

# A1 - Configuration de base et annexes d'un véhicule

## Informations à propos de l'organisation du wiki sur la configuration des préfabs de véhicules

Le wiki à partir de cette page va être organisé en fonction d'un code simple pour vous repérer dans le niveau de difficulté et votre progression ☐☐

- “ **A**=Configuration de base pour un véhicule
- B**=Configuration pour un véhicule qui nécessite un minimum de réflexion et qui dépend de votre véhicule.
- C**=Configuration pour un véhicule qui est complexe et qui nécessite de la recherche personnelle et qui varie en fonction des véhicules.
- D**=Système complexe pour un véhicule qui n'est pas généralisé à tous les véhicules et qui n'est pas évident.

**Il est important de savoir que tout est possible**, mais ce système de code permet de vous donner une idée de la difficulté. La seule limite que vous avez est l'habitude ainsi que l'expérience avec le workbench et blender, on ne peut pas tout dévoiler car ENFUSION et Blender restent des logiciels complexes ☐

Tous les préfabs non commenté dans cette page ne sont pas a modifier pour l'instant (si vous suivez le fil rouge du wiki) ou simplement pas a modifier sur toute l'intégration de

voiture (potentiel exception a des systèmes personnalisés).

# Introduction

Dans cette page, nous allons découvrir les composants qui permettent de réaliser un véhicule de base. A la fin de cette page, votre prefab base véhicule aura votre modèle 3D assigné ainsi que des configurations de base pour votre véhicule.

Il faut savoir que dans l'intégration d'un véhicule, ça ne sert à rien de réinventer la roue, on vous conseil de vous baser sur les véhicules vanilla ainsi que sur les fichiers de configuration déjà existant.

Après cette page, le véhicule ne sera pas encore jouable.

## MeshObjet

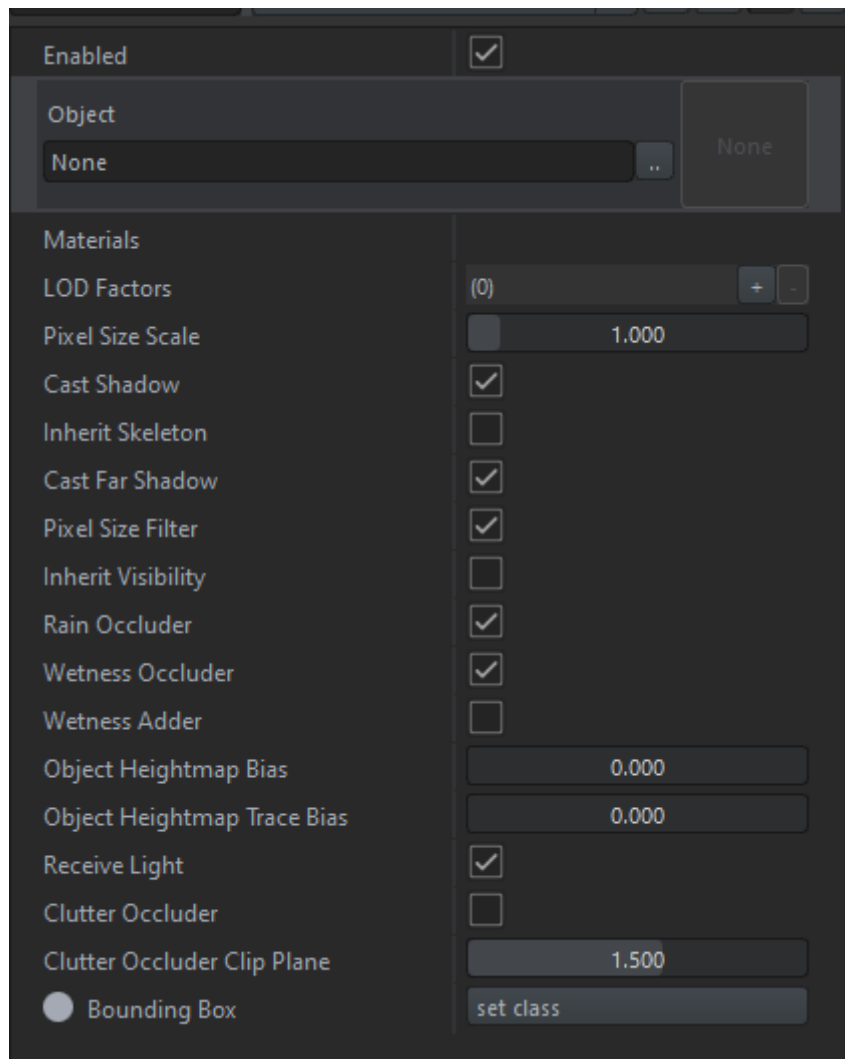
**Cliquez pour en savoir plus**

**Ce component permet de gérer le rendu de votre modèle 3D dans le prefab.**

Il vous suffit de définir dans le paramètre "Object", le chemin de votre modèle 3D de véhicule principal.

Vous pouvez ici régler un grand nombre de choses en lien avec l'affichage du modèle 3D en lui même.

Il y a également le menu des matériaux, ce menu va nous être utile pour créer des variants de votre véhicule en lui changeant de camouflage par exemple.



## RigidBody

### Cliquez pour en savoir plus

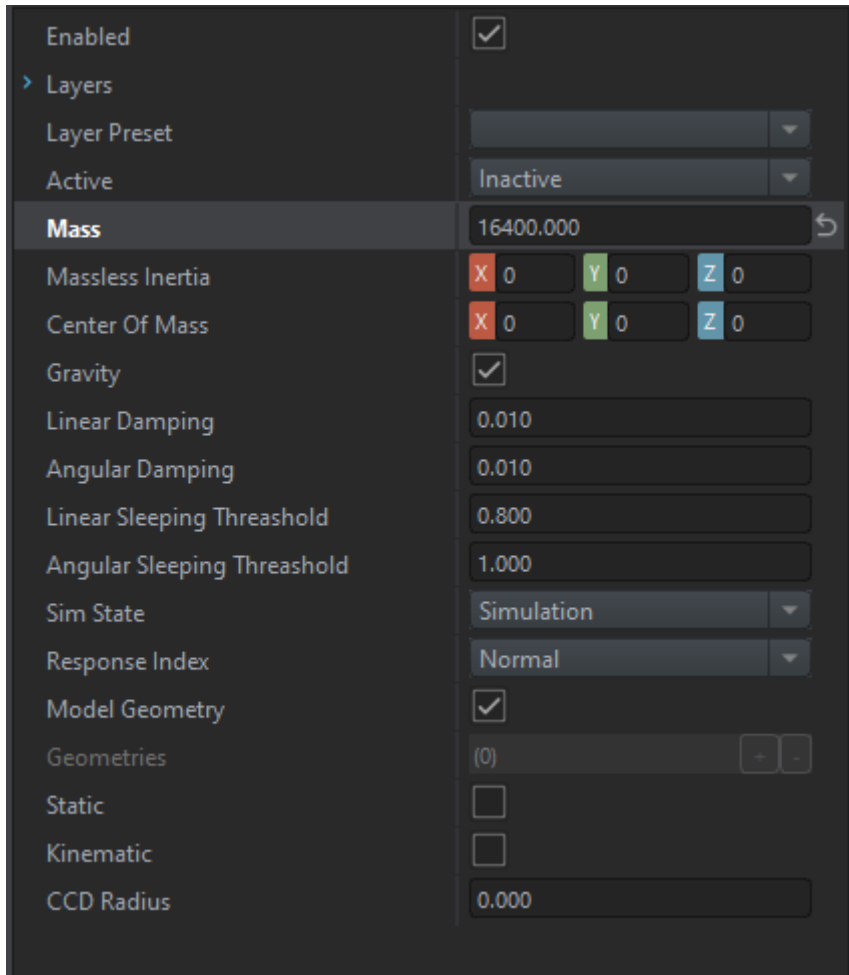
Le component RigidBody permet de régler tout ce qui est en lien avec le modèle 3D dans le jeu, cet à dire les collisions qu'il va avoir, sa masse, son centre de masse...

Avec les préfabs base, tout est déjà configuré pour un véhicule de son type. Vous pouvez déjà modifier le paramètre masse pour que la masse dans Arma Reforger correspondent avec celle dans la vraie vie.

**ATTENTION:** Le réglage du centre de la masse et du Massless inertia se font que une fois le véhicule fonctionnel ⚠

Si vous grillez des étapes, vous risquez de définir une valeur biaisé qui fait bugger la simulation de la masse.

Les valeurs du centre de la masse et du Massless inertia doivent être ronde pour éviter tous soucis: 1.432 →1.4



## SCR\_VehicleSoundComponent

**Cliquez pour en savoir plus**

Ce component va servir a assigner des sons à notre véhicule.

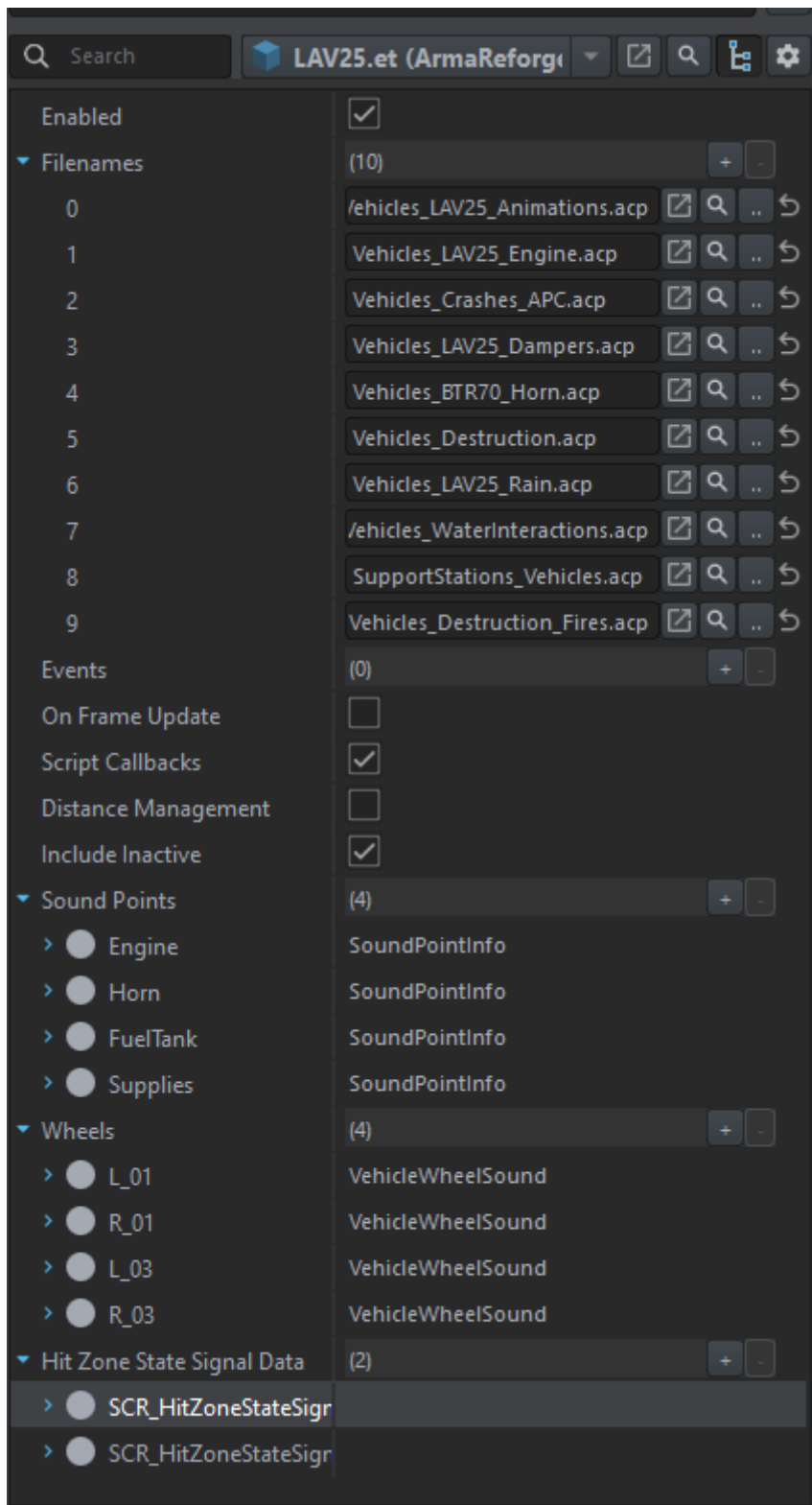
A part si vous voulez créer des sons personnalisés, je vous conseil de récupérer la même configuration que dans un véhicule vanilla, recherchez dans la barre de recherche un

véhicule similaire qui existe dans Arma Reforger puis copiez collez les propriétés de ce component.

La liste Filenames permet de définir la liste des projets son que le véhicule va utiliser.

Les sound Points permettent de définir une position ou les sons vont apparaître. PS: c'est le même fonctionnement pour les positions wheels.

Et pour finir HitZoneStateSignalData permet de définir une liste de signaux a donner aux projets son en fonction de l'état du véhicule (Ex: Roue abîmé).



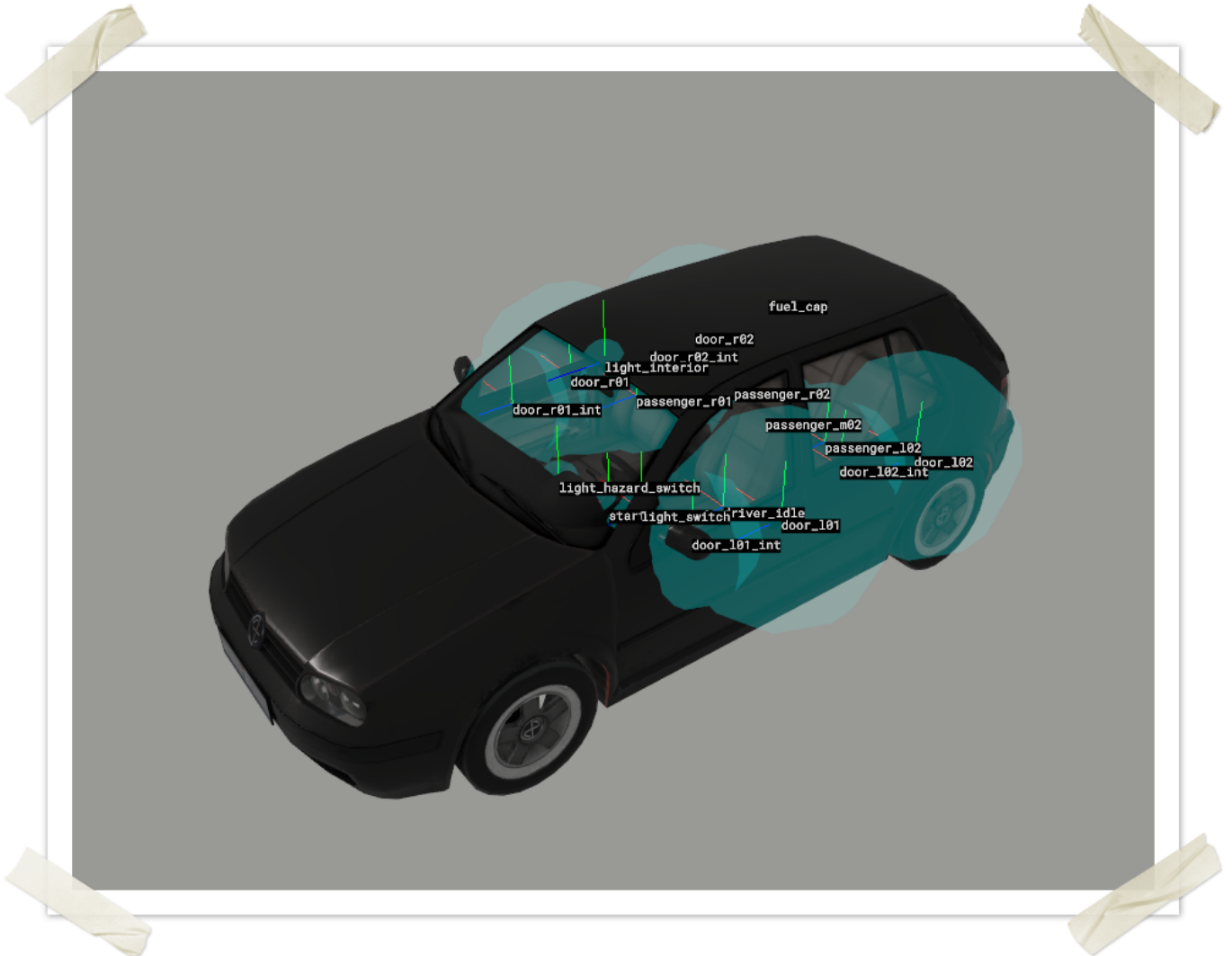
## ActionManagerComponent

Cliquez pour en savoir plus

Ce composant va nous servir à définir une liste d'actions disponibles pour notre véhicule.

Tips: Faites clic droit sur le component **ActionManagerComponent** puis cliquez sur *toggle context(s) radius visualization* et *toggle context(s) visibility angle visualization* pour rendre la tâche de créer des actions beaucoup plus simples.

Il faut savoir que les noms des actions ne sont que superficiel, vous pouvez très bien les nommer comme vous le voulez, mais dans tout le court de ce topic, les actions auront les noms définis plus bas.



*Aperçu avec les aides activées*

Ce component est découpé en deux parties:

- **Actions Contexts:** Liste des "actions visibles" par les joueurs.
- **Additional Actions:** Paramètre de ce que les actions vont réaliser une fois que le joueur ai interagit avec.

# Object Properties

Vehicle

ActionsManagerComponent

- Vehicle 3
- ActionsManagerComponent 3

Search

Enabled



▼ Action Contexts

(1)

> ● Unflipper

▼ Additional Actions

(11)

> ● SCR\_RefuelAtSupportStationA

> ● SCR\_ResourceEntityRefundAc

> ● SCR\_RepairAtSupportStationA

> ● SCR\_RepairAtSupportStationA

> ● SCR\_RepairAtSupportStationA

> ● SCR\_RepairAtSupportStationA

> ● SCR\_UnflipVehicleAction

> ● SCR\_UnflipVehicleAction

> ● SCR\_PushVehicleAction

> ● SCR\_ToggleHandbrakeAction

> ● SCR\_ToggleHandbrakeAction

Allow Cross Hierarchy



Is Ghost Entity



+ Add Component

Apply to prefab

Object Properties

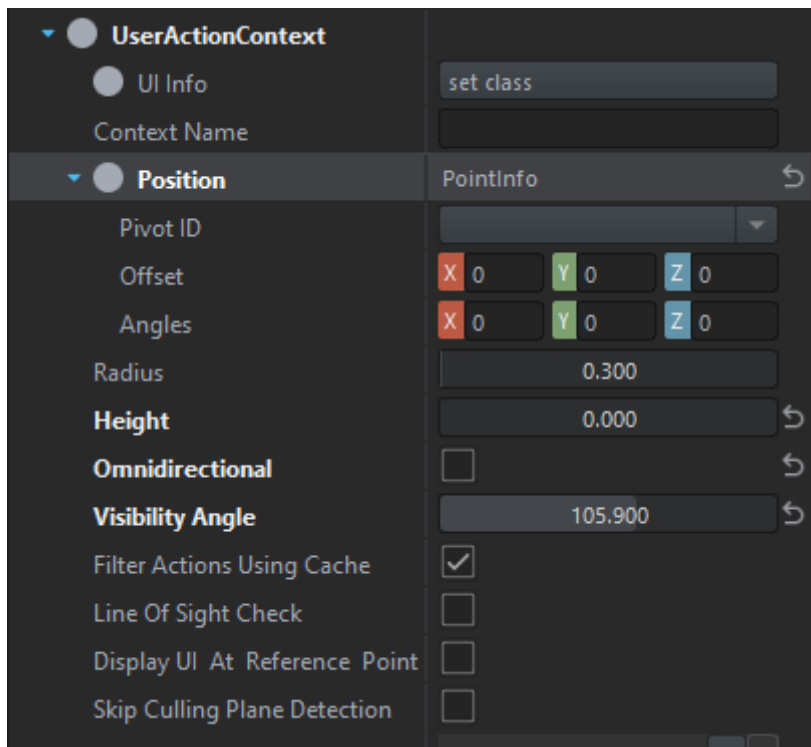
Move

Version 1.3.0.55 (submission)

Pour créer une nouvelle action, il vous suffit de cliquer sur le + à coté de la ligne "Action Contexts". Voici la description des options d'une action:

- **UI Info:** Permet de définir une liste d'informations en lien avec l'apparence de l'interaction (PS: Pour les véhicules, inspirez vous de ce qui a déjà été configuré, notamment sur les icônes).
- **Context Name:** Le nom de l'interaction dans l'éditeur.
- **Position:** La position de l'interaction (cliquez sur Set Class).
- **Radius:** Taille de l'action sous la forme d'une sphère.
- **Height:** Hauteur de l'action. Cette fonctionnalité permet aux actions de prendre une taille qui fait tout le long de votre véhicule par exemple.
- **Omnidirectionnal:** Si cette option est cochée, l'action est visible dans tous les angles possibles. Si elle n'est pas cochée, il vous suffit de régler le **Visibility Angle** pour définir un angle maximum où l'interaction est visible.
- **LineOfSightCheck:** Si cette option est cochée, la visibilité de l'action dépendra de la ligne de mire.
- **Display UI at reference point:** Si cette option est cochée et que le **Height** a une valeur au dessus de 0, l'UI de l'actions ne suivra pas les mouvements de la caméra du joueur et l'UI restera au point que vous avez défini plus haut.

Pour le cas des actions sur un véhicule, ne mettez pas de titre ou de description aux actions depuis sur l'UI Info, on verra plus tard pour que l'action ai un texte traduis et une icône.



Une fois que vous savez tout ça, il ne vous reste plus qu'à créer ce minimum d'actions dans la catégorie Action Contexts:

### Actions spécifique pour chaque compartiment:

Context Name	Position & description
driver	Action qui permet de changer de siège ou de monter si la porte est ouverte en place conducteur.
gunner	Action qui permet de changer de siège ou de monter si la porte est ouverte en place tireur. Placez cette action au niveau du siège tireur.
commander	Action qui permet de changer de siège ou de monter si la porte est ouverte en place chef de char. Placez cette action au niveau du siège chef de char.
passenger_l01, passenger_m01, passenger_r01, ...	Action qui permet de changer de siège ou de monter si la/les porte(s) est ouverte en place passager gauche, milieu ou droit. Placez cette action au niveau des sièges respectif de chaque action.
passenger_l02, passenger_m02, passenger_r02, ...	Action qui permet de changer de siège ou de monter si la/les porte(s) est ouverte en place passager arrière gauche, milieu ou droit. Placez cette action au niveau des sièges respectif de chaque action.

### Actions extérieure pour les portes:

Context Name	Position & description
door_L01, door_L02 ...	Action pour une porte simple, dans ce cas des portes situés à la gauche du véhicule.
door_R01, door_R02 ...	Action pour une porte simple, dans ce cas des portes situés à la droite du véhicule.
door_back_left, door_back_right ...	Action pour une porte simple, dans ce cas des portes situés à l'arrière du véhicule, soit à gauche soit à droite

### Actions intérieure pour les portes:

Context Name	Position & description
door_L01_int, door_R02_int ...	Action pour une porte simple, dans ce cas des portes situés à la gauche et à l'intérieur du véhicule.
door_R01_int, door_R02_int ...	Action pour une porte simple, dans ce cas des portes situés à la droite et à l'intérieur du véhicule.
door_back_int	Action pour une porte simple, dans ce cas des portes situés à l'arrière du véhicule et à l'intérieur. En général, une seule action suffit si vos deux portes sont proches.

**Conseil:** Définissez un Radius de visibilité pour les actions des portes. Par exemple, définissez un radius de 90°, vers l'intérieur pour les actions `door_L01_int`. Cela évite que l'action intérieur soit visible depuis l'extérieur.

### Actions extérieure du véhicule:

Context Name	Position & description
push_front	Action qui permet de pousser le véhicule depuis l'avant. Il faut la placer à l'avant, au niveau d'une zone sur laquelle les joueurs peuvent pousser le véhicule.
push_rear	Action qui permet de pousser le véhicule depuis l'arrière. Il faut la placer à l'arrière, au niveau d'une zone sur laquelle les joueurs peuvent pousser le véhicule.

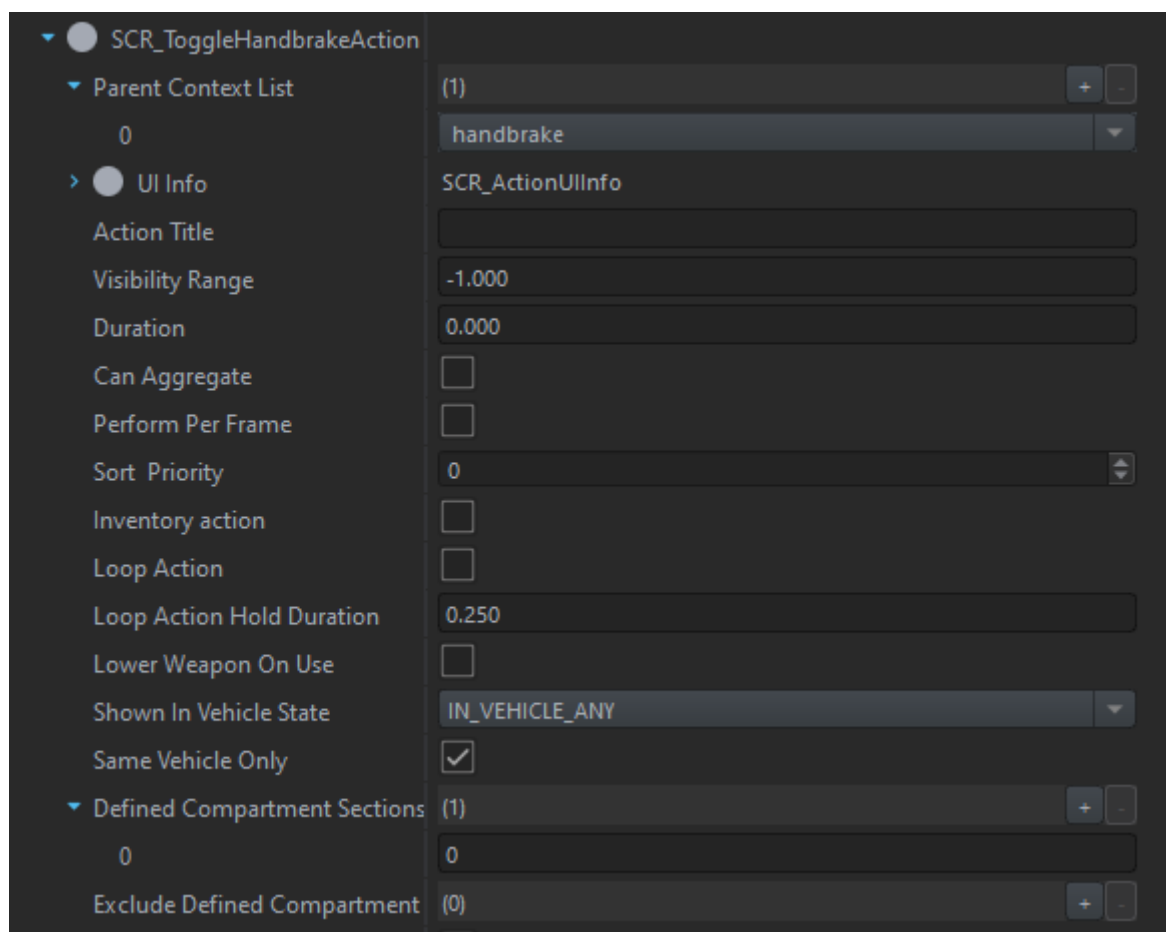
<code>Unflipper</code>	Action qui permet de retourner le véhicule depuis n'importe quel direction. Il faut la placer en x=0 y=1 z=0.
<code>fuel_cap</code>	Action qui permet de remplir le réservoir du véhicule. Placez la au niveau de la trappe à essence.
<code>engine_location</code>	Action qui permet de réparer le moteur du véhicule. Placez la au niveau d'une zone accessible par les joueurs, au niveau du moteur.
<code>hood_front</code>	Action qui permet de réparer le véhicule en globalité (boîte de vitesse, transmission...). Placez la au niveau d'une zone accessible par les joueurs, au niveau du capot en général.

### actions possibles réalisable par le conducteur:

Context Name	Position & description
<code>starter_switch</code>	Permet de démarrer le véhicule. Placez la au niveau du dispositif de démarrage du véhicule.
<code>light_switch</code>	Permet de contrôler les lumières du véhicule. Placez la au niveau du dispositif d'allumage des feux du véhicule. PS: Vous avez la possibilité de séparer l'action pour allumer les feux de route en créant une nouvelle action nommée: <code>light_HiBeam</code>
<code>light_hazard_switch</code>	Permet de contrôler les feux de détresse du véhicule. Placez la au niveau du dispositif d'allumage des feux de détresse du véhicule.
<code>handbrake</code>	Permet d'activer ou désactiver le frein parking du véhicule. Placez la au niveau du frein parking du véhicule.
<code>light_cabin_switch</code>	Permet d'activer ou désactiver les lumières intérieures du véhicule. Placez la au niveau du bouton pour allumer les lumières intérieures.

Une fois ces actions créés, il faut qu'on leurs affecte une tâche qu'elle vont réaliser une fois actionné par le joueur. Pour ce faire on va utiliser les Additional Actions. Une additional Action est simplement un script qui va être assigné a une ou plusieurs actions et qui va remplir définir ce que font les actions quand elle sont activés, en cours de chargement ou encore annulé par le joueur. C'est également dans les additional Actions qu'on va définir une liste de paramètre pour l'apparence des actions (texte traduit, icône, temps de chargement...).

Pour affecter une/des action(s) qu'on a crée plus haut, il suffit de l'ajouter dans la liste Parent Context List en faisant + et en sélectionnant l'action(s) concernée(s)



“ Pour des raisons de cohérence entre les véhicules, il est fortement conseillé de **prendre en modèle un véhicule d'Arma Reforger** et de faire clic droit puis copier sur une additional action dans un véhicule vanilla puis clic droit et coller dans sur une additional action, dans votre véhicule. Cela permet d'éviter d'avoir des soucis de configuration ou encore une cohérence dans les apparences de vos actions et celle des véhicules déjà existants.

**ATTENTION:** Lors du copier/coller, les class dans ce que vous avez copiée ne sont pas copiée, il faut copier/coller l'additional action et les class dans l'additional action ⚠

Les actions pour les portes et compartiments seront configurée plus tard dans un autre component.

Pour les véhicules, il en existe beaucoup, mais voici les plus importantes:

Action(s) a affecter	Nom de l'additional action	Description et informations
<code>fuel_cap</code>	<code>SCR_RefuelAtSupportStationAction</code>	Permet de remplir de carburant le véhicule.
Dans les préfabs de base pour les véhicules, ces additional actions sont déjà crée, il vous suffit de changer l'action déjà affecté par une action similaire ou la même avec une syntaxe différente.	<code>SCR_RepairAtSupportStationAction (x4)</code>	Permet de réparer certaines parties du véhicule en fonction de l'affectation de l'action.
<code>Unflipper</code> <code>Unflipper</code>	<code>SCR_UnflipVehicleAction (x2)</code>	Permet de retourner le véhicule si il se trouve sur le toit.
<code>push_front</code> , <code>push_rear</code>	<code>SCR_PushVehicleAction</code>	Permet de pousser le véhicule.
<code>handbrake</code> <code>handbrake</code>	<code>SCR_ToggleHandbrakeAction (x2)</code>	Permet de tirer le frein parking du véhicule.
<code>starter_switch</code> <code>starter_switch</code>	<code>SCR_EngineAction (x2)</code>	Permet de démarrer le véhicule.
Une action au choix au niveau d'une zone potentiel de stockage, exemple: <code>push_rear</code> <code>door_back</code>	<code>SCR_OpenVehicleStorageAction</code>	Permet d'ouvrir l'inventaire du véhicule.
Une action au choix au niveau d'une zone potentiel de stockage, exemple: <code>push_rear</code> <code>door_back</code>	<code>SCR_ResourceContainerVehicleLoadAction (x2)</code>	Permet en mode de jeu conflit, de placer du supply dans le véhicule. (Optionnel).

Vos actions sont désormais entièrement configurées et la plupart sont déjà prête à l'emploi ☐