**Introduction Unity 3D**

**INSTALLATION UNITY 2020.3.8**

Nouveau projet > 3D Core > Nom Projet > créer

Créer une plateforme > clique droit dans hiérarchie > 3D Object > cube > modifier pour faire une plateforme

Créer le joueur > 3D Object > capsule

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

Glissez le matériel sur le sol pour lui donner sa couleur

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” qui est ce que l’on appelle une « hitbox » dans le milieu du jeu vidéo, une zone invisible autour des objets

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, la force de saut, la vitesse de la caméra du joueur, vous permettant de modifier ses informations depuis unity directement dans l’inspector du joueur

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Voici un exemple de comment faire un mouvement à votre joueur, si la touche espace est pressée. On va chercher le component “Rigidbody” qui gère la physique du joueur et ajouter une force de saut à son Vector3.

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

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 3D à la physique du joueur. 

Puis on vient créer un nouveau vecteur 3D qui contient nos axes que l’on multiplie à la vitesse et au temps en jeu que l’on ajoutera à notre « RigidBody »



Vous avez dû remarquer que le joueur tombe et se mange le sol, pour régler ça inspector player > Rigidbody > Constraints > Freeze rotation  
  
Je vous laisse réfléchir à quelle rotation geler pour que le joueur ne soit pas bloqué.

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.  
  
On va maintenant faire en sorte que la souris suive le joueur.

Pour cela on va récupérer l’abscisse de la souris et l’utiliser pour créer un vecteur 3D et le multiplier à la vitesse de rotation et au temp de jeu

Une image contenant capture d’écran, texte, Police, ligne

Description générée automatiquement

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

Vous avez 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.

Une image contenant capture d’écran, texte, Police, ligne

Description générée automatiquementOn 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”

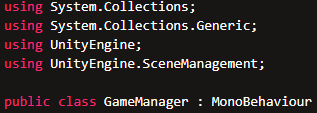
Il vous reste plus qu’à rajouter une condition pour que le joueur puisse sauter seulement s'il touche le sol !

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.

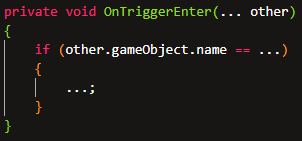
Il nous faut bien une fin !

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 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 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.

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, ou encore créer un fond de la map dans lequel le joueur ne doit pas tomber par exemple

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 Scene Manager, un système d’éléments hostiles, même des plateformes mouvantes ou sinon je te laisse imaginer des choses à rajouter à ton jeu !