

Unity : premiers pas avec MecAnim

Vous trouverez [ici un tutorial simple d'animation d'un ventilateur avec MecAnim](#). Ce tutorial montre la création d'animation d'une hélice avec des keyframes. La suite de ce TP se propose d'animer un personnage humanoïde en important des données depuis des fichiers de motion capture.

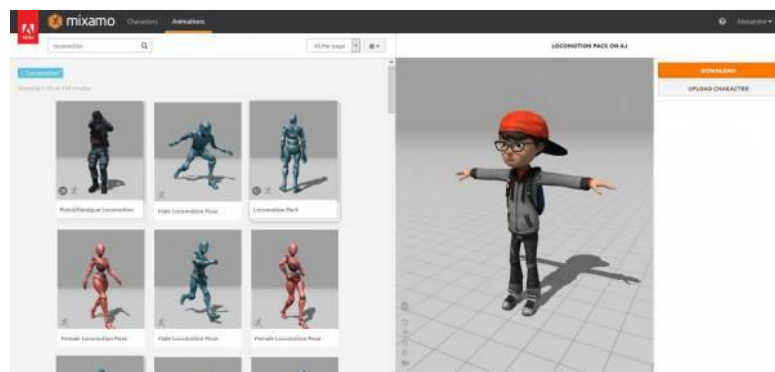
1. Contrôleur simple d'un personnage animé avec MecAnim

Les données : Mixamo

Pour aller plus vite, [vous pouvez télécharger le zip ici](#).

[Mixamo](#) est un site offrant des personnages (Maillage) et des animations.

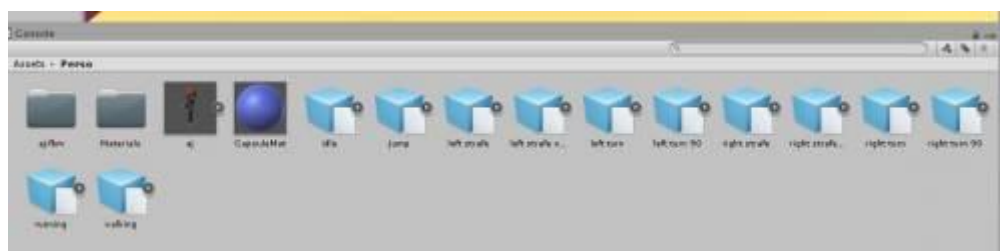
- Allez dans l'onglet "Character" sélectionnez le personnage qui vous intéresse.
- Allez dans l'onglet "Animation", tapez "locomotion pack" dans le moteur de recherche pour trouver un pack d'animation. Cliquez sur celui qui vous intéresse.
- Cliquez à droite sur "download". Le zip doit contenir le maillage et les animations.



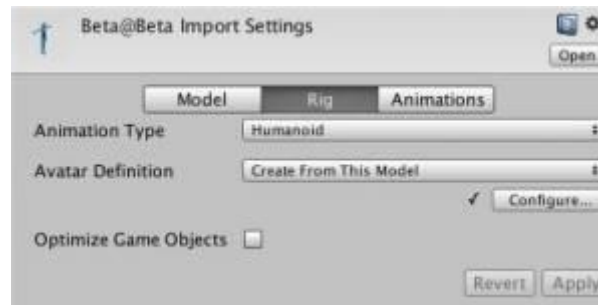
Il y a également des données dans https://www.assetstore.unity3d.com/l/assets_store_d'Unity.

Importer les données dans Unity

- Votre personnage va avoir de nombreux composants différents, il est important de créer un nouveau dossier. Menu Assets > Create > Folder, nommez le "Perso".
- Importer vos animations en faisant glisser/déposer depuis l'explorateur (ATTENTION : ne pas faire import qui ne prendra pas les fichiers meta)
- Sélectionnez toutes les animations, puis dans le panel "Inspector" à droite cliquez sur le bouton "Rig"

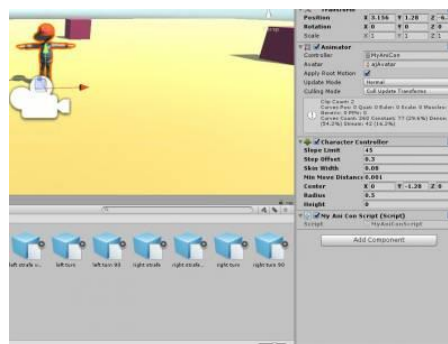


- et mettez le mode "humanoid" dans "Animation Type" (qui y est peut-être déjà), à faire également pour le mesh du personnage.

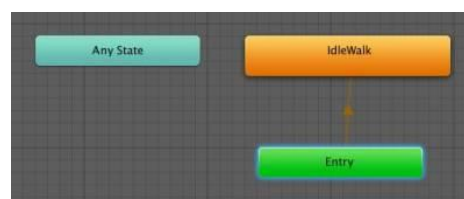


Contrôleur d'animation

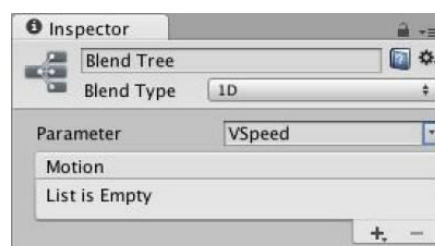
- Bouton droit, créez un "Animation Controller" nommé par exemple "MyAniCon".
- Créez un script C#, nommé "MyAniConScript".
- Placez votre personnage sur la scène, et sélectionnez le personnage.
- Dans le panel "inspector" à droite, placez le contrôleur d'animation "MyAniCon" dans Animator/Controller
- Puis faites "add component" pour ajouter le "MyAniConScript"



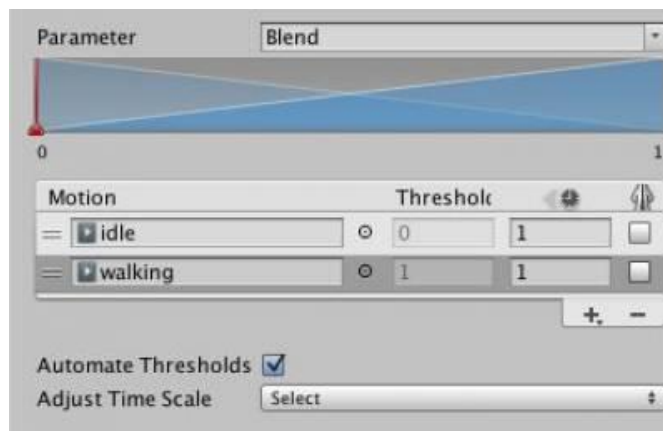
- Fenêtre "Animator", clique droit puis "Créer > Nouveau Blendtree"



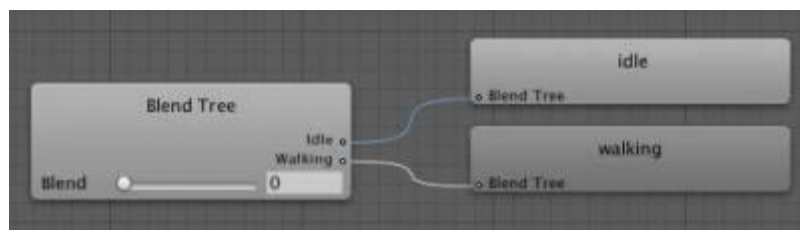
- Dans le coin supérieur gauche de la fenêtre "Animator", sélectionnez l'onglet Paramètres et créez un nouveau paramètre Float et nommez-le VSpeed



- Fenêtre Animateur, double-cliquez sur IdleWalk. Réglez le paramètre sur VSpeed. Cliquez sur le bouton '+' et dans Actifs, Animations, faites glisser l'animation Idle dans le dossier Animations Motion Field.
- Répétez le processus et faites glisser l'animation de marche dans le deuxième champ Animation.



- Votre BlendTree doit ressembler à ceci



- Pour passer d'un état à un autre nous avons besoin d'un script C#. Ouvrez MyAniConScript.cs et indiquez ceci

```
void Start () {
    myAnimator = GetComponent<Animator>();
    Debug.Log("MyAniConScript: start => Animator");
}

// Update is called once per frame
void Update () {
    myAnimator.SetFloat("vSpeed", Input.GetAxis("Vertical"));
    Debug.Log("vSpeed = " + Input.GetAxis("Vertical"));
}
```

- Ca bouge !!! mais seulement une fois (enfin ça dépend de vos animations d'entrée) ... Allez dans le "Blendtree" et pour chaque animation coché "Loop time"

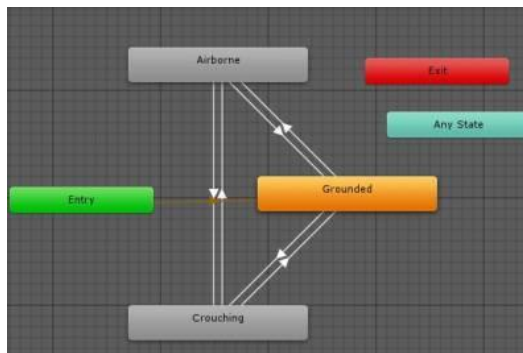


Droite et gauche

- Ajoutez un paramètre hSpeed
- Améliorer votre Blend Tree pour qu'il prenne en compte les virages à gauche/droite, la course, etc. Ne mettez dans le blendtree que les déplacements classiques.
- Dans l'Inspector passer le système d'interpolation en motion field "2D Freeform Cartesian". Le centre va correspondre à Idle, les points à droite au animation tournant à droite, ceux à gauche les virages à gauche et les points en haut pour les marches et courses.
- Remarque : les données ne comportent que des animation "tourner à droite", vous pouvez les inclure et cocher "Mirror" pour avoir les animations "tourner à gauche"



- [Lisez ceci pour comprendre la différence entre une transition et un BlendTree.](#)
- [Lisez également une discussion à ce sujet ici \(court mais important\).](#)
- Globalement vous allez inclure toutes vos animations dans un BlendTree quand l'animation désirée doit être continue et progressive, ce qui sera bien adapté pour la 'locomotion'. Les transitions seront utilisées pour passer à une animation directement, comme un saut, ou s'accroupir. Ci-dessous, me graphe d'animation comporte un état "Ground" pour les déplacements classiques sur le sol, un état "Crouching" pour les déplacements accroupis et un état "Airborne" pour les sauts.



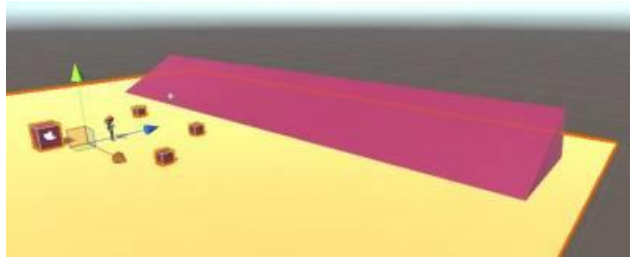
A faire

- Sur le même principe, améliorez votre contrôleur pour qu'il prenne en compte les sauts et les déplacements accroupis.
- Ajouter la cinématique inverse d'Unity pour que les pieds soient bien placés sur le sol. Voir la question suivante. Attention, ce point est différent de la partie du TP qui demande de recoder un algorithme de cinématique inverse en entier « Fabrik ».

2. Cinématique inverse pour placer les pieds sur le sol

Pour placer les pieds sur un sol non plat, nous allons regarder les fonctionnalités d'Unity pour faire de la cinématique inverse.

- [La doc d'Unity à propos de la cinématique inverse \(Inverse Kinematics\)](#)
- Ajoutez un plan incliné à votre scène



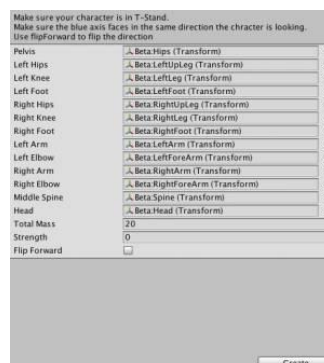
- Modifiez votre personnage pour que ses pieds soient bien placés sur le sol tout au long de la marche : l'avant du pied et le talon bien orientés sur le sol à chaque pas.
- [La doc sur l'IK](#)
- [Une vidéo tutorial sur l'IK](#)

3. Interaction avec l'environnement : la physique

- Créez deux cubes l'un sur l'autre comme ceci



- Ajoutez des Component "Rigid Body" à chacun des deux cubes
- Pour que le personnages interagissent avec les cubes, vous devez ajouter un "Ragdoll" (traduction : poupée désarticulée) à votre personnage. Ce processus va créer un Rigidbody pour chaque articulation de votre personnage.
 - Pour cela, bouton droit créez un ragdoll,
 - puis faites un glisser de chaque articulation de votre squelette dans la bonne case du ragdoll.



4. Idée : pour aller plus loin

- (facile) Ajoutez à votre personnage, la possibilité de shooter des sphères.
- (très facile) Construisez un mur de brique (avec un script, il ne s'agit pas de placer les dizaines de cube à la main 😊)
- Ajouter un 2e personnage fixe qui regarde toujours dans la direction de votre personnage : tourne la tête, puis éventuellement tourne le corps. Ce personnage déplace sa main vers le projectile que vous lui envoyez : [voir par exemple cette vidéo](#).