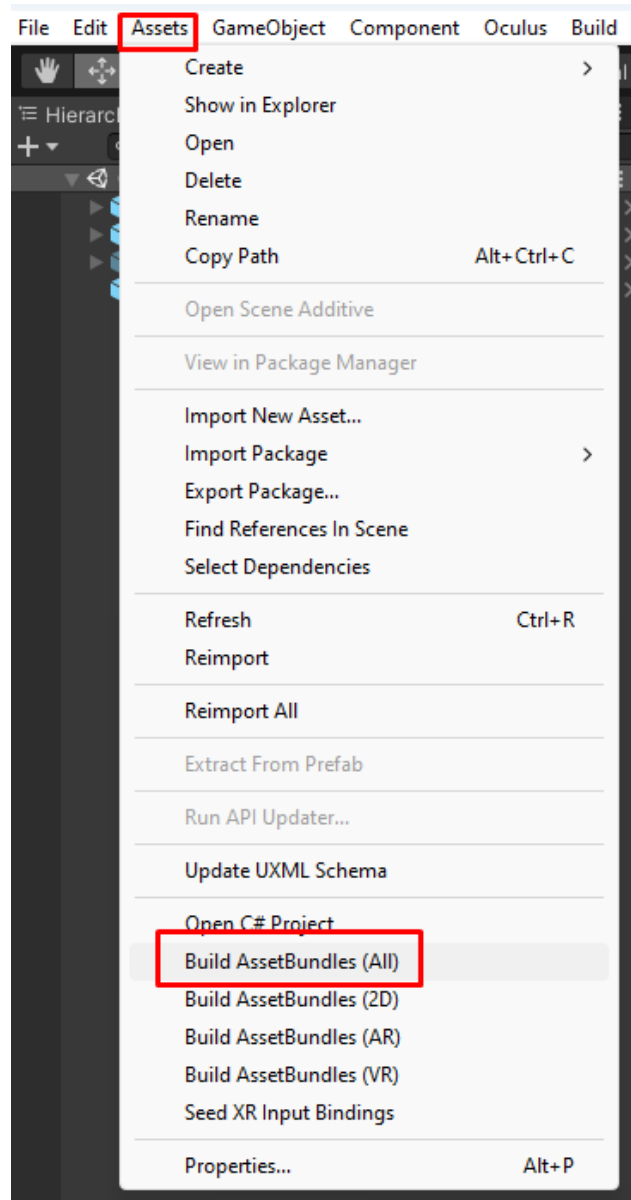
Tout en bas de l'inspecteur, à côté du label **AssetBundle**, on clique sur le dropdown qui actuellement montre **None**.



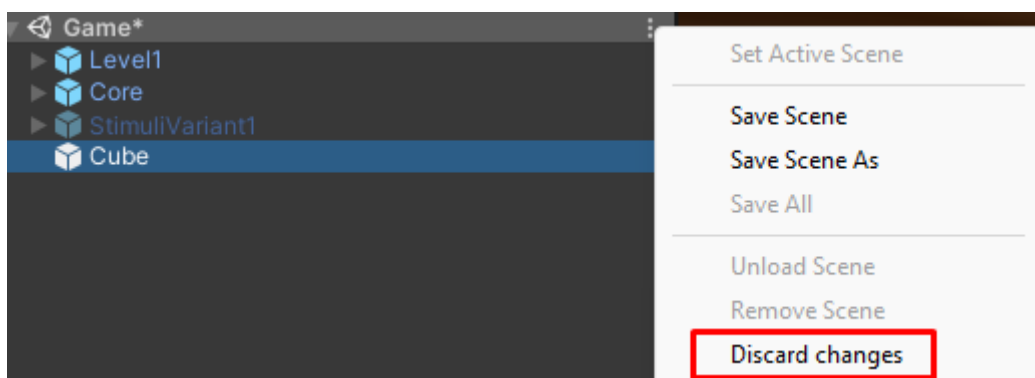
En cliquant sur **New...**, on peut donner un nom à l'AssetBundle. **Le nom donné ici sera utilisé comme nom de fichier et sera utilisé au moment de la création du scénario dans le champ "assetBundle"**



Maintenant, on peut maintenant régénérer toutes les asset bundels sous **Assets** → **Build AssetBundles**. Le nouveau bundle pourra être trouvé dans le dossier **Assets/AssetBundles** du projet Unity.



Pour terminer, on supprime l'objet qu'on a ajouté dans la scène. Au plus simple, cela peut être fait en supprimant toutes les modifications dans la scène.



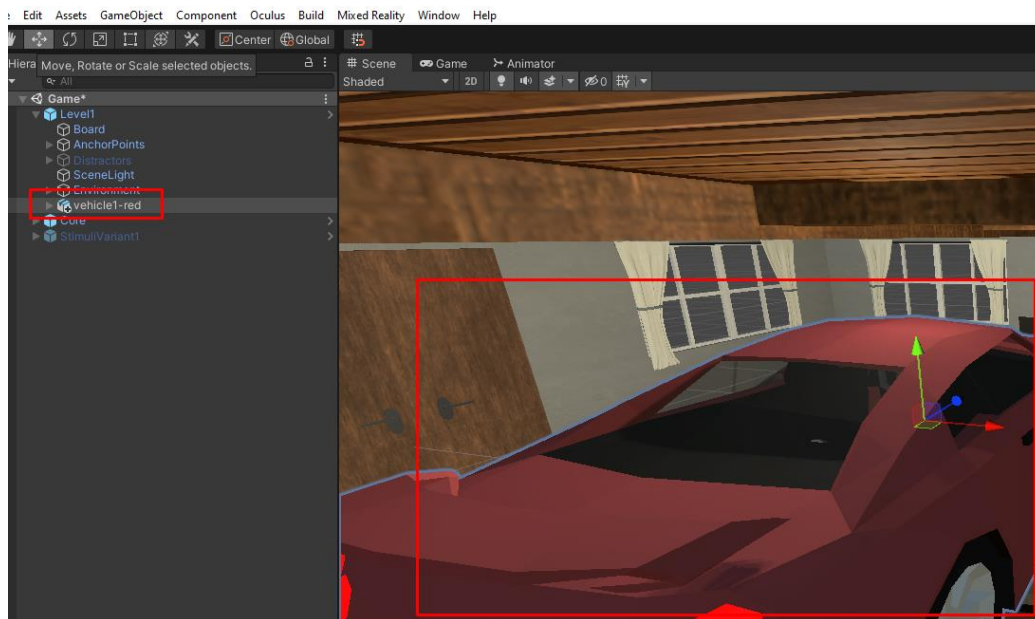
4.2 Interférence complètement animée

Ce chapitre explique comment créer une nouvelle ressource animée complet.

Ces interférences sont plus flexibles, cependant ils sont plus compliqués. Dans l'exemple nous allons ajouter une voiture qui avance et recule sur une table.

Contraintes : L'objet lui-même peut être aussi complexe comme souhaité (avec des enfants, la lumière et autre). Par contre, l'objet racine doit avoir un composant **Animator**. Ce composant sera activé au début de l'interférence. En plus, la position et la taille de l'objet doit être spécifié.

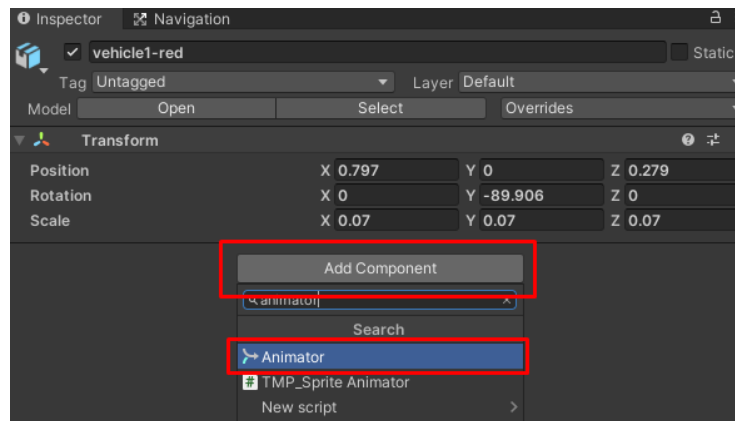
Pour commencer, on ajoute l'objet dans la scène. Dans ce cas, il est important de le faire en utilisant le niveau auquel l'interférence sera utilisée puisque son positionnement est relative à ce niveau. Dans l'arborescence il doit donc figurer comme enfant du niveau.



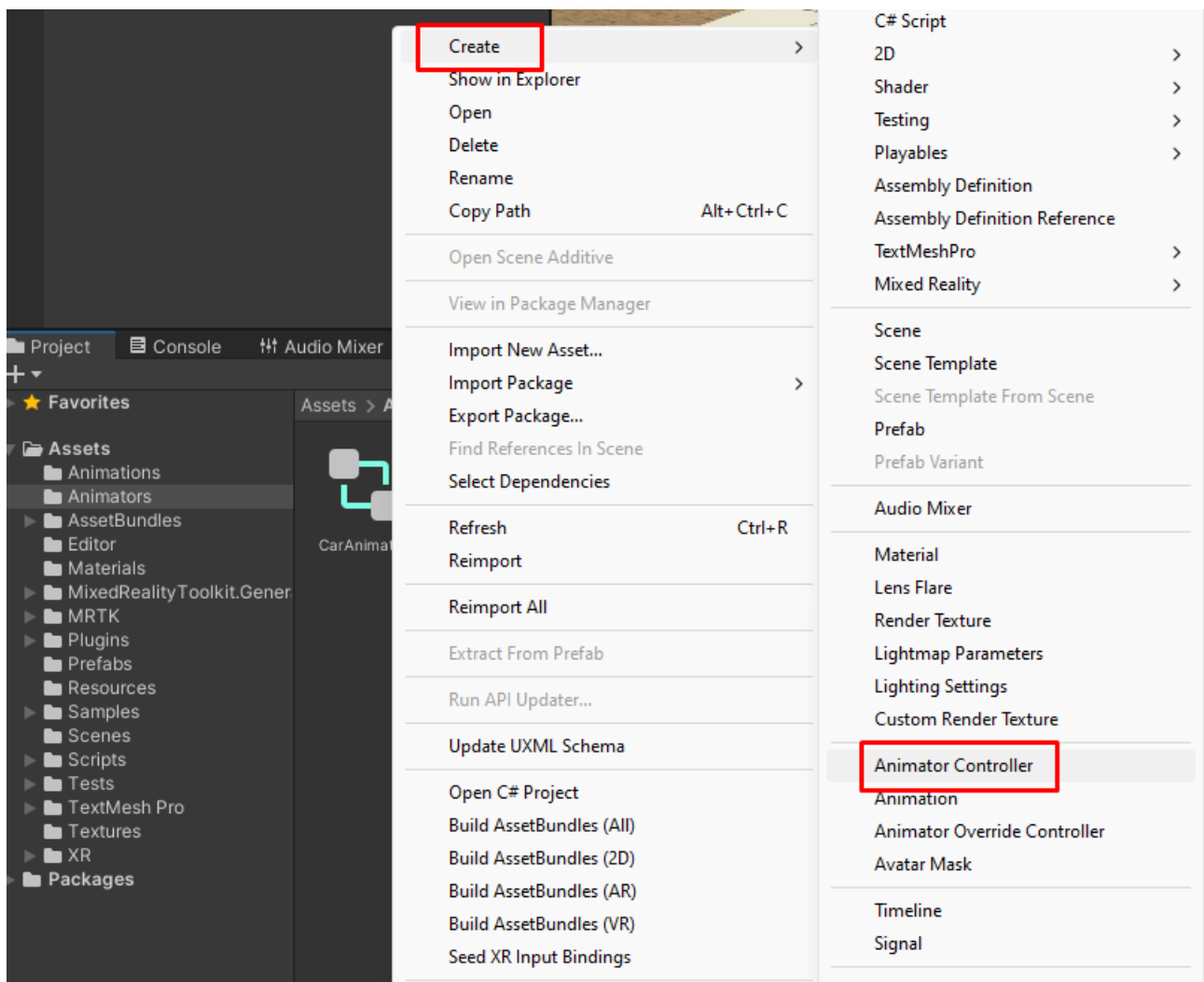
Maintenant on modifie son positionnement et sa taille pour que l'objet se retrouve à l'endroit où il doit apparaître.



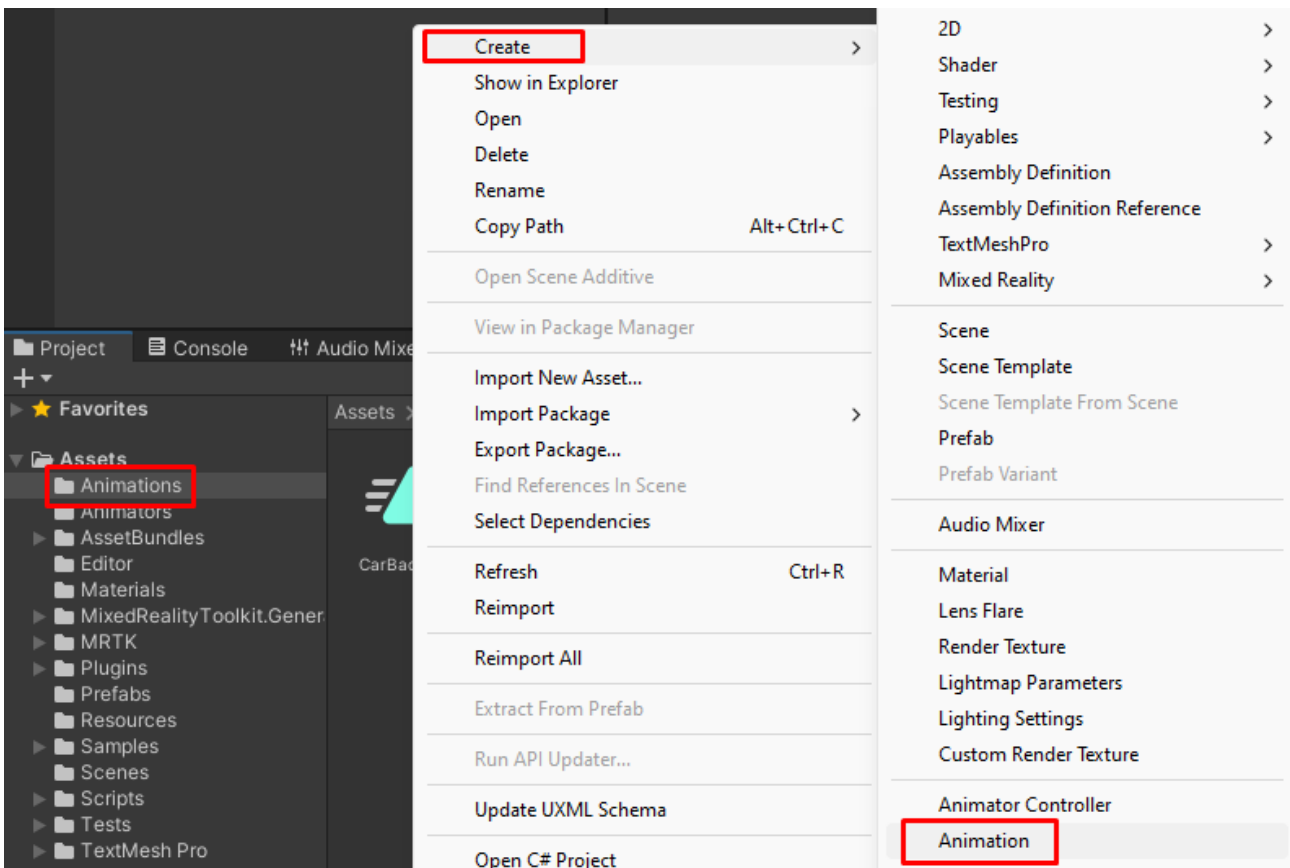
On ajoute maintenant le composant Animator à ce GameObject. Pour l'instant, il n'est pas encore configuré.



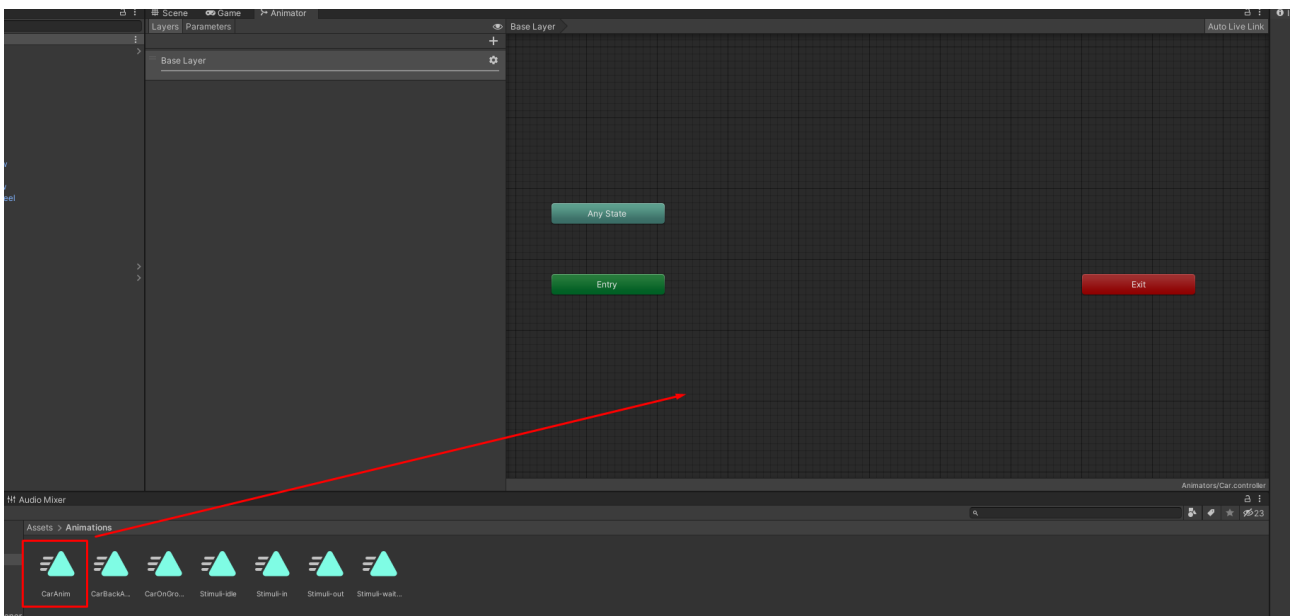
Nous allons maintenant créer l'animation qui sera joué au moment de l'interférence. Pour cela, nous devons tout d'abord créer un animator controller. On se déplace dans le dossier **Animators** et à l'aide d'un clic-droit on peut créer un contrôleur sous **Create** → **Animator Controller**. Il faut lui donner un nom et puis faire un double-clic sur cet objet. Le tab **Animator** devrait s'ouvrir.



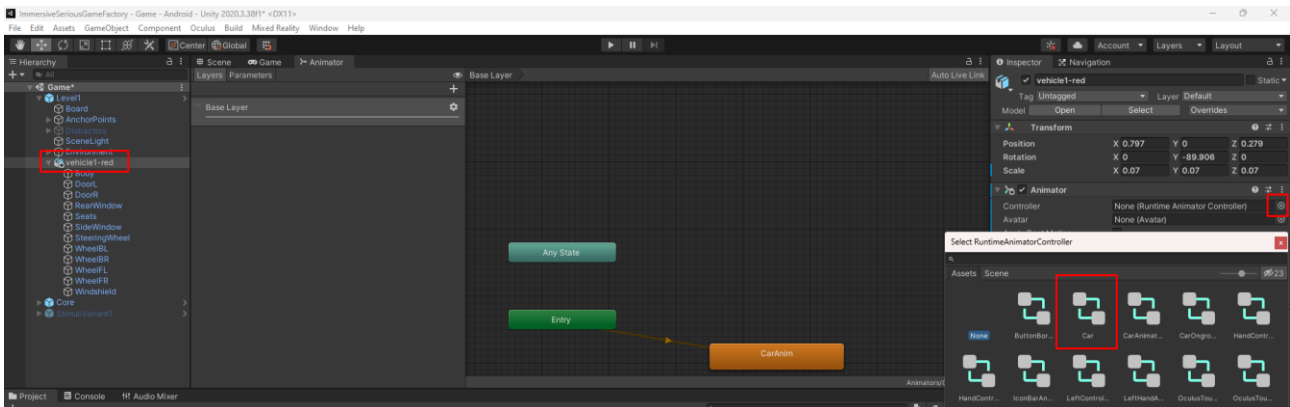
En plus du contrôleur nous avons besoin d'une animation. Celle-ci sera créé dans le dossier **Animatiors** avec un clic-droit, puis choisir **Create** → **Animation**.



On peut ensuite tirer cette nouvelle animation dans le tab de l'animator avec un drag and drop.

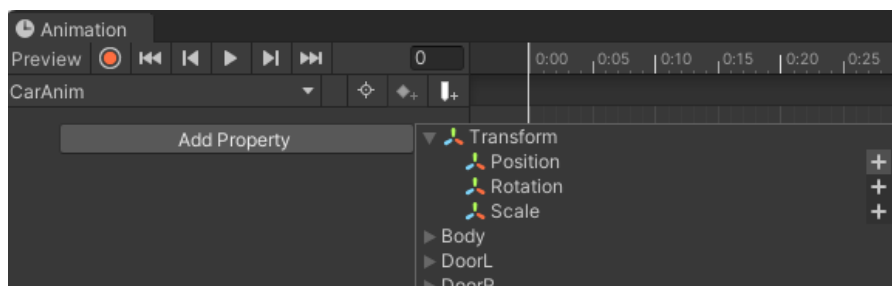


Dans l'arborescence de la scène, on peut maintenant sélectionner notre voiture. Dans l'inspecteur on peut cliquer sur le petit rond à côté de **Controler** et sélectionner le contrôleur que nous avons créé.

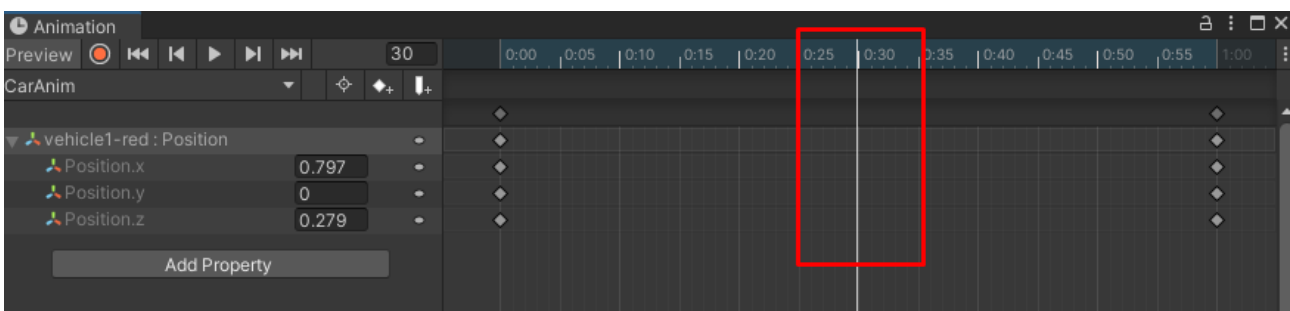


Avec un double-clic sur l'animation qu'on a ajoutée dans la fenêtre Animator qui figure comme rectangle jaune, la fenêtre **Animation** s'ouvre. Afin de pouvoir ajouter une nouvelle property à cette animation, il faut de nouveau sélectionner notre Objet dans l'arborescence de la scène.

Nous aimerions faire avancer la voiture. Pour cela nous ajoutons la property **Transform** → **Position** en cliquant sur le signe plus.



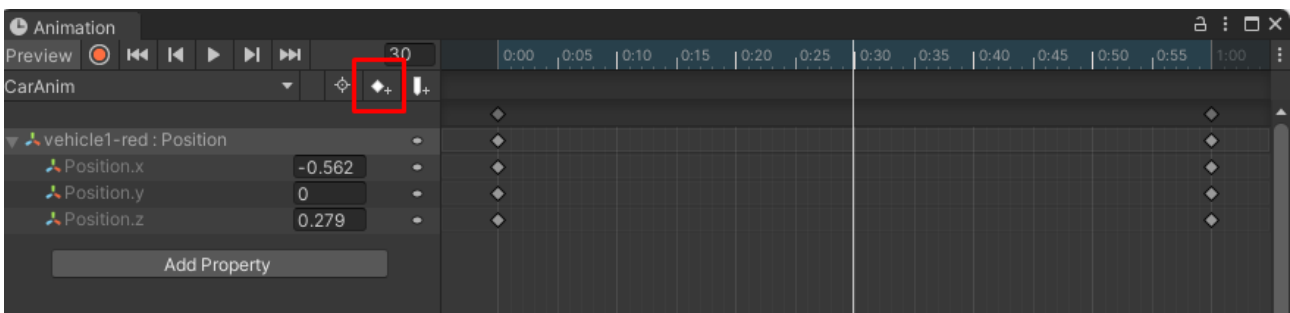
Maintenant, on pose le curseur de l'animation au milieu de la timeline.



En ouvrant le tab **Scene** on peut maintenant bouger l'objet à l'endroit intermédiaire.



Dans la fenêtre Animation on peut maintenant **ajouter un nouveau keyframe** (entouré en rouge). En cliquant sur le bouton play on peut voir l'animation complète.



Pour terminer, on peut maintenant suivre la procédure du chapitre précédent à partir du deuxième pas expliquant comment créer l'AssetBundle à partir de cet GameObject.

N'oubliez pas à supprimer tous les changements emportés à cette scène une fois que l'AssetBundle est built.

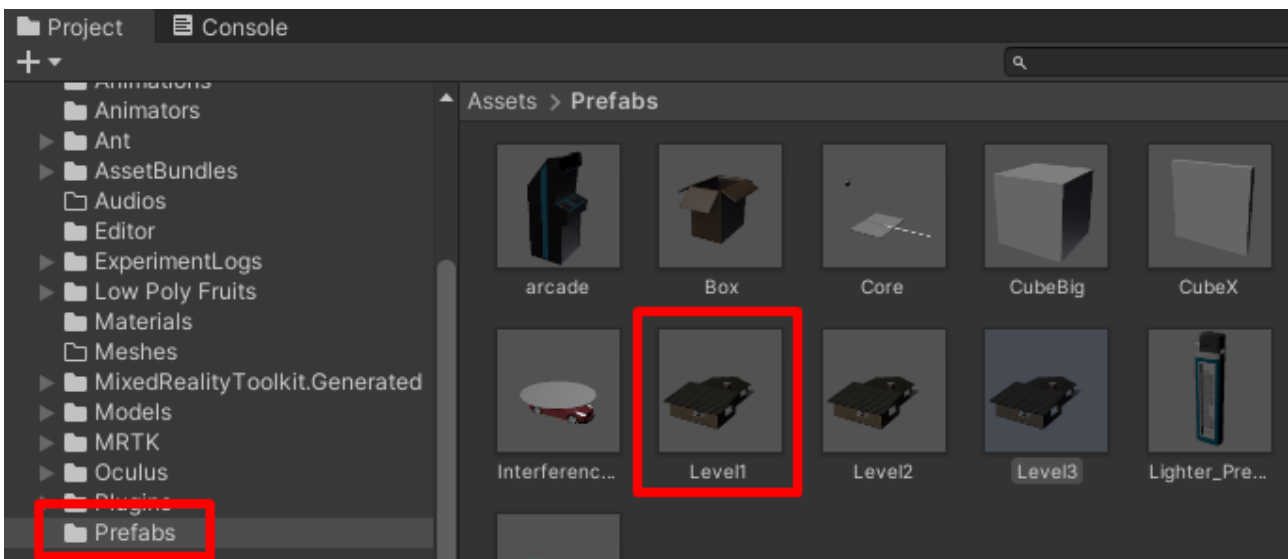
5 Créer un nouveau niveau

Le niveau par défaut "Level1" du jeu peut être remplacé par un niveau personnalisé. Cela est fait dans le fichier de configuration. Cette partie du fichier config est chargé au moment de lancer le jeu. Pour échanger le niveau il faut donc redémarrer le jeu, contrairement aux modifications qui concerne les autres parties de l'expérience qui sont chargé au moment de débiter une partie.

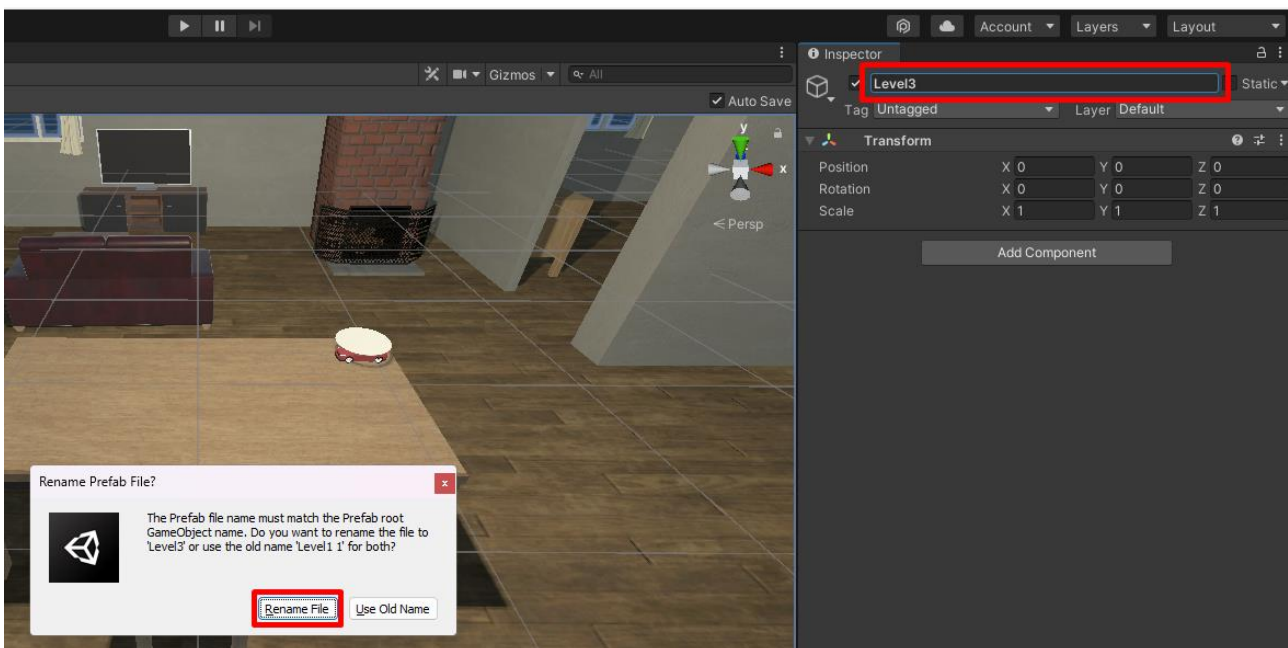
Attention : Les AssetBundles (Niveau, Stimuli et Interférence Animée) et les exécutables doivent être générés sous la même version (mineur) de Unity.

5.1 Créer une nouvelle variante du Level1

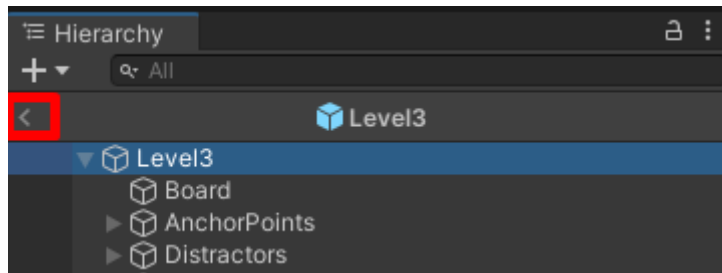
Dans un premier pas, faites une copie du prefab "Level1" en sélectionnant ce prefab et utiliser le raccourci de touche CTRL+D. Cela va créer un prefab portant le nom "Level1 1".



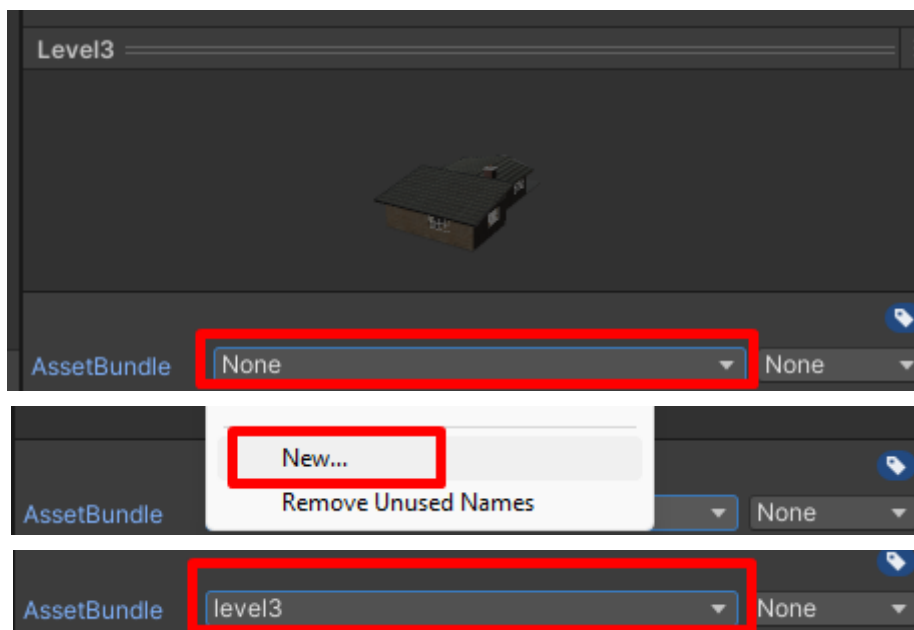
Maintenant, double-cliquez ce prefab et commencez à le renommer.



Ensuite vous pouvez faire toutes les modifications nécessaires à la scène. Une fois terminé, utilisez la petite flèche de gauche afin de sortir de la modification du prefab.



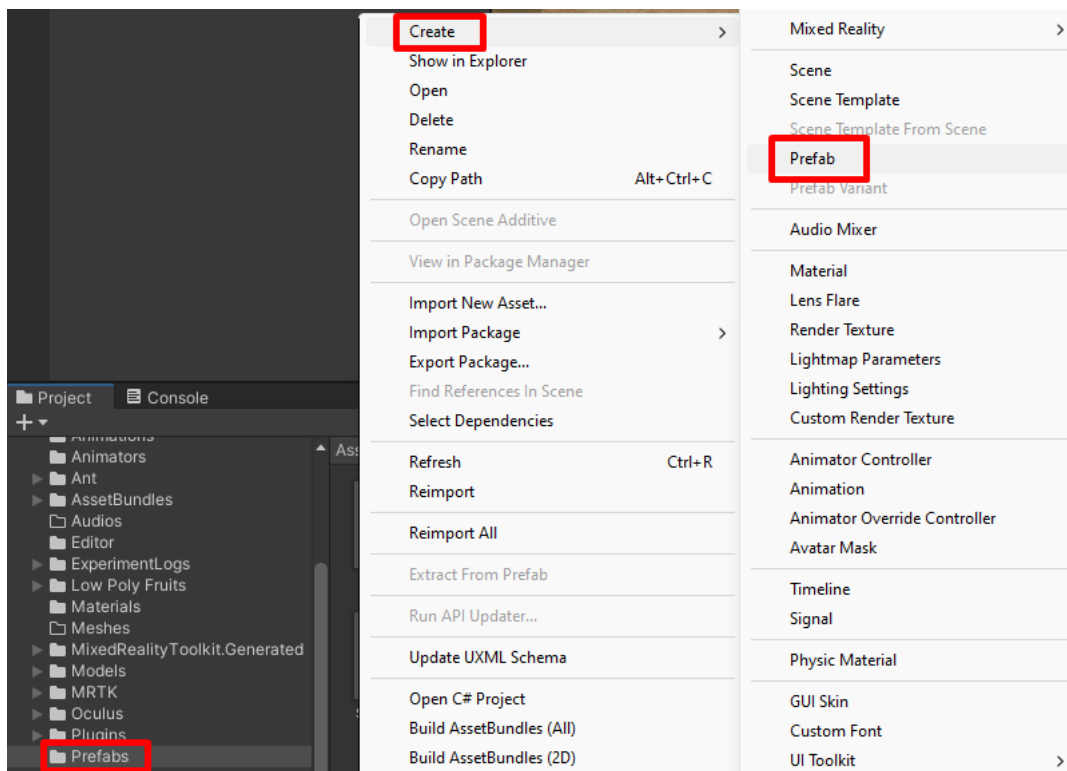
Resélectionnez ce prefab et nommez l'asset bundle en bas à droite. Pour cela, cliquez sur le dropdown qui actuellement montre "None", et sélectionnez "New...". Vous pouvez maintenant entrer un nom qui sera utilisé comme nom de fichier. Pour confirmer, appuyez sur la touche Enter.



Pour terminer, générez les asset bundles comme montré dans le chapitre 2.1.1.

5.2 Créer un tout nouveau niveau

Pour commencer, créez un nouveau prefab dans le dossier Prefab. Ensuite, ouvrez-le avec un double-clic et renommez-le comme montré dans le chapitre précédent.

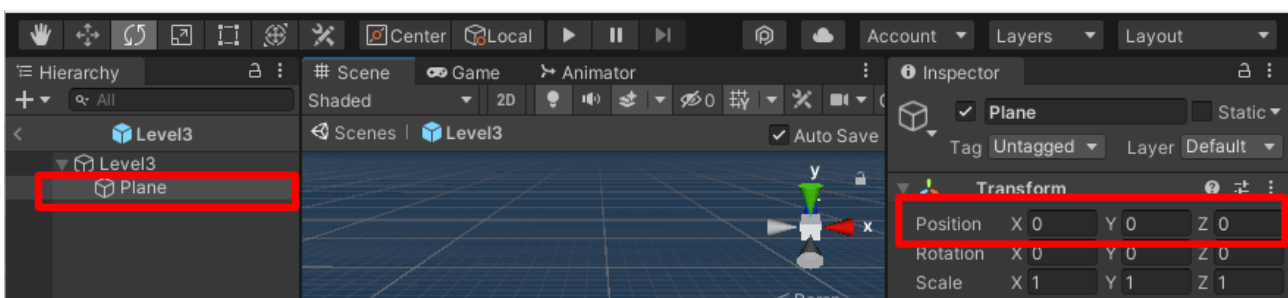


Des niveaux doivent respecter quelques contraintes afin de pouvoir être chargés. Les objets suivants doivent être présentes dans la hiérarchie du prefab.

- GameObject avec le tag "AnchorPoints" (voir chapitre 7 comment ajouter un AnchorPoint)
- GameObject avec le tag "Distractors"
- GameObject avec le tag "SceneLight" qui possède un composent "Light"

Une fois cette structure de base et mise en place, vous pouvez commencer à populer le niveau avec des objets. Vous pouvez vous servir des assets de l'asset store comme décrit dans le chapitre 7.

Afin que le niveau sera mise au bon endroit dans la scène et que le stimuli sera placé au bon endroit dans le niveau (préférable sur une table), il faut faire en sorte que le milieu de la surface de la table se trouve aux coordonnées (0,0,0). Pour vérifier cela, vous pouvez ajouter un point de référence : créez une nouvelle plane à la racine du niveau et placez-la à la position (0,0,0). Vous pouvez modifier les valeurs "Scale" afin qu'il ne prenne pas la taille de toute la scène. N'oubliez pas à l'enlever après avoir terminé le design du niveau.



6 Créer un nouveau stimuli

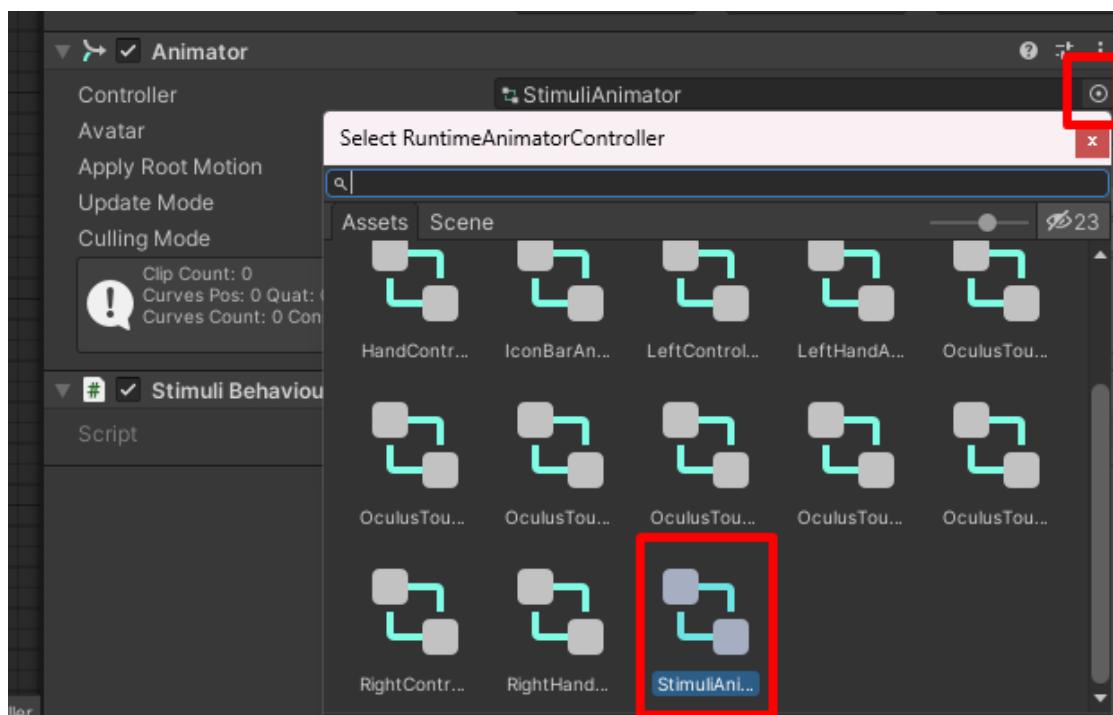
Attention : Les AssetBundles (Niveau, Stimuli et Interférence Animée) et les exécutables doivent être générés sous la même version (mineur) de Unity.

Le stimuli doit suivre les contraintes suivantes. Cette liste représente aussi la structure hiérarchique nécessaire.

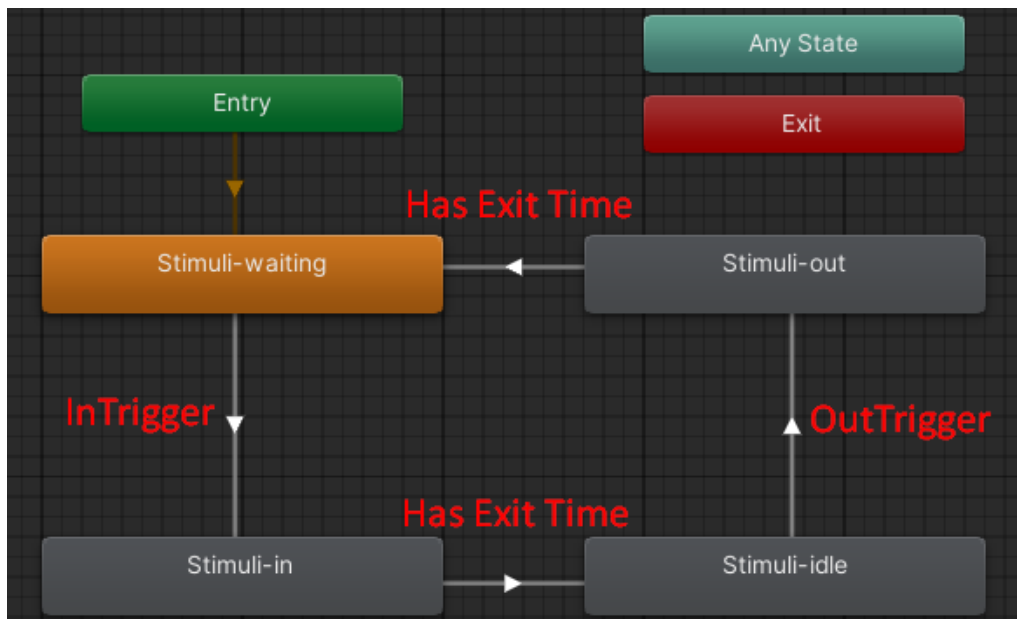
- GameObject avec le tag "StimuliRoot"
 - GameObject avec le tag "Stimuli" ayant un composant "Animator"
 - GameObject avec le tag "StimuliActive"
 - GameObject avec le tag "StimuliStop"
 - GameObject contenant une pile des stimuli

On commence donc à créer un nouveau prefab comme montré dans le chapitre précédent. En premier, il faut mettre en place la hiérarchie décrite.

Stimuli : Cela est le GameObject représentant le stimuli qui bouge devant le joueur. Tous ces enfants vont bouger avec. On peut donc ajouter des enfants qui font la représentation visuelle de l'objet. Le composant "Animator" peut être configuré afin d'utiliser le contrôleur d'animation par défaut portant le nom "Stimuli Animator". Pour utiliser cet animateur, sélectionnez le stimuli, cliquez sur le petit rond à droite du champ "Controller" et cherchez le "Stimuli Animator".



Si vous voulez créer votre propre animateur, veuillez respecter la machine d'état du contrôleur montré dans l'image suivante. Il faut donc ajouter les triggers "In" et "OutTrigger". En plus il faut faire en sorte que chaque animation dure exactement 1 seconde. Cela assure que le Fixation time soit bien respecté.



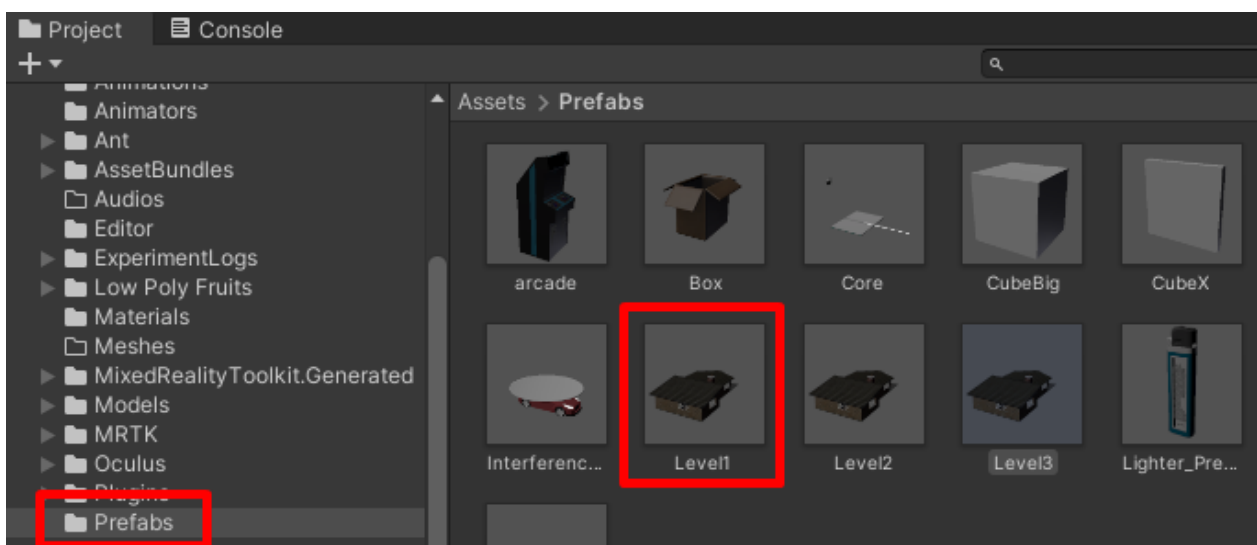
StimuliActive : Cela est le GameObject qui va être montré si le joueur fait une action. De nouveau, il peut contenir des enfants qui vont également être montrés après l'action du joueur.

StimuliStop : Cela est le GameObject qui va être affiché pour les stimuli stop après le Stop Signal Delay. De nouveau, il peut contenir des enfants qui vont également être montrés après le Stop Signal Delay.

Pille des stimuli : Cette partie est facultative. Cette pile va juste contenir plusieurs copies du stimuli (sans tag et animateur). Le vrai stimuli peut ensuite sortir de cet pile en éventuellement modifiant l'animateur.

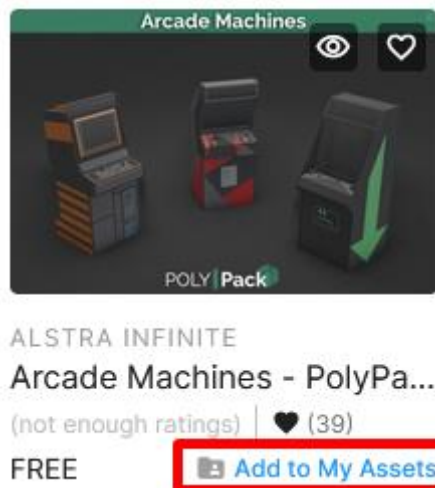
7 Ajouter un nouveau point d'encrage

Dans un premier pas, ouvrir le prefab "Level1" avec un double-clic. Il est possible que ce prefab soit affiché en très petit, et il faut donc bien zoomer pour le voir dans l'éditeur de scène.

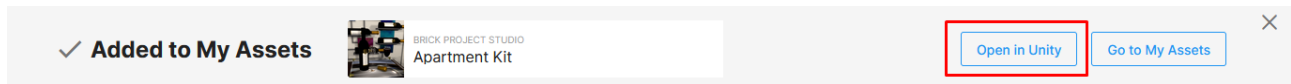


Ensuite, ajouter l'objet qui va être le nouveau point d'encrage. La manière la plus facile est de télécharger un objet depuis le Unity Asset Store : <https://assetstore.unity.com/>. Il en existe des nombreux asset packs gratuits.

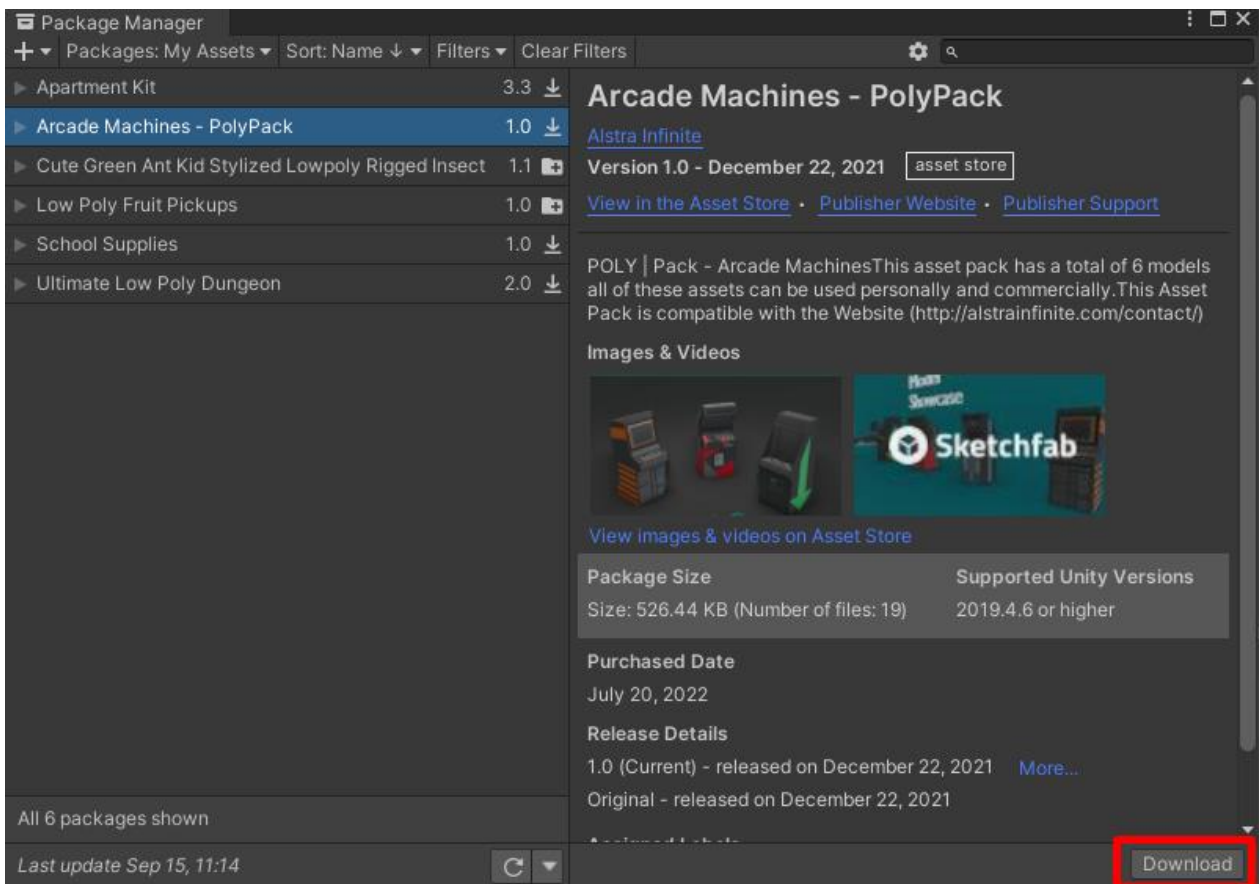
Après avoir trouvé un asset pack qui convient, cliquer sur « Add to My Assets ».



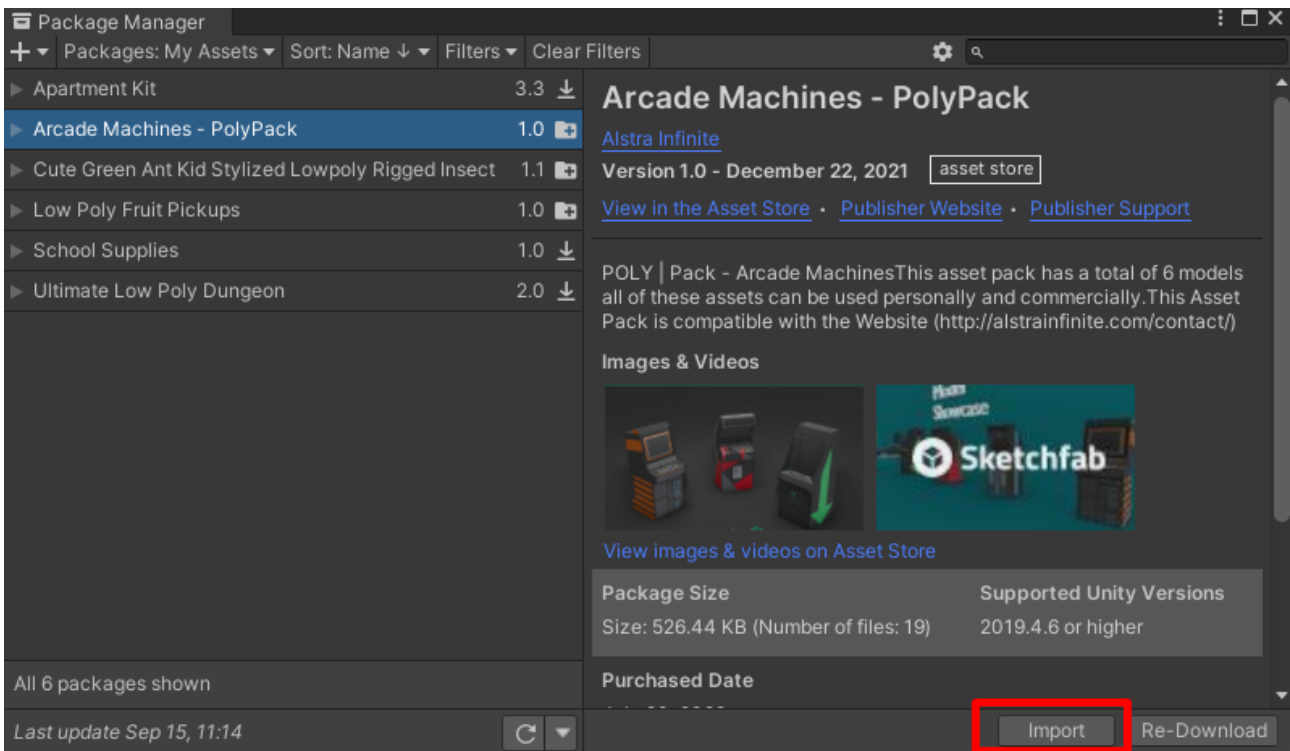
Et puis ouvrir en Unity.



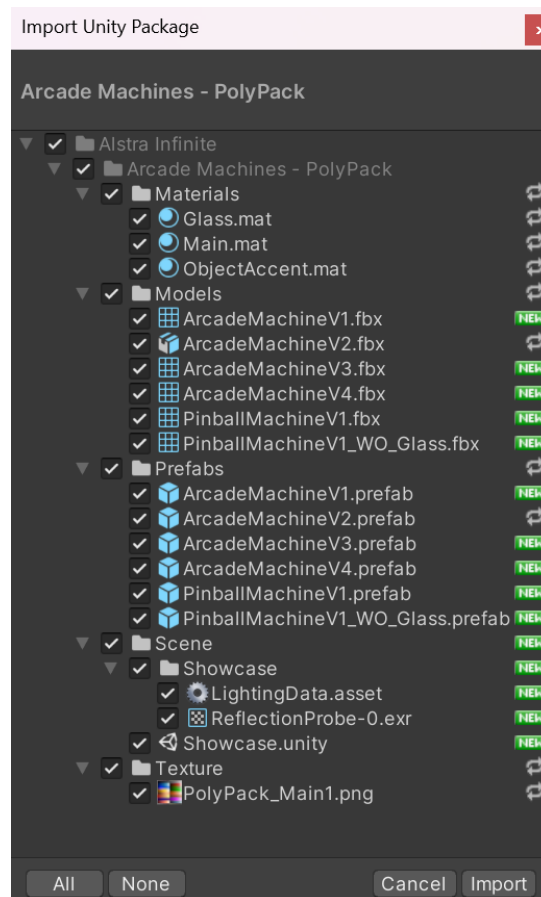
Cela va ouvrir le Package Manager de Unity qui nous permet de télécharger cet asset pack.



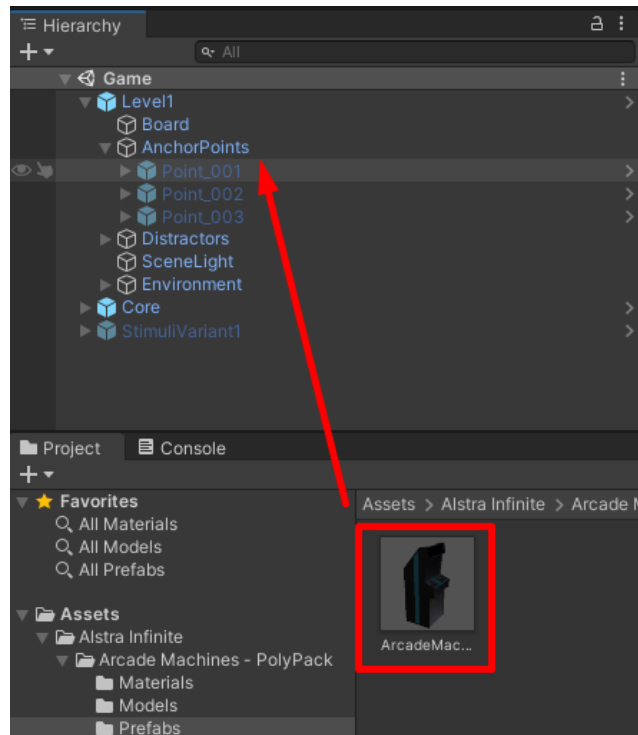
Après avoir téléchargé, on peut importer des objets.



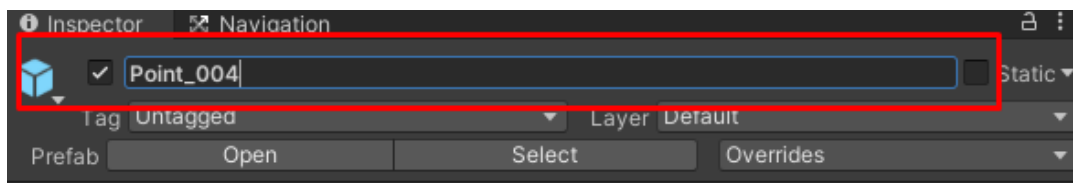
Dans ce dialogue, on peut choisir les objets à importer. Il est préférable de ne pas importer tout afin de ne pas utiliser trop d'espace. Prenez bien note de la structure des dossiers, c'est avec cette hiérarchie que les fichiers seront ajoutés au projet. Il faudra pouvoir localiser le dossier Prefabs.



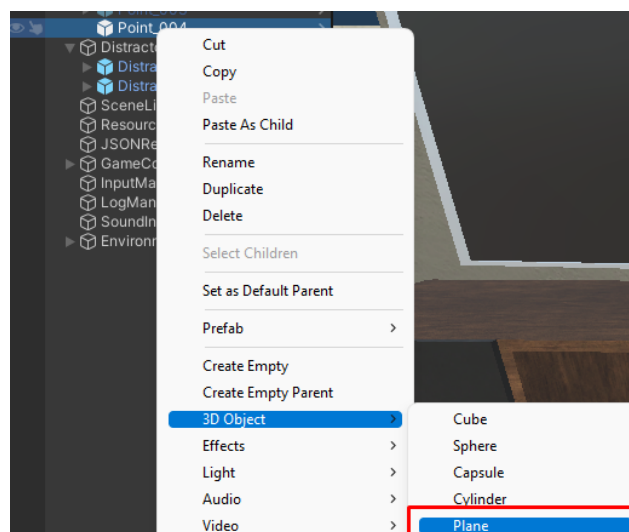
Les objets se retrouvent dans un sousdossier portant le nom de l'asset pack du dossier « Assets ». Dans celui on peut aller chercher le prefab. Il doit être ajouté dans la scène en faisant un drag & drop vers le GameObject « AnchorPoints ».



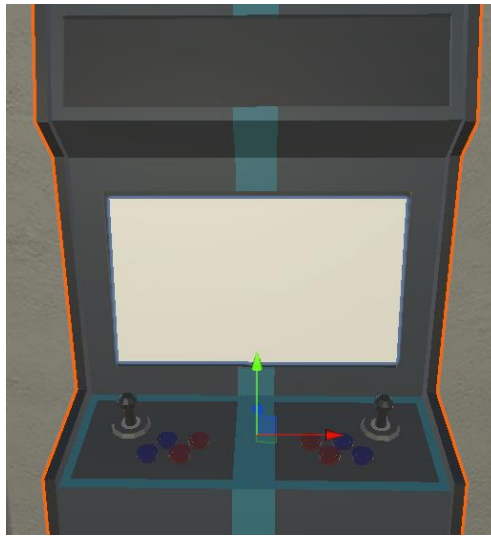
Cela va faire apparaître l'objet dans la scène à son origine. Il peut ensuite être placé à un point convenable. L'objet doit être renommé avec le schème « Point_xxx ». Dans ce cas il doit s'appeler « Point_004 » puisque c'est le prochain numéro libre.



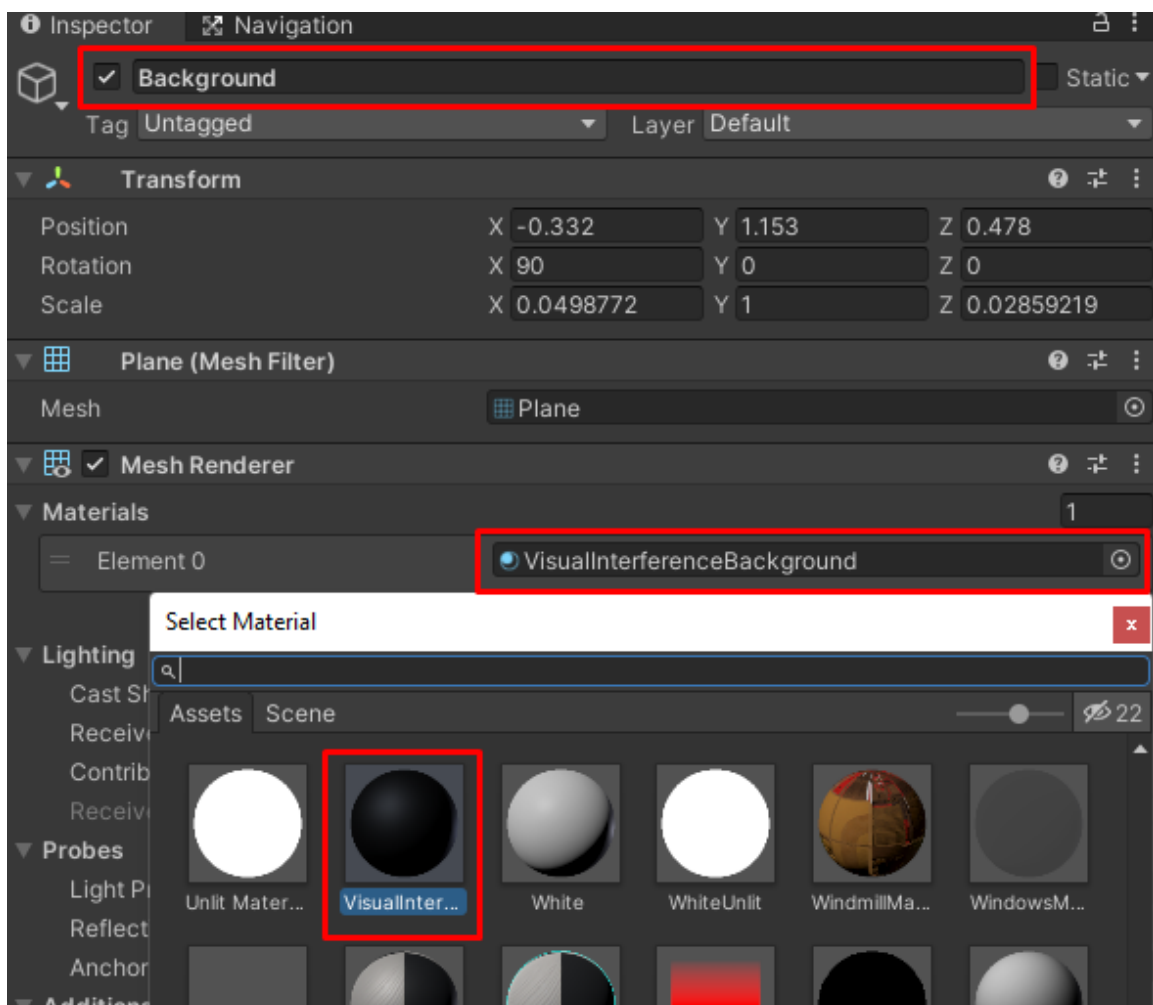
Il faut ensuite ajouter un objet « Plane » à ce point d'encrage en faisant un clic-droit sur celui-ci.



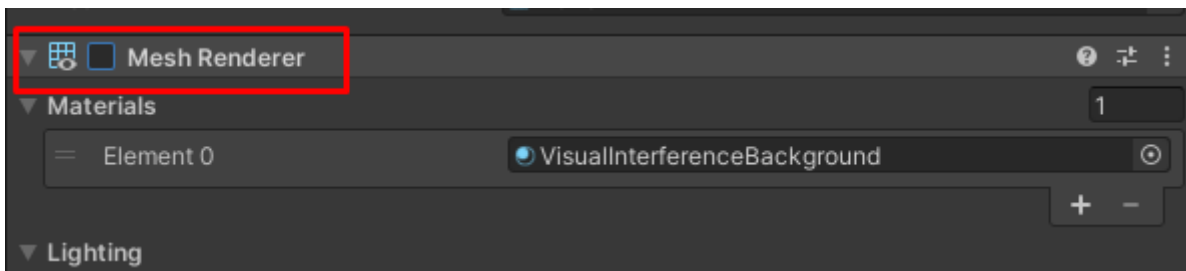
C'est sur cette plane que va être affichée l'interférence visuelle et elle doit donc être redimensionnée afin d'être à la bonne taille de l'objet. Pour cet objet d'exemple on aimerait que l'interférence couvre tout l'écran de la machine arcade. La plane (ici en blanc) est donc redimensionnée afin d'être à la même taille que cet écran.



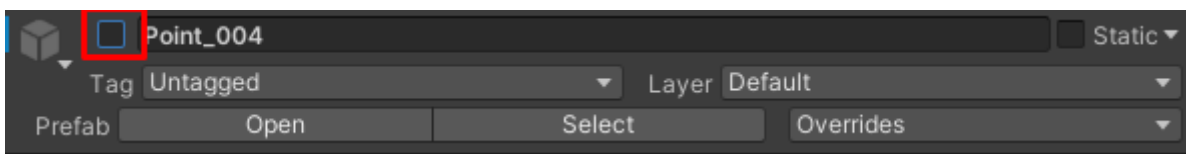
La plane est ensuite renommée en « Background » et le matériel du MeshRenderer est modifié. Ce matériel va être visible au cas où l'image ne peut pas couvrir tout l'espace prévu puisqu'il devrait être déformé autrement. Le matériel « VisualInterferenceBackground » est proposé pour cela mais tout autre matériel peut être utilisé à sa place.



Le MeshRenderer de la plane doit être désactivé.



Et le GameObject « Point_xxx » doit aussi être désactivé si l'objet ne doit pas être visible par défaut.



Pour terminer, il est recommandé de tester ce nouveau point d'encrage afin de vérifier l'orientation. Au cas où l'interférence visuelle n'ait pas la bonne orientation, il faut jouer avec la propriété « Rotation » de la plane. La manière la plus facile est d'utiliser l'outil « Rotate Tool ». Cliquer sur la plane et puis drag & drop (dans ce cas le demi-cercle bleu) pour faire un tour de 180°.

