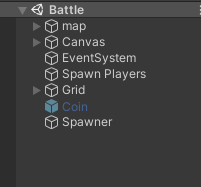
Documentation Technique

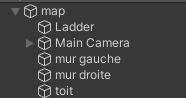
# Installation d'unity et des outils nécessaires

Pour ce projet on a utilisé unity. Pour utiliser unity il vous faudra unity hub ([ici](https://public-cdn.cloud.unity3d.com/hub/prod/UnityHubSetup.exe?_ga=2.92154458.1560024421.1634734962-664918836.1619425748)) et il vous faudra ensuite la version 2020.3.17f1 d'unity ([ici](unityhub://2021.1.17f1/03b40fe07a36)). Il vous faudra aussi installer un IDE C# (nous avons utilisé Visual Studio 2019 que vous pouvez trouvé [ici](https://visualstudio.microsoft.com/fr/vs/). Si vous utilisez cet IDE il vous faudra installez les outils de développement de jeux avec Unity si vous voulez avoir l'auto complétion).

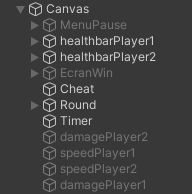
# Rangement de la hiérarchie

Pour le rangement dans la hiérarchie de notre scène principale (Battle.unity).



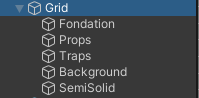


Nous avons la première partie "map" qui est composée des murs invisibles sur les côtés de la carte, de la camera, et de l'échelle.



Nous avons ensuite la partie suivante "Canvas" qui est composée de tout ce qui touche à l'interface utilisateur. Tel que: les barres de vie, le menu pause, le score, le timer…

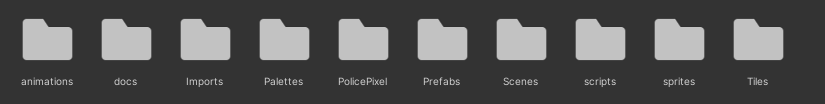
Il y a ensuite "EventSystem" qui gère tout plein d'évènement du jeu, nous n'y avons pas touché a par pour mettre la musique, et il y a "SpawnPlayers" qui est un GameObject qui est là que pour supporter le script du spawn de joueurs, il n'a aucun impact sur le jeu de façon visuelle.



Nous avons ensuite la partie "Grid" qui a permis de faire la map visuelle. C'est le GameObject qui gère la tilepalette qui est une manière de créé des maps. Pour ce faire il a fallu découper une image contenant chaque texture sous forme de carré, puis importer chaque texture - donc morceau de l'image initiale – dans la "Tile Palette" ce qui permet ensuite de les insérer sur la scène de jeu.  
En ce qui concerne les texture il est aussi possible de les superposer les unes sur les autres pour faire par exemple apparaitre l'échelle par-dessus le background, pour ce faire il faut faire plusieurs Tilemap et modifier leur "Order in Layer" : l'image avec un order à 4 apparaitra devant une à 0. Ensuite chaque partie de "Grid" est une partie de la map.

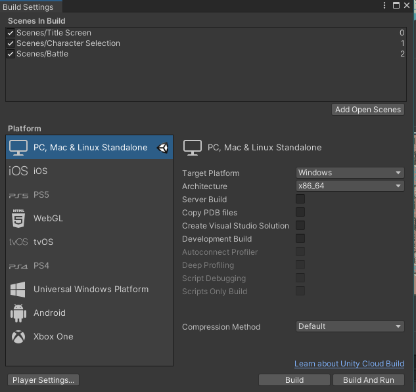
Il y a pour finir "Coin" et "Spawner" qui vont gérer la pièce de power-up.

# Rangement des dossiers



* Il y a le dossier "animations" qui contient toutes les animations, tant celles des joueurs que de la pièce que du texte.
* Il y a le dossier "docs", que vous avez du trouvé si vous lisez ça, car il contient toutes les documentations.
* Ensuite le dossier "Imports" qui contient toutes les ressources importées.
* Ensuite le dossier "Palettes" qui contient la TilePalette. Ensuite le Dossier "PolicePixel" qui contient une police modifiée.
* Ensuite le dossier "Prefabs" qui contient tous les prefabs du projet. Il y a la première section qui contient le prefab de la sélection des personnages et un dossier "Ressources" qui contient les prefab des joueurs disponibles dans le jeu. Et pour finir ce dossier, un sous-dossier "Choices" qui contient l'image pour le choix des personnages et d'affichage dans le jeu.
* Le dossier suivant est le dossier "Scene" qui contient toutes les scènes.
* Le dossier "scripts" quant à lui contient tous les scripts utilisé dans le projet.
* Le dossier "sprites" contient toutes les images permettant les animations et autres utilisations dans le jeu.
* Et pour finir le dossier "Tiles" qui contient toutes les parties de la Tile Palette.

# Créé un build utilisable

Pour créer un nouveau build il faut aller sous File -> Build Settings… qui vous ouvrira une fenêtre qui permet de gérer les paramètres du build.

Les scenes

Puis créé le build

La plateforme