

Unity Journey

Quoi de plus simple pour commencer que de remake le jeu qui a redéfini ce que pouvait être un jeu vidéo.

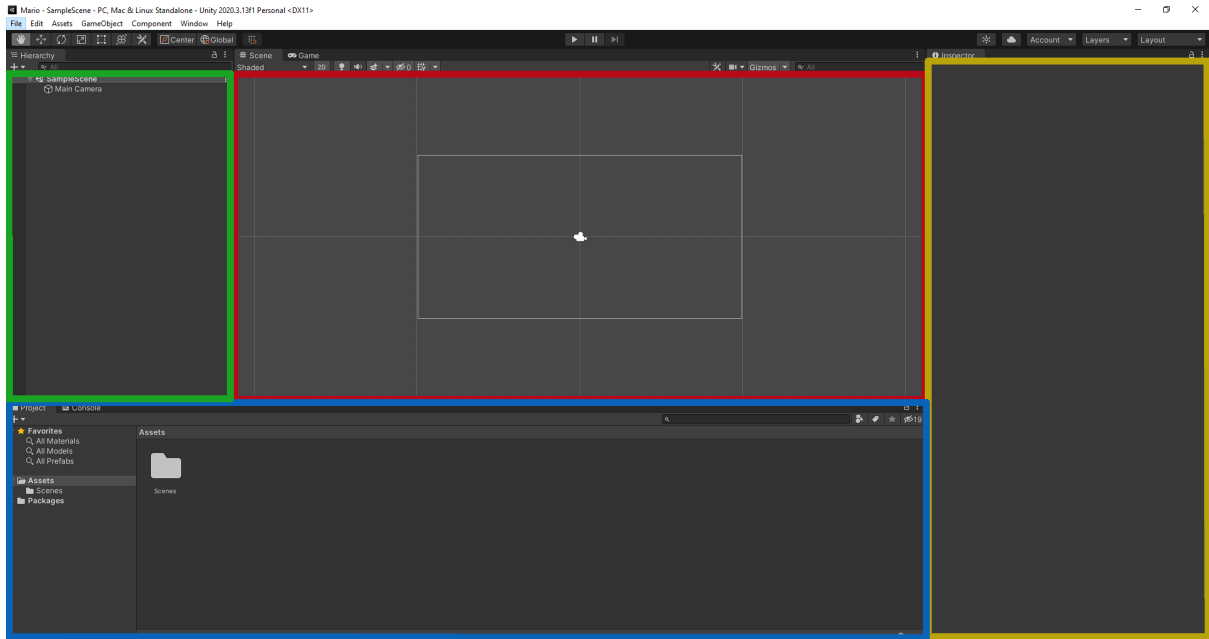
Pour ce faire, nous allons utiliser un des outils phare dans la création de jeux vidéo: Unity.



1. Lancer le projet

A partir de Unity Hub (le launcher) , créez un nouveau projet avec le template 2D

Une fois le projet ouvert, vous devriez avoir cette fenêtre devant vous:



La scène en Rouge:

Une vue directe sur le contenu de votre scène;
C'est ici que vous organisez vos niveaux par exemple.

L'inspecteur en Jaune:

Expose tous les composants de l'objet sélectionné;
C'est ici que l'on ajoutera des composants.

La hiérarchie en Vert:

Elle contient une liste de tous les objets contenus dans votre scène;

L'asset explorer en Bleu:

Il vous permet d'accéder à tous les fichiers de votre projet;
C'est d'ici que l'on importera nos sprites dans la scène

2. Créer les objets

Comme on peut le voir dans l'inspecteur, notre scène se compose seulement d'une caméra.

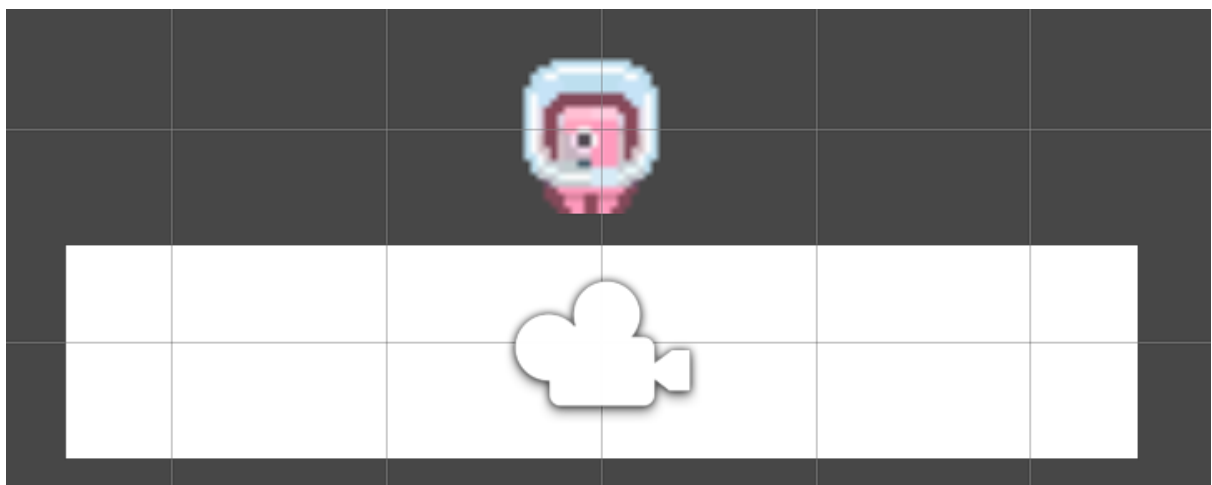
Nous allons donc ajouter notre premier objet: *Un Carré*
Clique-droit dans la hiérarchie->2D Object->Square

Vous verrez donc apparaître un carré dans votre scène, il servira de plateforme pour notre joueur.

Pour ce faire il faut télécharger le dossier "Ressources" situé dans le repo et le déplacer vers le dossier "Assets" du projet
Clique-droit dans l'explorer du projet->Show in Explorer

Une fois que vous avez les sprites dans votre projet, vous pouvez faire glisser le personnage de votre choix depuis le dossier "Characters" jusque dans votre scène

Votre personnage étant maintenant dans la scène, vous pouvez ajuster sa taille et sa position dans l'onglet inspecteur ("scale" pour la taille) de telle façon à ce qu'il soit par dessus un carré allongé.



3.Physique et collision

Nous avons maintenant peuplé notre scène, mais les objets n'interagissent pas encore entre eux.

Nous allons donc commencer par faire de notre personnage un objet physique.

Pour cela, sélectionnez le, puis dans l'inspecteur ajoutez lui le composant Rigidbody2D

Add Component->Rigidbody2D

Vous verrez que si vous lancez la scène, notre personnage est maintenant affecté par la gravité

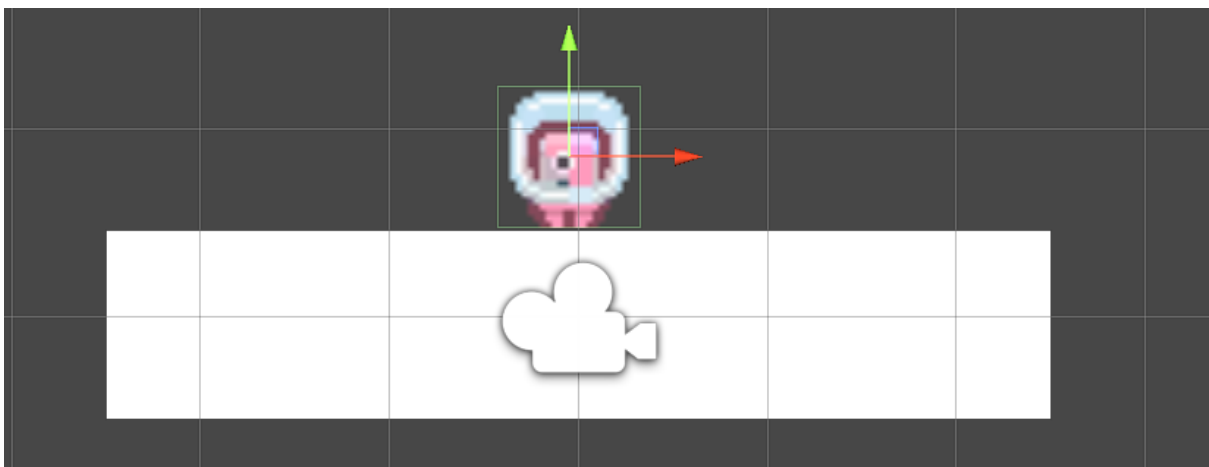
Icon Play au dessus de la scène

On remarquera également qu'il est passé à travers du carré, ceci est dû au fait que notre joueur et notre plateforme n'ont pas de collider pour déterminer leur collision, autrement appelée "hitbox".

Pour y remédier, on ajoutera aux deux le composant "Box Collider2D"

Add Component->Box Collider2D

Si vous lancez la scène à présent, le joueur devrait s'arrêter sur la plateforme.



4. Scripts

Après tout ce travail de mise en place, nous pouvons enfin rentrer dans la partie la plus intéressante : Le Code.

Pour commencer, nous allons ajouter un composant script au joueur.

Add Component->New Script->Movement

puis

Clique-droit sur le composant->Edit Script

ce qui devrait nous amener sur notre éditeur de code.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Movement : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        //
    }

    // Update is called once per frame
    // Message Unity | - references
    void Update()
    {
        //
    }
}
```

Unity utilise le C#, langage similaire au C++ syntaxiquement.

On peut voir dans l'éditeur que notre script correspond à une classe, et qu'il contient deux fonctions.

La fonction Start est appelée par Unity au lancement du jeu.

La fonction Update est appelée par Unity à chaque frame du jeu.

Exercice:

Utilisez ***Debug.Log("text");*** pour log "start" au lancement du jeu, et "update" chaque frame (vous verrez ce texte dans l'onglet console)

5.Inputs

Unity nous expose la classe Input pour nous donner des booléens correspondant au statut des différentes touches du clavier.

Par exemple, `"Input.GetKeyDown(KeyCode.Z)"` sera seulement vrai la frame où la touche z est enfoncée.

`("Input.GetKey(KeyCode.Z)"` fais de même, mais reste vraie tant que la touche est enfoncée).

Exercice:

Loggez "tap" quand on appuie sur la barre espace et "hold" quand on la laisse appuyée

6.Movement

Pour faire bouger notre personnage, il faut accéder à son composant Rigidbody2D et les méthodes associées.

Pour cela, on utilise `"GetComponent<TypeDuComposant>()"` qui nous renvoie l'instance du Rigidbody2D de notre personnage.

Rigidbody2D nous donne accès à plusieurs méthodes pour changer la vitesse de notre objet.

<https://docs.unity3d.com/ScriptReference/Rigidbody2D.html>

Par exemple:

`"Rigidbody2D.addForce(new Vector3(0, 1, 0))"` appliquera une force dans l'axe y.

`"Rigidbody2D.velocity = new Vector3(0, 0, 0)"` remettra la vitesse à 0 de l'objet à 0

Exercice:

- 1.Faites en sorte que le personnage saute à chaque fois que l'on appuie sur Espace.
- 2.Faites en sorte que l'on puisse se diriger vers la droite et la gauche avec Q et D.

7. La suite ?

Si tu es arrivé aussi loin, c'est que tu comprends les bases de Unity.

Tu dois sûrement avoir plein d'idées et les possibilités sont infinies

Ajoutes des ennemis, des animations, des collectibles, etc...

Unity a une documentation en ligne complète et une infinité de tutoriels pour t'accompagner.

