**Introduction Unity 3D**

**INSTALLATION UNITY 2020.3.8**

Nouveau projet > 3D Core > Nom Projet > créer

Créer une plateforme > right click dans hiérarchie > 3D Object > cube > modifier pour faire une plateforme

Créer le joueur > 3D Object > capsule

Faire des Matériaux > right click assets > Cérate > Materials > changer couleur dans l’inspecteur avec “l’albedo”

Glissez le matériel sur le sol pour lui donner sa couleur

Dans les assets créer des “Folder” pour organiser votre projet avec un folder “Materials” qui va contenir tous vos matériaux

Créer une physique au joueur : dans inspector > ajouter un component >”Rigidbody”

En appuyant sur “Play” on peut voir que la gravité est activée, le joueur tombe sur le sol et ne passe pas à travers car les deux ont le “collider” une hitbox

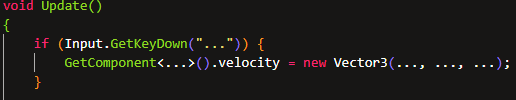
En utilisant les outils changer la position et la rotation de la caméra pour l’adapter au joueur

Créons un script pour les mouvements du joueur inspector > nouveau component > new script > “PlayerMovement” et ouvrez-le

Start() pour ce qui se fait au début Update() pour ce qui se fait en permanence

Maintenant vous allez créer des variables de type “public float” pour la vitesse et la force de saut du joueur, vous permettant de modifier ses informations depuis unity directement dans l’inspector du joueur

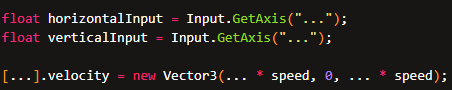
Voici un exemple de comment faire un mouvement à votre joueur, si la touche pressée est “space”. On va chercher le component “Rigidbody” qui gère la physique du joueur et lui ajouter un vecteur de x = 0 y = “Force de saut” et z = 0



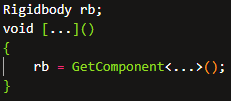
Bravo votre joueur peut sauter !!

Pour le faire bouger dans les autres directions vous devez récupérer les axes Horizontal et Vertical puis comme vu ci-dessus, ajouter un vecteur à la physique du joueur.

Réfléchissez à comment remplir les […] pour faire en sorte de faire bouger votre joueur



Vous avez dû remarquer que le joueur tombe et se mange le sol, pour régler ça inspector player > Rigidbody > Constraint > Freeze all rotation

Maintenant vous avez remarqué un nouveau problème : lorsqu’on appuie sur “Sauter” on ne peut pas se déplacer en même temps. Pour ça on va d’abord récupérer la variable Rigidbody dans le code pour ne pas à avoir à appeler ‘GetComponent<Rigidbody>()’ des dizaines de fois.

On va donc créer une variable rb et l’assigné dans la fonction approprié.

Maintenant que vous avez cette variable vous pouvez remplacer tous vos “GetComponent<Rigidbody>()” par “rb”.

Pour régler le problème, on va devoir remplacer toutes les position 0 de nos vecteur3 par les coordonnées actuelles du joueur comme dans cet exemple :

Vous avez aussi remarqué que la caméra ne suit pas le joueur. Dans le menu “hiérarchie” glisser la caméra dans “joueur” et placer la en fonction de vos préférences derrières le joueur.

Maintenant utiliser la création d’objet unity pour créer une map à l’aide de plateforme, une sorte de parcours

Vous avez aussi dû remarquer un gros problème : on peut sauter à l’infini !

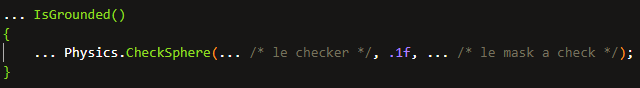
Pour régler ce problème on va créer un “Empty Object” qu'on va faire glisser dans le joueur dans le menu “hiérarchie” puis on va le placer au pied du joueur dans la scène.

Ensuite vous aurez besoin de créer ce que l’on appelle un “Layer Mask”, dans le menu inspector de vos objets qui servent de sols, il faudra créer un “Layer” qu’on appellera “Ground” et l’appliqué à toute votre scène.

Dans votre code créer ces deux variables



Ensuite vous devrez créez une fonction qui renverra Vrai si notre joueur touche bien le sol si non : faux.

On utilisera la fonction Physics.CheckSphere pour cela et n’oubliez pas d’assigner les bonnes valeurs aux variables sur l’inspector du joueur dans la section “script”

Il vous reste plus qu’à rajouter une condition pour que le joueur puisse sauter seulement s'il touche le sol !

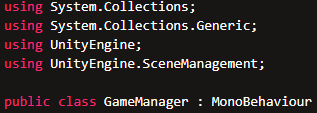
On va régler un dernier problème, le joueur reste collé aux parois des plateformes !

Pour régler ça on va aller créer dans les assets un Matériel Physique et mettre toutes ses valeurs à 0 puis glisser ça sur le joueur sur la scène.

Il nous faut bien une fin !

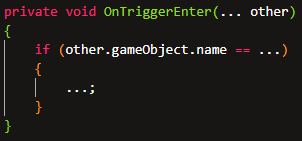
Vous allez créer une plateforme de fin et lui créer un script qu’on appellera “Game Manager”.

Vous pourrez effacer tout ce que Unity a écrit à votre place et écrire cela à la place :



On va ensuite essayer de détecter le joueur sur la plateforme, pour cela on va utiliser une fonction d’Unity et les tags.

Tout d’abord on va sur Unity, on sélectionne le joueur, dans l’inspector on va lui attribuer le tags “Player” et tu vas modifier la case “Is trigger” du “Collider” de ta plateforme finale pour permettre au joueur de passer à travers tout en temps détectant sa présence

Voici la fonction qui prend en argument un component extérieur le “Collider” du joueur. Cette fonction va vérifier que le Game Object rentrant en contact avec la plateforme est bien du tag “Player”.

Une fois cette fonction vérifier, elle exécutera un redémarrage de la scène grâce au “SceneManager” et la fonction “LoadScene”. Je te laisse regarder comment utiliser ces fonctions !

On va refaire la même chose mais cette fois quand le joueur tombe de la map. Pour cela, rajoute cette ligne en haut du script contrôlant le joueur



Tu vas devoir rajouter une condition pour vérifier en permanence que la position de ton joueur n’est jamais inférieure à la position des éléments de ta map

BRAVO !!! T’AS REUSSI CETTE INTRODUCTION A UNITY 3D !

Si tu as le temps je te propose de faire un menu de fin grâce au Scène Manager, un système d’ennemis, même des plateformes mouvantes ou sinon je te laisse imaginer des choses à rajouter à ton jeu !