

## Tuto Nav Mesh Surface

### I - High Level NavMesh Building Components

Tout ce qui tourne autour de l'intelligence artificiel et notamment du pathfinding est très limité sur Unity notamment avec les NavMeshs qui sont unique à une scène.

Récemment un développeur d'Unity (Mike Geig) à sortie une version modifié du NavMesh Components actuel (voir lien GitHub).

Grâce à cette nouveauté il est possible désormais d'avoir plusieurs NavMesh qui seront spécifique à des objets.

Ce qui permet d'aller plus loin dans le développement de l'IA.

Dans ce tutoriel je vais essayer de vous présenter les différentes nouveauté et possibilité à travers des exemples.

[Documentation Officiel](#)

[Vidéo Officiel](#)

### II - Installer le Component

Requiert : Unity 5.6 minimum\_

[Lien GitHub Officiel](#)

[Lien GitHub de la version utilisé pour ce tutoriel](#)

Une fois télécharger, rendez vous dans le dossier Assets du fichier téléchargé:

..\NavMeshComponents\Assets\

Copier le dossier nommé "NavMeshComponents" (Dans Assets) et coller le dans le dossier Assets de votre Projet.

Puis relancer Unity.

### III - Exemple 1 Voiture/Piéton

Dans cet exemple nous allons créer deux zones distinctes une réservées aux voitures et l'autre aux piétons.

Objectif de l'exemple : Créer deux NavMeshs sur une seule scène

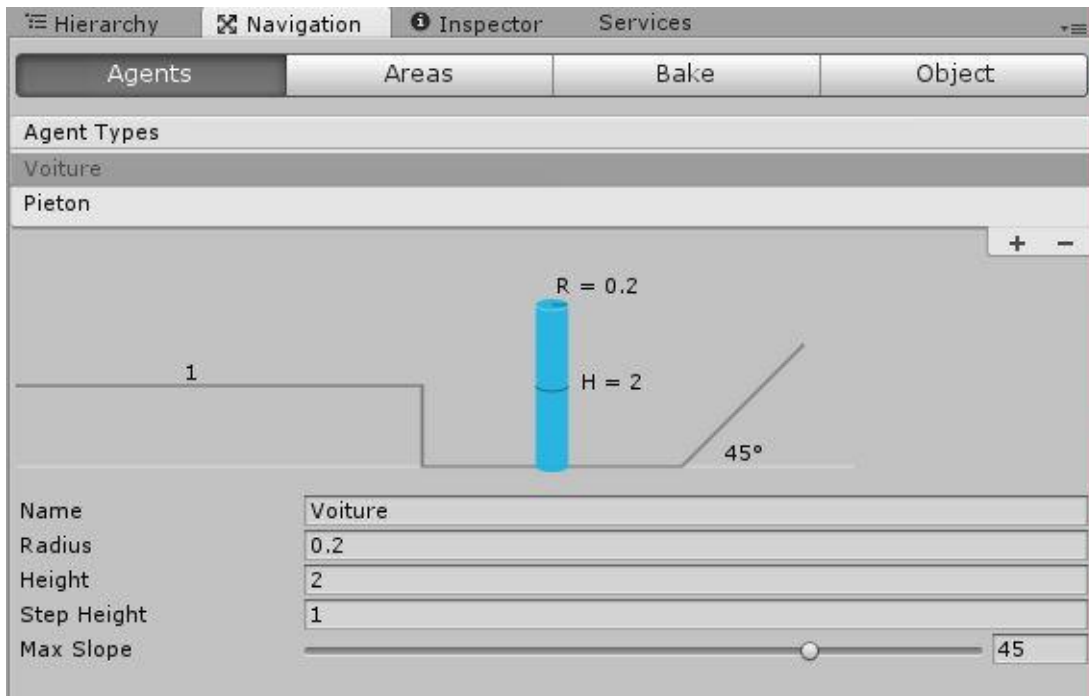
#### Étape n°1 - Créer les Agents :

Les *Agents* vont vous permettre de choisir qui peut faire certaines actions sur des zones prédéfinies.

Allez dans la fenêtre *Navigation* puis dans l'onglet *Agents*

Vous devez créer deux *Agent Types* (une voiture et une piéton)

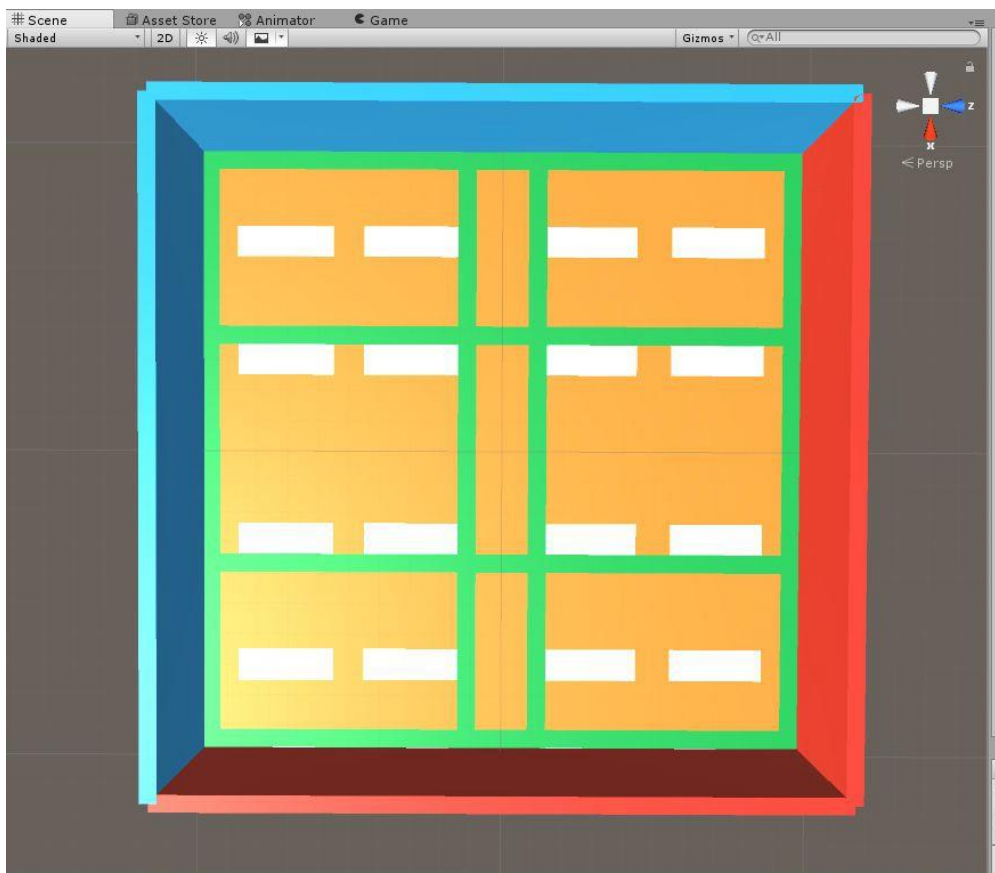
Pour les valeurs libre à vous de modifier selon votre scène.



### Étape n°2 - Dessiner les Zones respectives :

Dans cet exemple j'ai décidé de donner une couleur à mes zones afin de pouvoir les différencier facilement.

Pour ce faire j'ai placé une série de "Plane" qui vont délimiter cette zone.



En vert la zone pour les piétons et en orange la zone pour les voitures, les rectangles blancs étant des "obstacles", qui ici pourrait représenter des bâtiments.

### Étape n°3 - Créer les Zones respectives :

Dans l'étape 2 nous avons matérialisé nos zones, il faut maintenant les spécifier à Unity.

Pour cela allez dans la fenêtre *Navigation* puis dans l'onglet *Areas* :



	Name	Cost
Built-in 0	Walkable	1
Built-in 1	Not Walkable	1
Built-in 2	Jump	2
User 3	Pieton	1
User 4	Voiture	1
User 5		1
User 6		1

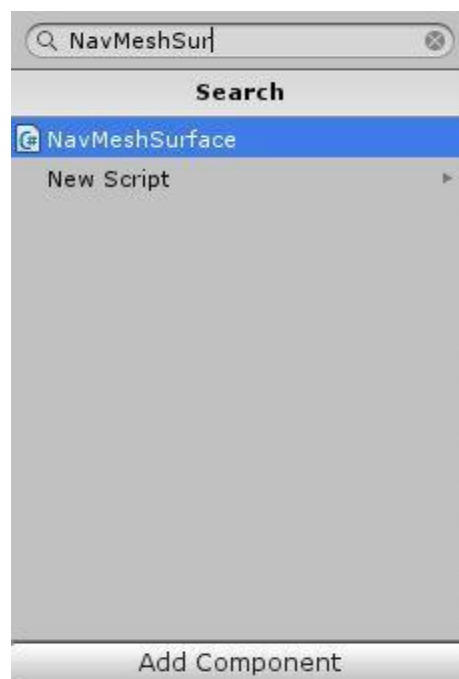
Les *Built-in* sont les valeurs par défaut et inchangeable (seul le coût est modifiable).

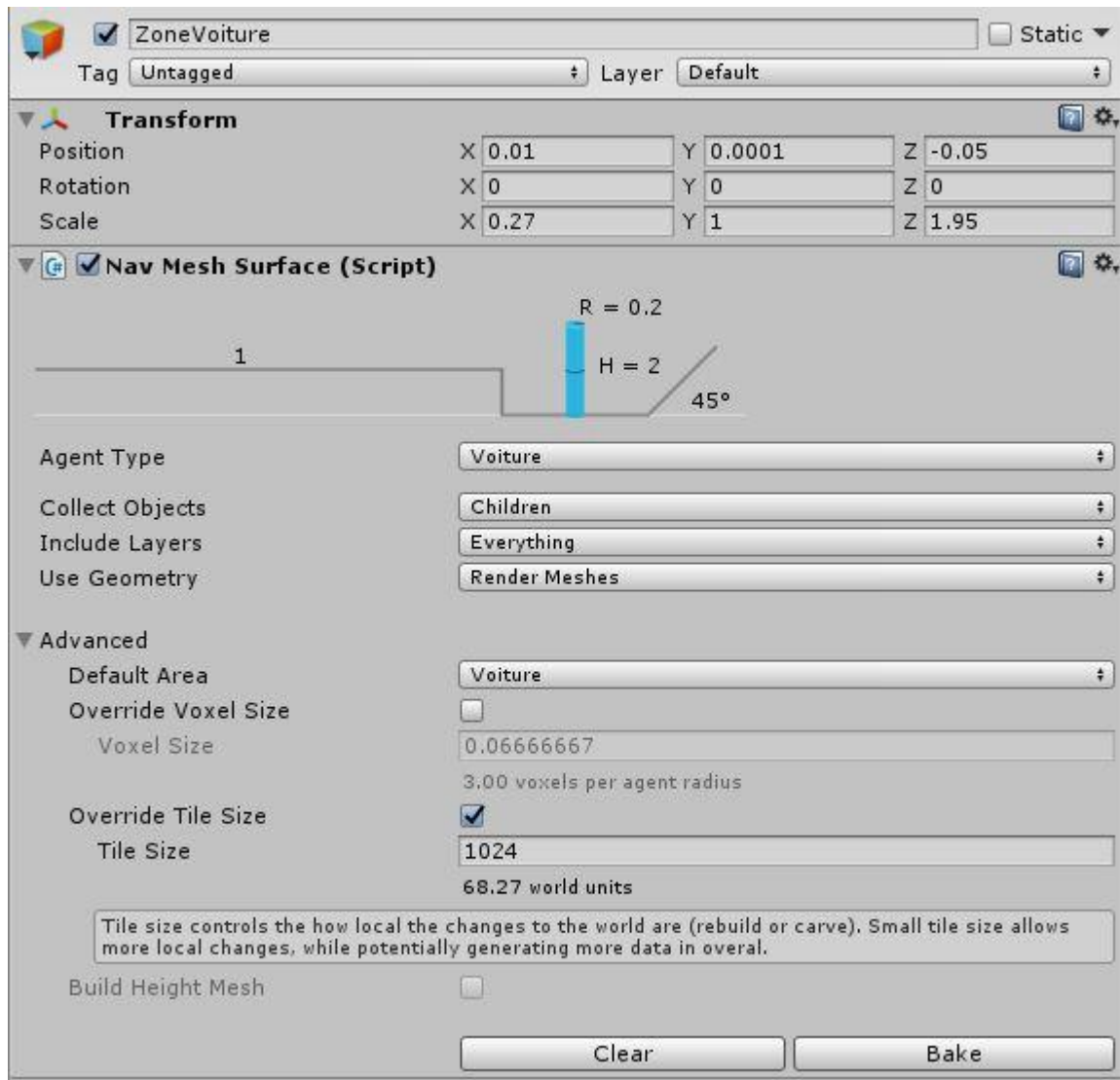
Ajoutez les zones piétons et voitures dans le champs *Name*.

Le Coût (*Cost*) n'est pas utile ici mais il sert à influencer le pathfinding.

### Étape n°4 - Ajouter le Component :

Maintenant que nous avons créer et dessiner les zones il faut les liers en ajoutant le Script "*NavMeshSurface*" :





Ajoutez le Script sur les deux zones.

Ensuite pour chaque zones choisissez le *Agent Type* en conséquence.

Pour les autres paramètres je vous met la traduction de la documentation :

*Parameters*

- **Agent Type** – Le Type d'Agent que va utiliser ce NavMeshSurface. Le Type d'Agent est utilisé pour les paramètres de génération (Bake), ainsi que pour faire correspondre un NavMeshAgent à une surface appropriée lors du pathfinding.
- **Collect Objects** – définie sur quel objet la génération (Bake) va se baser
  - *All* – utilise tous les objets actif.
  - *Volume* – utilise tous les objets actifs qui dépasse le volume délimité (défini plus loin)
  - *Children* – utilise tous les objets actifs qui sont des enfants du component NavMeshSurface, en plus de l'objet sur lequel le component est placé.

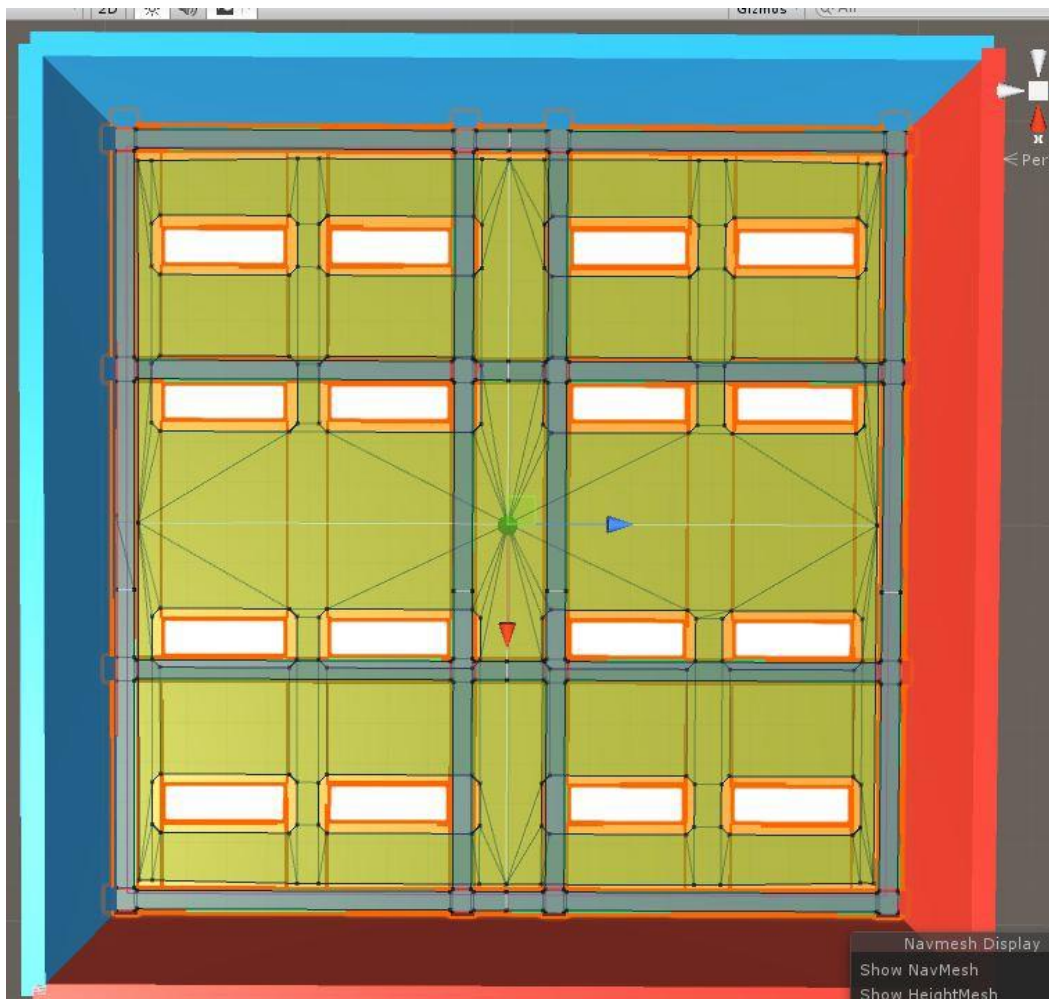
- **Include Layers** – Définit les couches sur lesquelles les objets doivent être inclus dans la génération (Bake). Cela permet d'éliminer davantage les objets d'inclusion dans la génération (Bake), par exemple Des effets ou des personnages animés.
- **Use Geometry** – Sélectionne quelle géométrie est utilisée pour la génération (Bake).
  - *Render Meshes* – utilise la géométrie des rendered meshes et des terrains.
  - *Physics Colliders* – utilise la géométrie des collisions et des terrains. Lors de l'utilisation de la physique, il s'agit habituellement d'une meilleure option que Render Meshes. De cette façon, les agents seront plus proches des limites physiques de l'environnement.

Dans la partie "Advanced" il faut changer "Default Area" et mettre la zone voulue.

Pour finir on clique sur le bouton "Bake" pour générer le NavMesh de la zone :

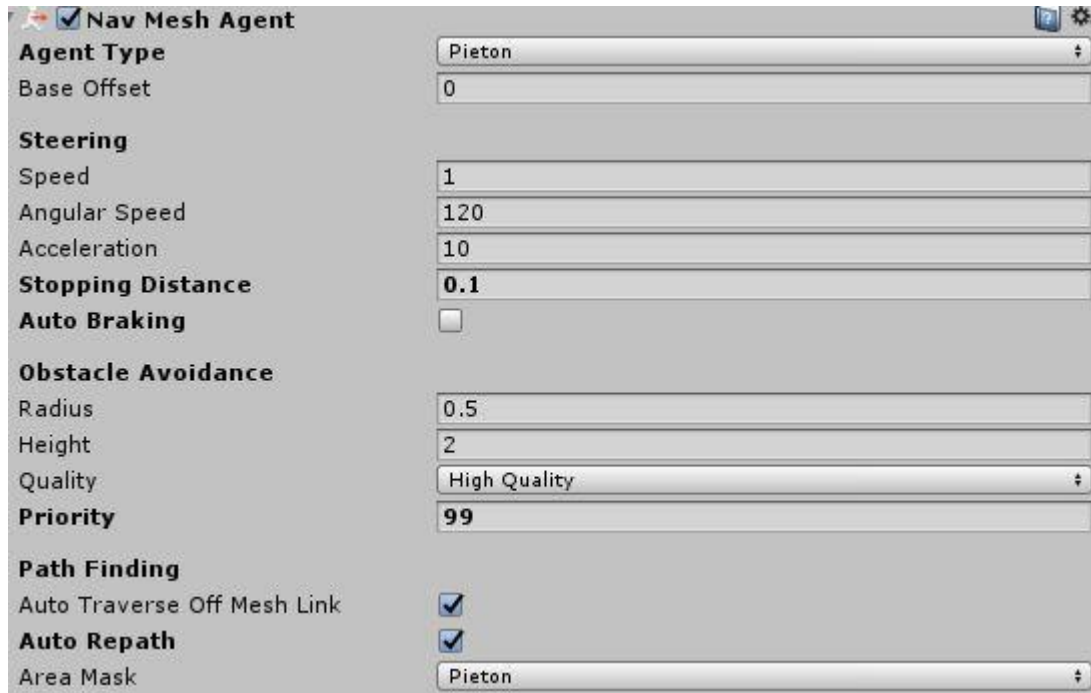


Vous devez normalement obtenir un résultat similaire :



## Étape n°5 - Associer une Zone avec un NavMeshAgent :

Maintenant que les zones sont faites et que les NavMesh sont générés il faut placer les NavMeshAgents sur les IA afin que Unity puisse associer l'IA et sa zone :



Les deux paramètres à modifier pour que les zones soient respectées sont :

- *Agent Type* : Ceux créés ci-dessus
- *Area Mask* : Zone à attribuer à l'*Agent Type*

L'exemple 1 est terminé je vous invite à regarder la vidéo officielle ainsi que la documentation associée en savoir plus sur ce component.

[Documentation Officiel](#)

[Vidéo Officiel](#)

*Tutoriel réalisé par notre stagiaire Pierre VERHAEGHE élève ingénieur,  
Bonne chance à lui pour la suite de ses études et merci pour ce pratique tuto ☺*