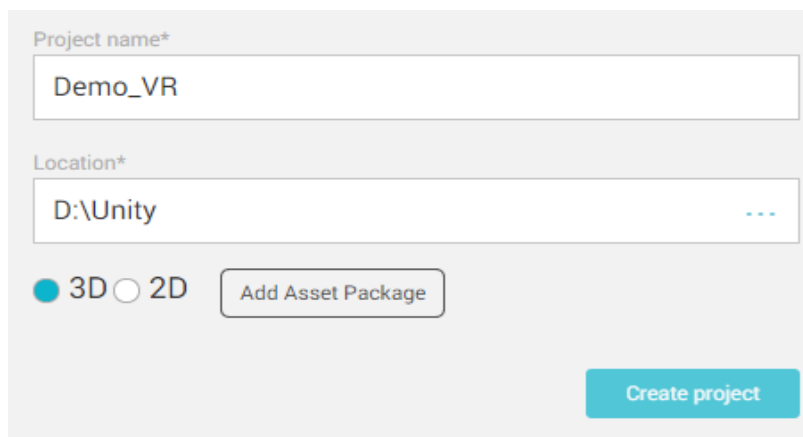


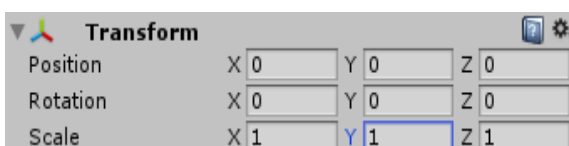
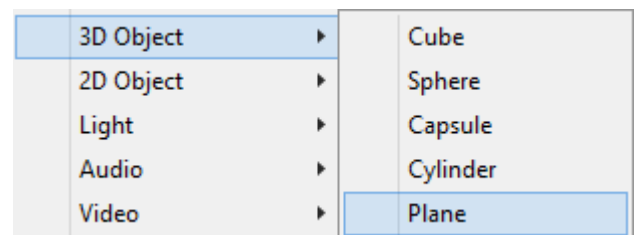
Nous avons dans les fiches précédentes comment réaliser notre premier jeu sur Windows et sur Android. Nous allons à présent voir comment réaliser notre premier jeu sous Windows avec un déplacement à la première personne (FPV : First Person View) où vous verrez au travers des yeux du personnage. Nous modifierons ensuite notre jeu pour l'exporter sur Android avant de le transformer en un jeu en réalité virtuelle (VR).

Commençons par créer un nouveau projet :



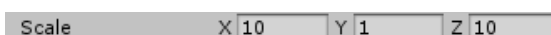
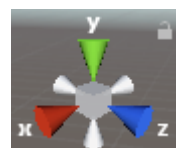
Cliquez sur *Create project* et laissez Unity charger votre environnement de travail.

Comme toujours, commencez par réaliser un clic droit dans la « *Hierarchy* » et créez un support « *Plane* » à votre jeu.



Vous allez le régler à une taille « humaine » pour pouvoir y évoluer normalement par la suite. Pour cela cliquez sur le *Plane* dans votre « *Hierarchy* » puis observez l'*Inspector* pour y trouver le « *scale* » (échelle) dans la partie *Transform*.

D'après la représentation des plans situés en haut à gauche de la scène, vous interviendrez sur les axes x et z.



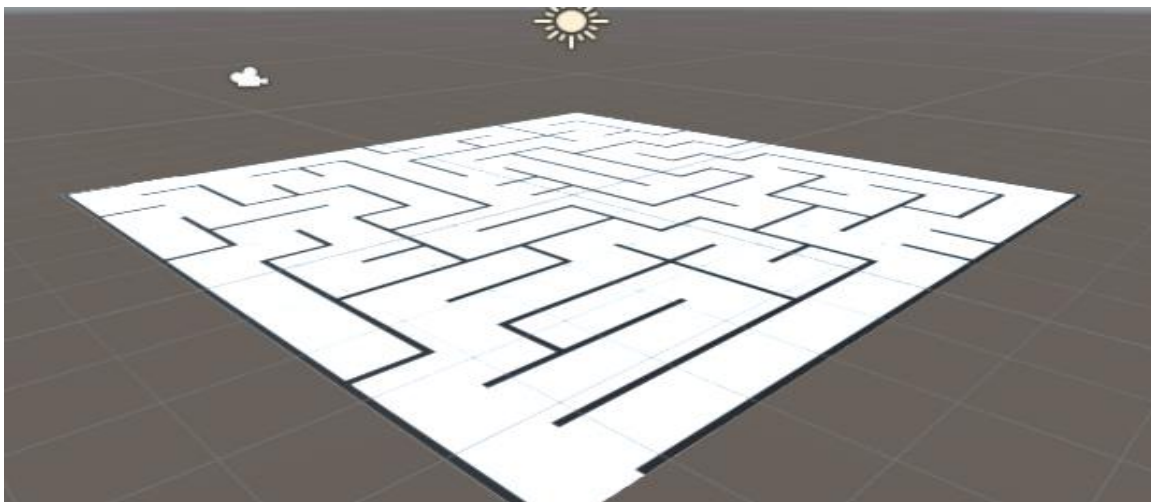
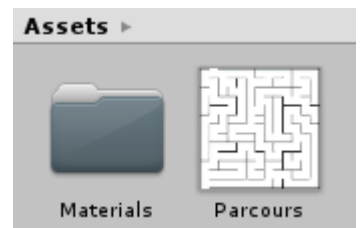
Réglez alors l'échelle à 10 pour que la largeur et la longueur soient multipliées d'autant.



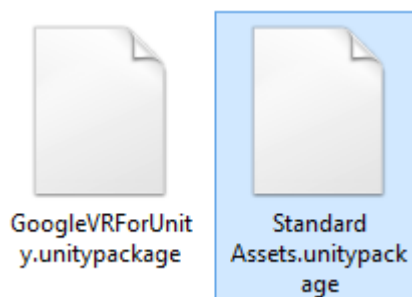
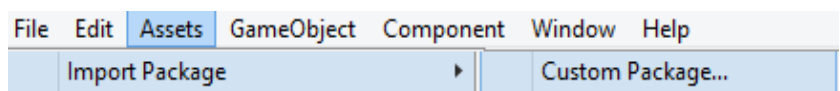
Prendre le fichier « *Parcours.jpg* » présent dans le dossier *Demo\_VR* et faites le glisser dans la fenêtre « *Assets* » normalement présente en bas de votre écran. L'image apparaît alors automatiquement.

Faites la glisser sur votre plane pour alors voir apparaître un nouveau dossier « *Matérials* » dans la fenêtre « *Assets* » .

Vous verrez aussi votre plan devenir un vrai labyrinthe en 2D, il nous servira de modèle pour monter les murs de notre labyrinthe :

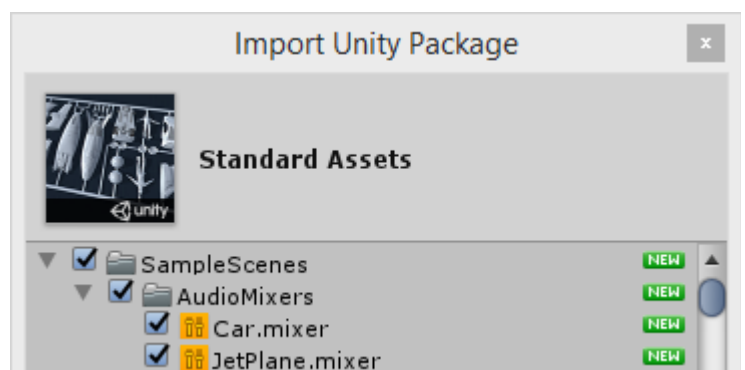


Rendez-vous dans l'onglet « *Assets* », choisissez « *Import Package* » puis « *Custom Package* »

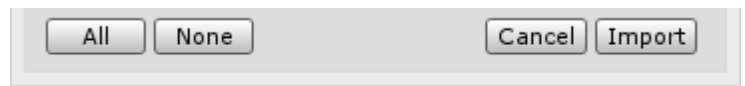


Une nouvelle fenêtre s'ouvre et rendez-vous dans le dossier du jeu contenant le fichier : « *Standard Assets.unitypackage* »

Laissez Unity décompresser son contenu jusqu'à voir apparaître :

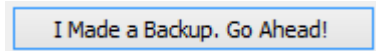


Cliquez alors sur « *Import* » pour tout importer dans votre projet.



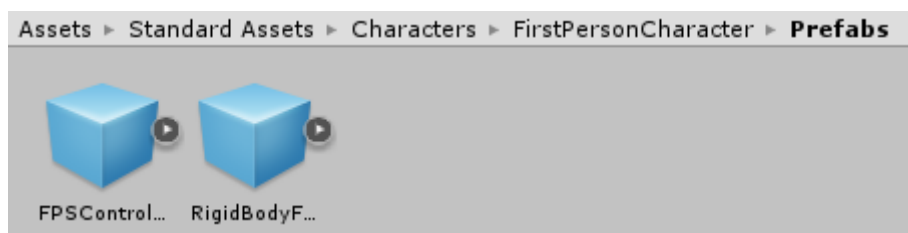
Attendez à nouveau que tout soit copié dans votre projet.

Unity peut alors vous demander une petite mise à jour des fichiers importés, pour cela cliquez sur le bouton ci-contre :



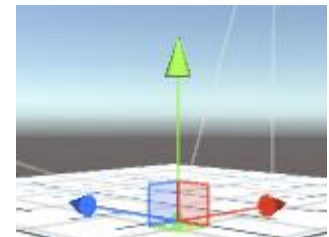
De nouveaux dossiers apparaissent encore dans votre fenêtre projet.

Suivez le chemin ci-dessous afin de trouver les « *Prefabs* » :



Prenez le « *RigidBodyFPSController* » et faites le glisser dans votre « *Hierarchy* ». Unity place automatiquement votre *Préfab* à l'altitude 0 de votre scène, autrement dit si vous lancez le jeu, votre personnage tombera et vous ne pourrez pas jouer.

Pour éviter cela, cliquez juste sur votre *Préfab* et en restant cliqué sur la flèche verte (axe y) vous pourrez le faire monter ou descendre pour que son centre de gravité (origine du repère ici) soit au-dessus du plan.



De plus, ce *Préfab* contient déjà une caméra liée à votre mouvement, il vous faudra alors supprimer la *MainCamera* déjà présente dans votre liste en cliquant simplement dessus, appuyer sur la touche Suppr de votre clavier et valider sa suppression à l'écran.

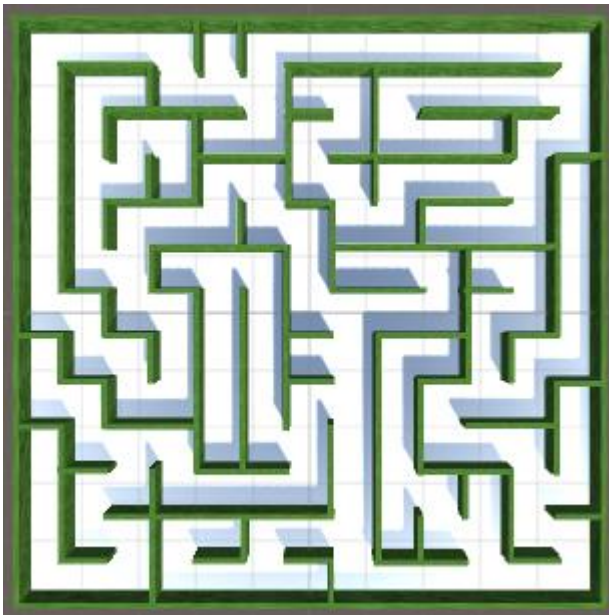


Pour tester si vos commandes fonctionnent, vous pouvez appuyer sur le signe « *Play* » en haut de votre écran, votre scène passe automatiquement en mode « *Game* » et vous voilà dans votre jeu.



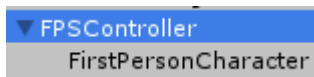
Pour en ressortir, appuyer sur la touche Echap puis cliquez à nouveau sur Play pour arrêter la simulation.

**Attention,** toutes les modifications que vous apporterez pendant que le bouton Play est enfoncé ne seront pas prises en compte.



Maintenant il est temps, de créer les murs, pour cela, vous créerez des cubes les uns après les autres, vous pourrez y appliquer une texture pour les rendre plus réalistes. Vous aurez à créer autant de cubes que de murs en prenant les dessins au sol comme référence.

Le parcours est terminé, vous pouvez déplacer votre « personnage » et vous amuser à l'intérieur du labyrinthe. Vous pouvez l'exporter sur Windows pour le tester et le faire tester à vos camarades. Nous allons à présent essayer de transformer notre jeu en réalité virtuelle. Vous allez donc modifier le jeu pour qu'il soit multiplateforme, c'est-à-dire exportable à la fois sur Windows et Android.



Dans la « *Hierarchy* » cliquez sur *RigidBodyFPSController* ou *FPSController* en fonction du choix réalisé précédemment. Vous trouverez alors dans l'*Inspector* le détail du *Préfab*.



Vous chercherez le script *FirstPersonController* que vous allez modifier en double-cliquant sur la zone grisée.

Une interface d'édition de code s'ouvre, il s'agit en général soit de MonoDevelop soit de VisualStudio. L'objectif de cette modification est de faire tourner notre personnage dans le sens de la caméra afin de faciliter son déplacement dans le jeu.

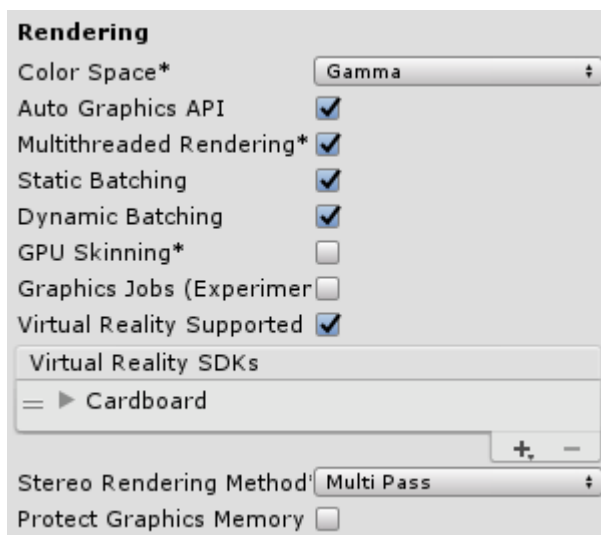
Pour cela vous chercherez la fonction : `private void FixedUpdate()` qui est la fonction mettant à jour le jeu à chaque frame (image). Vous allez modifier le mouvement désiré en celui de la caméra, pour cela cherchez à présent le vecteur « ***desiredMove*** » :

```
Vector3 desiredMove = transform.forward*m_Input.y + transform.right*m_Input.x;
```

Que vous modifierez en rajoutant la transformation de la caméra :

```
Vector3 desiredMove = Camera.main.transform.forward*m_Input.y + Camera.main.transform.right*m_Input.x;
```

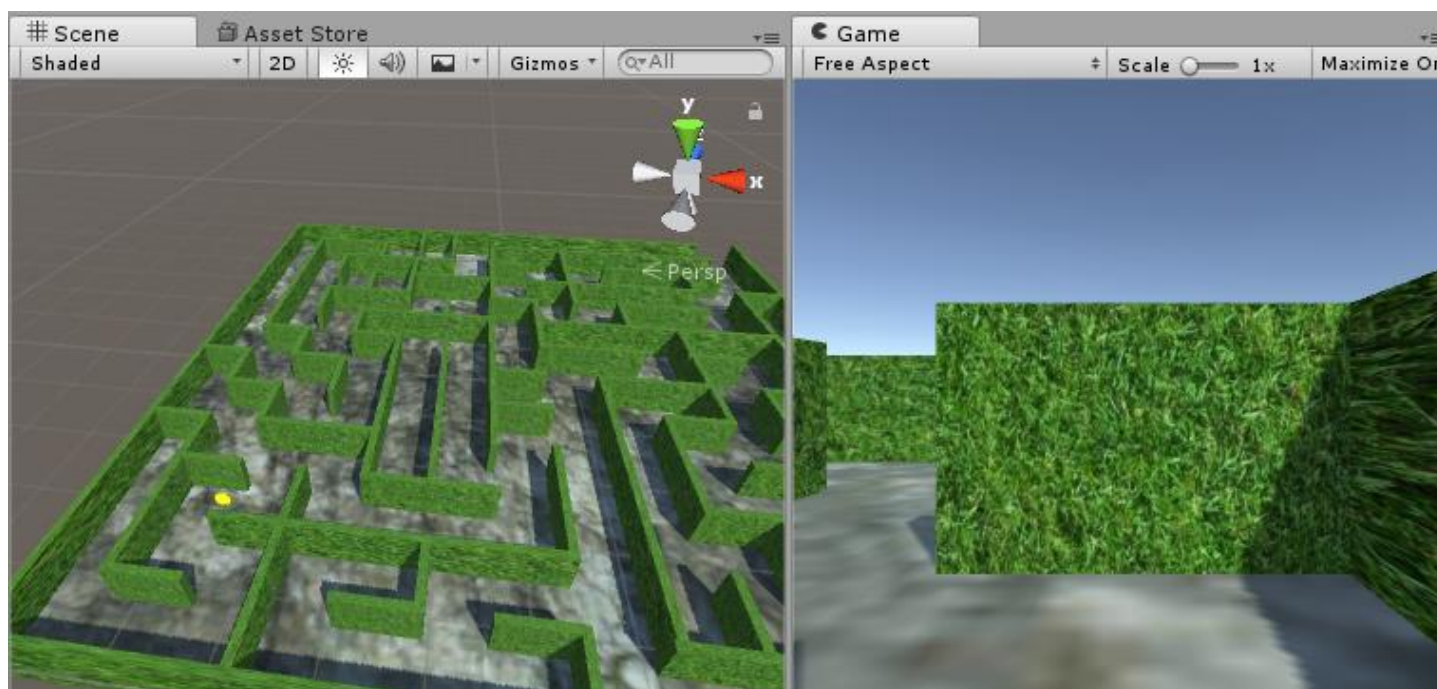
Vous pouvez à présent enregistrer et quitter l'éditeur de code pour aller exporter votre jeu. Pour cela comme expliqué dans la fiche précédente, rendez-vous dans **File** puis **Build Settings**.



Puis choisissez Android et cliquez sur Player Settings. Dans l'Inspector, vous trouverez la section **Rendering** où vous cochez **Multithreaded Rendering** pour améliorer l'affichage graphique sur votre périphérique Android. Pour ce qui concerne la VR, vous cliquerez sur **Virtual Reality Supported** puis en appuyant sur le **+** situé à la ligne inférieure vous choisirez **CardBoard**.

Voilà votre premier jeu FPV pour Android est prêt, il ne vous reste plus qu'à le compiler pour l'envoyer sur votre téléphone.

**Attention**, si votre Unity venait à se quitter seul pendant la compilation, il peut être nécessaire de cliquer sur « Switch Platform » d'attendre quelques minutes et une fois que Unity a fini de se mettre à jour pour Android, vous pouvez lancer la compilation en cliquant sur Build.



*Dans une prochaine fiche, nous pourrons définir un scénario de jeu : ramasser des pièces, compter le score, mesurer le temps nécessaire pour trouver la sortie du labyrinthe, ajouter des ennemis...*