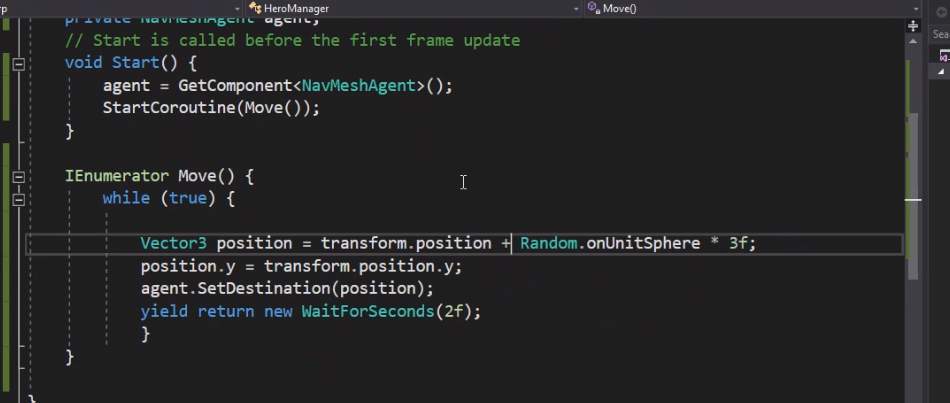
Serialized Sauvegarde avec dictionnaires

Voir les scripts car il y a quelques erreurs dans mes screenshots !

# NavMesh Random

Après la boucle retourne au gestionnaire de Coroutine.

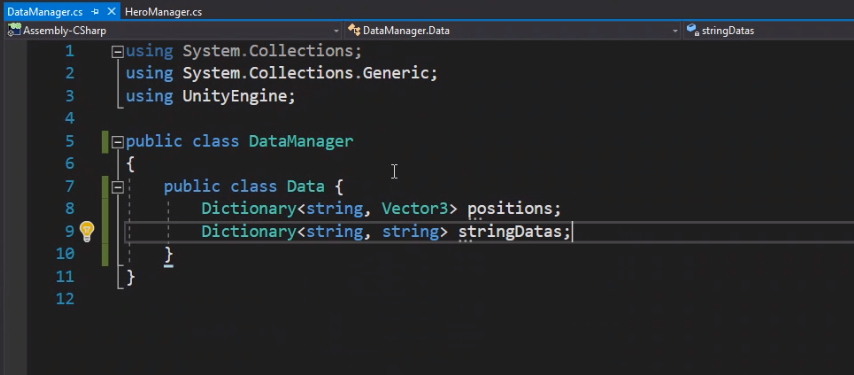


Coroutine lui demande de faire la fonction Move()

Donc il y a 2 secondes entre chaque boucle while (qui est randomisé d’ailleurs).

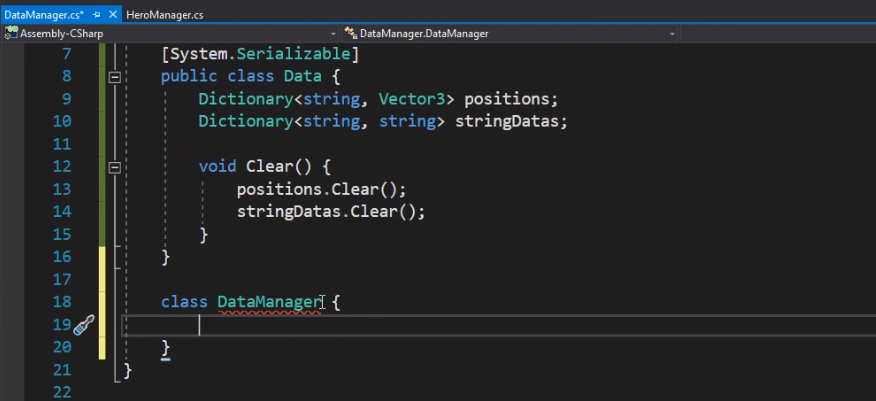
# DataManager

Création de dictionnaire de Data. Autant de dictionnaires de String, int, bool… qu’il nous faut.

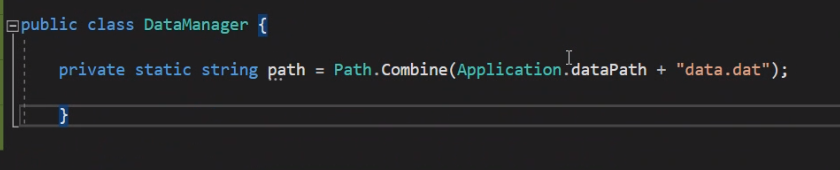


Oublier dans le screenshot :

Mettre les dicos en public !!

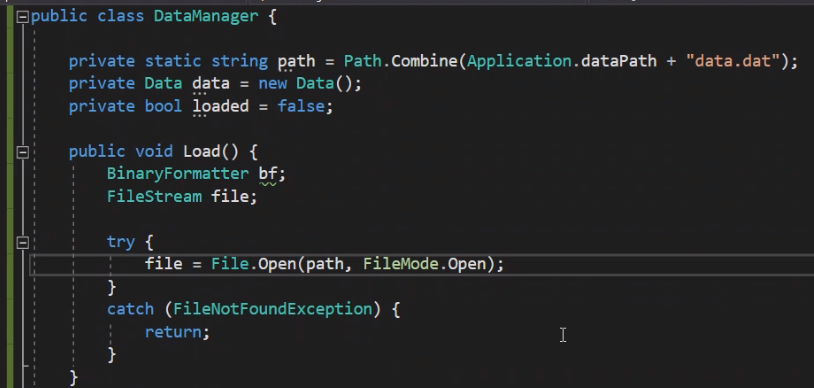
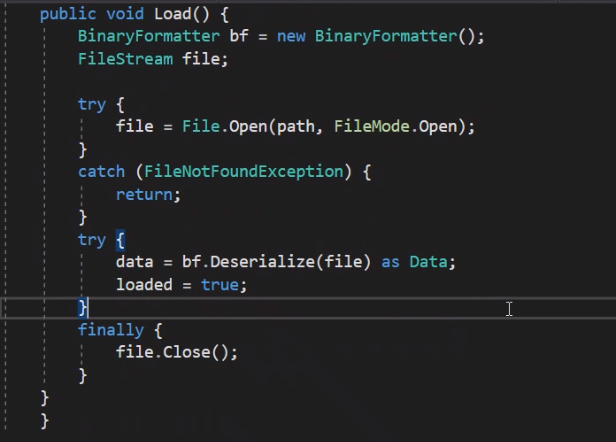


* Une petite fonction Clear() pour reset les dico si nécessaire.
* Nous créons une nouvelle classe DataManager.



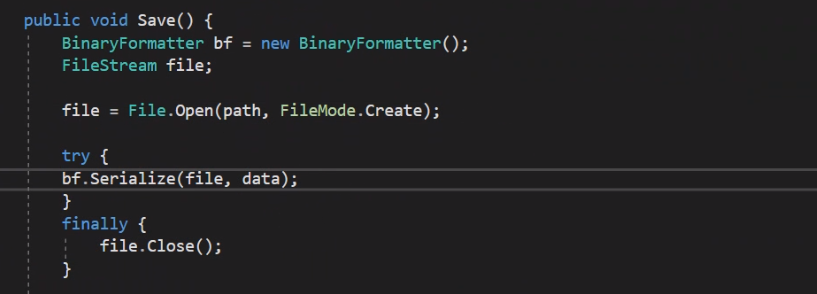
Chemin vers les data.   
using UnityEngine.IO ;

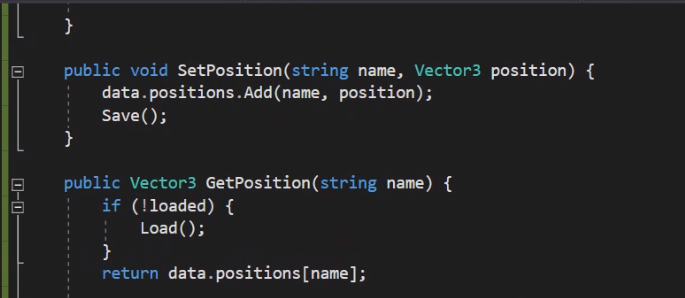
On fait ensuite une fonction Load() et une fonction Save()



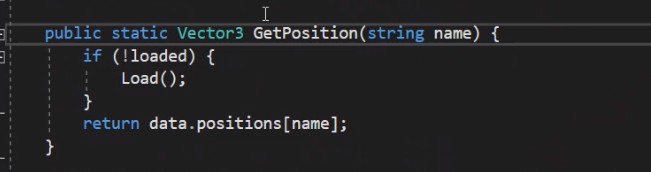
* On crée le fichier data.dat
* On désigne le lieux du fichier
* On return s’il n’y a pas de fichiers à Load.
* On prend les fichiers si y a des fichiers.
* On met le bool ‘loaded’ à true pour checker (pas obligatoire juste par pratique)
* On ferme le fichier de data (pas oublier sinon bug !!)

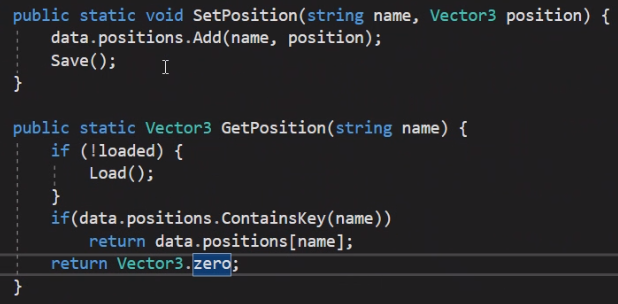
On crée la fonction Save() qui fonctionne de la même manière.

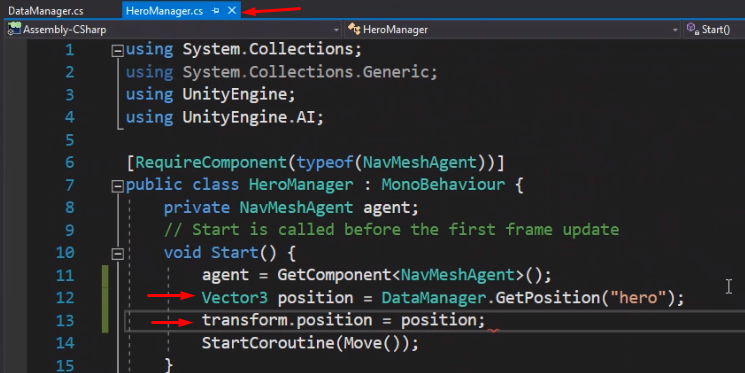




**METTRE TOUTES LES FONCTIONS EN PUBLIC STATIC SINON CA MARCHE PAS !!! Car sinon on ne peut pas l’utiliser dans une méthode qui est static.**







A chaque fois qu’il a fini un cycle, il save sa position :

### Rappel : Les bibliothèques dans C#

