Le son

Jouer des sons dans Unity

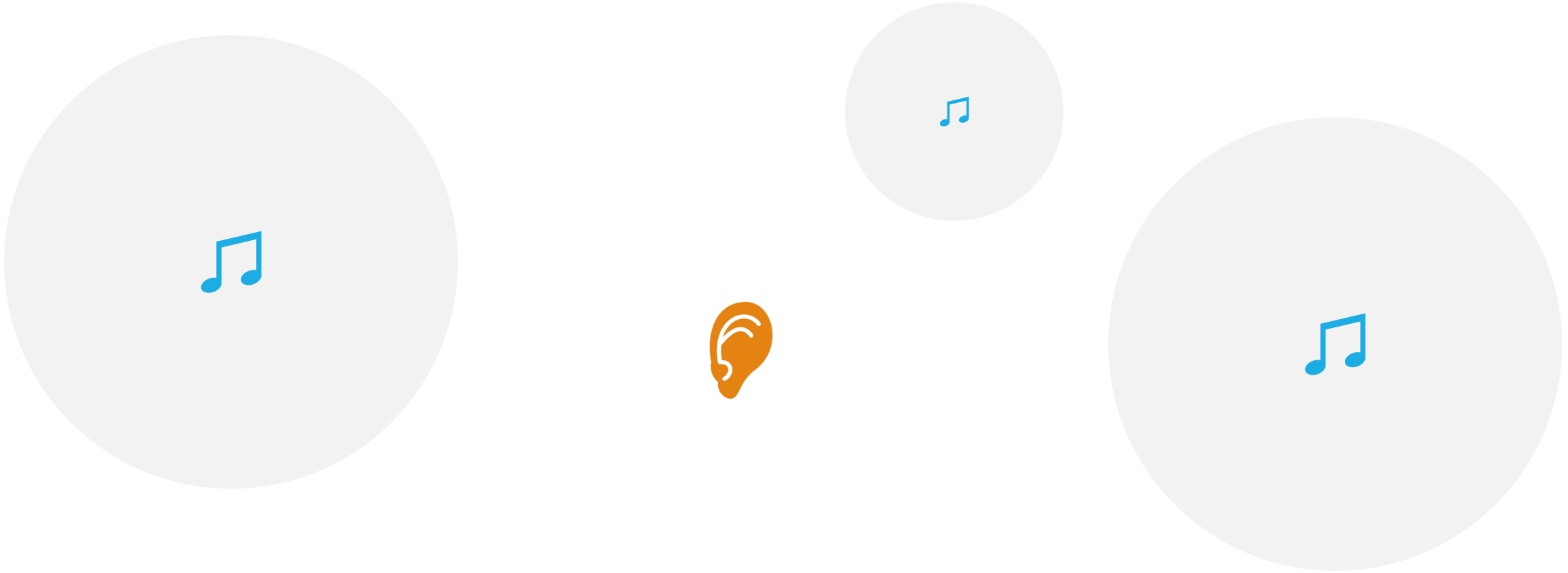
1. Une « oreille »
2. Un son

Jouer des sons dans Unity

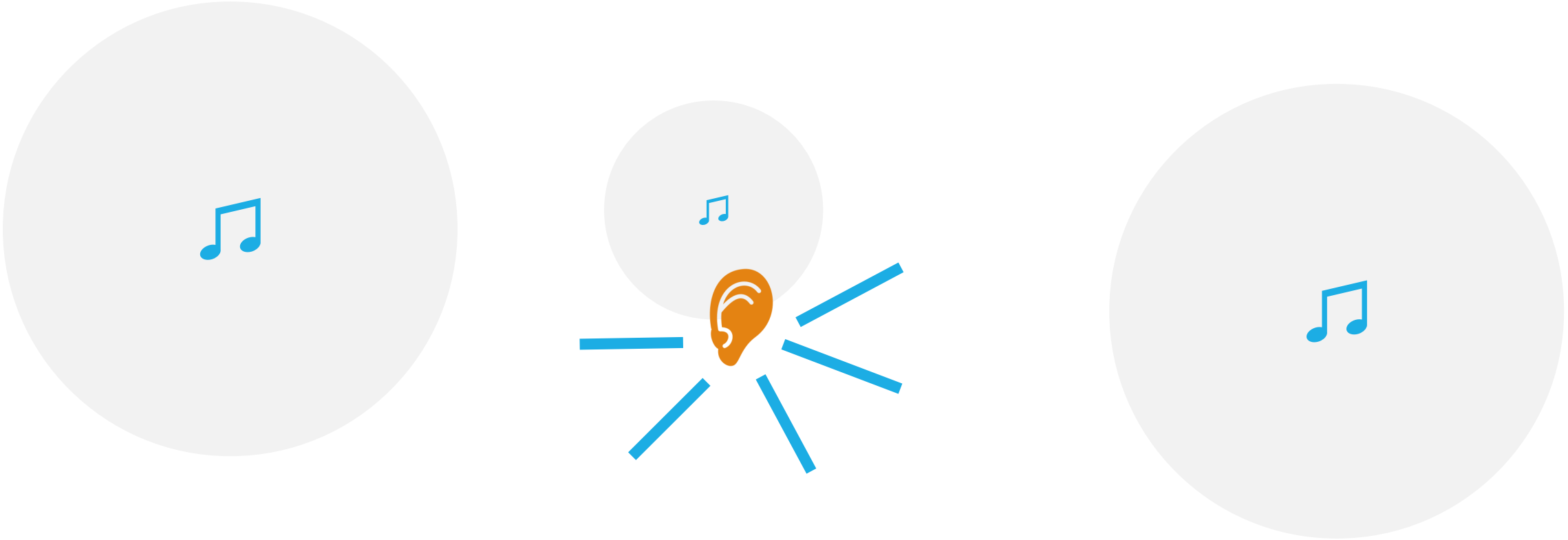
1. Une « oreille » ➔ AudioListener

2. Un son ➔ AudioSource

Les sons dans l’espace



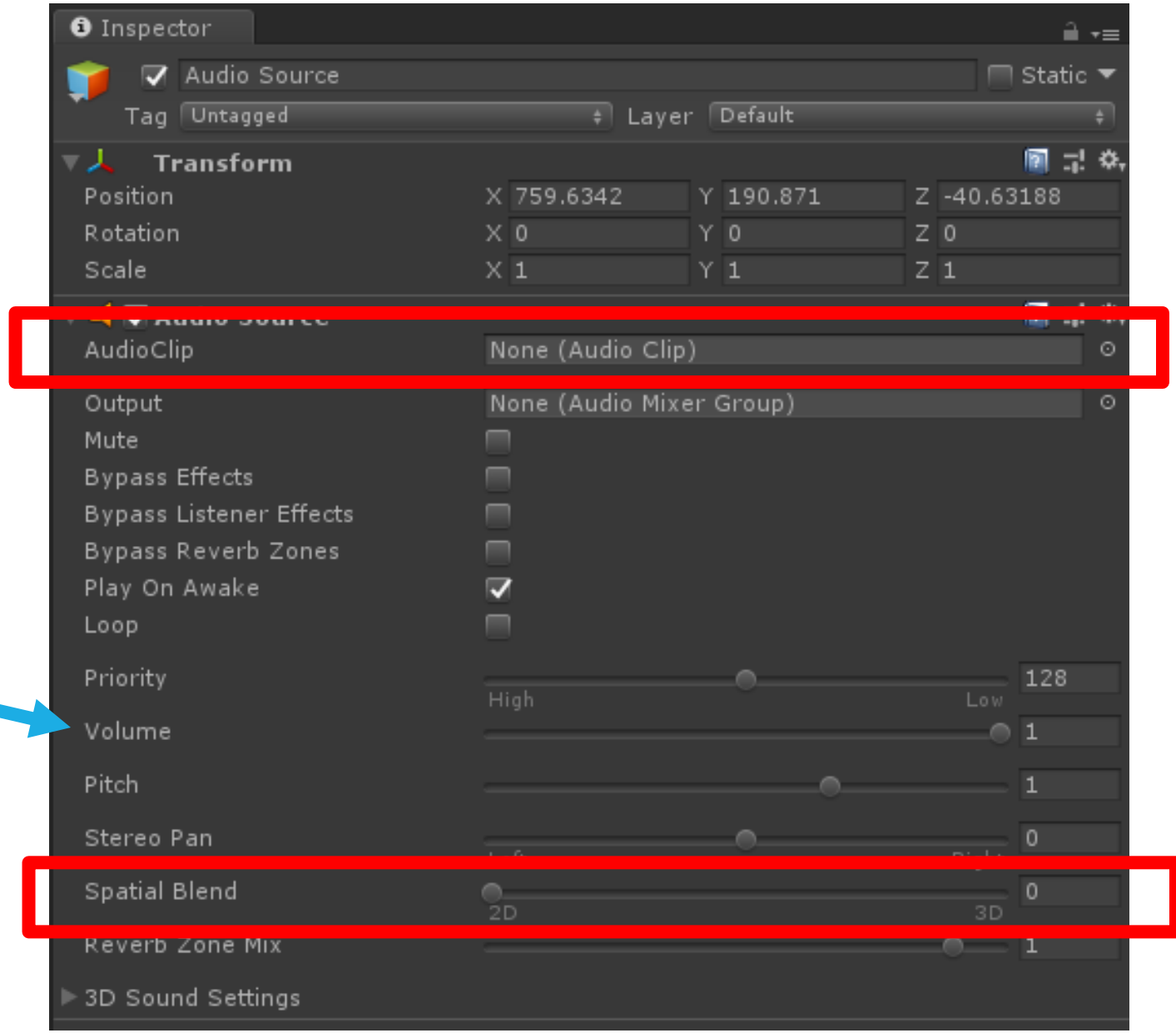
Les sons dans l’espace



L’AudioClip est le fichier sonore à jouer

Gérez le volume ici

Un son peut être « 2D » ou « 3D »



Gérer plus de 500 sons dans un projet

Question

Comment est-ce que vous feriez pour permettre à l'utilisateur de gérer le volume sonore ?

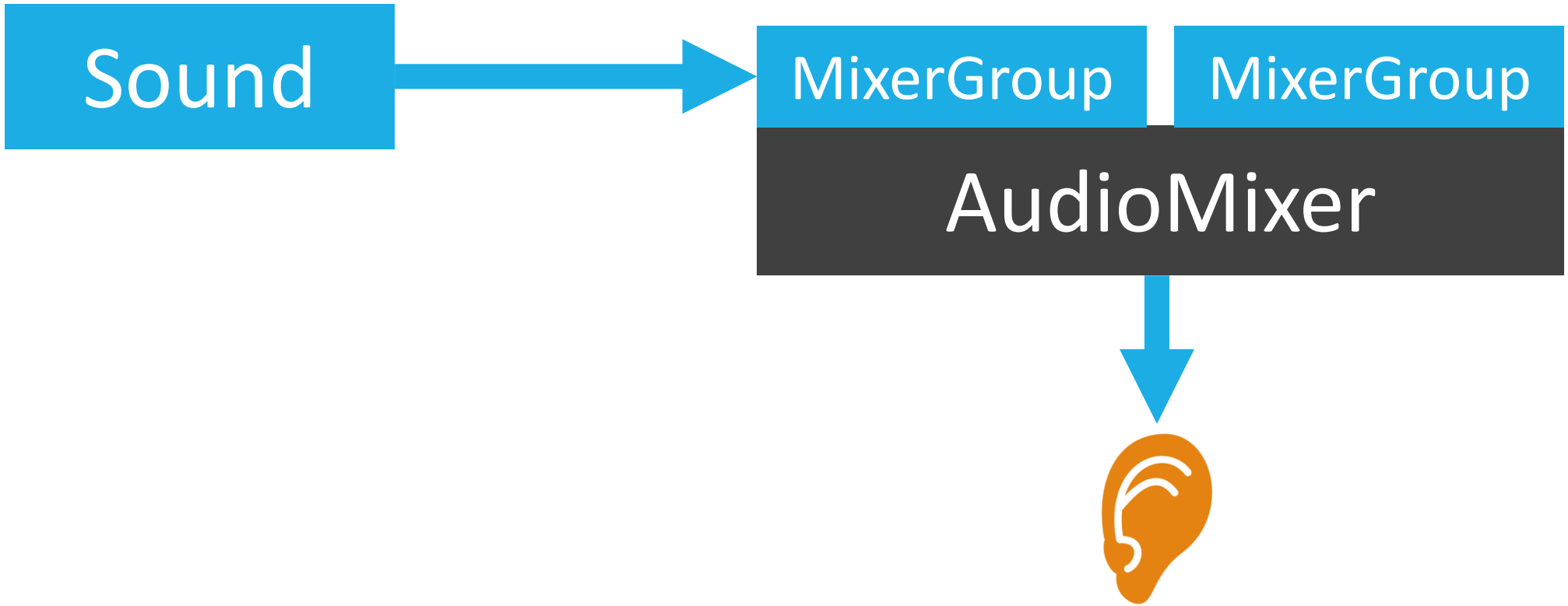
L’AudioMixer à la rescousse

IN

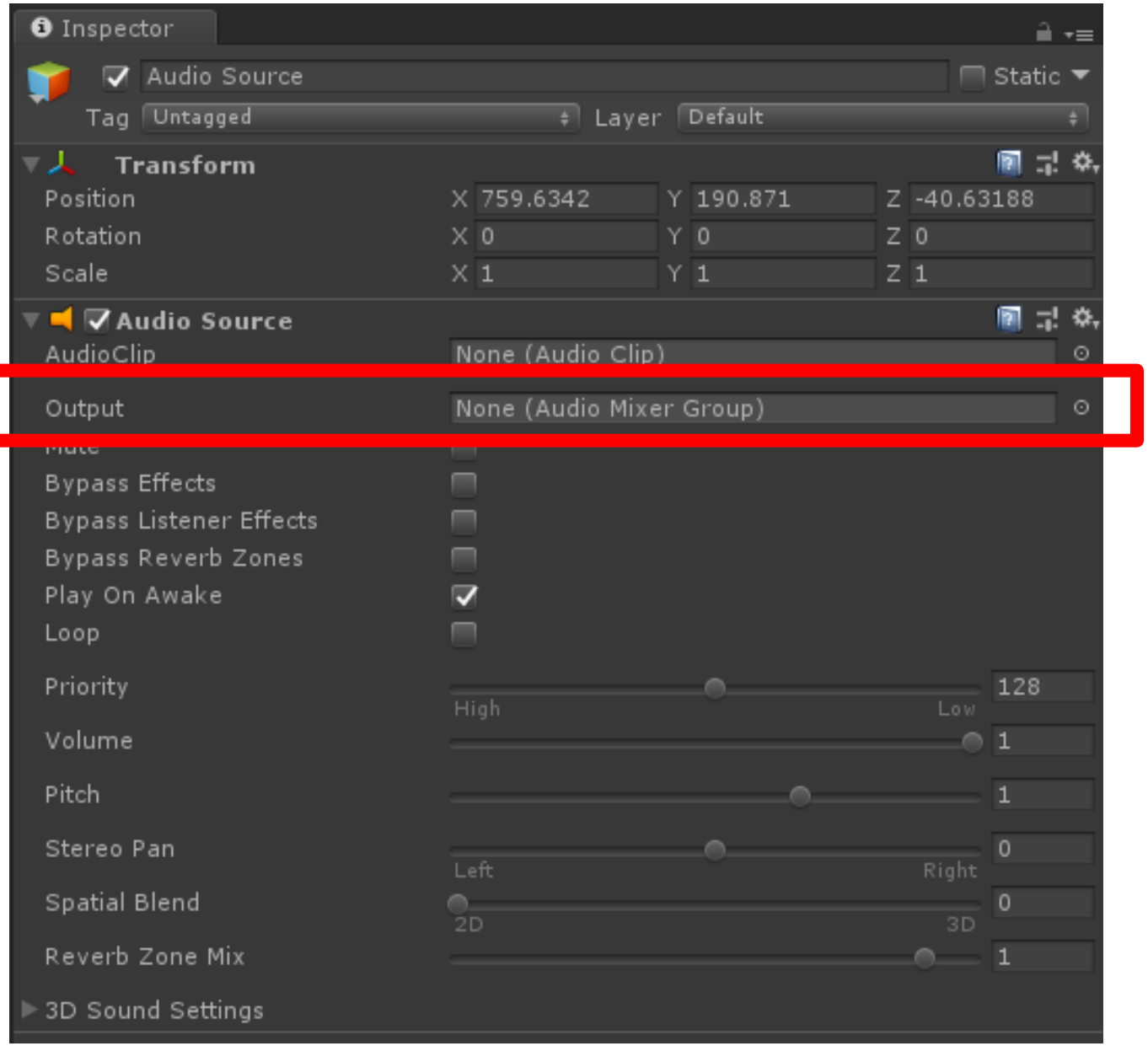
OUT

L’AudioMixer à la rescousse

L'AudioMixer à la rescousse

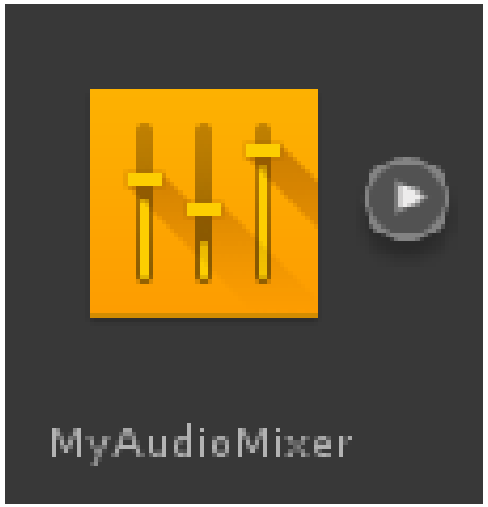


La sortie de l'AudioSource… c'est le groupe du mixer



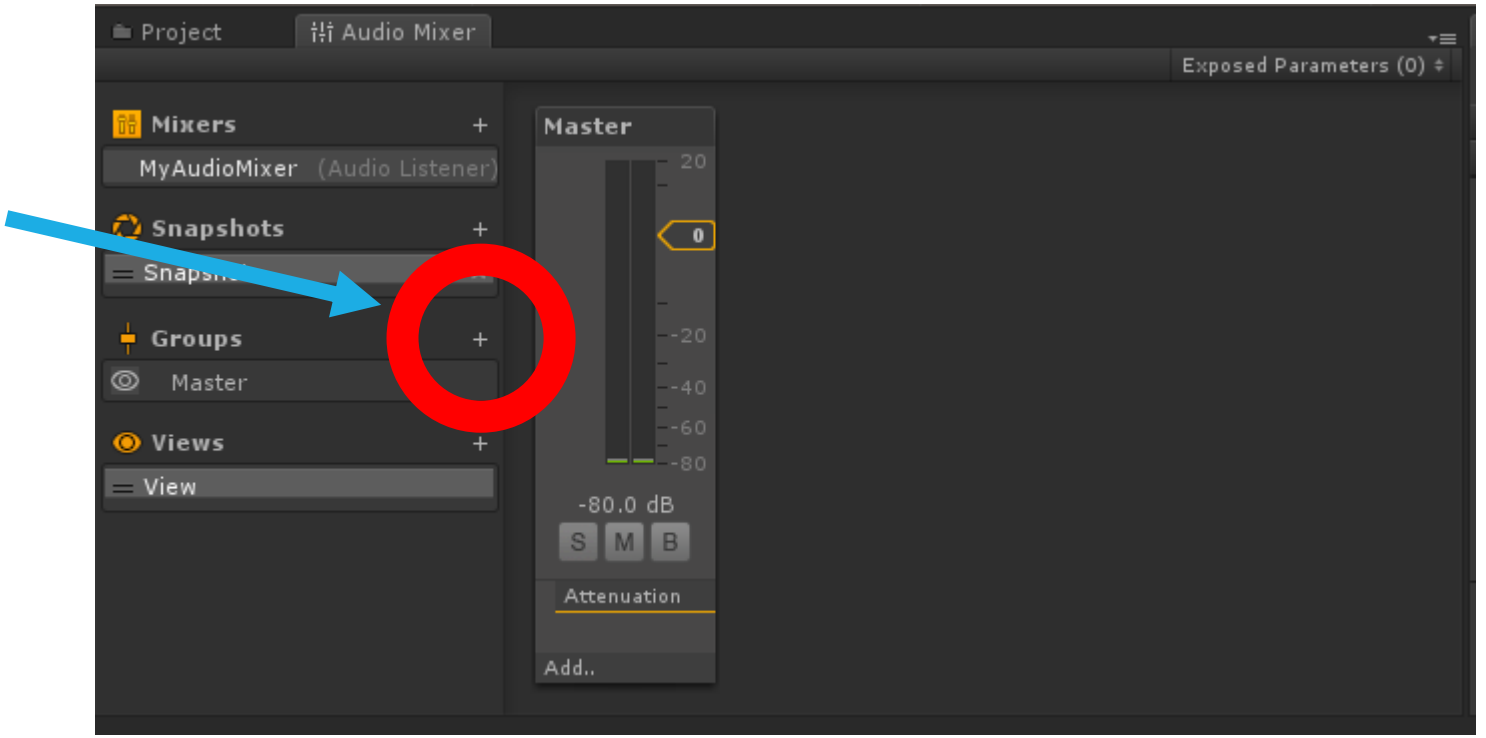
Créer un AudioMixer

- C’est un asset projet
- Menu de creation d’assets Create > AudioMixer
- Double click sur l’asset pour ouvrir la fenêtre du Mixer



Créer un AudioMixer

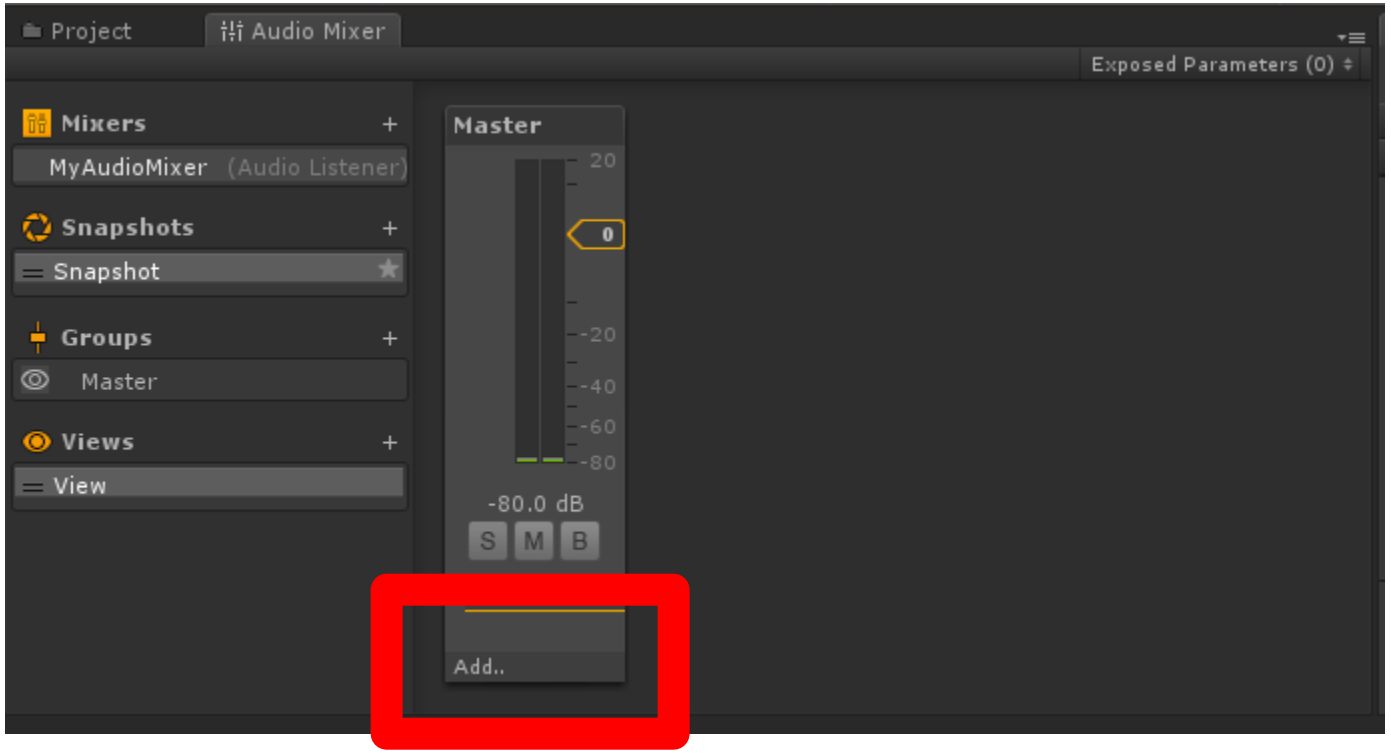
Ajoutez autant de groupes que vous voulez



Groupes et sous-groupes

Le son passe du groupe d'entrée jusqu'au groupe Master

Ajouter des effets



A propos du volume

- Va de -80db à +20db
- ce sont des **Décibels**

Pour convertir facilement d'une valeur arbitraire vers des decibels :

https://answers.unity.com/questions/283192/how-to-convert-decibel-number-to-audio-source-volu.html

Les Animations

Principes de base

1. Les points clé

2. L’interpolation

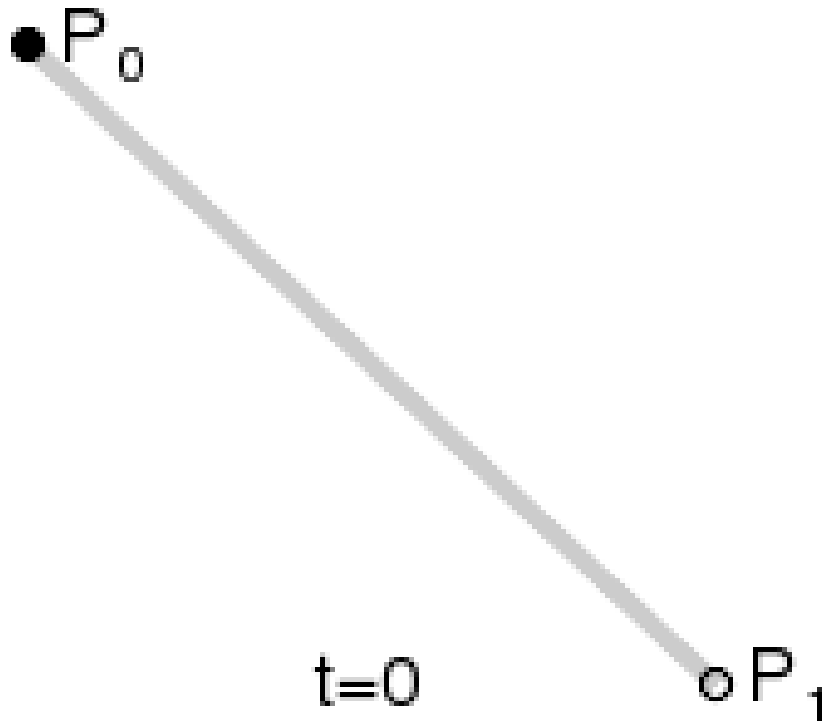
Les points clé

- On va découper un mouvement pour ne garder que les poses « représentatives » du mouvement
- Un point clé est un indicateur temporel sur la timeline de l'animation
- Il est possible d'attacher des modification de pose à un point clé

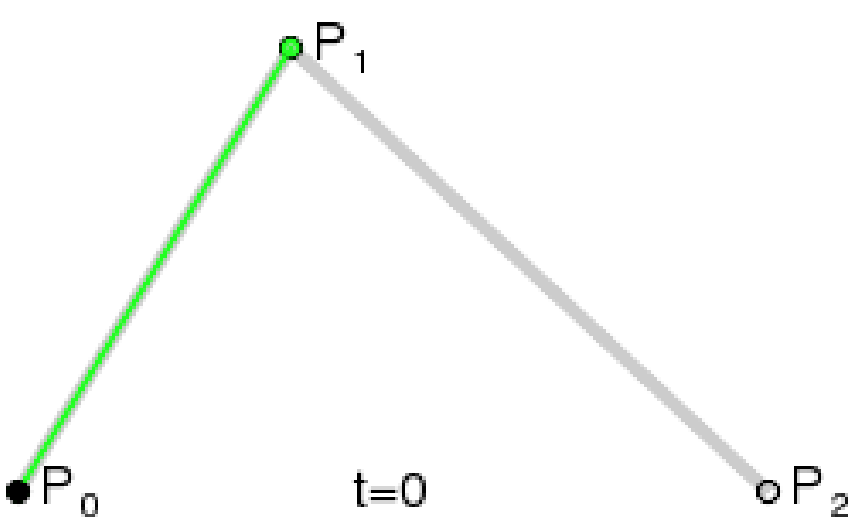
Les points clé : le stop motion

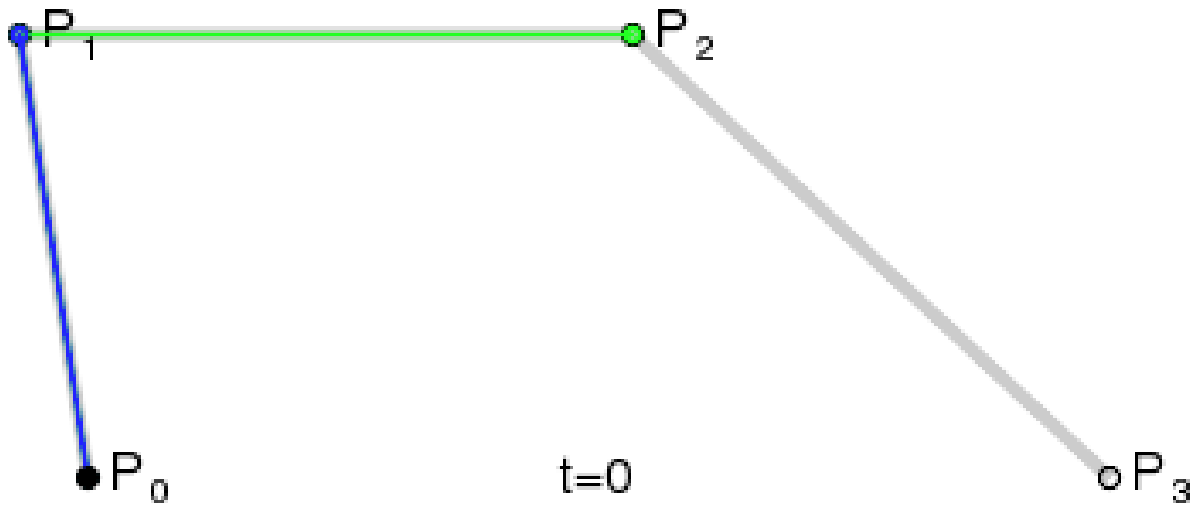
Interpolation

1. Valeur de départ
2. Valeur de fin
3. Delta (0-1)

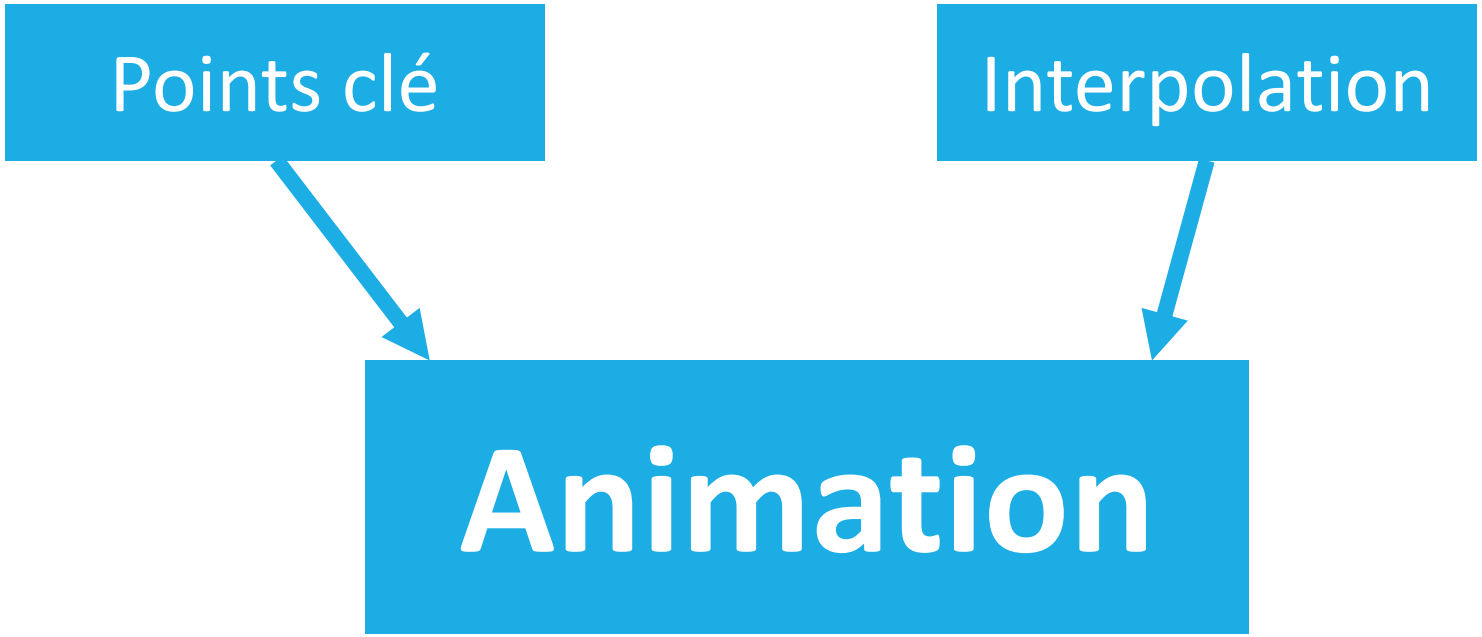


Interpolation complexe : Courbes de Bézier





En résumé



L’animation dans Unity

- Unity intègre un système d’animation versatile
  - C’est une machine à états
  - Avec un peu d’imagination… on peut l’utiliser pour d’autres sujets
- Avec ce système, vous pouvez :
  - Importer des animations depuis l’extérieur
  - Créer des animations directement depuis l’éditeur

Une machine à états

Une machine à états , c’est quoi :

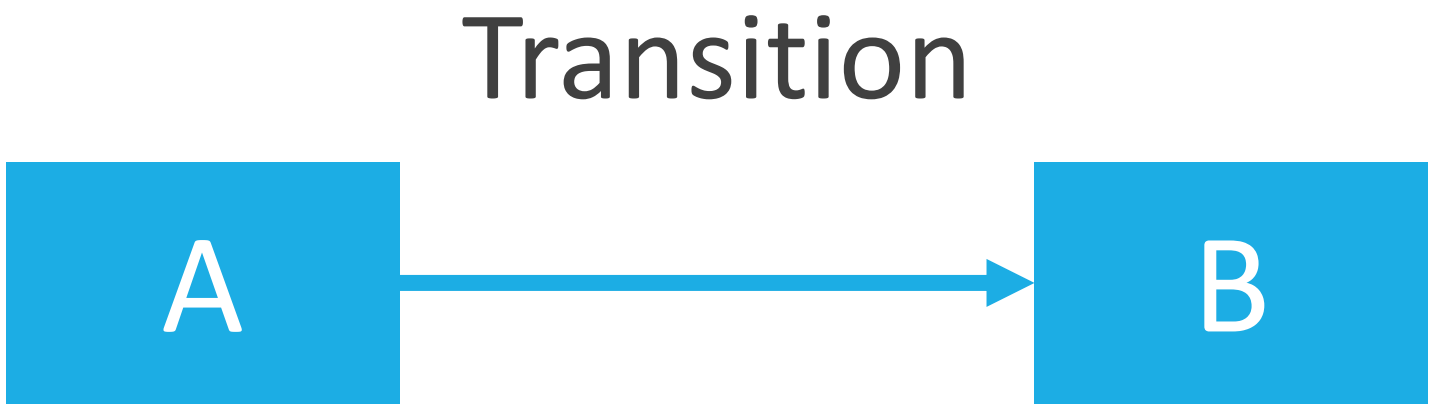
- Elle a plusieurs « états »
- Elle est **TOUJOURS** sur **UN SEUL** état à la fois
- Elle peut passer d’un état à un autre

Exemple de machine à états

Une barrière de parking a deux états :

- Ouvert
- Fermé
- … et une **transition** entre les deux

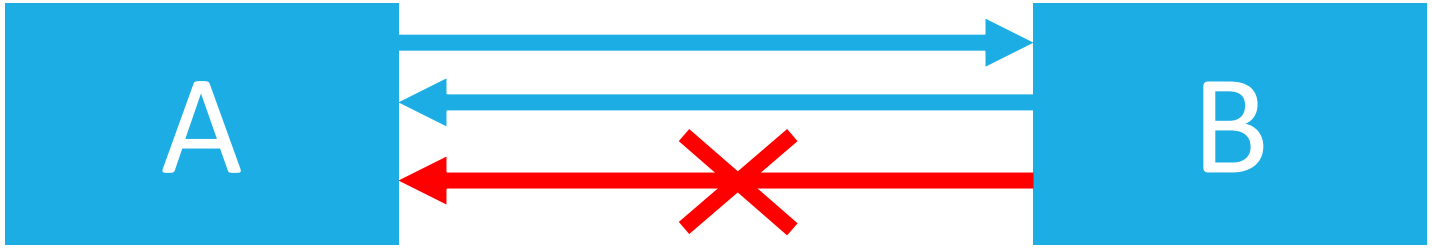
Une machine à états



Les transitions sont
à sens unique

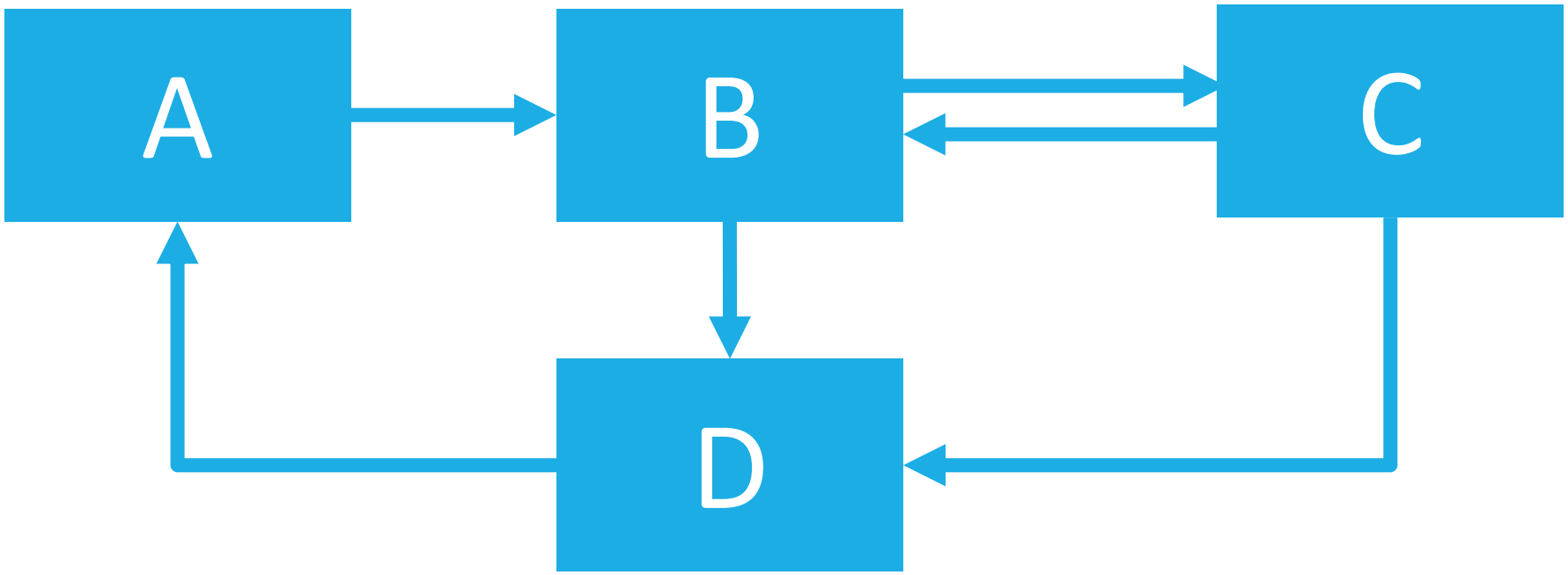
Une machine à états

Transitions



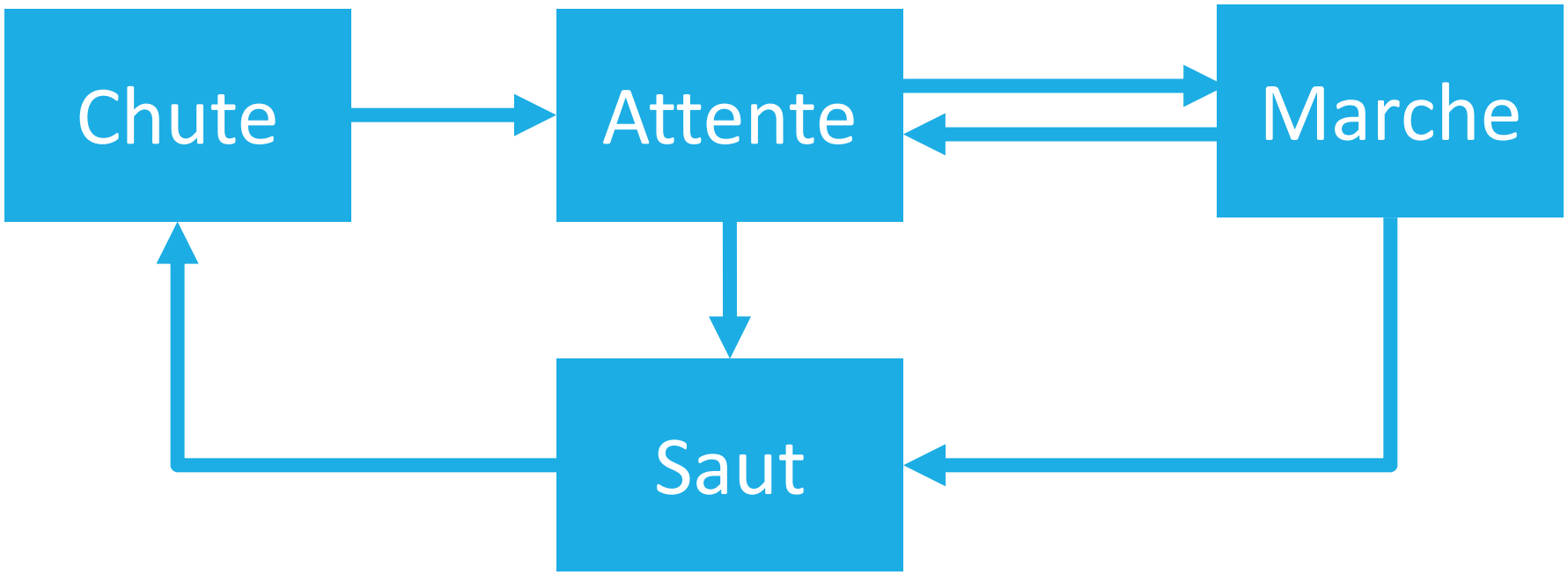
On ne peut créer qu'une seule transition dans un sens donné

Une machine à états



On peut avoir autant d’états qu’on veut

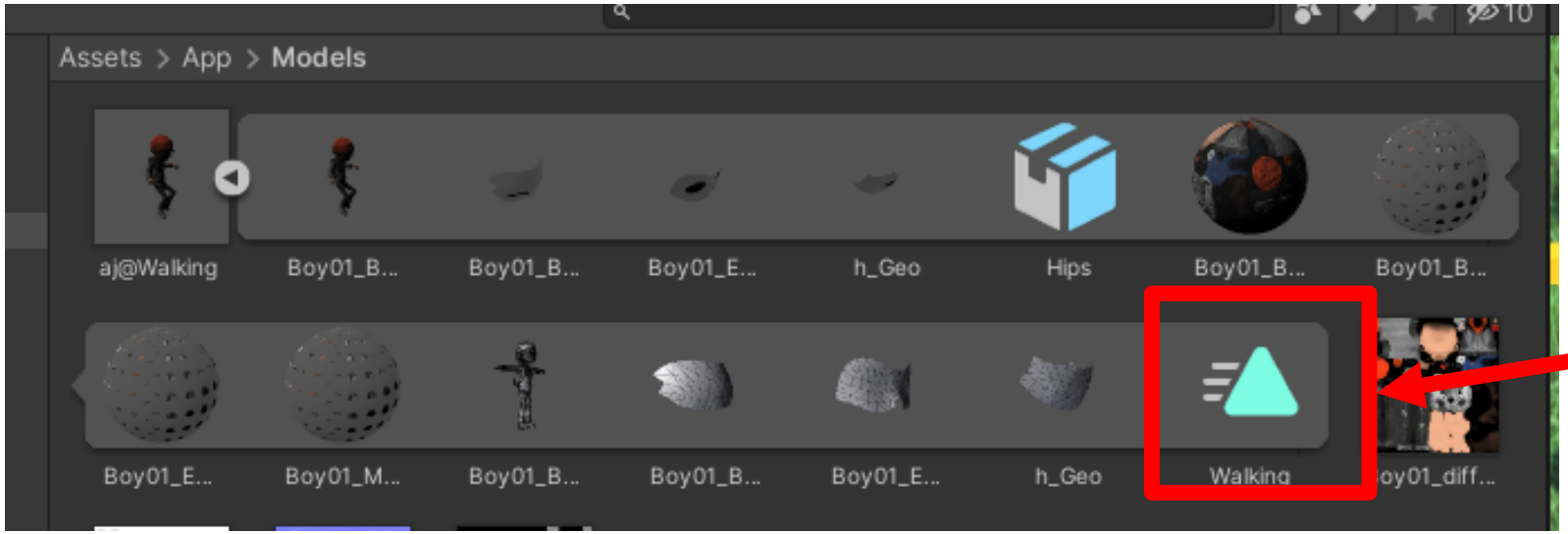
Une machine à états



On peut avoir autant d’états qu’on veut

Importer des animations

Importer des animations



Animation embarquée dans le modèle 3D

Il peut y en avoir plusieurs

L'Animator et l'Animator Controller

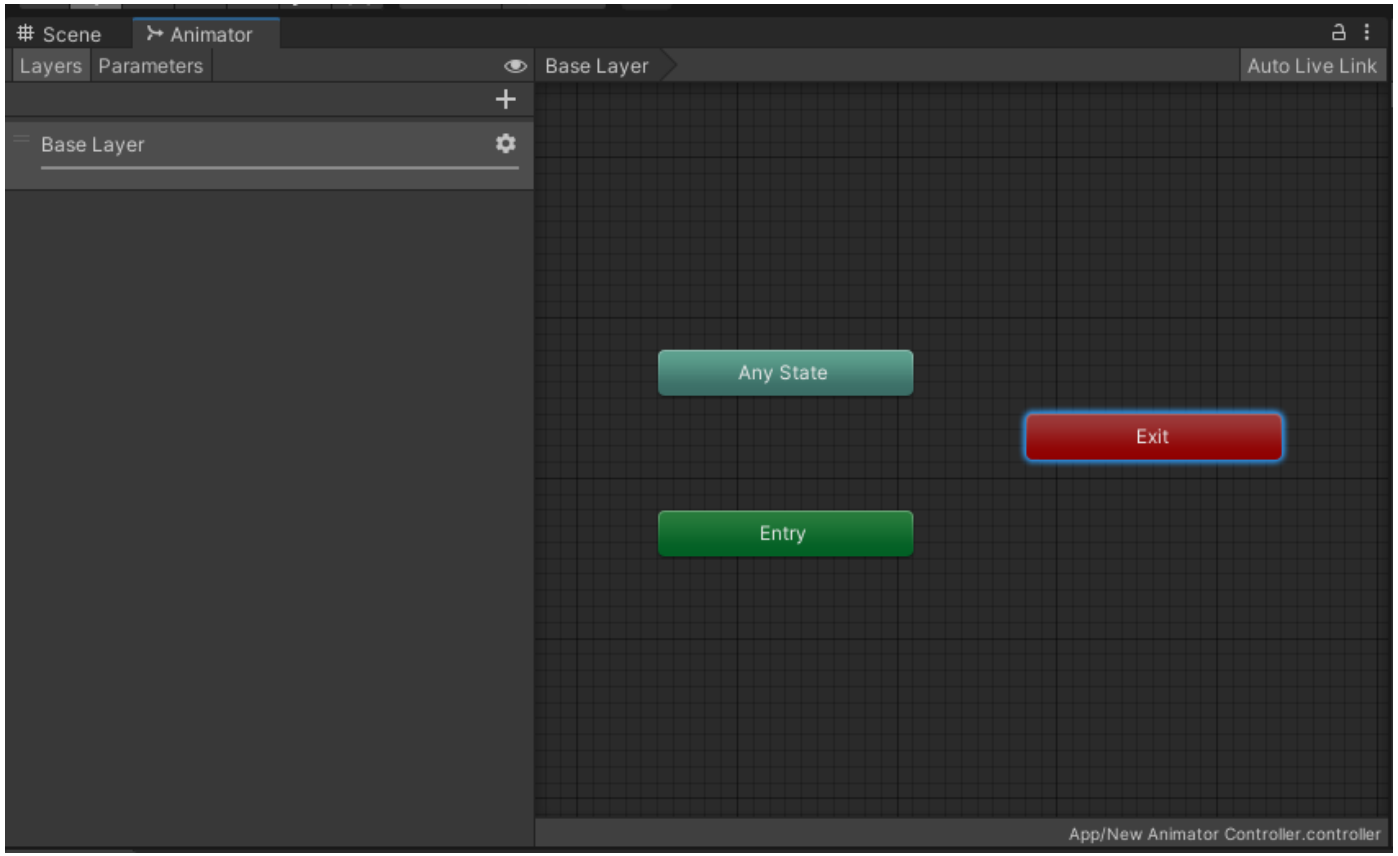
- L'**Animator** est **un composant** Unity qui permet de gérer les états d'animation d'un objet

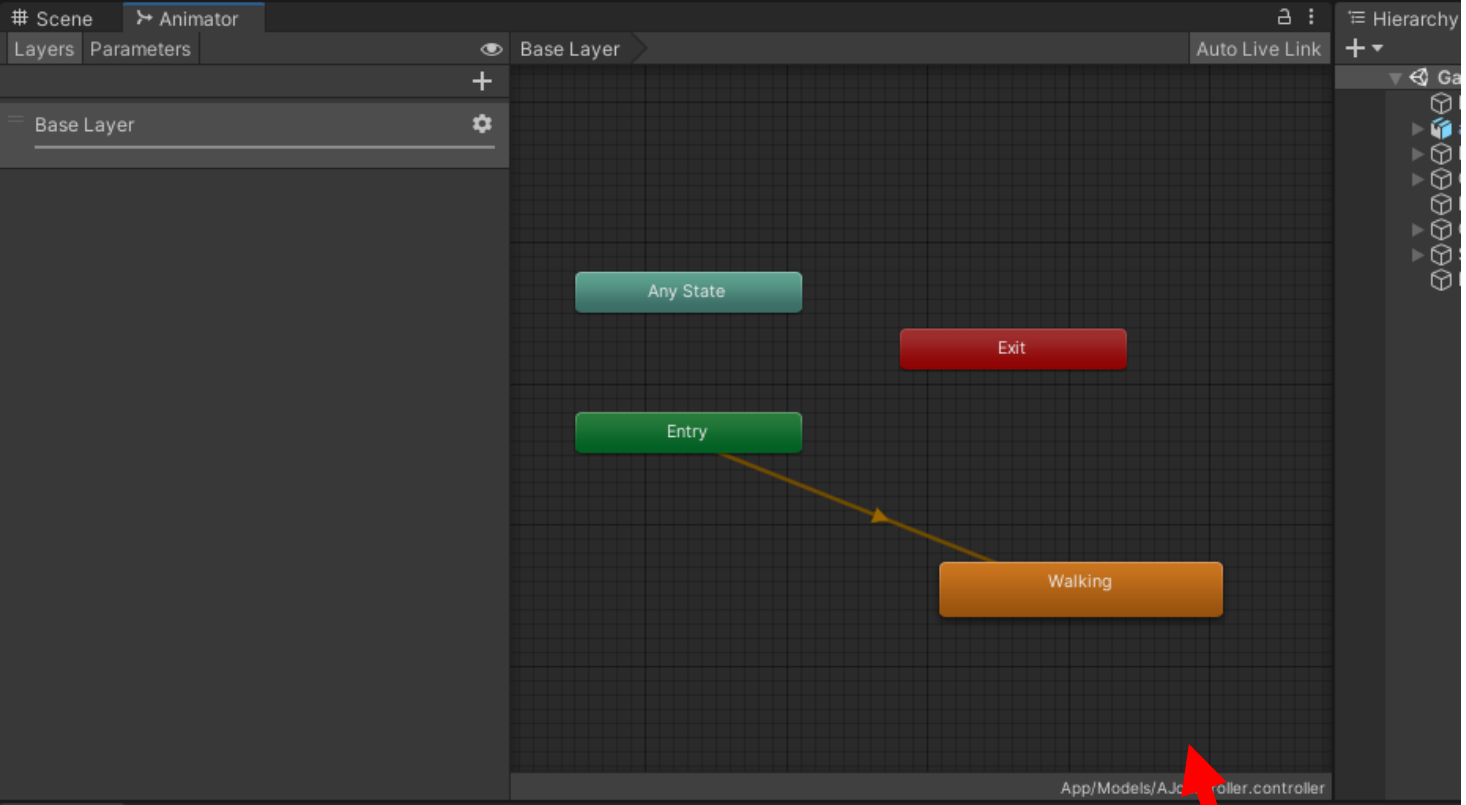
- L'**Animator Controller** est **un asset** qui stocke le graphe de de définition des états et de leurs transitions

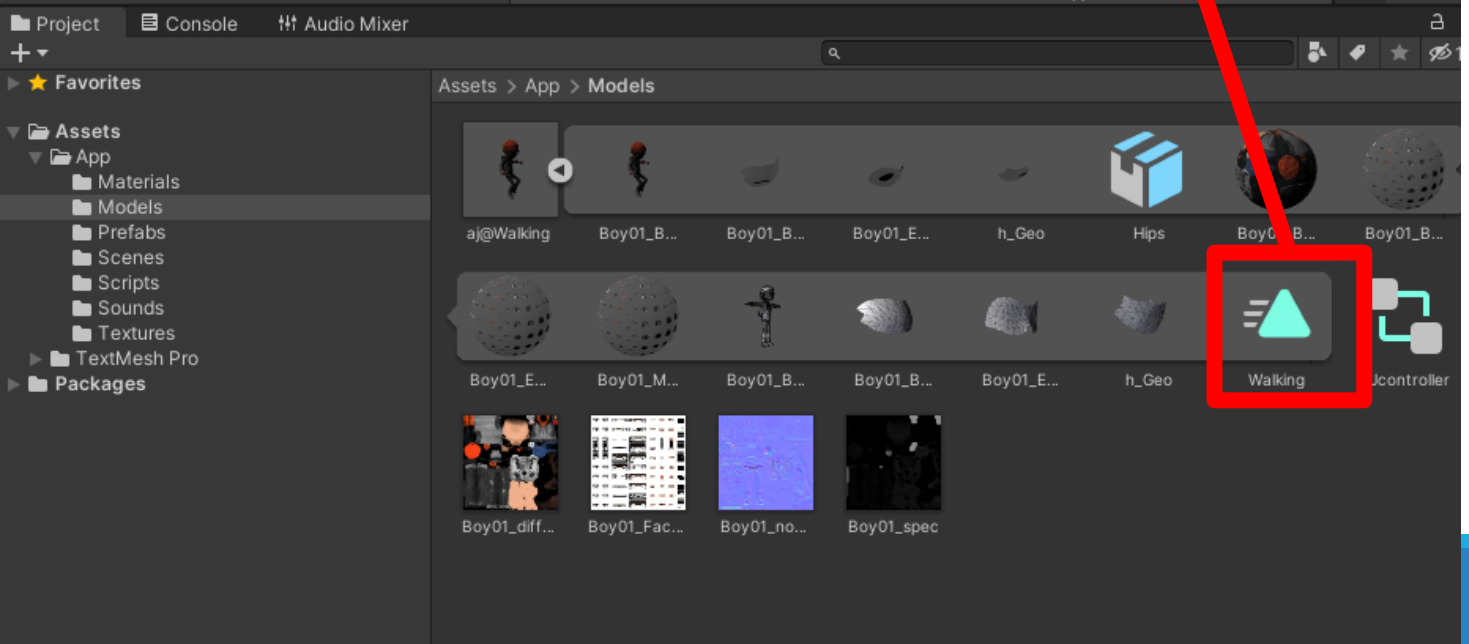
L'Animator et l'Animator Controller

- Pour gérer des animations sur un objet il faudra :
  - Créer l'**Animator Controller**
  - Le référencer dans le composant **Animator** attaché au GameObject

La fenêtre de l’Animator

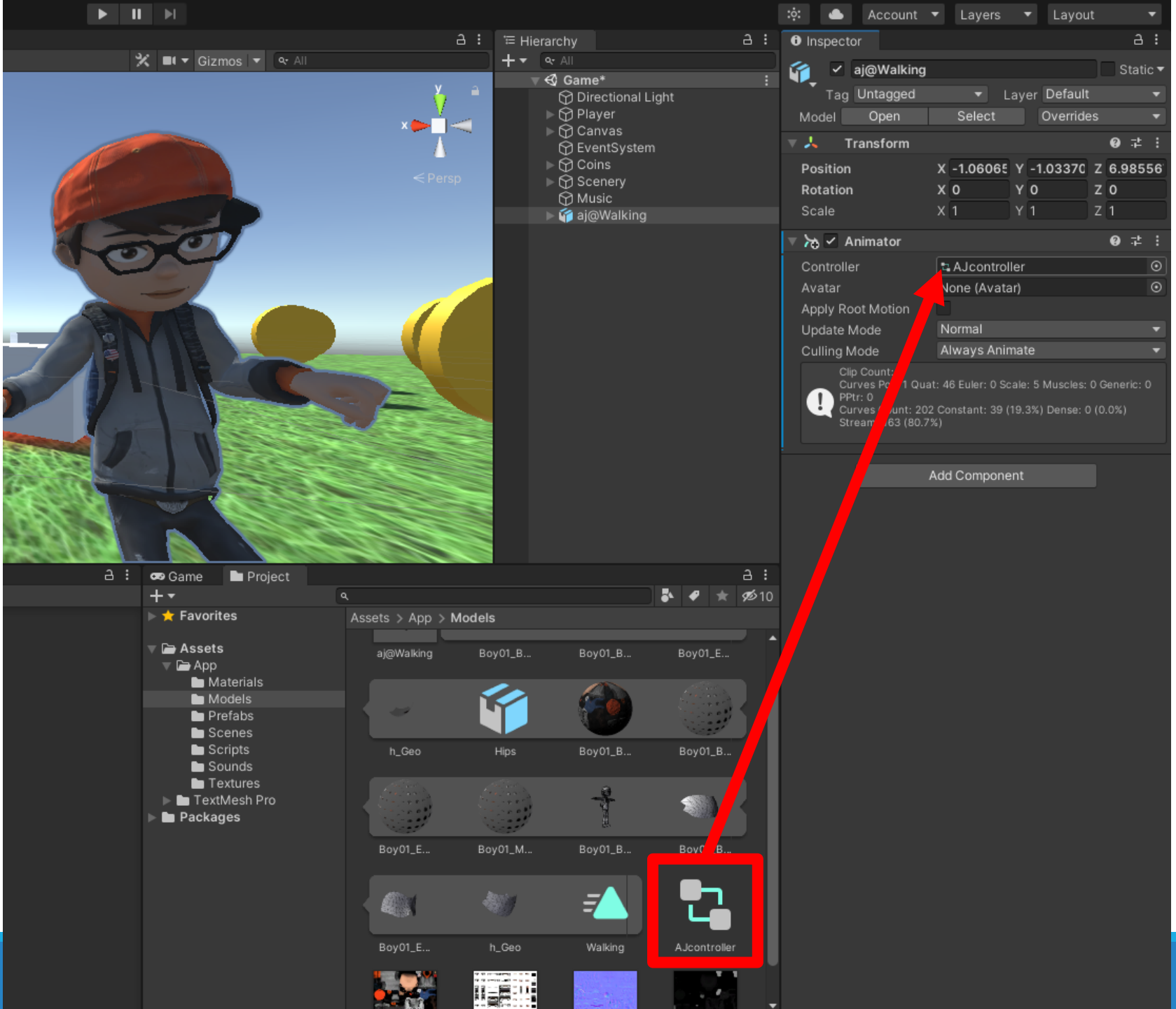






Pour ajouter une animation au Controller, faites un drag'n'drop depuis la fenêtre Project

Un nouvel état associé à cette animation sera créé

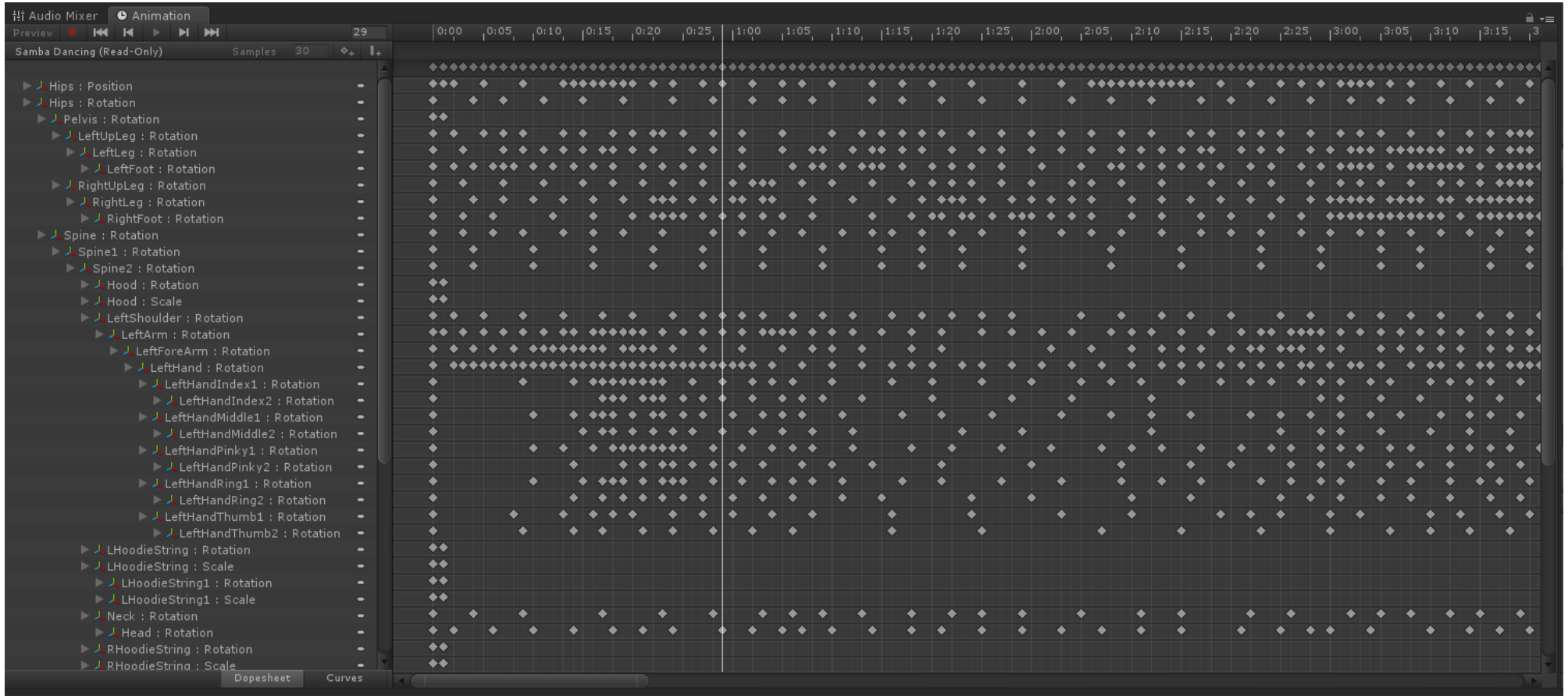


Il ne faut pas oublier de lier l'Animation Controller sur le composant Animator !

Créer des animations dans Unity

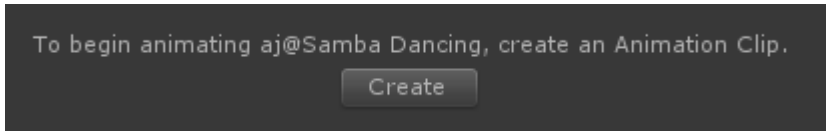
PARCEQUE POURQUOI PAS ?

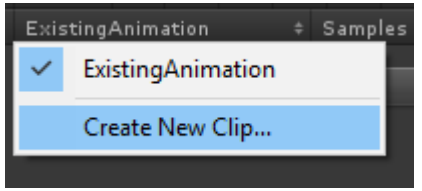
La fenêtre d’animation



Créer une animation

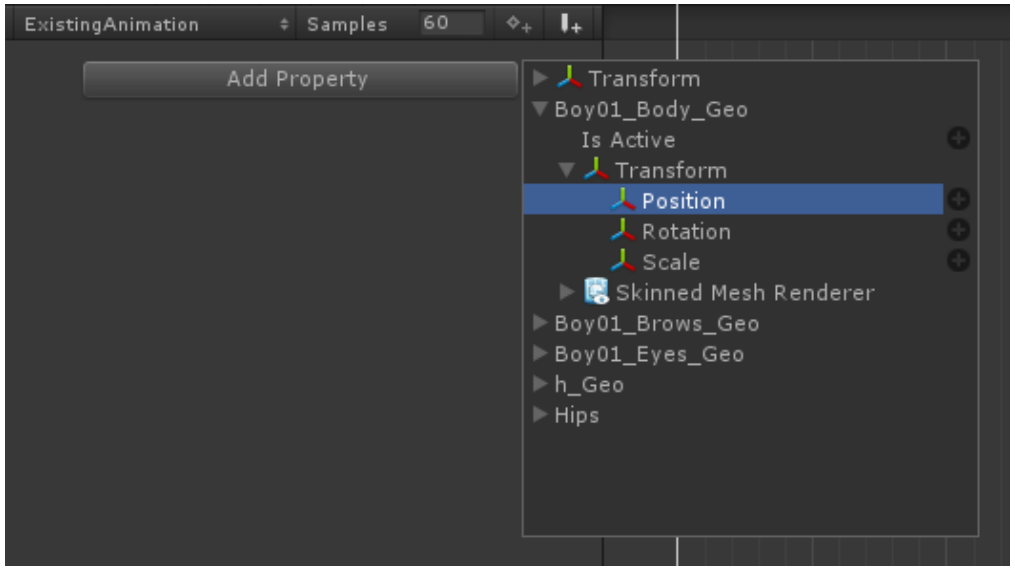
- Les animations sont stockées sour forme d'Assets
  - Note : les animations importées sont stockées dans le modèle 3D (en general) et **ne peuvent pas être modifiée**
- Vous devriez les créer via la fenêtre d'animation :

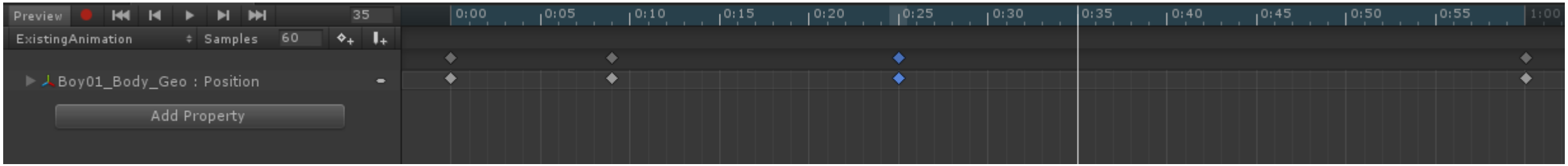




Deux façons de créer des animations

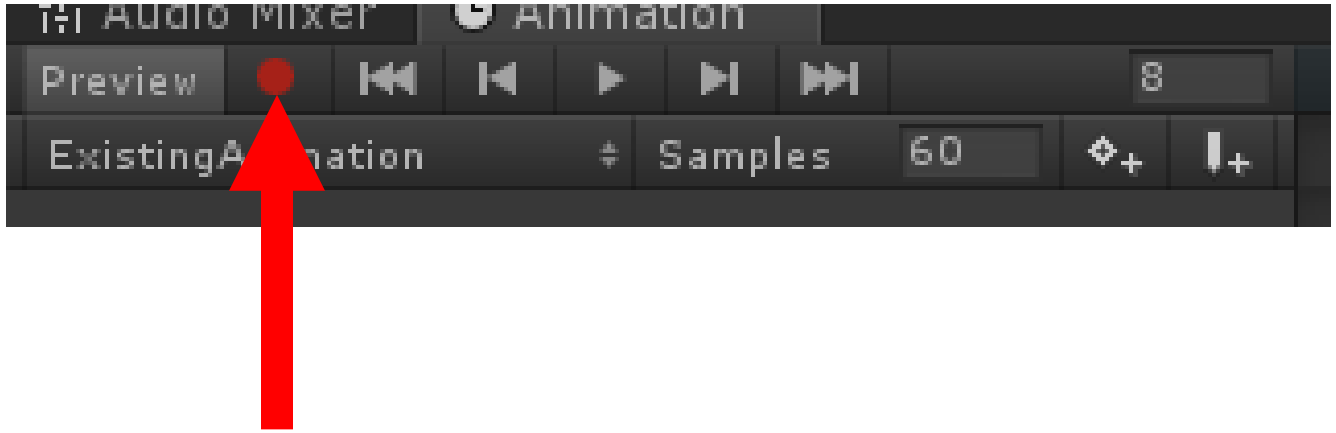
1. Paramétrer chaque point clé manuellement





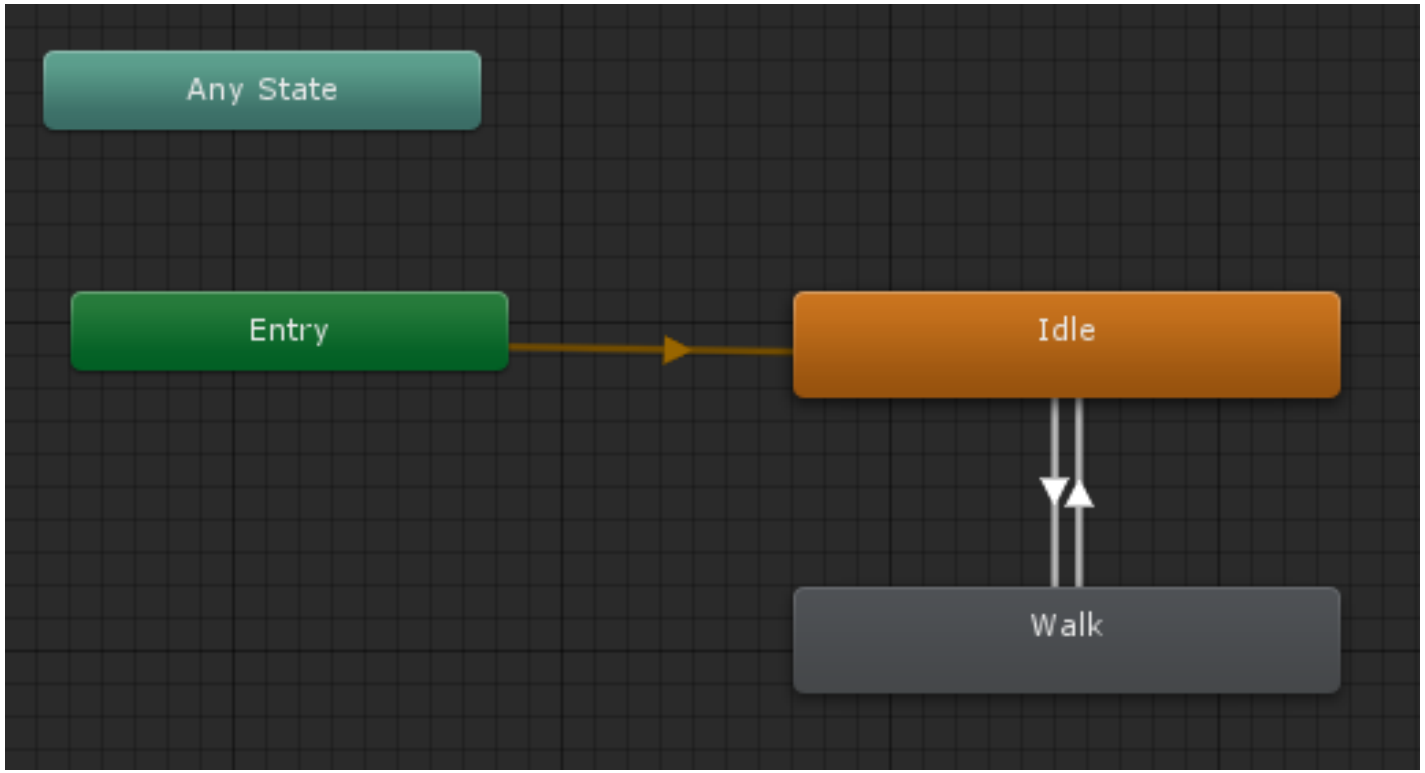
Deux façons de créer des animations

2. Enregistrer les points clés en faisant des modifsdirectement sur la scene #easyMode

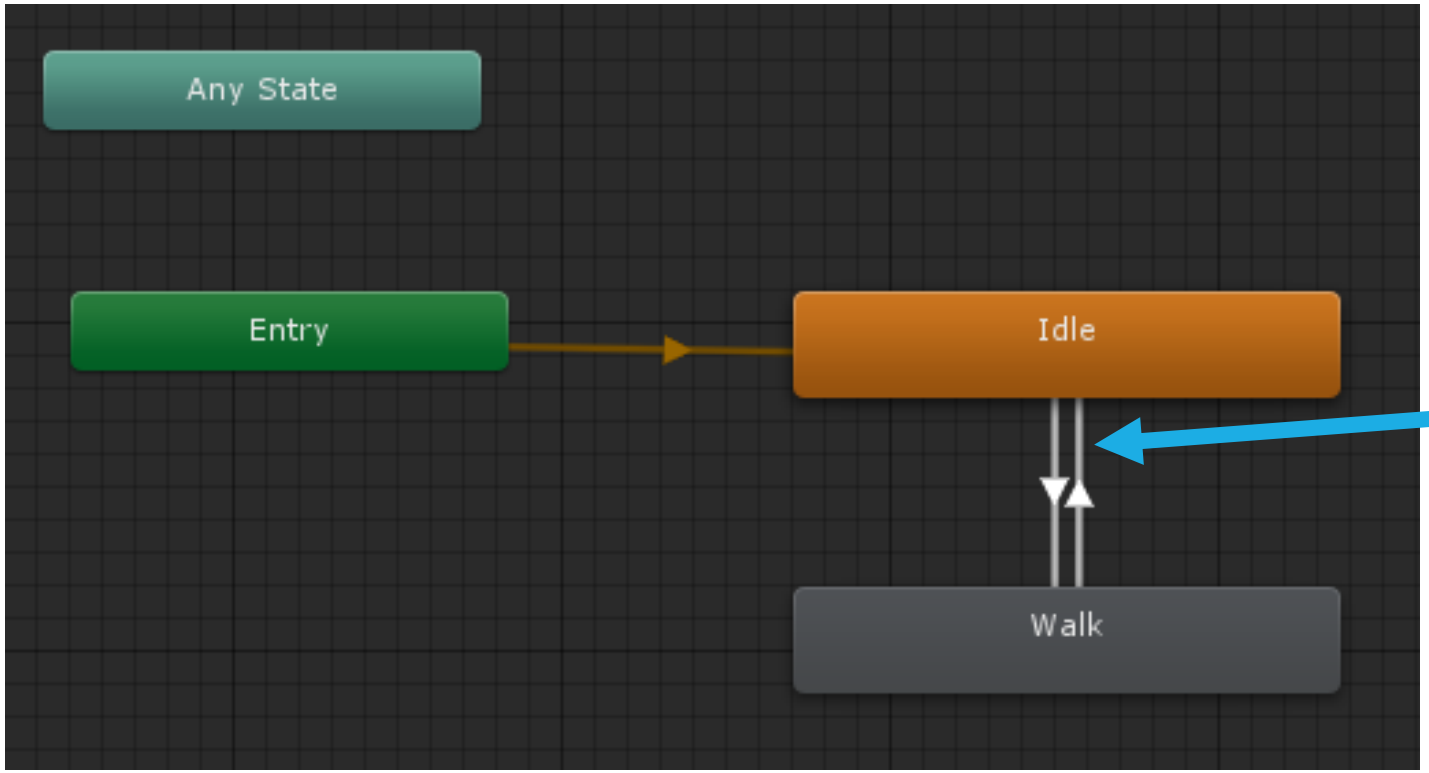


Piloter des animations

Piloter l’Animator

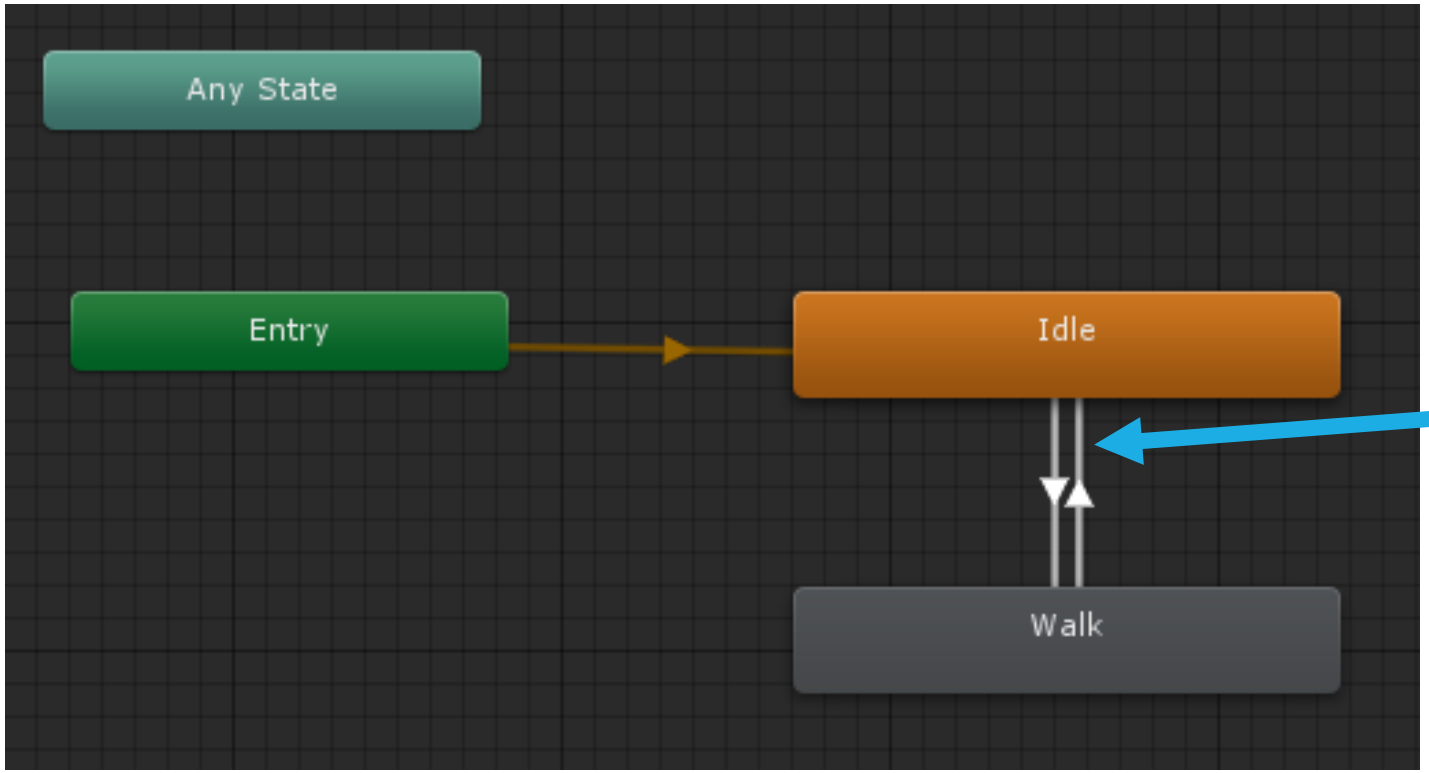


Piloter l'Animator



C'est une transition

Piloter l'Animator

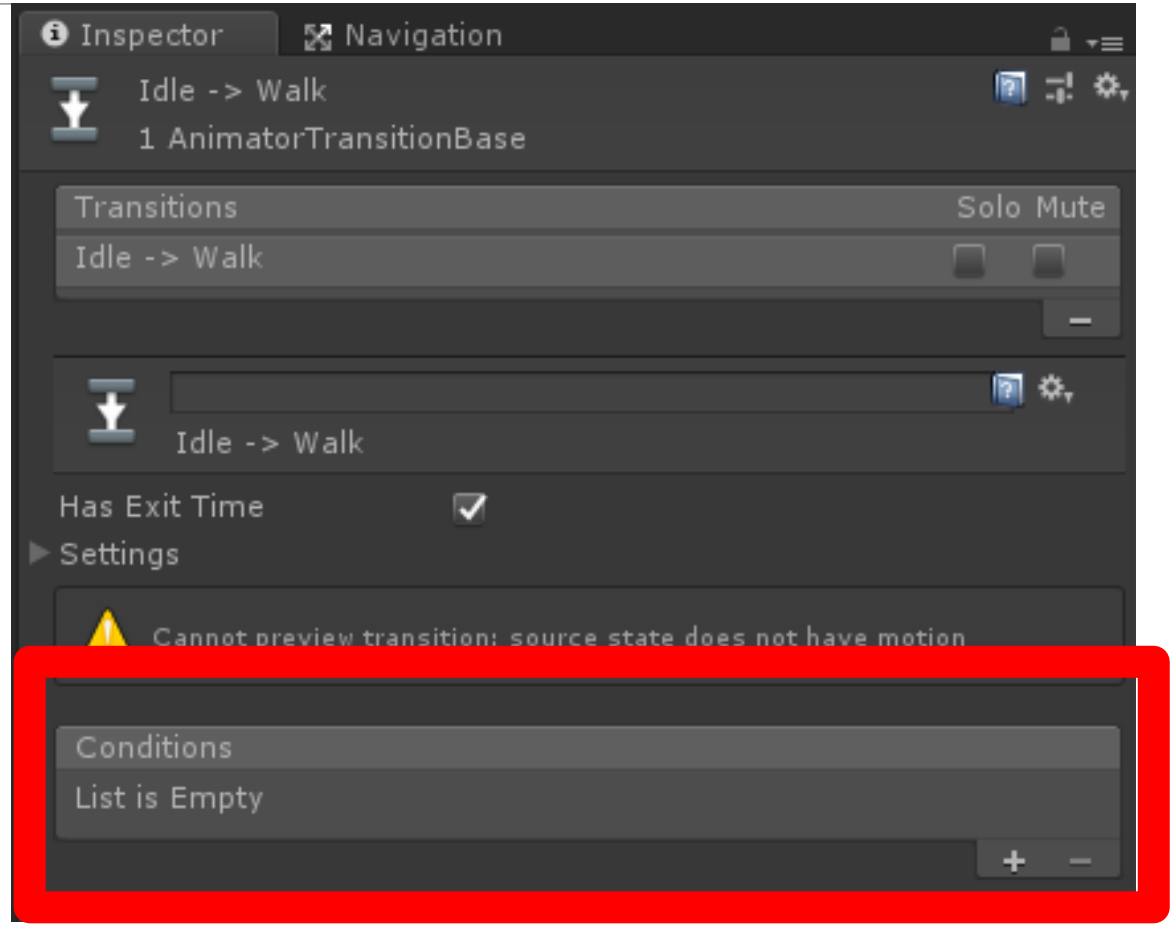


C'est une transition

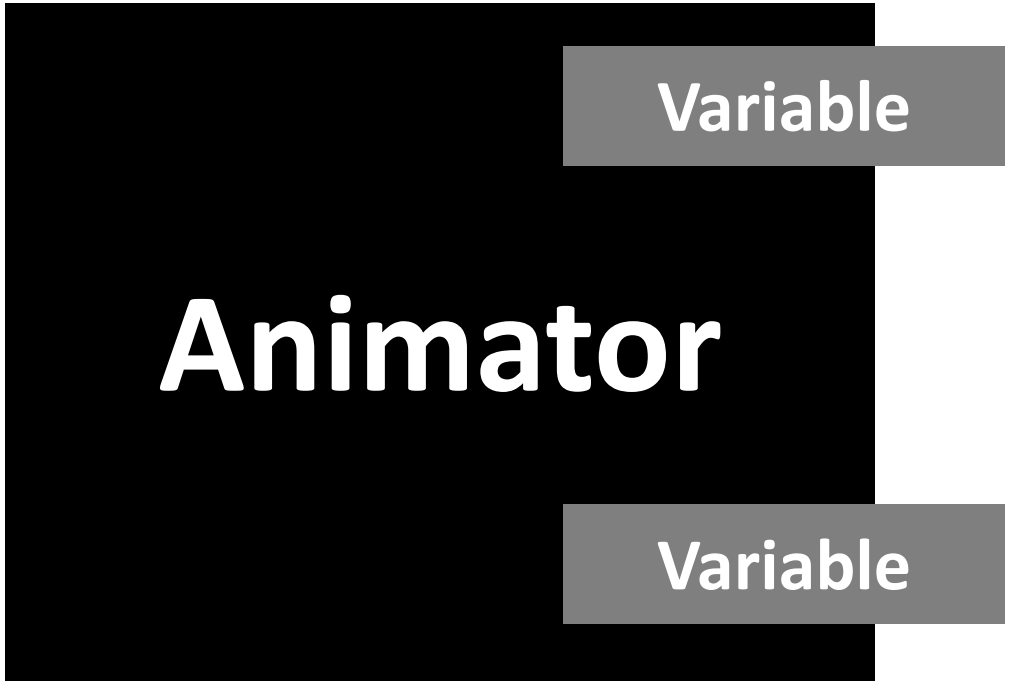
On peut y ajouter des conditions

Conditions de transition

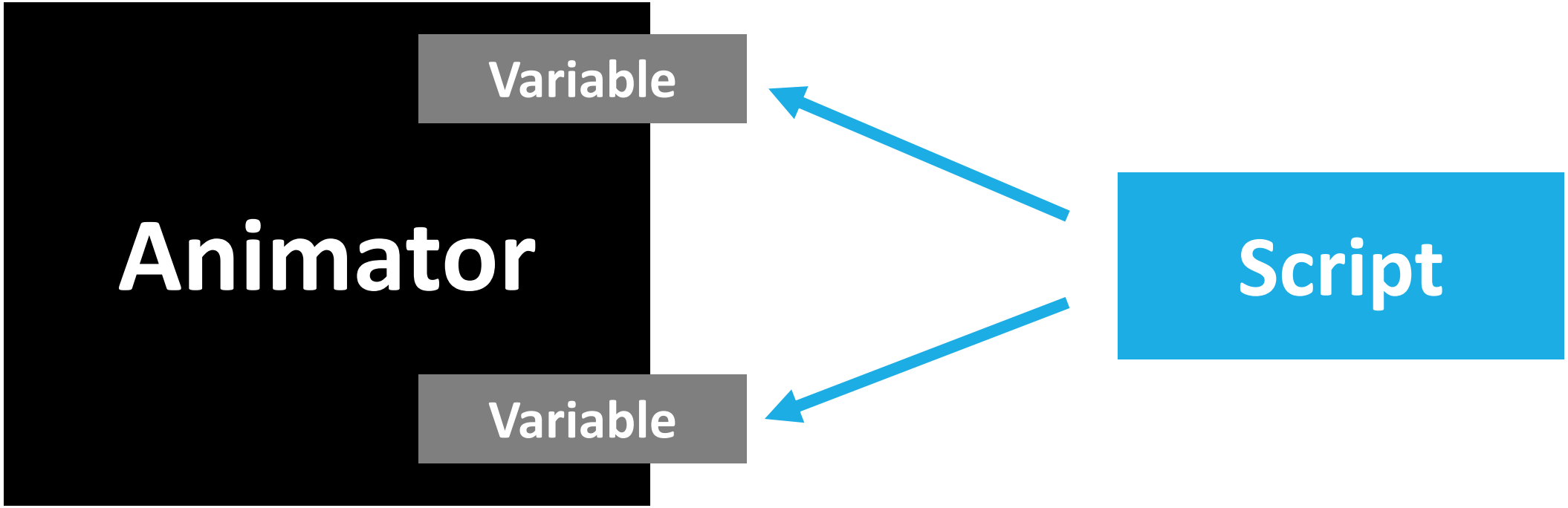
- On peut ajouter des conditions de transition basées sur des variables



Les variables dans l'Animator

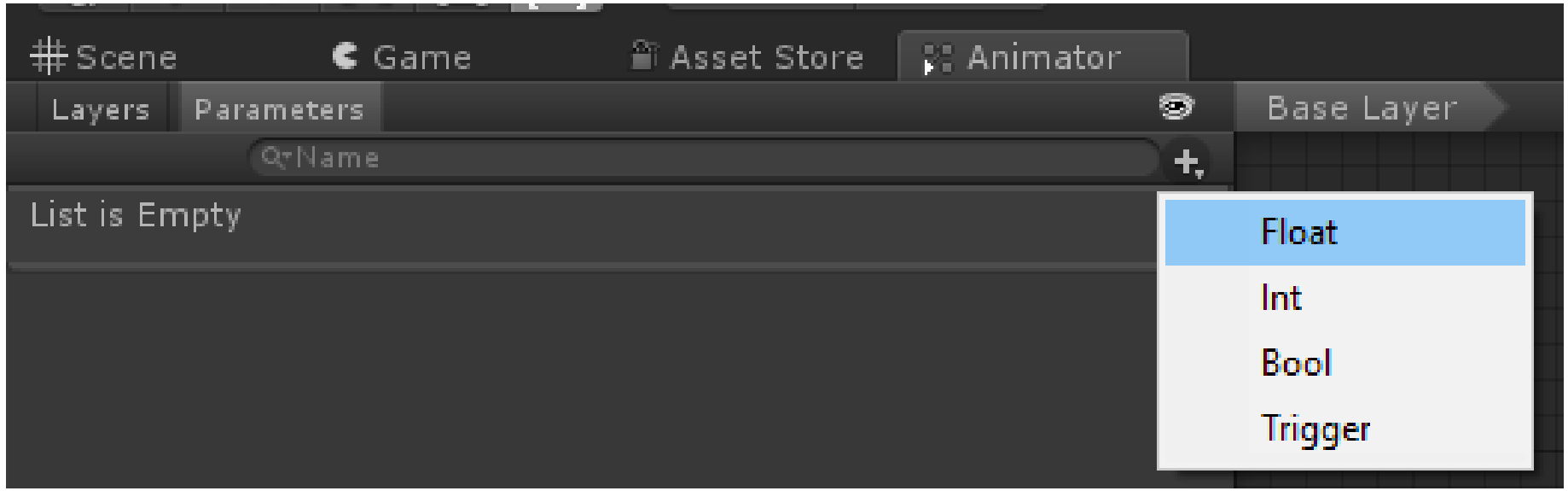


Les variables dans l'Animator



Utiliser les variables dans l'Animator

Paramétrer les variables dans l'Animator



Types de variables

- Float
- Int
- Bool
- Trigger

Types de variables

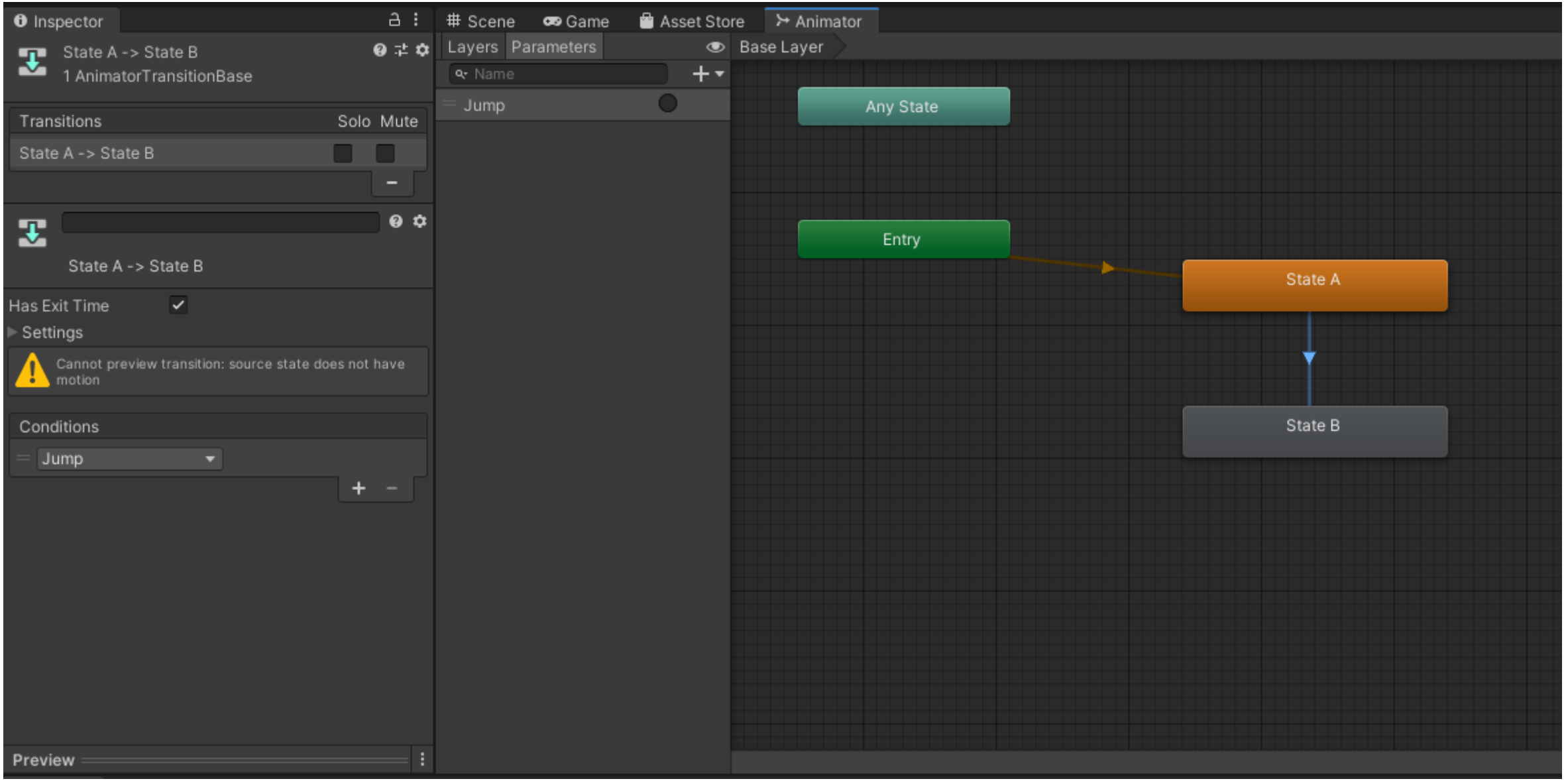
- Float → 3,1415
- Int → 42
- Bool → true / false
- Trigger → true / false

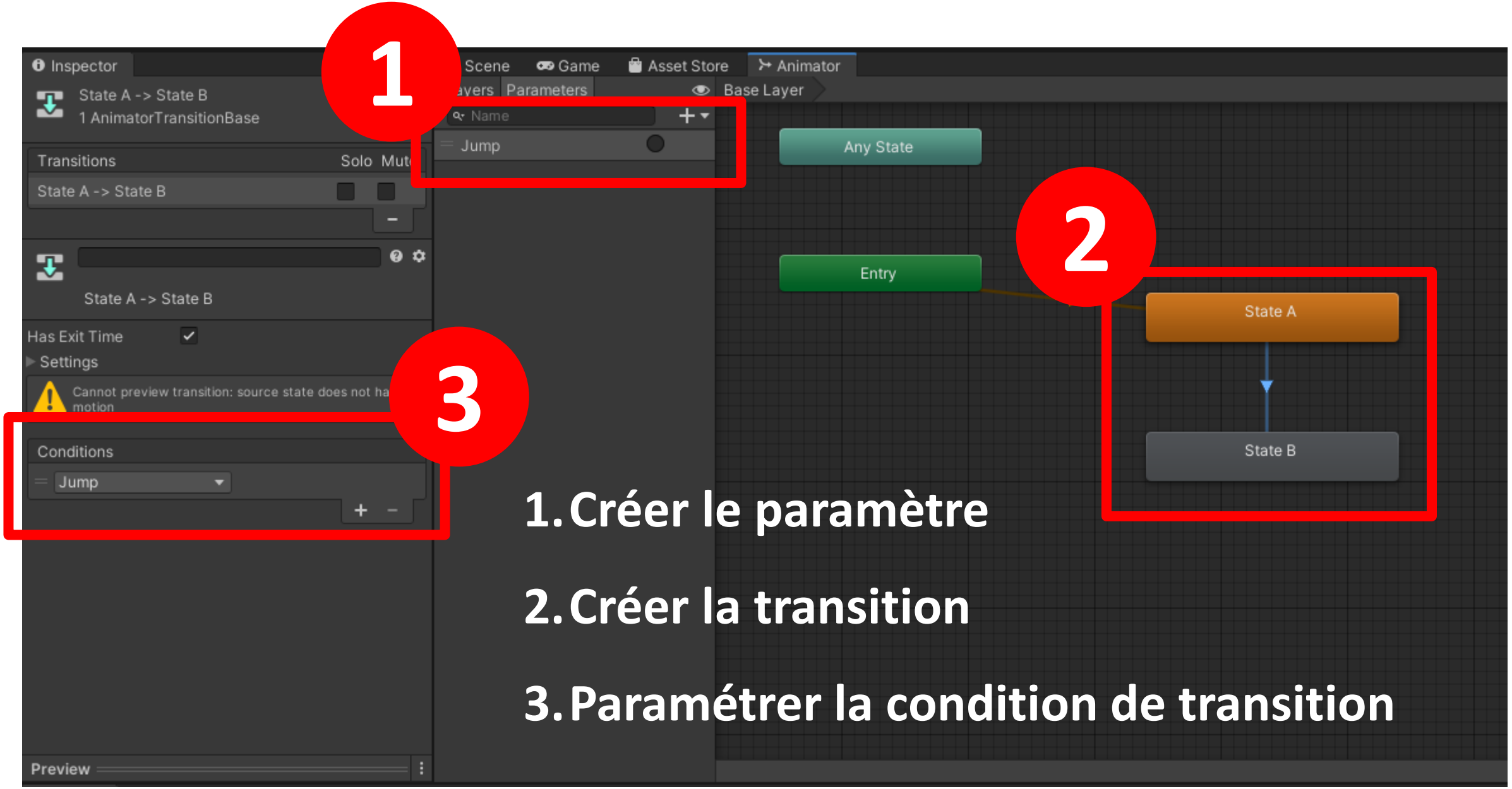
Types de variables

- Float → 3,1415
- Int → 42
- **Bool** → true / false
- **Trigger** → true / false

Bools VS Triggers

- Les variables Bool **CONSERVENT** leur Valeur après usage dans une transition
- Les variables Triggers sont **RESET** après usage dans une transition


Inspector
State A -> State B
1 AnimatorTransitionBase
Transitions
Solo
Mute
State A -> State B
State A -> State B
Has Exit Time
Settings
Cannot preview transition: source state does not have motion
Conditions
Jump
Preview
Scene
Game
Asset Store
Animator
Layers
Parameters
Name
Jump
Base Layer
Any State
Entry
State A
State B


1
Inspector
State A -> State B
1 AnimatorTransitionBase
Transitions
Solo
State A -> State B
State A -> State B
Has Exit Time
Settings
Cannot preview transition: source state does not ha
motion
3
Conditions
Jump
Preview
Scene
Game
Asset Store
Animator
Parameters
Base Layer
Name
Jump
Any State
Entry
2
State A
State B
1. Créer le paramètre
2. Créer la transition
3. Paramétrer la condition de transition

Piltage depuis un script

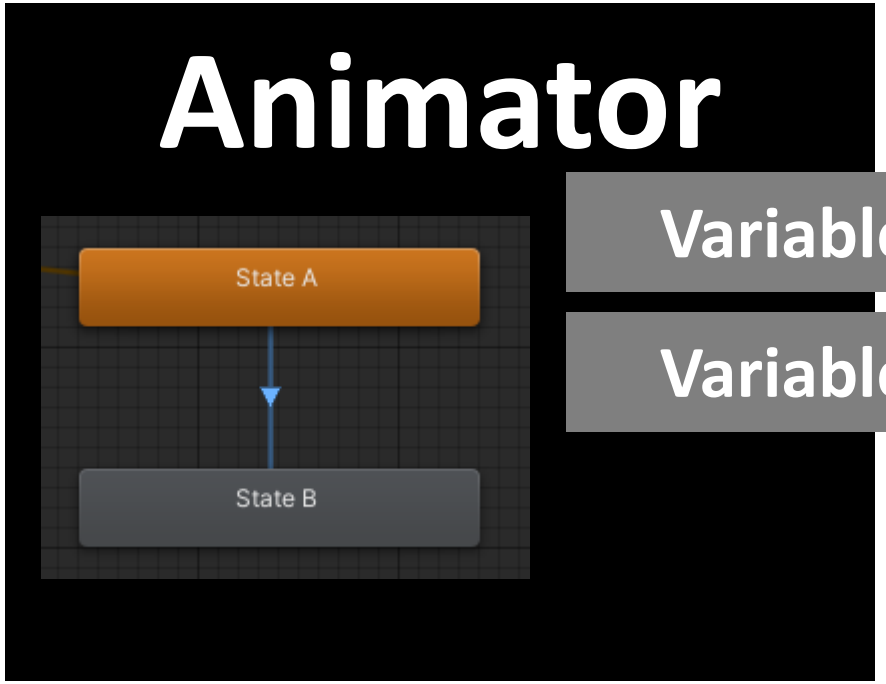
Il vous faut :

- Une reference vers l'Animator
- Initialiser la Valeur des variables que vous voulez modifier

Piltage depuis un script

```
Animator animator = GetComponent<Animator>();

animator.SetFloat("myFloat", 3.1415f);

animator.SetInteger("myInt", 42);

animator.SetBool("myFirstBool", true);
animator.SetBool("mySecondBool", false);

animator.SetTrigger("myTrigger");
```

L'Animator est une boite noire



Script

```
Animator animator = GetComponent<Animator>();

animator.SetFloat("myFloat", 3.1415f);

animator.SetInteger("myInt", 42);

animator.SetBool("myFirstBool", true);
animator.SetBool("mySecondBool", false);

animator.SetTrigger("myTrigger");
```

GO