

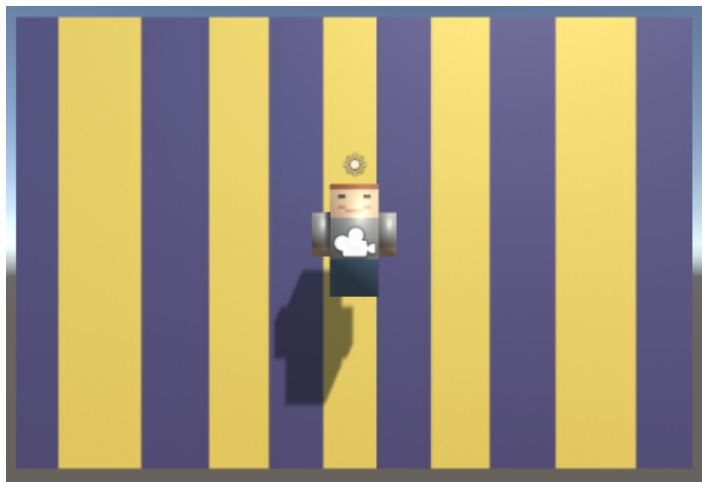
UNITY - TP3.2 - ANIMATION

Pour créer le 3^{ème} niveau de jeu, commencez par télécharger les ressources du TP. Importez le package CakeBoxBoy dans votre projet. Ouvrez la scène et lancez le jeu : lorsque vous cliquez sur l'arrière-plan vous lancez une part de gâteau ! La musique d'ambiance est définie dans l'objet **Tent**. L'AudioSource contient la musique (AudioClip) qui se déclenche automatiquement au lancement de la scène (PlayOnAwake) et qui tourne en continu (Loop).

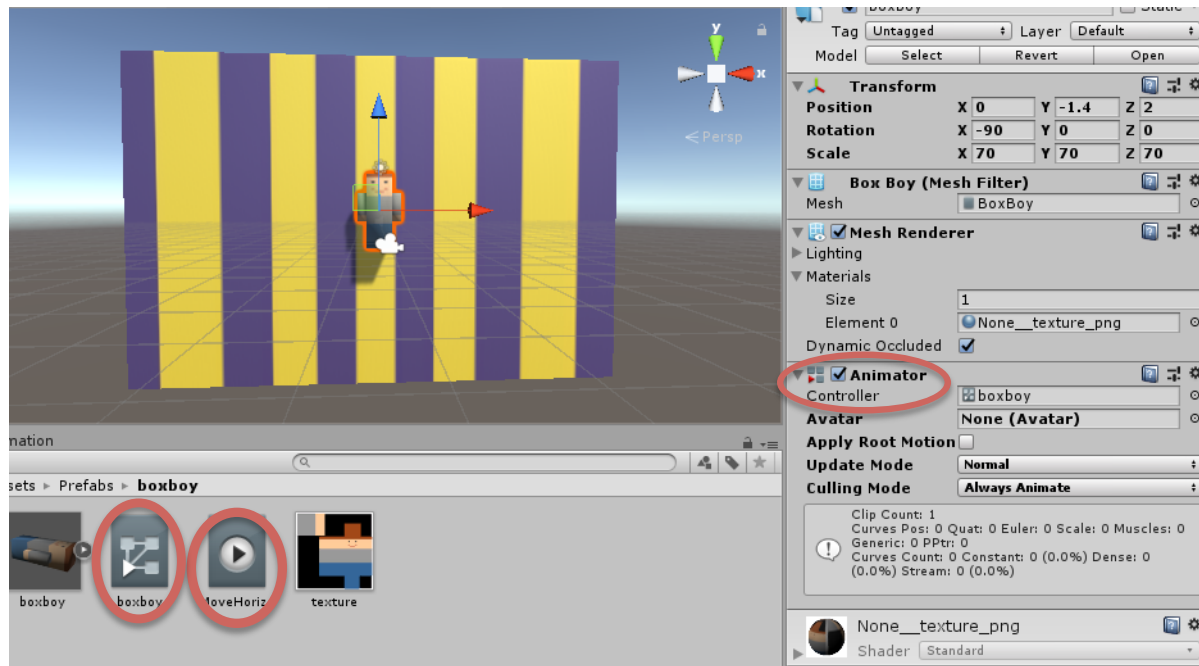
Nous allons ajouter une cible au jeu : boxboy.

PREMIERE ANIMATION : CIBLE EN MOUVEMENT

Alignez la scène avec la vue caméra : sélectionnez la **Main Camera** et puis cliquez sur le menu *GameObject > Align view to Selected*. Ajoutez à la scène le modèle BoxBoy Modifiez les paramètres pour le centrer sur l'arrière-plan (j'ai appliqué un facteur d'échelle 70). Lorsque le personnage est touché par un gâteau, un son doit se déclencher. Glissez le son Splatter du dossier Audio dans l'objet BoxBoy. Désactivez-le dans l'Inspector, il sera activé dans l'animation.



Gardez BoxBoy sélectionné et ouvrez l'onglet Animation grâce au menu *Window > Animation > Animation*. Cliquez sur le bouton **Create** et donnez le nom MoveHorizontally à l'animation. Unity crée deux fichiers : MoveHorizontally.anim et boxboy.controller. Il ajoute également un Animator Component au personnage.



MoveHorizontally est un **Animation Clip**, il stockera l'animation d'une ou plusieurs propriétés de l'objet : position, orientation, taille, matériau, couleur, volume d'un son, variables dans les scripts, etc.

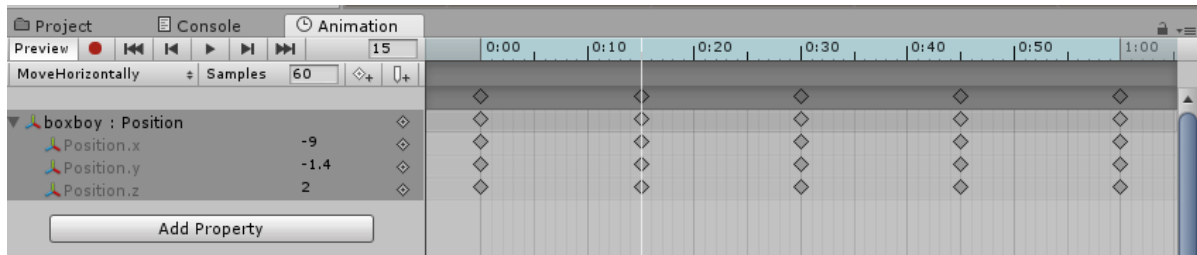
L'onglet Animation contient sur la gauche un bouton d'enregistrement, une liste déroulante avec les animations disponibles, le bouton de propriétés pour ajouter une propriété à animer, la durée de l'animation en frames (Samples 60), la frame actuelle (0), et deux boutons, un pour créer une key frame et le second pour créer un événement (un point dans le temps où une fonction sera appelée). Sur la droite vous avez la TimeLine, permettant d'avoir l'aperçu de la disposition dans le temps des keyframes.



CREATION DE L'ANIMATION

Dans l'onglet d'animation cliquez sur **Add Property**. Sélectionnez **Transform > Position** dans la liste déroulante et puis cliquez sur le signe plus. Vous pouvez dérouler la propriété pour voir les coordonnées x, y et z, et animer une seule coordonnée.

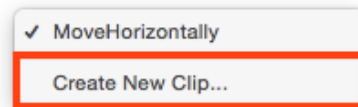
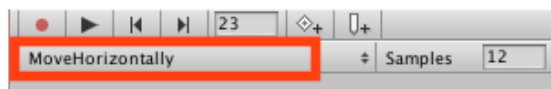
Appuyez sur le bouton rouge pour lancer l'enregistrement et placez-vous sur la timeline à 0 :15. Vous pouvez modifier la position du personnage dans l'éditeur ou dans l'inspecteur. Entrez -9 dans Position.x (pour avoir BoxBoy à gauche de la scène, mais la valeur dépend des transformations que vous avez réalisées sur le personnage). Cela ajoute un keyframe représenté par des losanges sur la Timeline. Ajoutez d'autres positions-clés pour faire faire au personnage un aller-retour.



Testez votre animation en cliquant sur le bouton Play. Modifiez le nombre de samples à 10 ou autres pour réduire la vitesse (ce nombre est inversement proportionnel au temps entre les frames de l'animation).

ANIMER UN SON

Dans l'onglet Animation, cliquez sur MoveHorizontally. Dans la liste déroulante, choisissez Create new Clip.



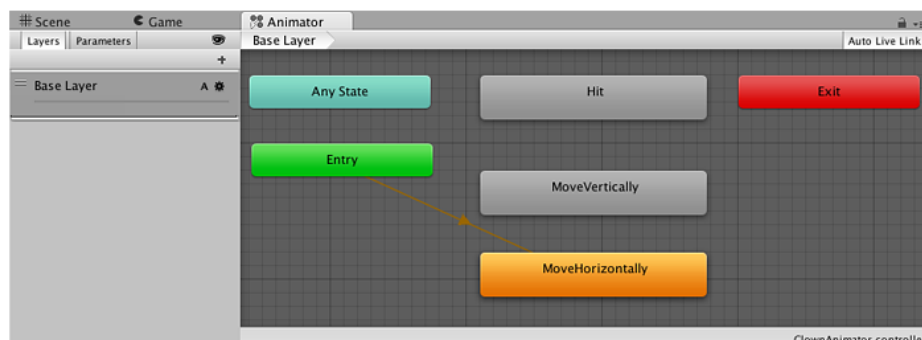
Donnez le nom Hit à la nouvelle animation. Ajoutez la propriété **AudioSource.Enabled**. Dans la Timeline, déplacez le curseur à 0:10 et activez dans l'inspector l'**Audio Source**. Déplacez ensuite le curseur de la Timeline à 1:00 et désactivez l'**Audio Source**.

A vous ! Créez un nouveau Animation Clip, MoveVertically, pour déplacer le personnage du haut en bas dans la scène et puis de retour au centre. N'oubliez pas de modifier le nombre de samples.

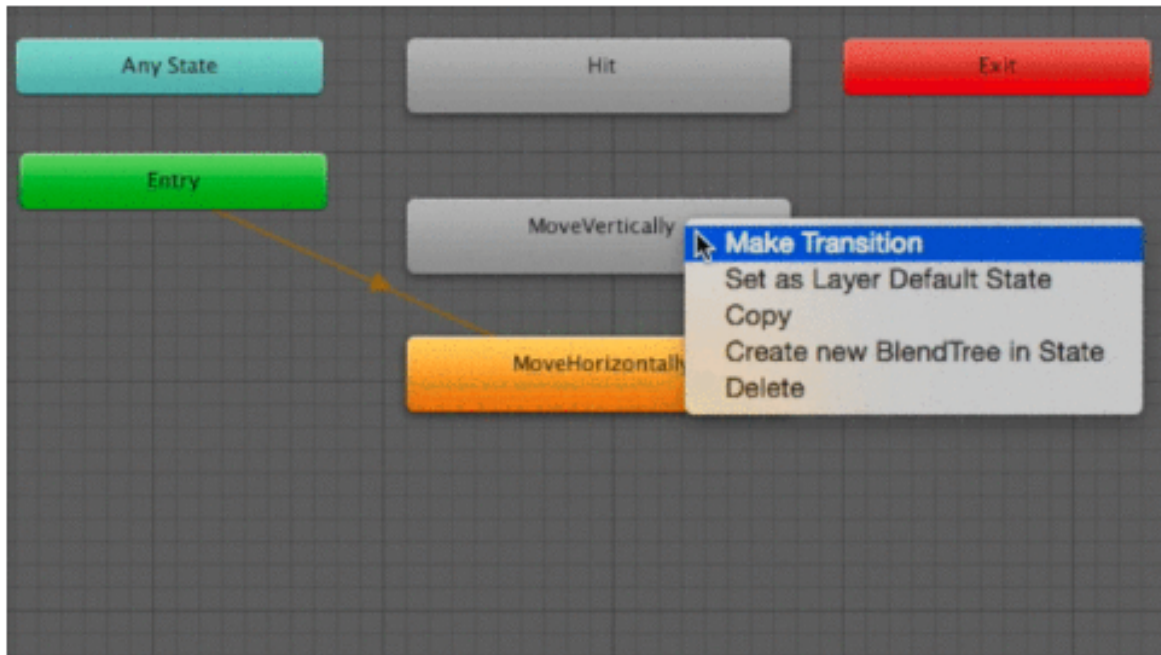
TRANSITIONS ENTRE ANIMATIONS

Unity utilise une machine à états pour associer les diverses animations. Sélectionnez le personnage et puis allez dans le menu **Window > Animation > Animator** pour ouvrir l'onglet Animator. Vous verrez les 3 animations que vous avez créées ainsi que les états Any State, Entry et Exit. Une flèche associe Entry à MoveHorizontally, ce qui signifie que MoveHorizontally est l'état par défaut. Il est coloré en orange.

Vous pouvez modifier la position des boîtes. Vous pouvez également naviguer dans l'onglet avec Alt + souris.

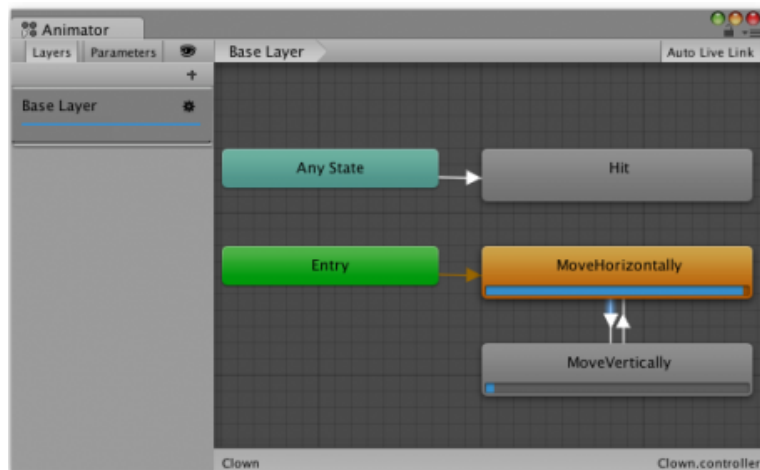


Pour créer une transition, faites clic droit sur MoveHorizontally, sélectionnez Make Transition et puis cliquez sur MoveVertically. Faites de même pour créer une transition dans l'autre sens.

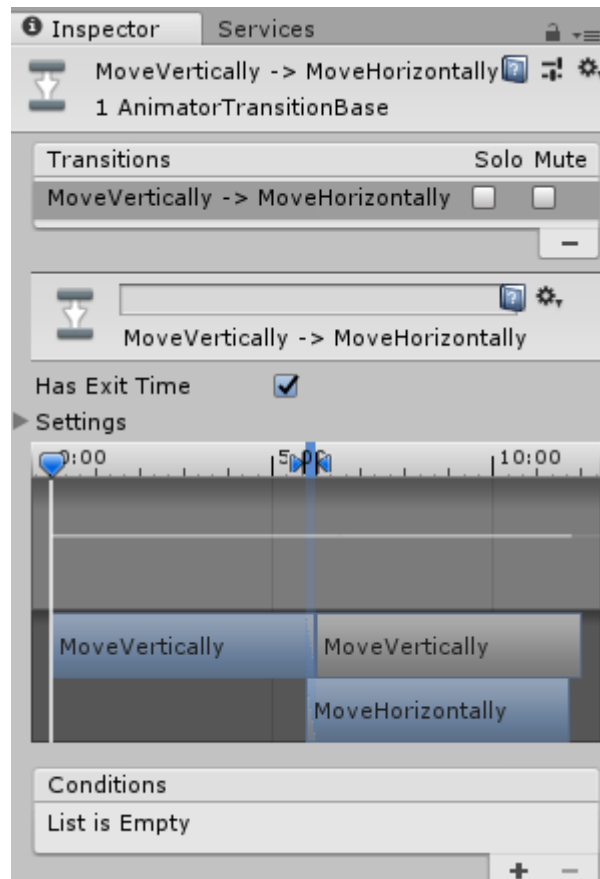


Testez le jeu ! Maintenant le personnage se déplace de gauche à droite et puis de haut en bas, en continu.

Pour mieux comprendre les transitions, regardez dans l'onglet Animator pendant que le jeu s'exécute. Vous verrez une barre de progression dans l'état actif. Avec les paramètres par défaut, une animation commence un peu avant que l'autre soit finie. Ce mélange peut être intéressant dans certains cas, par exemple, un personnage qui marche et qui se met à courir.



Il est possible de configurer les transitions en cliquant sur les liens :



Has Exit Time détermine si la transition peut être lancée à tout moment ou uniquement après l'**Exit Time** spécifié, à partir duquel la transition peut commencer. Par exemple, un Exit Time de 0.75 signifie que la transition commence quand l'animation est à 75%. Les marques bleues vous permettent de modifier le début et la durée de la transition.

Testez le jeu et lancez quelques gâteaux sur le personnage ! ☺

LE ANY STATE

Nous souhaitons que l'animation Hit se déclenche à chaque fois que le personnage est touché peu importe son état. Pour ce faire, on va utiliser la boîte **Any State**, un raccourci pour créer des transitions entre tous les états vers un état précis. Quand l'animation Hit finit, le personnage doit revenir à l'état MoveHorizontally.

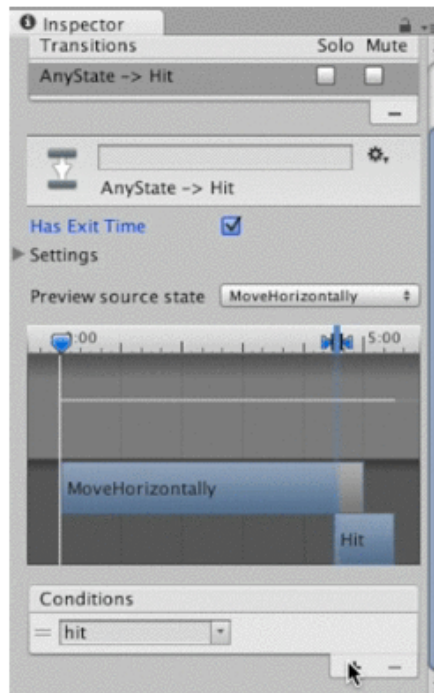
Définissez les transitions indiquées dans le paragraphe précédent

CHANGEMENTS D'ETAT EN UTILISANT DES PARAMETRES ET DES CONDITIONS

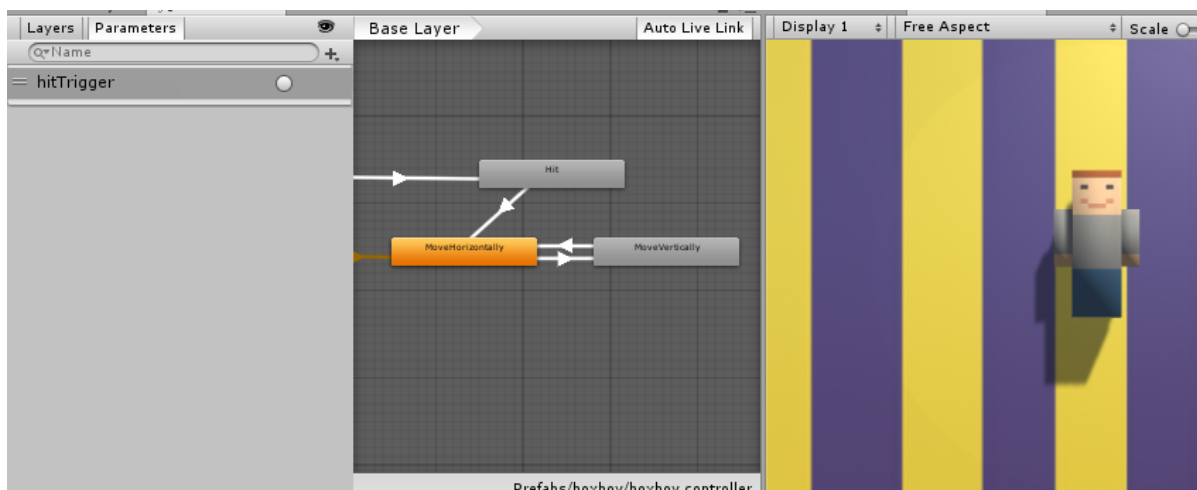
Maintenant, il est nécessaire de définir les conditions pour déclencher l'état Hit. Les paramètres et les conditions rentrent en jeu. Un paramètre est une variable que l'on peut utiliser pour définir une condition permettant de déclencher une transition.

En haut à gauche de l'Animator, sélectionnez l'onglet Parameters. Cliquez sur le bouton Plus et sélectionnez Trigger dans la liste déroulante. Donnez un nom au trigger, par exemple, hitTrigger.

Sélectionnez la transition entre **Any State** et **Hit**. Dans l'inspector, section Conditions vous verrez le message **List is Empty**. Cliquez sur le bouton Plus pour ajouter le trigger comme une condition.



La transition va se déclencher lorsque le paramètre hitTrigger aura la valeur True. Pour tester manuellement, mettez coté à coté les vues Game et Animator. Lorsque vous cliquez sur le paramètre hitTrigger, vous devez entendre un son et le personnage doit revenir à son état initial.



DECLANCHER UNE ANIMATION DANS LE CODE

Pour pouvoir détecter que le personnage est touché, nous avons besoin de colliders. Ajoutez un BoxCollider au personnage (**Add Component > Physics**). Créez un nouveau script C# « CollisionBehaviour » associé au personnage.

Lorsqu'un autre collider touche celui du BoxBoy, la fonction OnCollisionEnter est appelée. Ajoutez le code suivant au script CollisionBehaviour :

```
void OnCollisionEnter(Collision collision)
{
    Animator persoAnimator = gameObject.GetComponent<Animator>();
    persoAnimator.SetTrigger("hitTrigger");
}
```

Testez le jeu !

Réalisez la transition entre la scène BoxBoy et la scène animation.

Déposez sur Moodle le compte rendu de votre TP ainsi qu'un lien vers le téléchargement de votre projet.

Bon week-end !

REFERENCES

TP librement traduit de <https://www.raywenderlich.com/116652/introduction-unity-animation-system>