

UNITY - TP2

FIRST PERSON CONTROLLER

Unity inclut plusieurs controllers qui facilitent la navigation dans la scène. Ouvrez votre scène Terrain et glissez l'objet `FirstPersonController` à partir de la fenêtre Projet, **Assets > Characters > FirstPersonCharacter > Prefabs > FPSController**.

Désactivez l'Audio Listener et le script de contrôle de la camera principale pour éviter les conflits. Exécutez votre jeu et contrôlez votre « avatar » avec les flèches (ou les touches W, A, S et D), la souris (contrôle du « regard ») et la barre d'espace (saut). Testez les limites du contrôleur en essayant de lui faire escalader une montagne ☺

Dans l'Inspector, vous pouvez modifier la taille du contrôleur (height), le centre de la capsule (Center) et son rayon. L'option `SlopeLimit` limite les surfaces que le joueur peut emprunter. Si l'on indique 30, par exemple, le joueur ne pourra pas emprunter les surfaces dont l'angle par rapport au plan est supérieur à 30 (une valeur de 90° est conseillée).

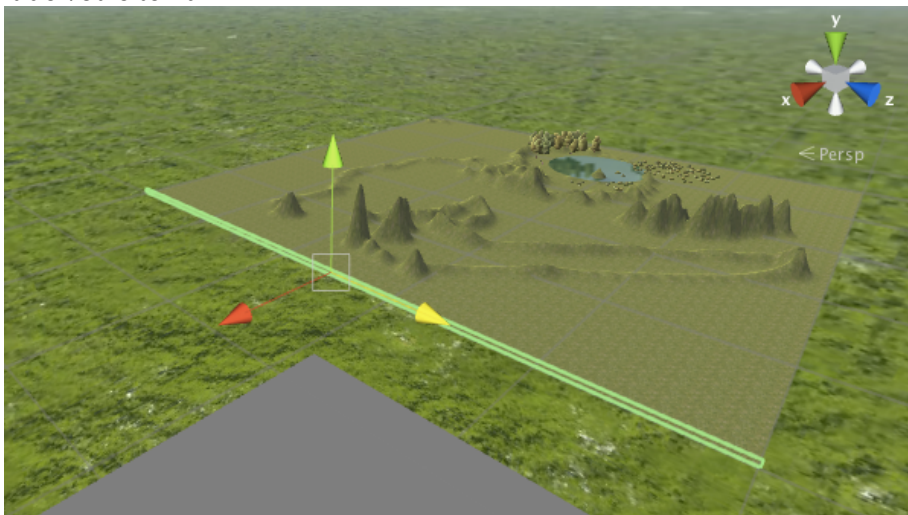
- L'option `StepOffset` indique que le personnage pourra emprunter une marche seulement si elle est au maximum à la hauteur indiquée (il est conseillé d'avoir une valeur entre 0,1 et 0,4 pour un personnage de 2m)
- Le paramètre `Min Move Distance` définit le déplacement minimum du personnage (valeur conseillée : 0)
- Le paramètre `Skin Width` donne le chevauchement maximal possible pour deux colliders (voir ci-dessous). Il est conseillé de donner à ce paramètre une valeur supérieure à 0,01 et environ de 10% du rayon (une valeur très basse peut bloquer le personnage).

Si ce n'est pas encore arrivé ☺ sortez le contrôleur du terrain...que se passe-t-il ?

LES COLLIDERS

La détection de collisions est un élément essentiel d'un environnement virtuel réaliste. Pour éviter à notre avatar de traverser les arbres et de tomber dans le vide, nous allons construire 4 murs autour du terrain.

Créez un objet `GameObject` vide en utilisant **GameObject > Create Empty** et renommez-le « frontières ». Dupliquez l'objet « frontières » et renommez la copie « mur1 ». Glissez le nouvel objet sous « frontières ». Maintenant sélectionnez l'objet « mur1 » et cliquez sur **Add Component > Physics > Box Collider**. Dupliquez l'objet « mur1 » 3 fois et renommez les objets « mur2 », « mur3 » et « mur4 ». Modifiez les attributs de chaque mur pour qu'ils correspondent à la taille et l'emplacement de votre terrain :

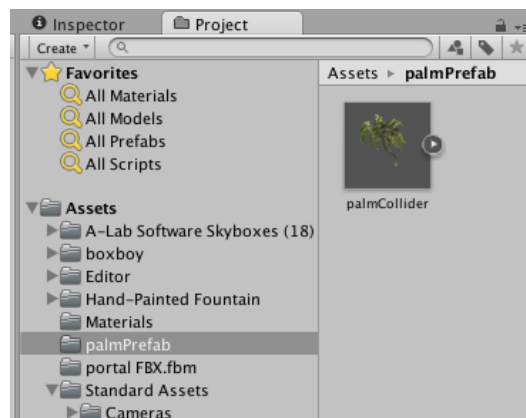


Testez votre jeu !

Pour les arbres, nous allons créer un nouvel objet à partir de l'arbre de base et lui rajouter un Collider qui sera placé autour du tronc uniquement. Au travail !

Dans la fenêtre Project, sélectionnez l'arbre Palm dans **Assets > Standard Assets > Environment > SpeedTree > Palm** et glissez-le vers la scène (près de votre contrôleur pour vous faciliter la tâche). Ajoutez à l'arbre sélectionné un collider de type capsule : **Add Component > Physics > Capsule Collider**. Testez le jeu et essayez de traverser l'arbre, ce n'est plus possible ! Vous pouvez modifier le collider de manière à entourer uniquement le tronc de l'arbre en modifiant son rayon.

Maintenant, comment remplacer tous les arbres de notre scène ? Dans la fenêtre Assets créez un nouveau dossier « palmPrefab ». Créez dans ce dossier un nouveau Prefab, en faisant clic droit et puis **Create > Prefab**. Glissez ensuite l'arbre dans le Prefab et effacez-le de l'hérarchie. Glissez-le à nouveau dans la scène mais cette fois à partir du prefab.



Pour modifier les autres arbres, sélectionnez le terrain et cliquez ensuite sur l'outil « Place Trees » dans l'Inspector. Cliquez sur l'arbre « Palm » et puis sur « **Edit Trees > Edit Tree** ». Dans la fenêtre, cliquez sur le cercle à côté de l'arbre et sélectionnez ensuite l'arbre prefab que vous avez créé. Validez et vérifiez dans l'Inspector que la case « Enable Tree Colliders » est cochée. Si vous avez d'autres types d'arbre, il faudra refaire la procédure pour chacun.

Testez votre jeu !

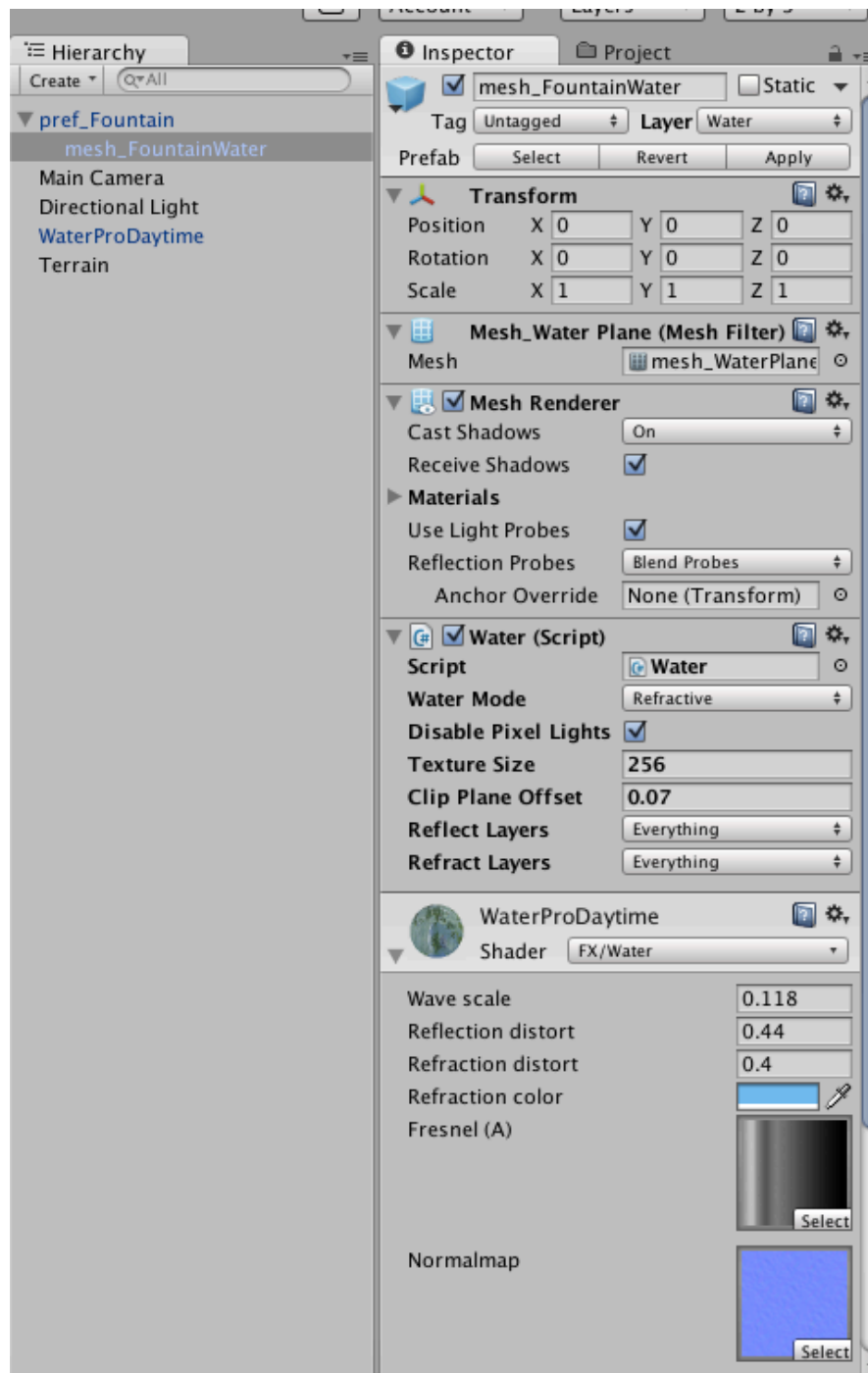
LE PORTAIL MAGIQUE

Nous allons maintenant faire un lien entre les deux scènes que nous avons créées la semaine dernière. Le nom de chaque scène est important, je l'appellerai la scène 1 «BoxBoy» ☺

Téléchargez de l'asset store le modèle de fontaine suivant (« Hand painted fountain ») :

<https://www.assetstore.unity3d.com/en/#!/content/41694>

Ce modèle permet l'ajout de l'eau grâce au maillage (plat, orienté horizontalement) mesh_FountainWater. Attachez à cet objet le script Water, disponible dans **Standard Assets > Environment > Water > Water > Scripts** (ou en faisant click sur All Scripts dans l'onglet Projet). Attachez également un matériau au mesh_FountainWater, par exemple, WaterProDaytime. Vous pouvez modifier dans le shader la couleur de l'eau et d'autres propriétés :



Sélectionnez l'objet Fontaine et dans l'Inspector, activez l'option IsTrigger. Cette option permet d'appeler une fonction lorsque le collider de l'objet est activé. Créez un nouveau script, passage.js :

```
var levelToLoad : String = "BoxBoy"; //le nom de votre scène
function OnTriggerEnter( other : Collider ) {
    Application.LoadLevel( levelToLoad );
}
```

Le script permet, lorsqu'il détecte la collision entre le FPC et la fontaine, de charger une scène dont le nom est donné. Associez le script à la porte. Utilisez ensuite le menu **File > Build Settings** pour rajouter la scène au jeu (Add Current). Faites de même pour la scène BoxBoy.

Testez votre jeu !

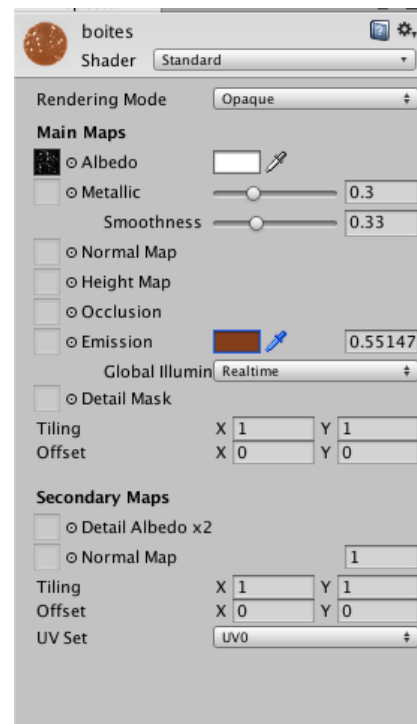
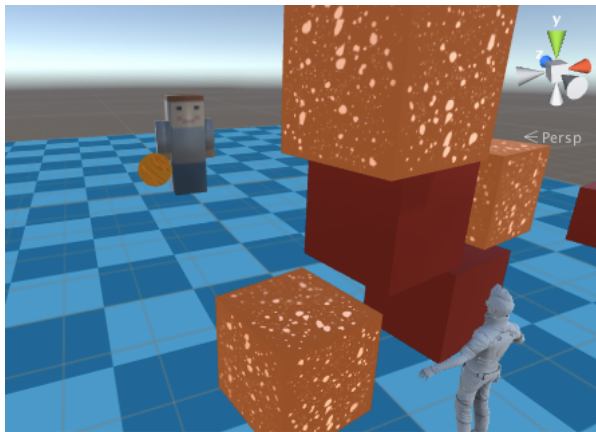
Maintenant lorsque vous vous approchez du portail magique vous êtes télé-transporté auprès de BoxBoy ☺ !

LE MAGICIEN BOXBOY

Ouvrez votre scène BoxBoy. Enlevez le script de contrôle par les flèches de BoxBoy et créez un ThirdPersonCharacter (*Standard assets > Characters > ThirdPersonCharacter > Prefabs > ThirdPersonController*).

Le but du « jeu » dans ce niveau est de se déplacer dans l'espace en évitant les obstacles et en essayant de trouver le BoxBoy magicien pour récupérer la lampe magique. Tout un programme ! ☺

Nous allons d'abord améliorer l'apparence des obstacles, en utilisant des Materials et Shaders, comme dans le TP précédent. Créez un Material (*clic droit sur Project*, puis *Create > Material*) pour les cubes, un autre pour la sphère et un dernier pour le plan, avec les couleurs et textures de votre choix :



Séparez la sphère de Boxboy et associez-lui un Rigidbody. **Testez la scène.** La sphère ne se comporte pas de manière très réaliste, elle ne rebondit pas ! Nous allons améliorer cela grâce aux Physics Materials. Créez un nouveau Physics Material et associez- le à la sphère. Modifiez dans le PhysicsMaterial la propriété bounciness et testez le jeu. Vous pouvez augmenter l'intensité des rebondissements en modifiant la propriété bounceCombine à « maximum ». Voilà qui est mieux !!

Créez un prefab pour la sphère et effacez-là de la scène. Créez un nouveau GameObject vide (GameObjectControl) et créez un nouveau script pour programmer le placement aléatoire des « météorites » :

```
public var prefabSphere : GameObject;
public var numberOfObjects = 20;
public var minX = -25 ; //adaptez aux dimensions de votre plan !
public var maxX = 25;
```

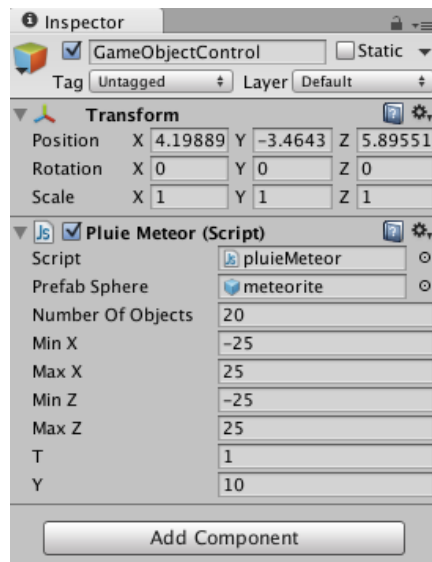
```

public var minZ = -25 ;
public var maxZ = 25 ;
public var t = 1 ; // temps d'attente entre la création d'une sphère et la suivante
public var y = 10; // position Y des sphères (hauteur)

function Start () {
    for (var i = 0; i < numberOfObjects; i++) {
        yield WaitForSeconds(t);
        var randomPos = Vector3(Random.Range(minX, maxX), y, Random.Range(minZ, maxZ));
        Instantiate(prefabSphere, randomPos, Quaternion.identity);
    }
}

```

Associez le script au GameObjectControl. Modifiez les paramètres pour associer le prefab sphère à la variable prefabSphere du script et pour affiner l'effet.



Ajoutez maintenant à l'objet BoxBoy un Boxcollider. Activez l'option isTrigger qui nous aidera à reconnaître le contact avec notre avatar.

LA LAMPE MAGIQUE

Importez le package MagicLamp (*Import package > Custom* ou glisser-déposer) et ajoutez à la scène l'un des modèles disponibles (LOD0 1898 tris, LOD1: 1200 tris, LOD2: 800 tris). Ajoutez à la lampe un système de particules *Component > Effects > ParticleSystem*. (<http://docs.unity3d.com/Manual/class-ParticleSystem.html>). Décochez l'option Looping pour que les particules s'envolent à tout jamais. Glissez votre lampe magique dans un nouveau prefab et effacez-la ensuite.

Créez le script ci-dessous et associez-le à BoxBoy. Il nous permet de détecter la présence de l'avatar, d'afficher la lampe et de laisser l'avatar la prendre :

```

public var prefablamp : GameObject;
var lamp : GameObject;
function OnTriggerEnter( other : Collider ) {
    if (other.gameObject.CompareTag ("Player")) {
        var pos = Vector3(transform.position.x, transform.position.y+2,
            transform.position.z-2);
        lamp = Instantiate(prefablamp, pos, Quaternion.Euler(-90,90,0));
    }
}

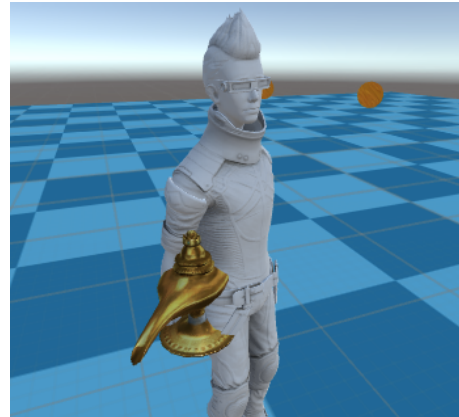
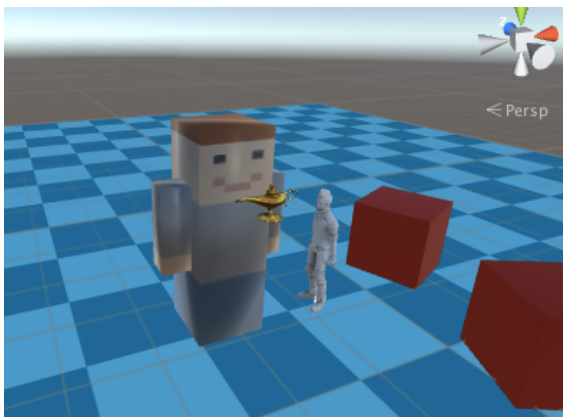
```

```

    }
};
// faisons disparaître le magicien ! et attrapons la lampe !
function OnTriggerExit (other : Collider) {
    if (other.gameObject.CompareTag ("Player")) {
        Destroy(gameObject);
        lamp.transform.parent = other.gameObject.transform;
        lamp.transform.localPosition=Vector3(0.5,0,0);
    }
}

```

Modifiez les valeurs des transformations pour qu'elles correspondent à l'échelle et position de vos modèles.



PUBLICATION DU JEU

Pour pouvoir montrer votre jeu à votre entourage (pour que mamie soit fière ☺) et pouvoir jouer en dehors de l'environnement Unity, il vous faut sélectionner **File > Build Settings**. Choisissez la plateforme d'export : Unity peut exporter vers Windows, Mac et le web grâce à l'Unity Web Player. Le Web Player est un plugin fonctionnant sur divers navigateurs, vous pouvez le trouver ici : <http://unity3d.com/webplayer/>

Bravo ! Vous avez complété la première étape de la découverte Unity ☺ Envoyez une capture d'écran de vos deux scènes à nancy.rodriguez@lirmm.fr. Si vous avez la possibilité de mettre votre jeu en ligne, envoyez aussi votre url !!

PS : Pour améliorer l'apparence du Player, vous pouvez télécharger les assets ici : <https://www.assetstore.unity3d.com/en/#!/content/21064>, et importer les textures du ThirdPersonCharacter (Ethan). Ajoutez ensuite la texture EthanAlbedoSpecular.png de Third Person Character Textures à « l'Albedo » du matériau EthanWhite.



BON WEEK-END ET À BIENTÔT !