



Unity Basics

IL FAUT BIEN PARTIR DE QUELQUE PART
HEIN ?

C'est quoi Unity ?

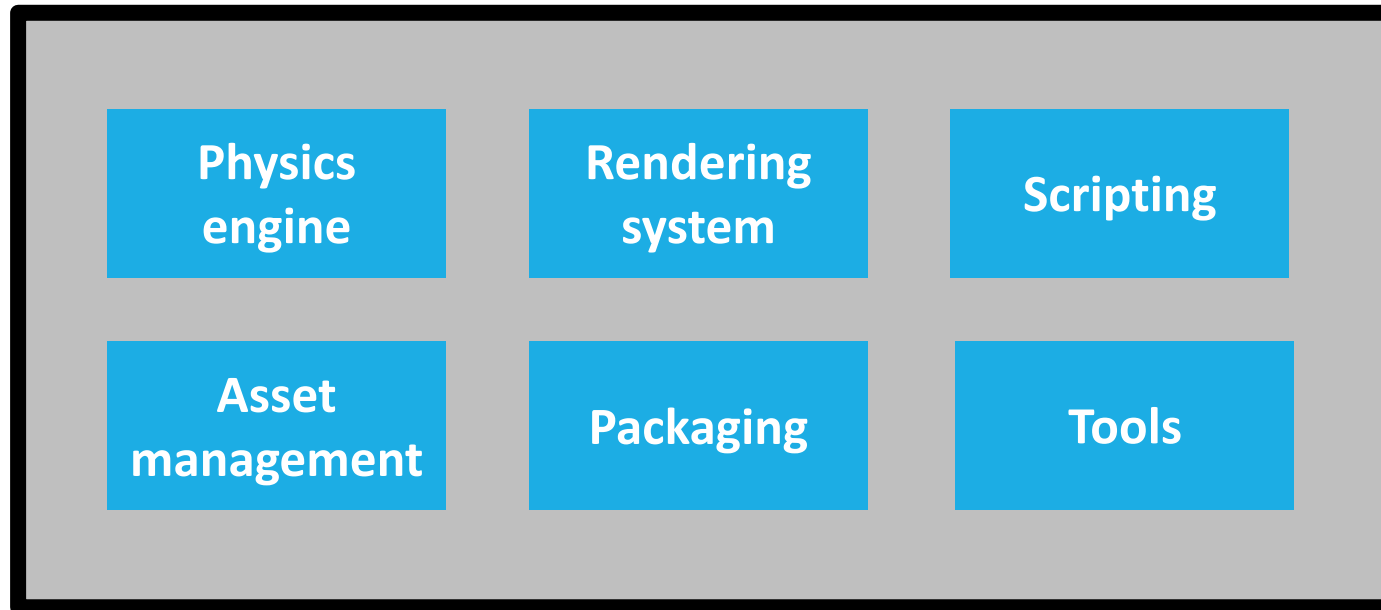
C'est quoi Unity ?

Un moteur de
jeux vidéo

C'est quoi un moteur de jeux vidéo ?

The Unity logo is displayed in white text on a black rectangular background. The word "Unity" is written in a bold, sans-serif font, with the 'y' having a distinctive upward curve.

C'est quoi un moteur de jeux vidéo ?



Pourquoi
apprendre ça ?

Pourquoi apprendre ça ?

- Un jeu mobile sur deux est produit avec Unity
- Plus de 75% des applications et jeux VR sont produits avec Unity
- C'est une techno cross-platform (on en reparlera)
- Le temps réel avec Unity, c'est trop cool*

*Vous avez le droit de considérer cet argument comme subjectif



La Philosophie de Unity

MAITRISEZ LA, ET VOUS SEREZ INARRÊTABLE*

** Ce n'est même pas une blague*

La Philosophie de Unity

- Un éditeur léger
- Des outils modulaires
- Une facilité de personnalisation
- Construit autour de 4 points fondamentaux

Les fondamentaux de Unity

1. Scenes

2. GameObjects

3. Components

4. Assets

La Scene

Une **Scene** est un espace
virtuel sur lequel vous pouvez
mettre des « trucs » dessus

La Scene

- Vous pouvez avoir plusieurs scènes dans un projet
 - Utilisez-les pour classer / catégoriser votre contenu
- Vous pouvez passer d'une Scene à une autre
- Les Scenes sont la colonne vertébrale de vos projets

Unity

Le GameObject

**Un GameObject est un
élément sur la Scene**

Le GameObject

- Un **GameObject** est une référence dans l'espace virtuel de la Scene
- Un **GameObject** peut bouger sur une **Scene** ou rester immobile
- Vous pouvez mettre autant de **GameObjects** que vous voulez sur une **Scene**

Le Component

Un **Component** est attaché à
un **GameObject** afin de lui
donner un comportement

Le Component

- Unity est livré avec plein de **Components** prêts à l'emploi
- Vous pouvez créer vos propres **Components** pour ajouter les mécaniques dont vous avez besoin

Les Assets

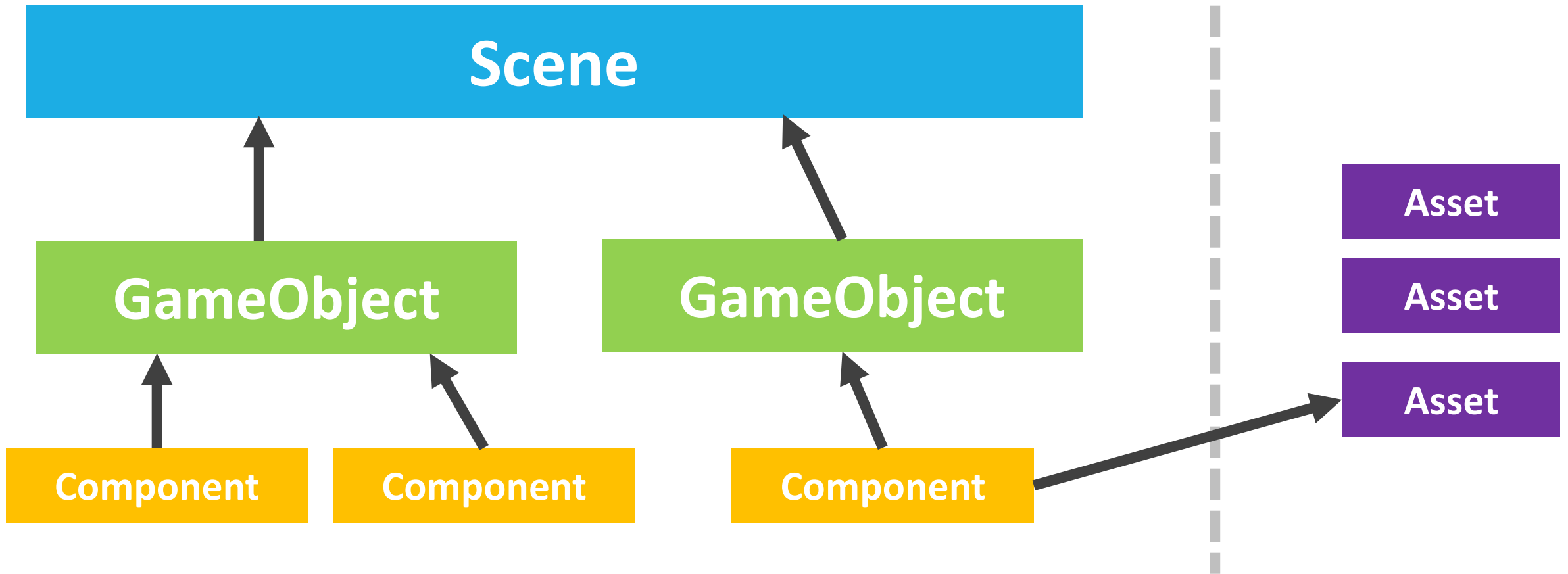
Un **Asset** sont une ressources
que vous ajoutez à votre projet

Unity

Les Asset

- Un Asset peut prendre plusieurs formes :
 - Une texture
 - Un modèle 3D
 - Un son
 - Un script
 - ...

Les Fondamentaux



Les Fondamentaux

Un **GameObject** a TOUJOURS
un **Component** bien précis
appelé « *Transform* »

Les Fondamentaux

- Le **Transform** stocke des infos spatiales du **GameObject** auquel il est attaché :
 - Sa position
 - Sa rotation
 - Son échelle

Les fenêtres principales

Les fenêtres principales

- **Project** : Affiche vos Assets
- **Scene** : La représentation 3D du contenu de votre scène courante
- **Hierarchy** : La représentation sous forme d'arbre du contenu de votre scène courante
- **Inspector** : Affiche les détails de l'élément sélectionné
- **Game** : C'est là que vous pourrez tester votre prod depuis l'éditeur

Les fenêtres Unity

- Toutes les fenêtres de l'éditeur peuvent être ouvertes et/ou fermées
- Vous pouvez les ancrer dans la disposition où vous voulez ou en faire des fenêtres flottantes
- Vous pouvez enregistrer des dispositions pour les recharger plus rapidement ensuite

Raccourcis utiles

- Outil de déplacement : W
- Outil de rotation : E
- Outil d'échelle : R
- Aimanter sur la grille : CTRL + tool
- Focus sur l'élément sélectionné : F*



*Sert aussi à rendre hommage, mais pas dans Unity

Les Primitives

- Les Primitives sont des formes 3D simples que vous pouvez rapidement ajouter sur une scène :
 - Cube
 - Sphere
 - Cylinder
 - Capsule
 - Quad
 - Plane

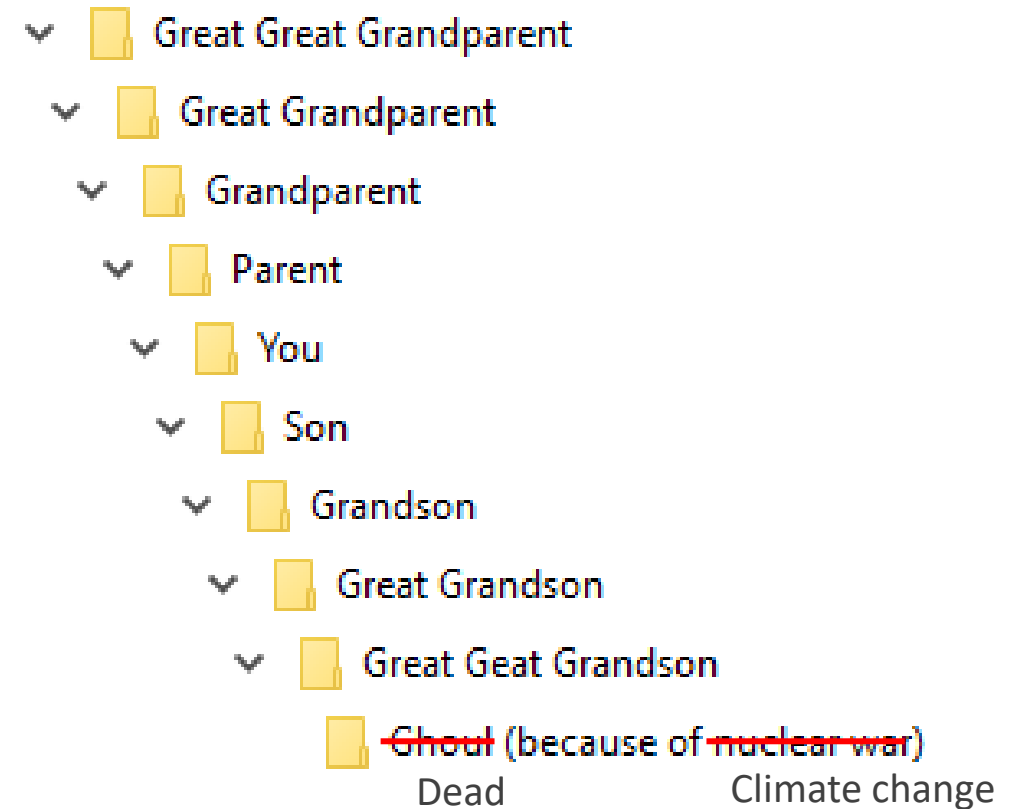
Now the real stuff

LE TITRE REND MIEUX EN ANGLAIS



La hierarchie

Peut servir de
« dossiers » sur
votre Scene



Espace Relatif vs absolu

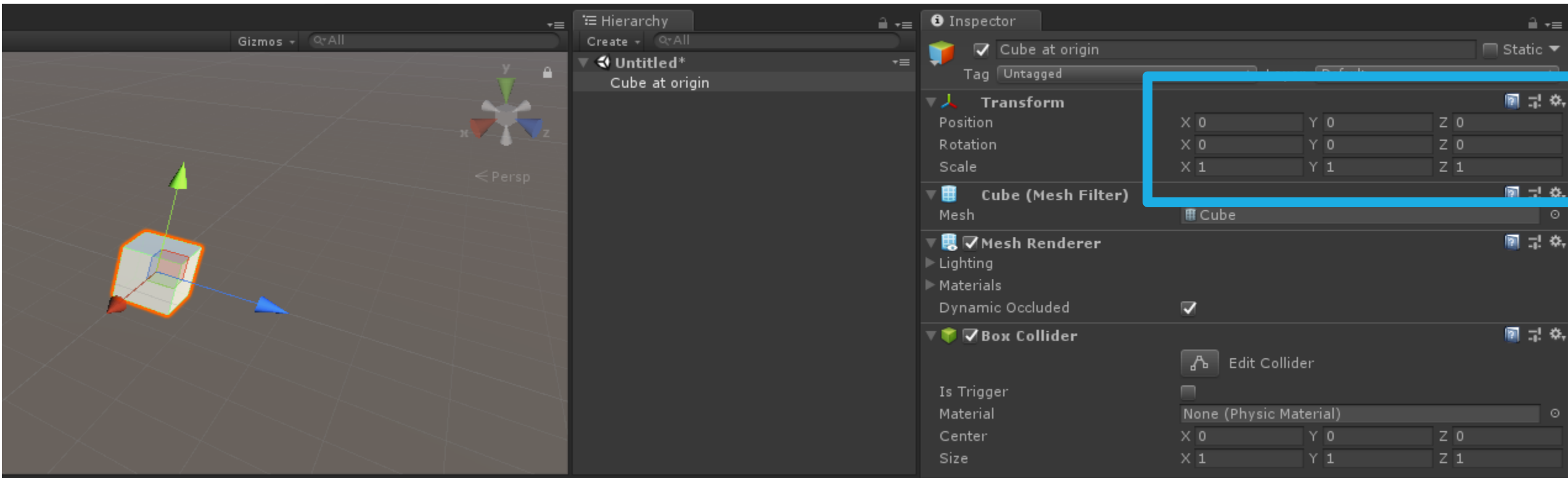
GameObject à la racine

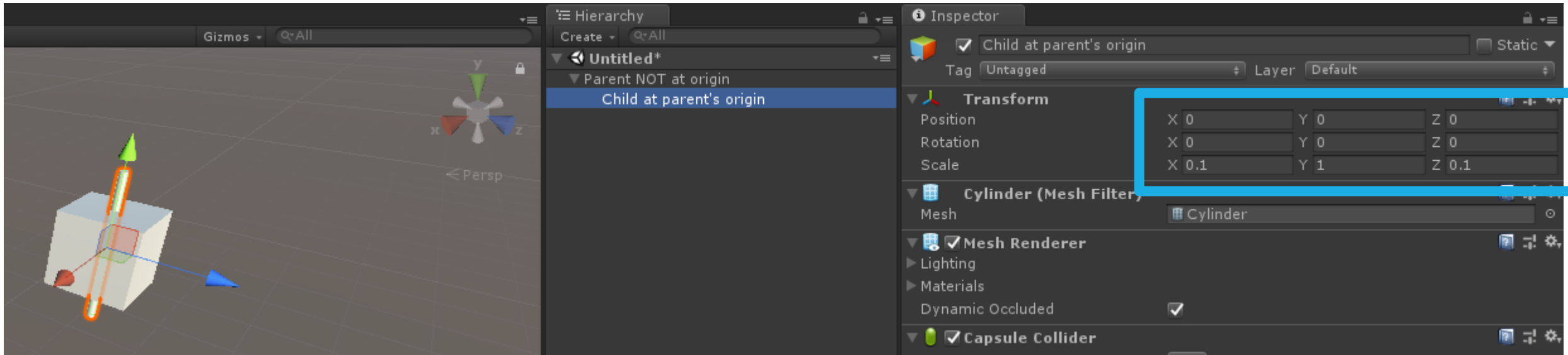
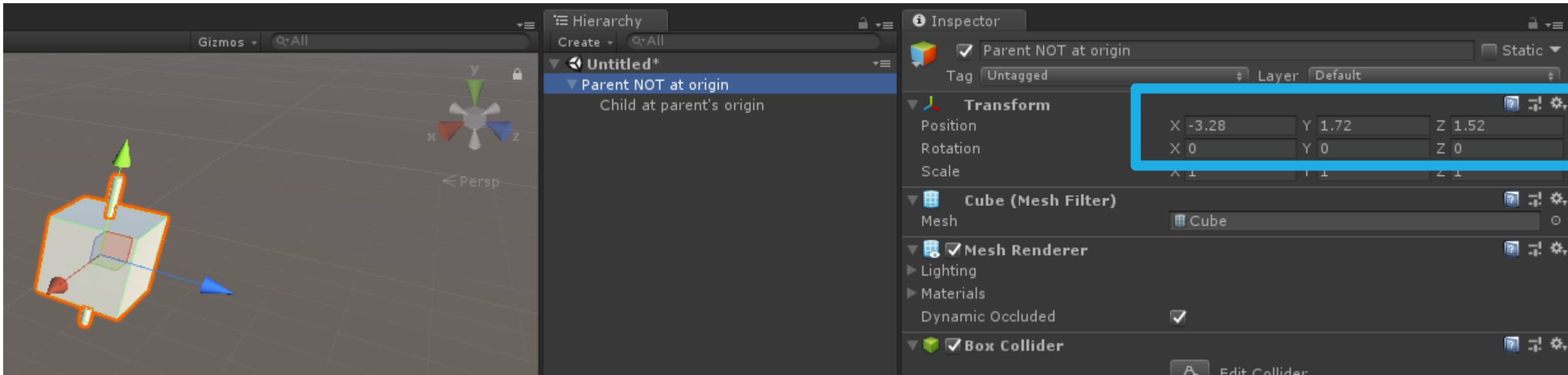
Les valeurs du Transform sont
relatives à la scène
(valeurs absolues)

GameObject enfant d'un autre GameObject

Les valeurs du Transform sont
relatives au Transform parent
(valeurs relatives)

Espace Relatif vs absolu



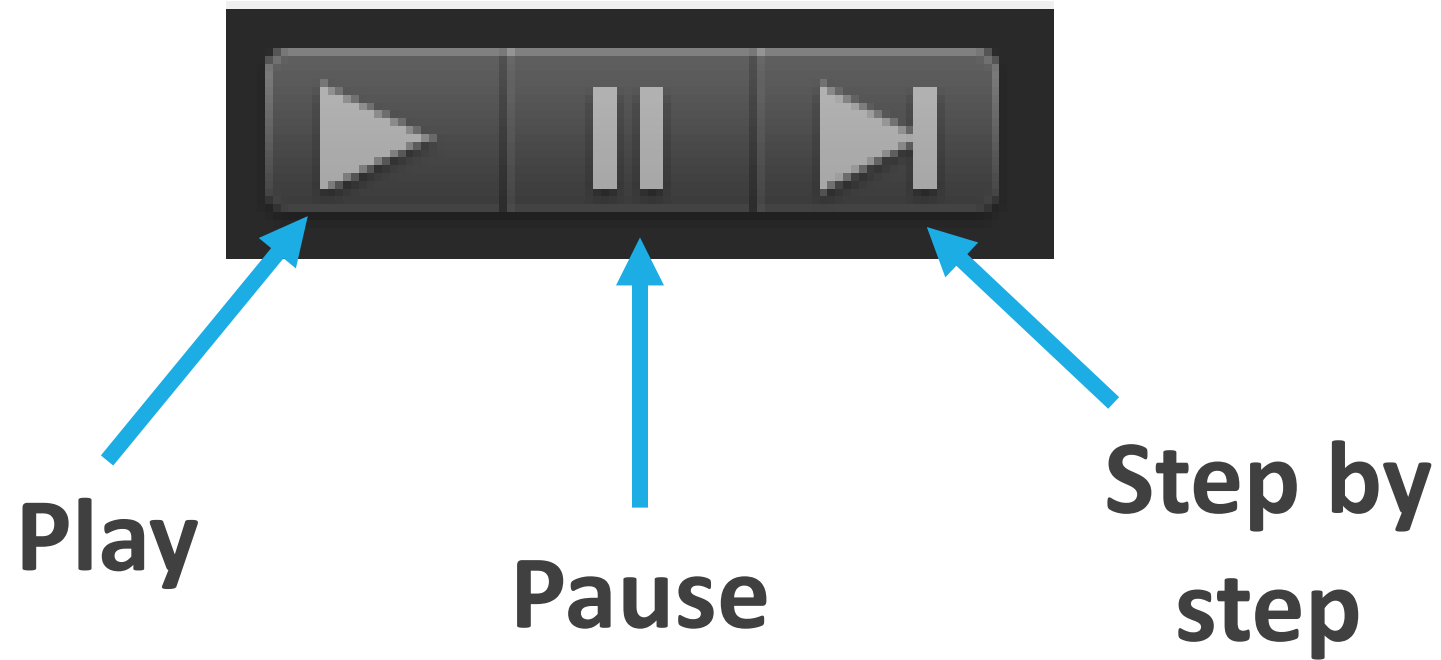


Espace Relatif vs absolu

En bougeant un parent, tous les enfants
bougeront aussi sur la scène

Par contre, la position relative de l'enfant
par rapport à son parent restera la même

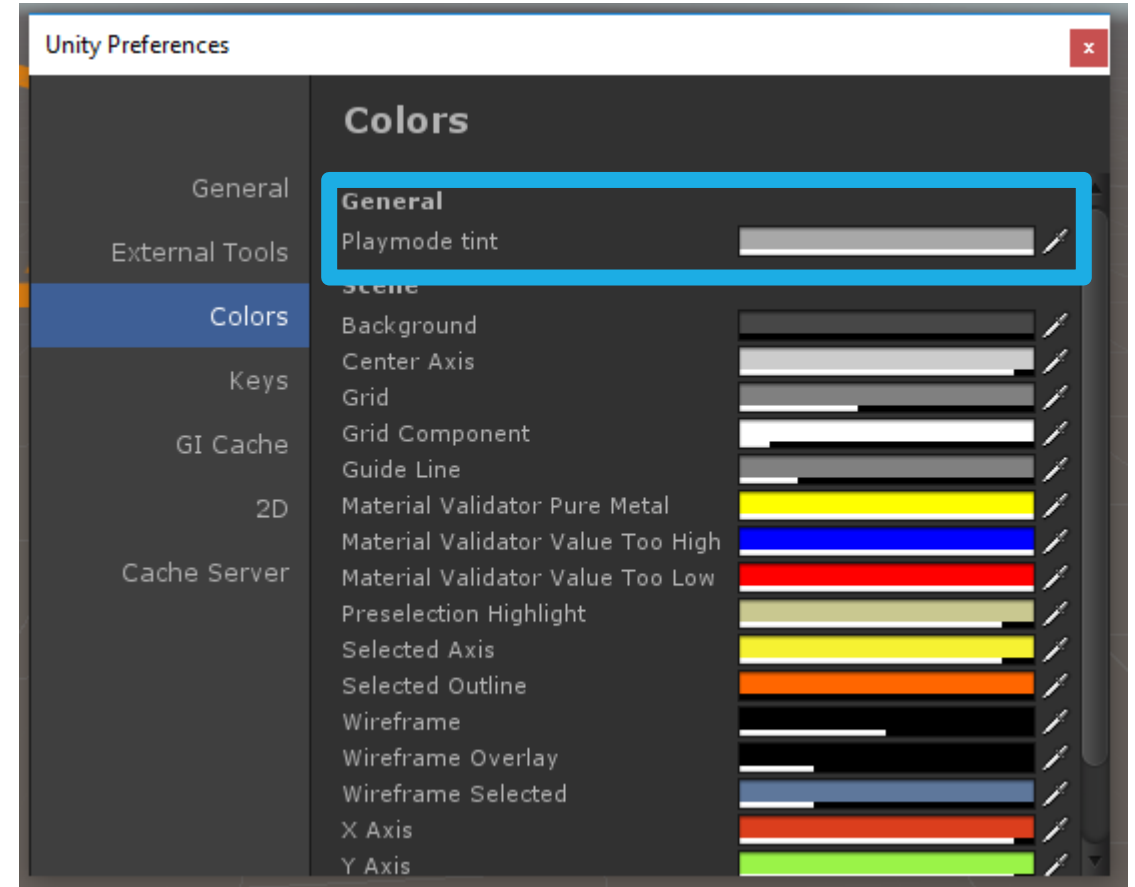
Le mode Play



Le mode Play



**Play-mode
in progress**



Le Scripting dans Unity

Comment ça marche

Unity a une boucle interne de rendu

```
while(true)
{
    // Do things
    // Do other things
    // ...
}
```

Comment ça marche

- Le moteur gère différentes chose à différents moments
- Vous pouvez « brancher » votre code à des moments précis de l'exécution du moteur pour faire ce que vous avez à faire

La classe MonoBehaviour

- **MonoBehaviour** est la classe mère de tous les components dans Unity
- Vous pouvez construire vos propres composants en créant une classe qui hérite de **MonoBehaviour**

Les « messages » MonoBehaviour

- Le moteur va essayer d'appeler des fonctions sur les MB en fonction de son ordre d'exécution
- Voici les plus courants :
 - Start : appelé une fois au lancement
 - Update : appelé à chaque frame de rendu (~60x/seconde ou plus)

Liste complète de l'ordre d'exécution ici : <https://docs.unity3d.com/Manual/ExecutionOrder.html>

```
public class BestScriptEver : MonoBehaviour
{
    0 references
    private void Start()
    {
        Debug.Log("Hello! My name is" + name);
    }

    0 references
    private void Update()
    {
        Debug.Log("Beep");
    }
}
```


Héritage de MonoBehaviour

```
public class BestScriptEver : MonoBehaviour
{
    0 references
    private void Start()
    {
        Debug.Log("Hello! My name is" + name);
    }

    0 references
    private void Update()
    {
        Debug.Log("Beep");
    }
}
```

Propriété héritée du
parent

Start est appelée
UNE FOIS

Update est
appelée à
CHAQUE FRAME

```
public class BestScriptEver : MonoBehaviour
{
    0 references
    private void Start()
    {
        Debug.Log("Hello! My name is" + name);
    }

    0 references
    private void Update()
    {
        Debug.Log("Beep");
    }
}
```

Créer un Script dans Unity

1. Click droit sur la fenêtre
2. Aller dans le sous menu « Create »
3. Cliquer sur « C# Script
4. Donner un nom au script*
5. Double cliquer sur l'asset créé pour ouvrir Visual Studio

* ATTENTION : pas d'espaces, pas de caractères spéciaux dans le nom des composants

Scripting API

UnityEngine

- + UnityEngine.Accessibility
- + UnityEngine.AI
- + UnityEngine.Analytics
- + UnityEngine.Android
- + UnityEngine.Animations
- + UnityEngine.Apple
- + UnityEngine.Assertions
- + UnityEngine.Audio
- + UnityEngine.CrashReportHandler
- + UnityEngine.Diagnostics
- + UnityEngine.Events
- + UnityEngine.Experimental
- + UnityEngine.iOS
- + UnityEngine.Jobs
- + UnityEngine.LowLevel
- + UnityEngine.Lumin
- + UnityEngine.Networking
- + UnityEngine.ParticleSystemJobs
- + UnityEngine.Playables
- + UnityEngine.PlayerLoop

MonoBehaviour

class in UnityEngine / Inherits from: [Behaviour](#) / Implemented in: [UnityEngine.CoreModule](#)

[Leave feedback](#)

Description

MonoBehaviour is the base class from which every Unity script derives.

When you use C#, you must explicitly derive from MonoBehaviour.

This class doesn't support the [null-conditional operator](#) `?.` and the [null-coalescing operator](#) `??`.

For code samples, see the individual MonoBehaviour methods.

Note: There is a checkbox for enabling or disabling MonoBehaviour in the Unity Editor. It disables functions when unticked. If none of these functions are present in the script, the Unity Editor does not display the checkbox. The functions are:

[Start\(\)](#)
[Update\(\)](#)
[FixedUpdate\(\)](#)
[LateUpdate\(\)](#)
[OnGUI\(\)](#)
[OnDisable\(\)](#)
[OnEnable\(\)](#)

See Also: The [Deactivating GameObjects](#) page in the manual.

La documentation officielle est détaillée

The answer to life, the universe
and everything, lies in the Unity
Documentation

... and sometimes in the Unity Answers board

Liens utiles

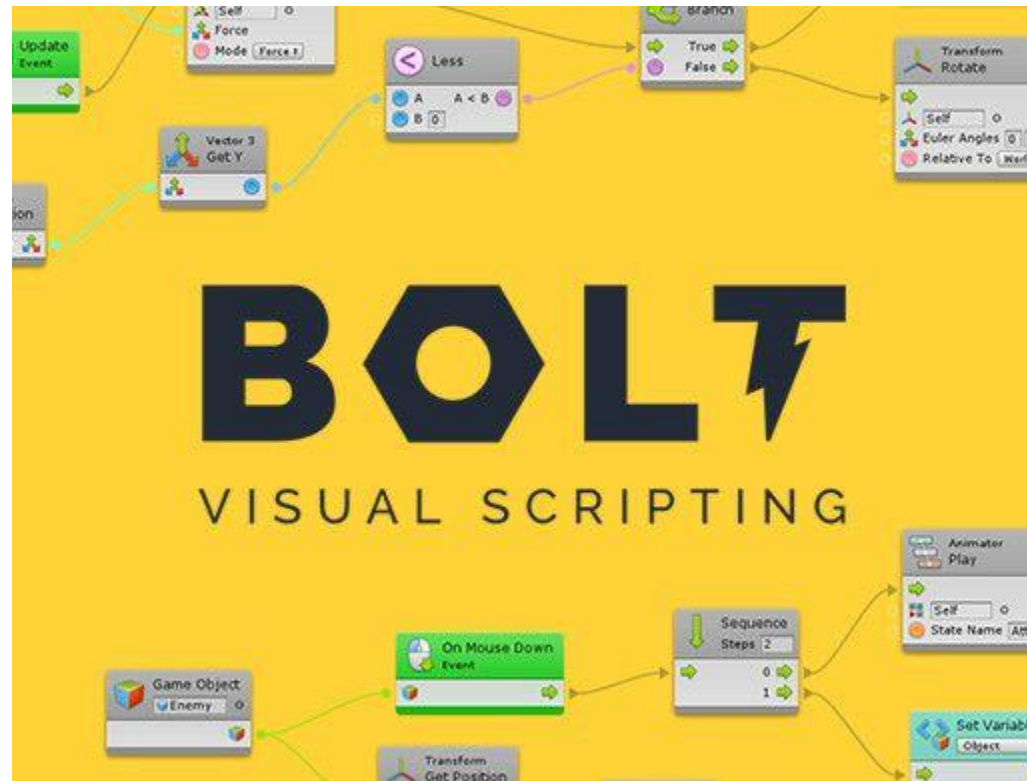
➤ Documentation:

- Unity Manual: <https://docs.unity3d.com/Manual/index.html>
- Scripting API: <https://docs.unity3d.com/ScriptReference/index.html>
- Faites attention à la version que vous utilisez lorsque vous regardez la documentation en ligne

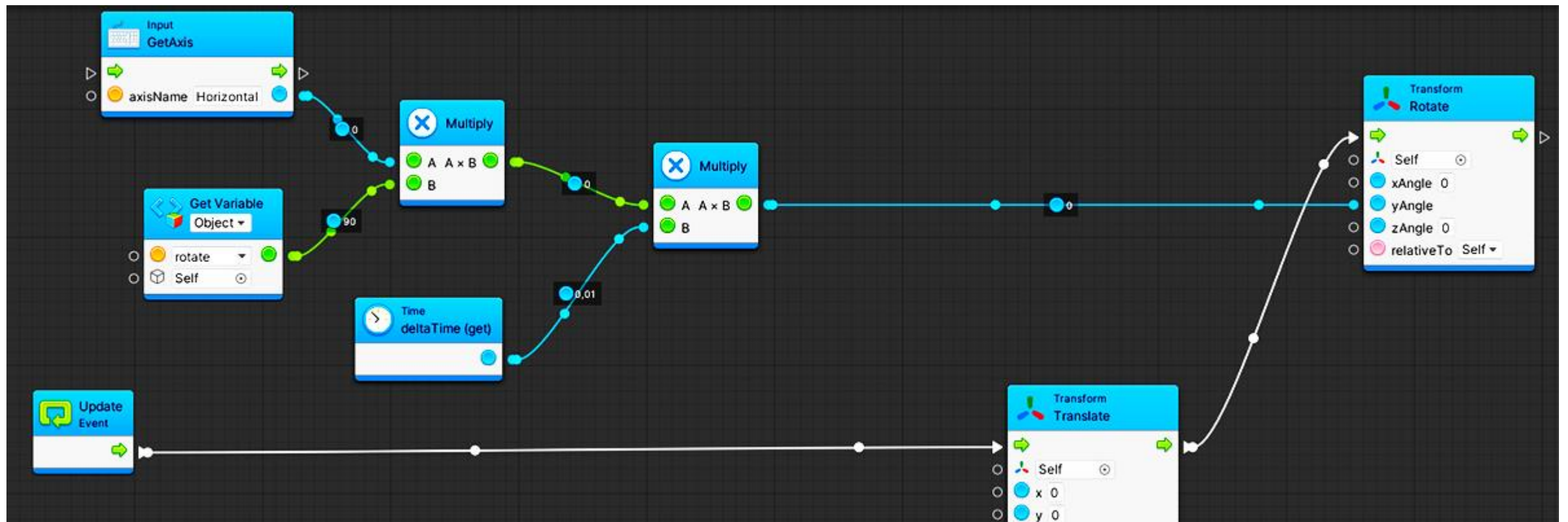
➤ Unity Answers:

- C'est grosso modo une FAQ
- Si la réponse à votre problème n'est pas dans la documentation, vous avez 99% de chances de la trouver ici
- <https://answers.unity.com/index.html>

Approche « No-Code » avec Bolt



Approche « No-Code » avec Bolt



GO
