

### TP3 Réalisation d'une scène virtuelle

**L'objectif du TP est de concevoir une scène virtuelle avec un décor, un personnage et interagir avec le personnage.**

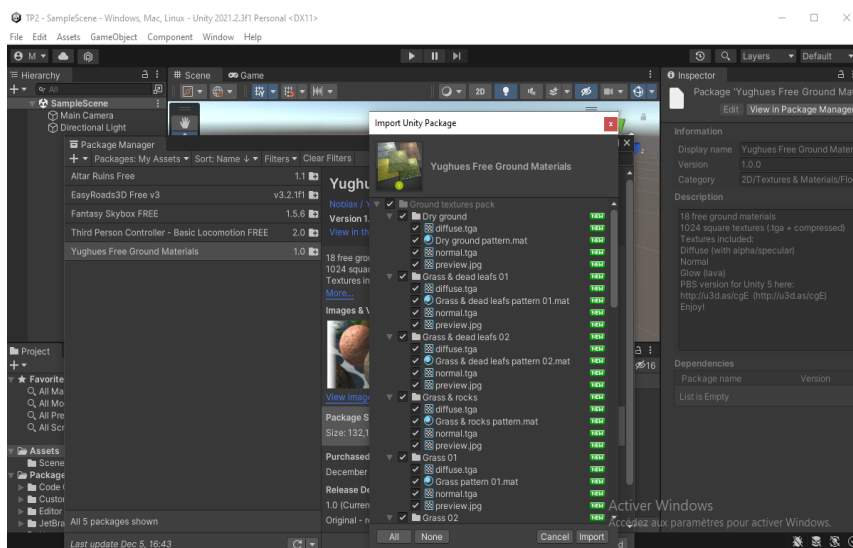
Vous pouvez télécharger Unity et Unity hub via les liens suivant : <https://unity.com/fr/download>

Télécharger les Assets via l'Asset Store <https://assetstore.unity.com/>

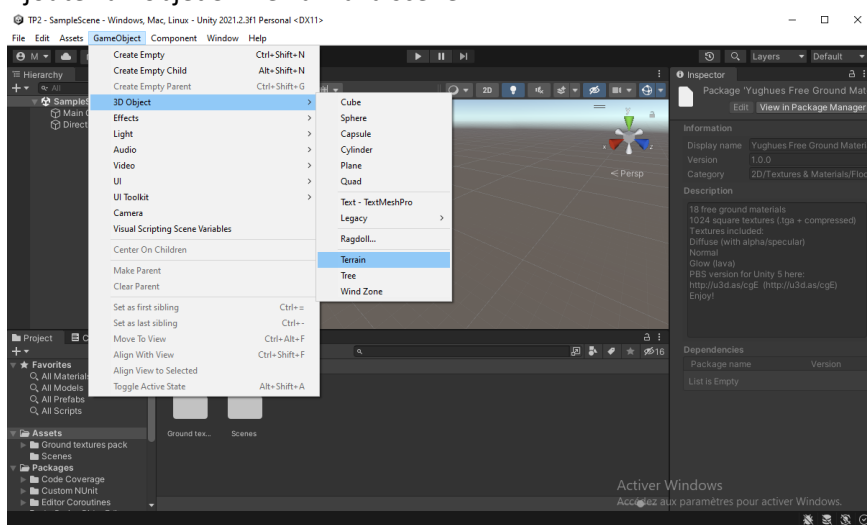
#### Exercice 1 : réalisation d'une scène qui comprend un terrain, Arbre, bâtiment

1. Démarrez un nouveau projet en sélectionnant 3D comme modèle de projet.
2. Ajoutez le package choisi (préalablement téléchargé de l'Asset Store exemple *Yughues free Ground Materiel* pour les textures des sols)

**Window -> Package Manager -> My Assets**

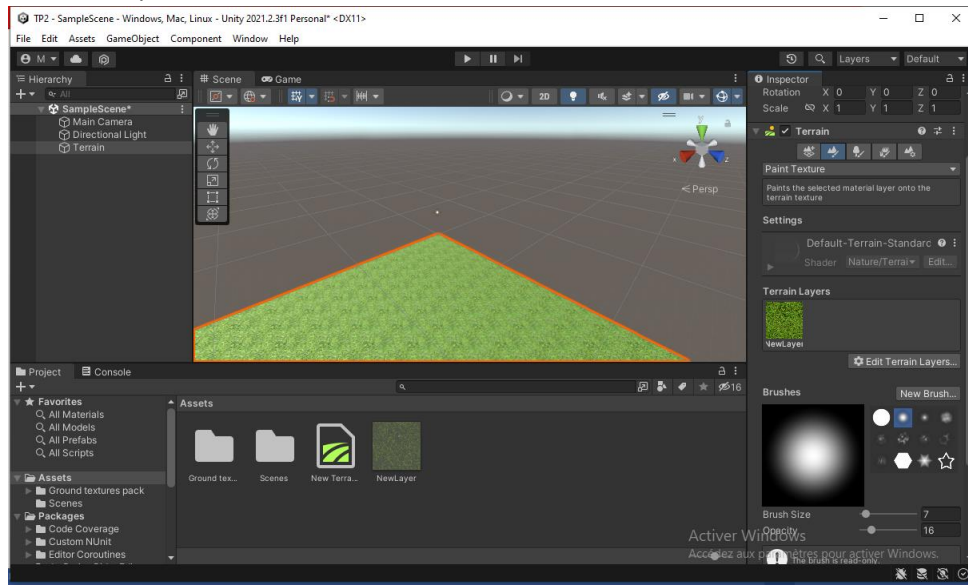


3. Ajoutez un objet 3D Terrain à la scène

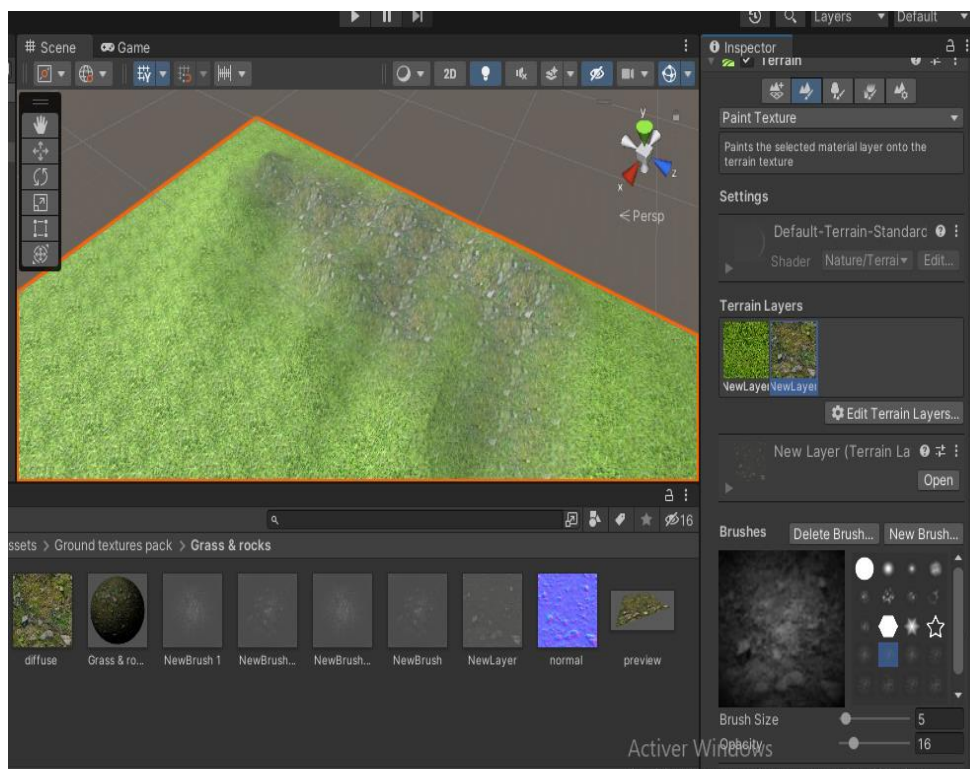


4. Ajoutez une texture au terrain

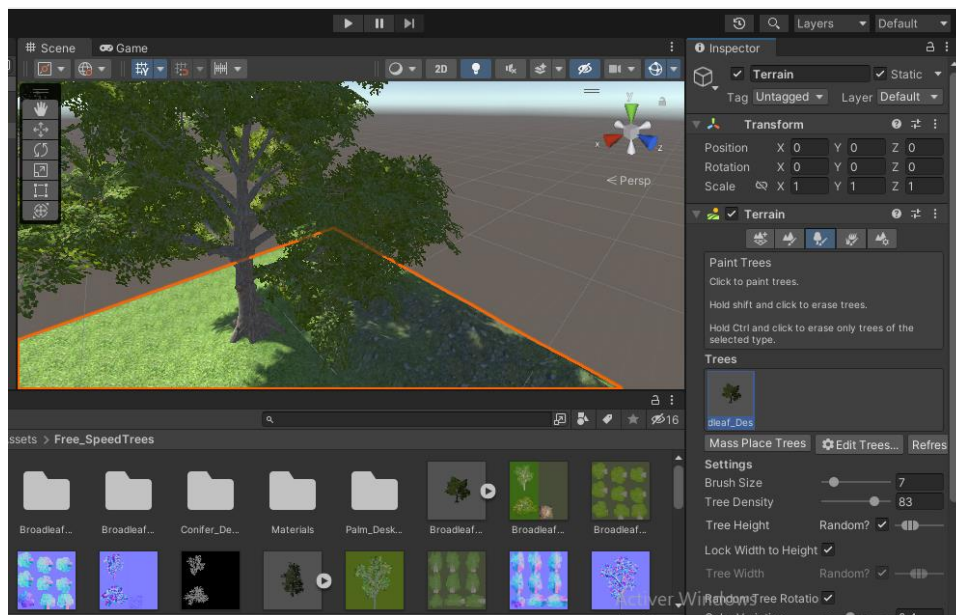
Dans la fenêtre Propriétés **Paint Terrain ->Edit Terrain Layers->Create Layers** et sélectionnez la texture que vous souhaitez



5. Ajoutez des courbures à votre terrain dans **Paint Terrain ->Raise or Lower Terrain -> Click gauche** sur le terrain pour ajouter les courbures suivant la dimension sélectionnée
  - Vous pouvez changer la texture des courbures pour ajouter des détails à votre terrain.



6. Ajoutez un arbre sur le terrain **Paint Trees->Edit Trees->Add Tree** et choisir le modèle suivant les Assets que vous avez (vous pouvez importer les Assets au préalable de l'Asset store)



7. Ajoutez des objets et des décors autant que nécessaire en glisser déposer, en suite modifier a souhaits jusqu'à obtenir le résultat attendu.



## Exercice 2 :

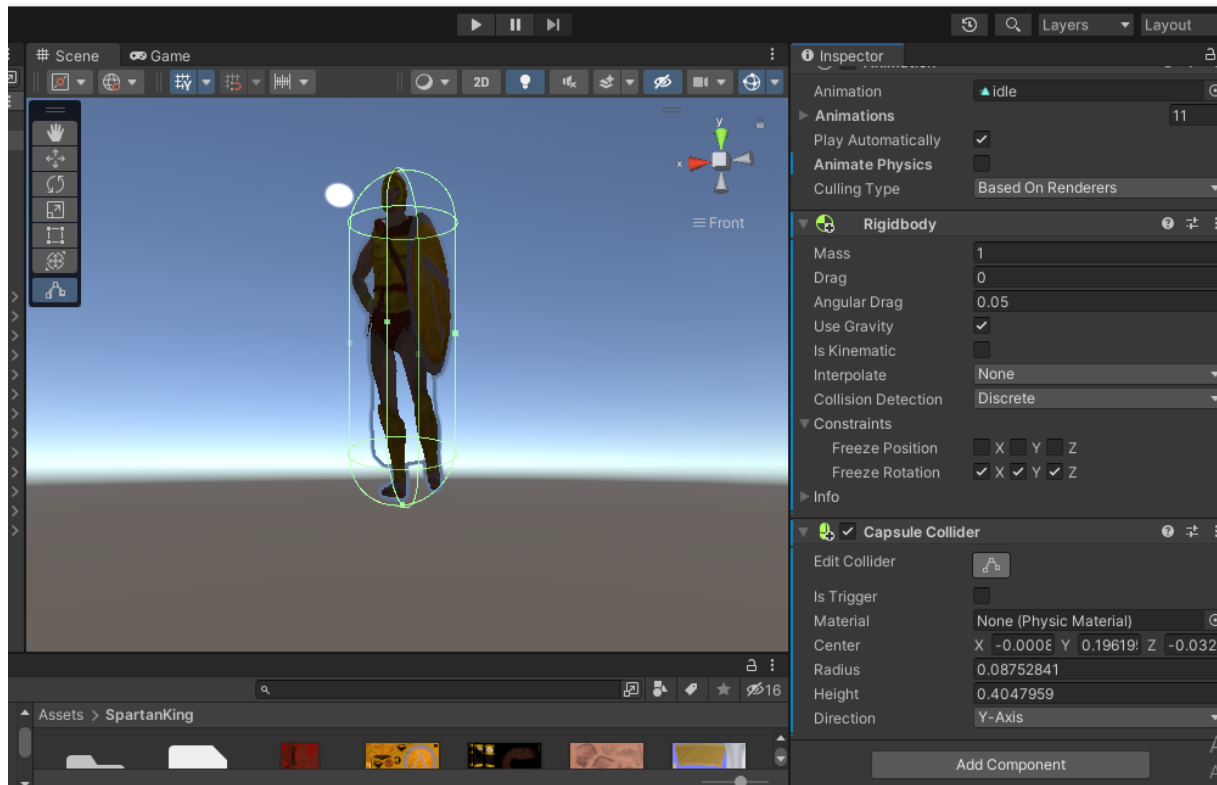
### Ajout d'un personnage

1. Ajoutez l'Asset du personnage souhaité (exemple : Animated Sparten King de l'Asset Store).
2. Glissez déposez le Préfab du personnage dans la scène.
3. Déplacer la camera de la scène dans le personnage pour suivre son déplacement (la camera devient enfant de l'objet personnage) pour rendre l'animation a la 3eme personne.
4. Ajoutez des caractéristiques physiques au personnage

**Propriétés->Add Component ->Physique->Rigid Body**

en faisant attention à Activer **Freez Rotation** pour éviter que le personnage ne soit en déséquilibre.

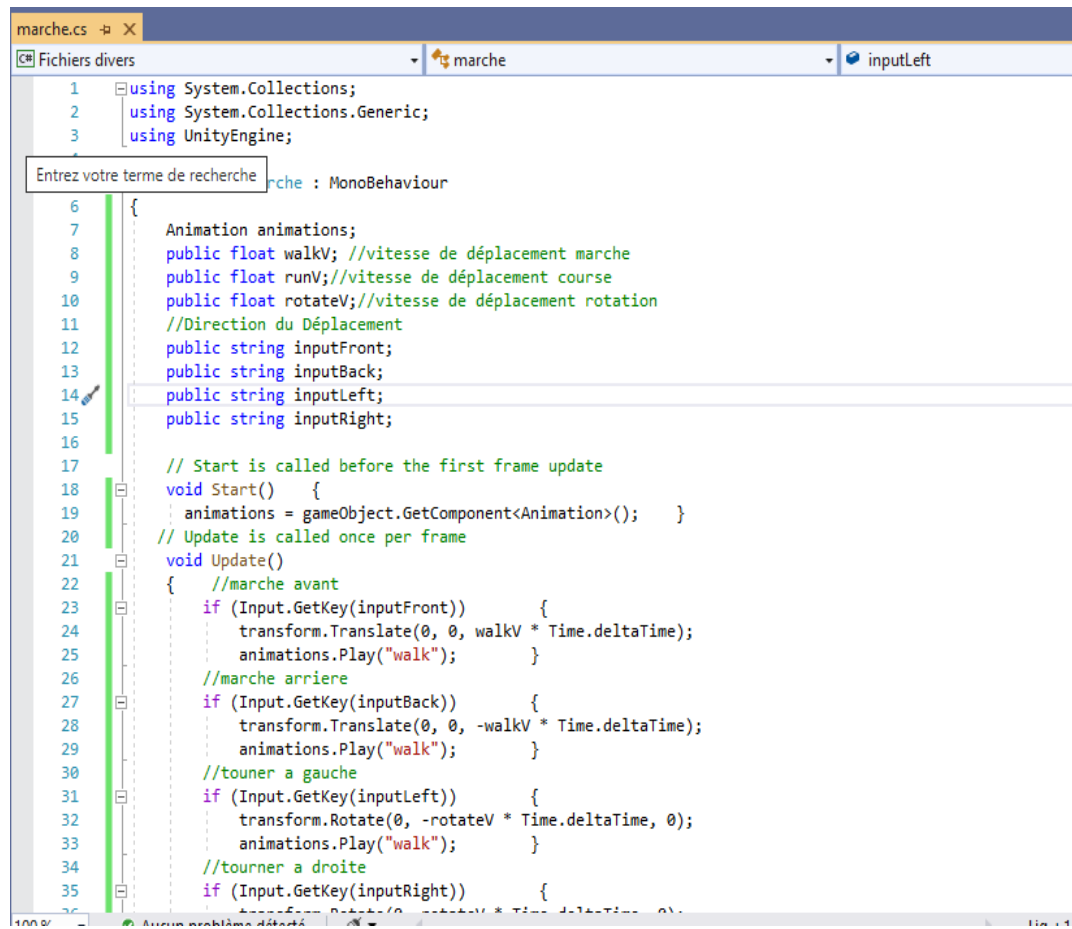
5. Ajoutez un Volume Englobant pour détection de collision, la forme la plus adaptée à notre personnage est une capsule, **Propriétés->Add Component ->Physique->Capsule Collider**  
Adaptez la forme de la capsule au plus proche du personnage en activant **Edit Collider**



**Animation du personnage :** le personnage se déplace en fonction des touches du clavier appuyées.

6. Créez un script C#. **Asset->click Droit->Create->C# Script** en suite glissez déposer le script sur le personnage pour le rattacher au personnage.
7. Ouvrez le script dans **Visual Studio** pour programmer les animations.
8. Déclarez les variables nécessaires : Animation, Direction du déplacement, vitesse de déplacement.
9. Spécifiez les initialisations au démarrage de l'application dans la fonction **Start()**.
10. Définir le comportement du personnage dans la fonction **Update()**.  
Enregistrez le code saisi, la compilation se fait sur Unity automatiquement.
11. Donnez des valeurs aux variables définies dans le script (la vitesse de marche et course et les touches du clavier pour la direction du déplacement)
12. Complétez le script pour ajouter d'autres animations : course, Rotation .....





```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;

4
5
6 {
7     Animation animations;
8     public float walkV; //vitesse de déplacement marche
9     public float runV; //vitesse de déplacement course
10    public float rotateV; //vitesse de déplacement rotation
11    //Direction du Déplacement
12    public string inputFront;
13    public string inputBack;
14    public string inputLeft;
15    public string inputRight;
16
17    // Start is called before the first frame update
18    void Start() {
19        animations = gameObject.GetComponent<Animation>();
20    }
21    // Update is called once per frame
22    void Update()
23    {
24        //marche avant
25        if (Input.GetKey(inputFront)) {
26            transform.Translate(0, 0, walkV * Time.deltaTime);
27            animations.Play("walk");
28        }
29        //marche arriere
30        if (Input.GetKey(inputBack)) {
31            transform.Translate(0, 0, -walkV * Time.deltaTime);
32            animations.Play("walk");
33        }
34        //tourner a gauche
35        if (Input.GetKey(inputLeft)) {
36            transform.Rotate(0, -rotateV * Time.deltaTime, 0);
37            animations.Play("walk");
38        }
39        //tourner a droite
40        if (Input.GetKey(inputRight)) {
41            transform.Rotate(0, rotateV * Time.deltaTime, 0);
42        }
43    }
44 }
```

