

Documentation "InputManager"

I) Utilisation

Pour utiliser l'InputManager, dans vos scripts, vous devez inclure la ligne suivante :

using SimulatorInstitut;

C'est dans ce *namespace* que se trouve l'InputManager.

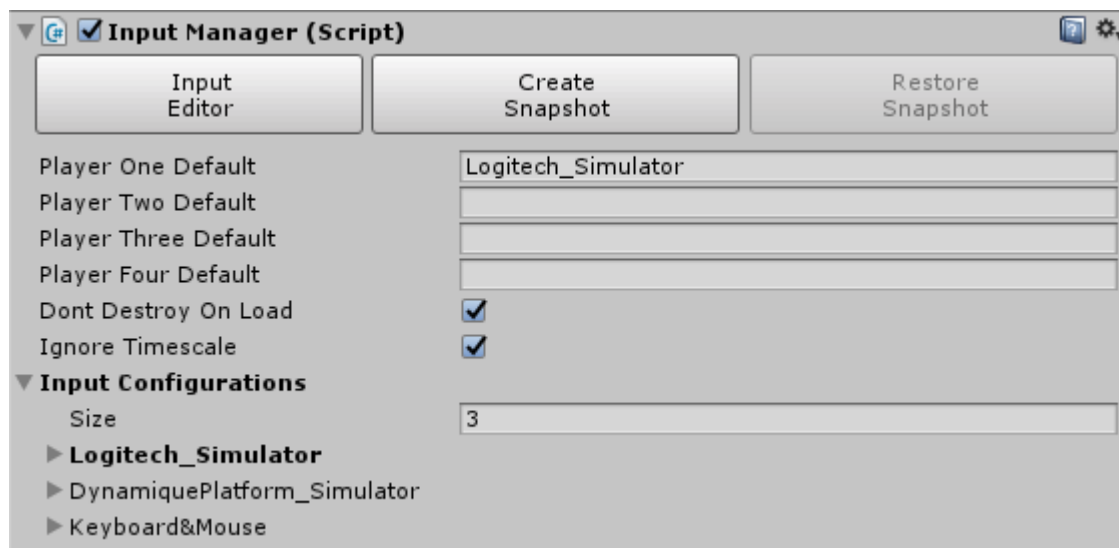
Pour utiliser les différentes fonctions de l'InputManager, vous devez faire précéder chaque nom de fonction de "*SimulatorInstitut.InputManager*". Les fonctions possibles sont :

- Reinitialize()
- ResetInputConfiguration(PlayerID playerID)
- GetInputConfiguration(string name)
- GetInputConfiguration(PlayerID playerID)
- GetAxis(string name, PlayerID playerID = PlayerID.One)
- GetAxisRaw(string name, PlayerID playerID = PlayerID.One)
- GetButton(string name, PlayerID playerID = PlayerID.One)

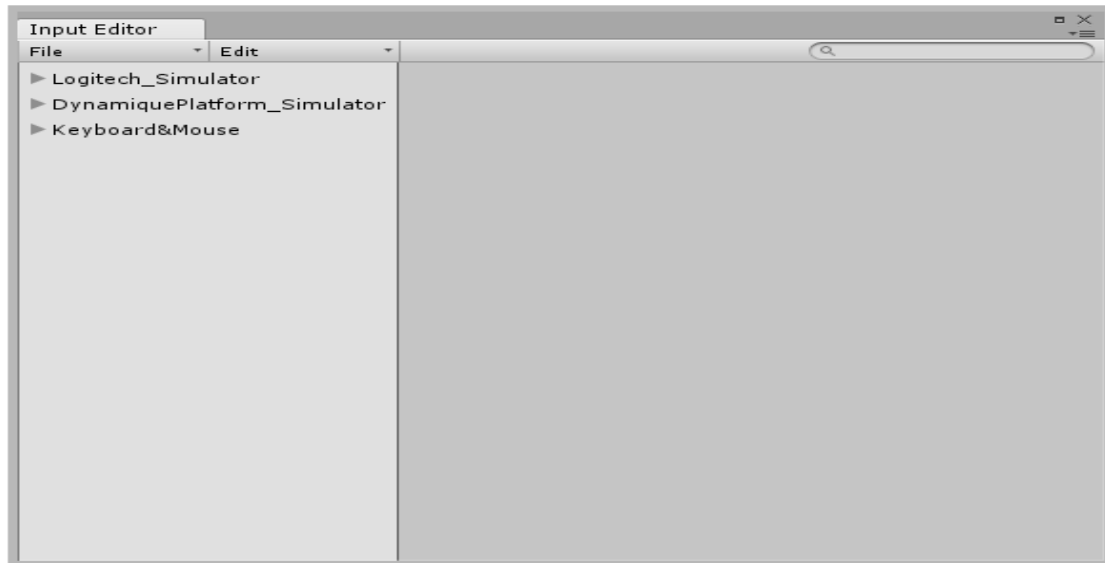
II) Ajout d'une configuration

Pour commencer, mettre le prefab "InputManager" dans votre scène. S'il n'y a pas de prefab créer un objet vide et mettre le script "InputManager" dessus.

On obtient donc un objet symbolisant notre InputManager dans notre scène avec le menu suivant :



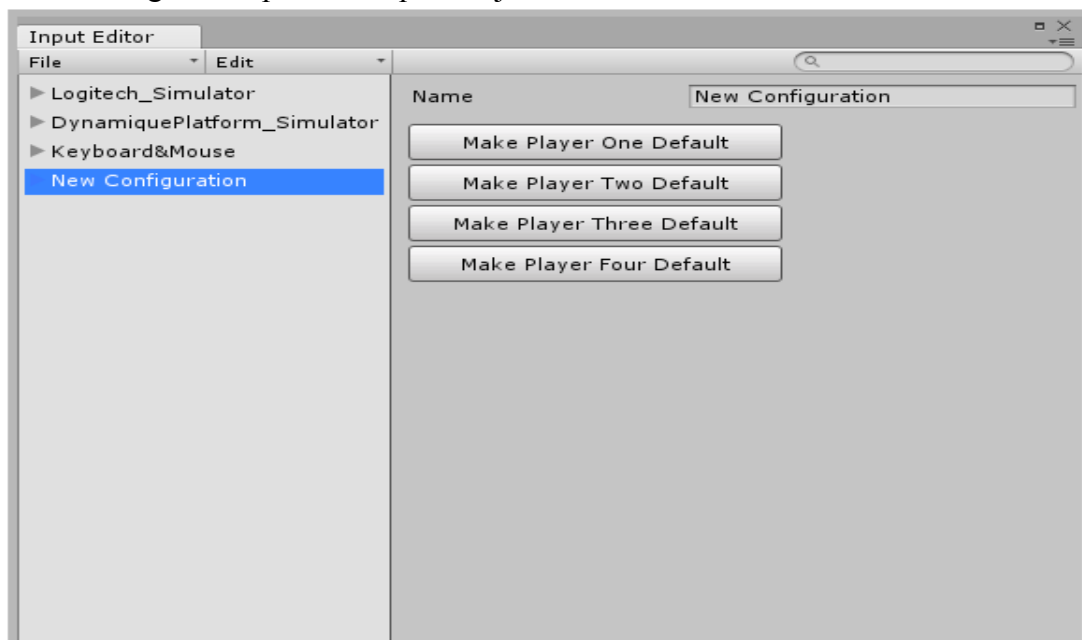
Nous avons différentes informations à notre disposition, comme les différentes configurations par défaut pour les joueurs et la liste des configurations. Ce qui nous intéresse est le bouton "Input Editor" qui va nous permettre de créer de nouvelles configurations.



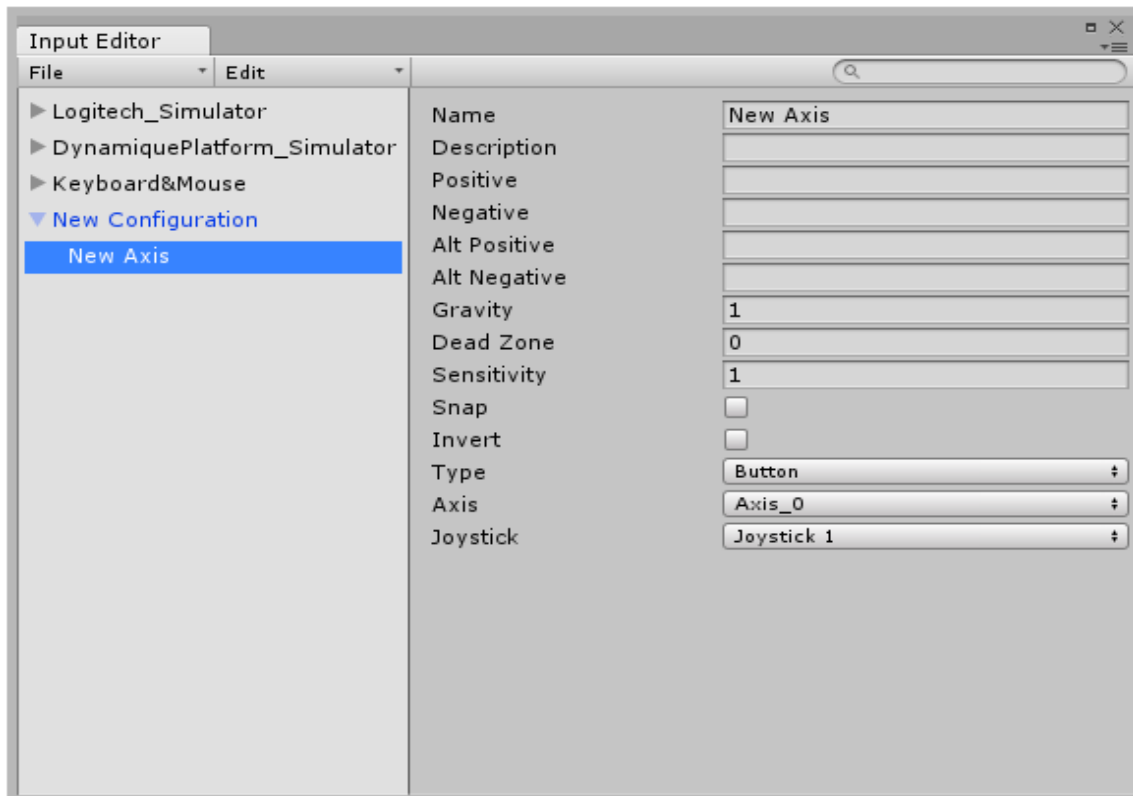
Le sous-menu "File" permet d'importer/d'exporter les configurations et de réinitialiser les configurations.

Le sous-menu "Edit" permet de créer de nouvelles configurations et de nouvel axe pour ces configurations.

Pour créer une configuration, on va donc faire Edit>New Configuration. On choisit ensuite un nom pour la nouvelle configuration. En option, on peut choisir s'il faut que la configuration devienne la configuration par défaut pour le joueur voulu.



On ajout ensuite les axes à cette configuration en faisant Edit>New Axis. Ensuite, on paramètre l'axe comme on le souhaite.



Une fois notre nouvelle configuration terminée, il faut exporter la nouvelle liste de configuration dans le dossier "StreamingAssets" de l'application, en remplaçant le fichier "input_config.xml". Ainsi l'InputManager chargera bien la nouvelle configuration au démarrage de l'application.

III) Ajout d'un périphérique

Pour ajouter un nouveau périphérique, il faut au préalable avoir une DLL (en 64bit de préférence) ou un script C# qui contrôle ce périphérique. Si vous avez fait une DLL, il faut faire un *wrapper* en C# pour la DLL.

A ce stade, vous devez avoir une classe en C# qui contrôle le nouveau périphérique. Il faut que cette classe hérite de la classe "CustomJoystick". C'est la classe générale qui regroupe tous les périphériques que l'InputManager doit contrôler.

La nouvelle classe doit implémenter quatre nouvelles fonctions :

- *"public override void InitCustomJoystick()"* -> Démarre la communication avec le périphérique.
- *"public override void StopCustomJoystick()"* -> Coupe la communication avec le périphérique.
- *"public override double getJoystickActionState(string action)"* -> Retourne le statut de l'action demandée en paramètre.
- *"public override string getName()"* -> Retourne le nom du périphérique.

Une fois la nouvelle classe terminée, avec les fonctions ci-dessus correctement implémentées, il faut modifier le fichier "InputManager.cs" qui se trouve dans le dossier Script/InputManager/RunTime/ .

Dans ce fichier repérer le bloc d'instruction suivante :

```
if(string.Equals(_playerOneConfig.name, "Logitech_Simulator")){  
    joystickDevicePlayerOne = new LogitechJoystick ();  
}  
else if (string.Equals(_playerOneConfig.name, "DynamiquePlatform_Simulator")){  
    joystickDevicePlayerOne = new DynamiquePlatformJoystick();  
}  
else{  
    joystickDevicePlayerOne = new KeyboardJoystick();  
}
```

Dans ce bloc d'instruction vous devez rajouter après le dernier *else if* :

```
else if (string.Equals(_playerOneConfig.name, "Nom_Nouvelle_Config")){  
    joystickDevicePlayerOne = new Classe_Nouvelle_Config();  
}
```

Vous remplacer le *Nom_Nouvelle_Config* par le nom de la nouvelle configuration que vous avez donnée dans l'Input Editor, et la *Classe_Nouvelle_Config* par le nom de la classe qui gère le nouveau périphérique.