

HAI935I - Vision, Réalité virtuelle et augmentée  
TP1

Ingo Diab

2023

## Table des matières

<b>1</b>	<b>Scene &amp; Textures/Matériaux</b>	<b>3</b>
<b>2</b>	<b>Physique</b>	<b>3</b>
<b>3</b>	<b>Prefabs</b>	<b>4</b>
<b>4</b>	<b>Skybox</b>	<b>6</b>
<b>5</b>	<b>Scripts</b>	<b>7</b>
<b>6</b>	<b>Third Person Character</b>	<b>8</b>
<b>7</b>	<b>Dragon</b>	<b>10</b>
<b>8</b>	<b>Musique ambiante</b>	<b>10</b>
<b>9</b>	<b>Création de Catbots autonomes</b>	<b>11</b>

## 1 Scene & Textures/Matériaux

Dans cette partie nous avons appris à poser un cube dans la scène ainsi qu'un plane que nous avons rescale en guise de sol. Nous avons appliqué une texture au material du cube pour lui donner cet aspect de boîte et nous avons appliqué une texture (trouvé sur le workshop unity) sur le plane pour lui donner un aspect de sol prototype. Nous avons aussi créé deux matériaux, un rouge et un bleu, en modifiant la couleur albedo du material.

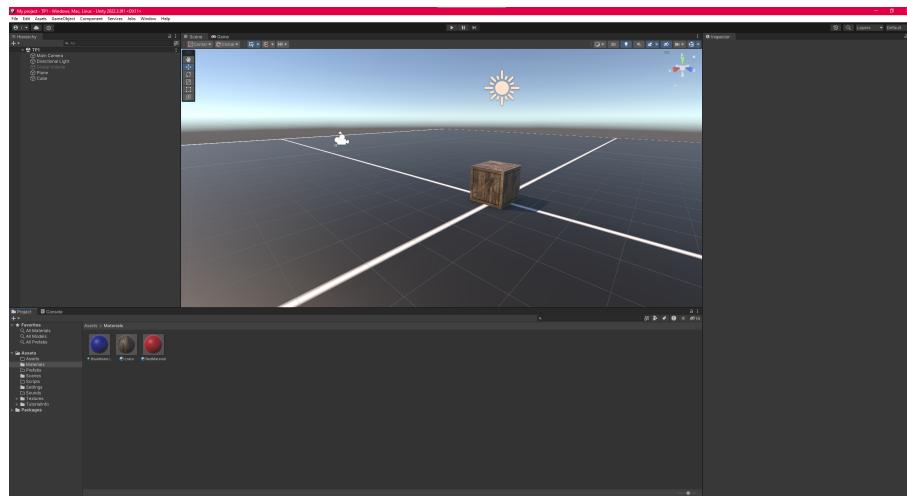


FIGURE 1 – Un cube et un sol

## 2 Physique

Dans cette partie nous avons placer un Rigidbody sur le cube afin qu'il soit affecté par la physique. Lorsque nous lançons le jeu, les cubes sont soumis à la gravité.

### 3 PREFABS

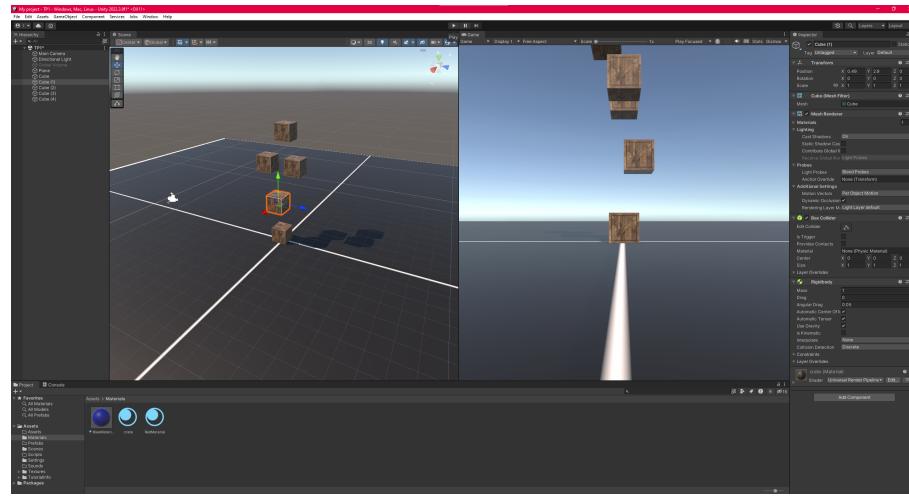


FIGURE 2 – Les cubes avant de lancer le jeu

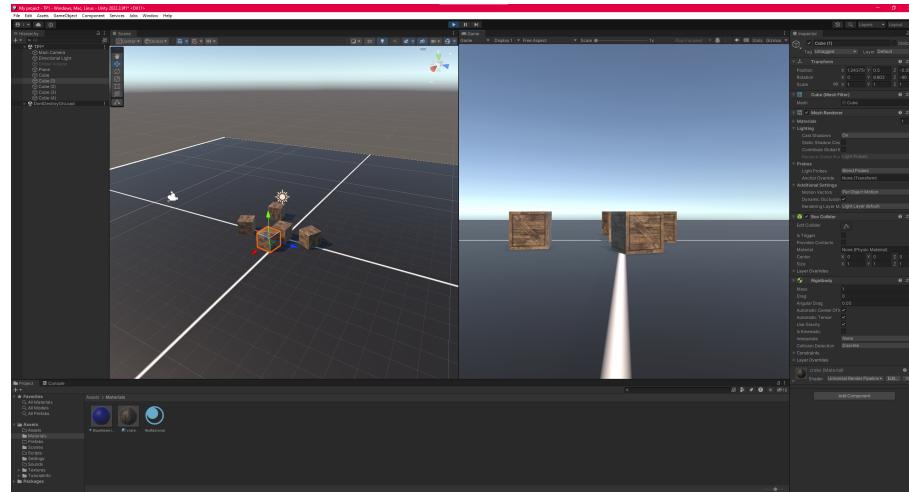


FIGURE 3 – Les cubes après avoir lancé le jeu

## 3 Prefabs

Dans cette partie nous avons créé un prefab de Catbot, un prefab composé de 4 roues (rouges), d'un cube (bleu) et d'un second cube (rouge) parenté au premier. Nous l'avons ensuite drag&drop depuis la hierarchie jusque dans la liste de nos assets.

### 3 PREFABS

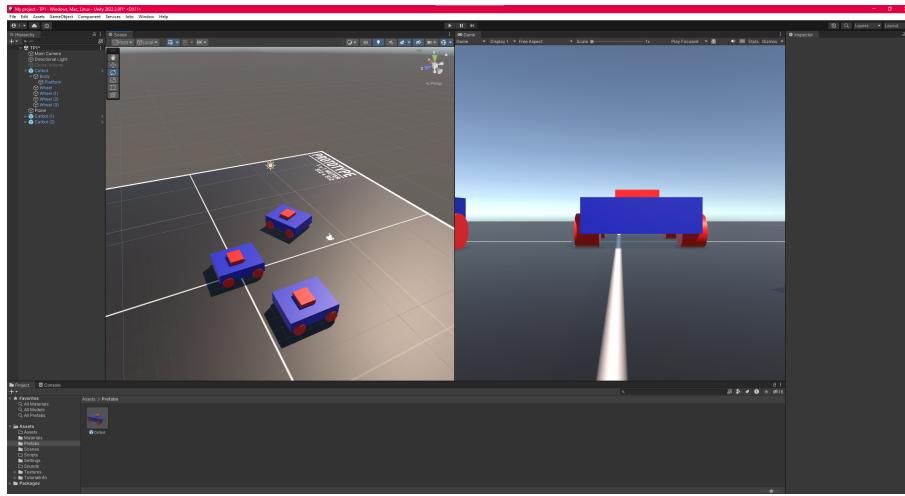


FIGURE 4 – Trois Catbots

Nous avons ensuite récupéré la MeowHead fournit. Nous l'avons rescale pour que sa forme soit cohérente et nous avons appliqué le material fournit avec la MeowHead. Nous l'avons ensuite déplacé pour la mettre juste au dessus la plateforme de la Catbot et nous avons "Apply all" cette instance pour que toutes les autres instances aient la MeowHead.

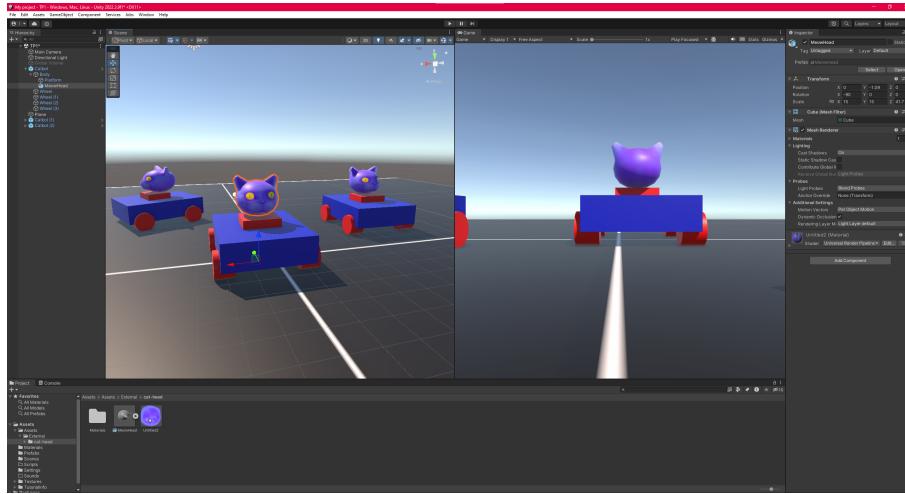


FIGURE 5 – Trois Catbots avec MeowHead

## 4 Skybox

Nous pouvons appliquer une skybox à partir des Lighting Settings (ou en faisant un drag&drop de la skybox dans la scène Unity). J'ai utilisé une skybox disponible dans un pack sur le shop Unity.

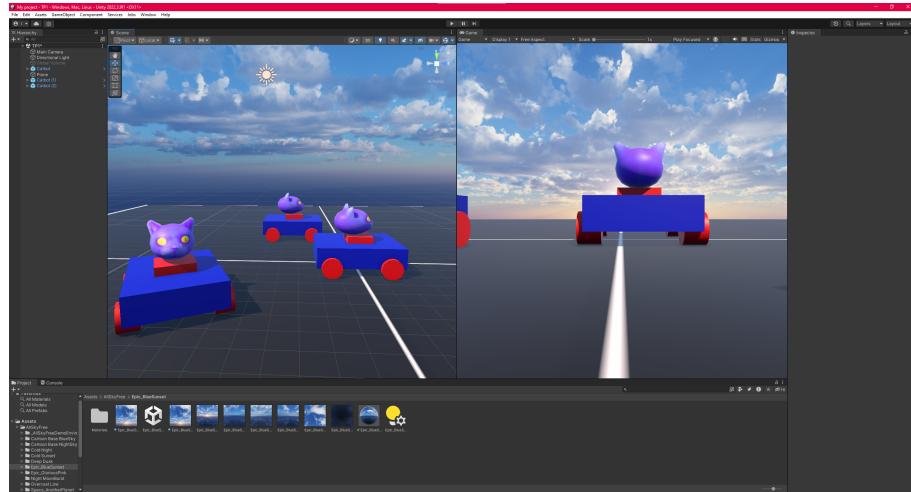
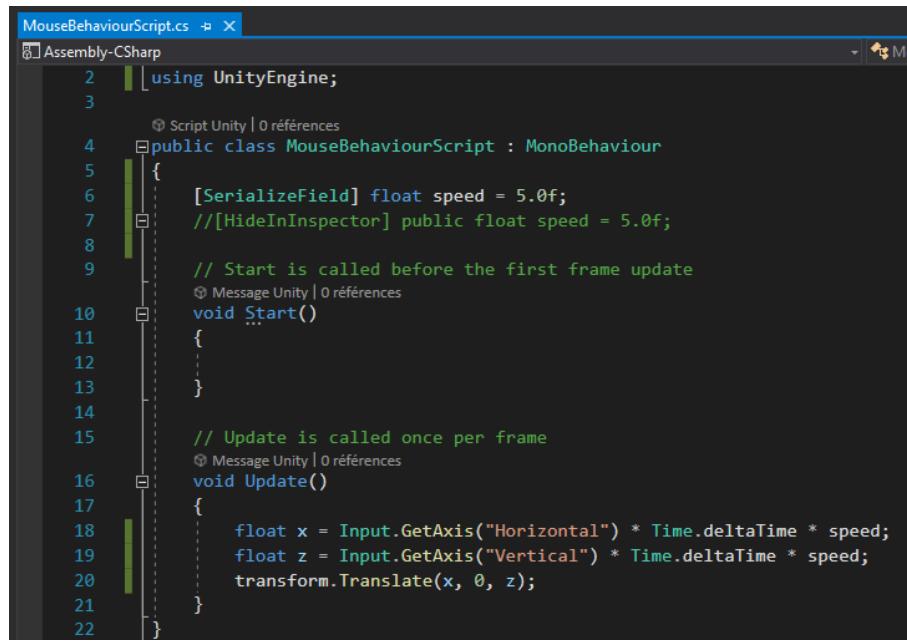


FIGURE 6 – Skybox ciel

## 5 Scripts

En ajoutant un script C# à notre GameObject Catbot, nous avons pu le faire avancer à l'aide des flèches.



```
MouseBehaviourScript.cs  X
Assembly-CSharp
  2  | using UnityEngine;
  3
  4  | public class MouseBehaviourScript : MonoBehaviour
  5  {
  6      [SerializeField] float speed = 5.0f;
  7      //#[HideInInspector] public float speed = 5.0f;
  8
  9      // Start is called before the first frame update
 10     void Start()
 11     {
 12     }
 13
 14
 15     // Update is called once per frame
 16     void Update()
 17     {
 18         float x = Input.GetAxis("Horizontal") * Time.deltaTime * speed;
 19         float z = Input.GetAxis("Vertical") * Time.deltaTime * speed;
 20         transform.Translate(x, 0, z);
 21     }
 22 }
```

FIGURE 7 – Script

En parentant un Catbot à celui qui possède le script, nous pouvons faire bouger les deux Catbots.

## 6 THIRD PERSON CHARACTER

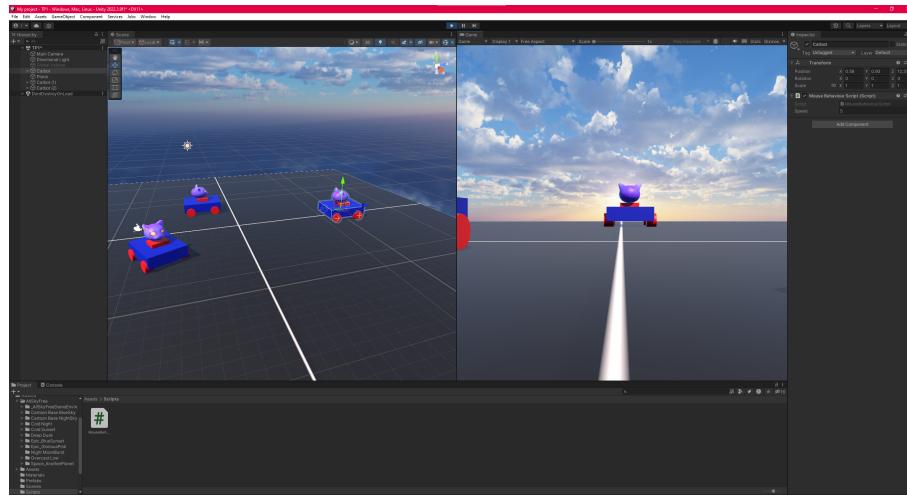


FIGURE 8 – Catbot qui se déplace

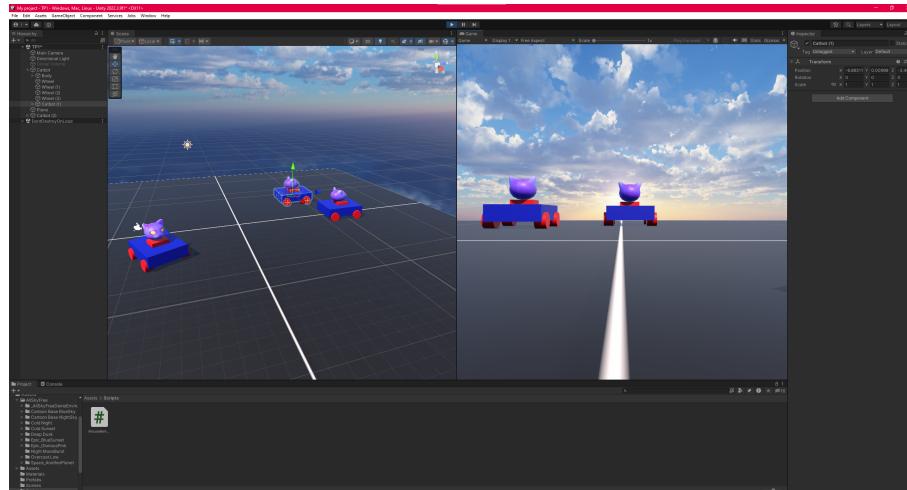


FIGURE 9 – 2 Catbots qui se déplacent

## 6 Third Person Character

A l'aide du template TPC, nous pouvons instancier un personnage et le faire bouger/sauter, le tout avec des animations. En parentant la caméra au TPC et changeant la position locale de la caméra, nous pouvons avoir une caméra qui suit le personnage en mode 3e personne.

## 6 THIRD PERSON CHARACTER

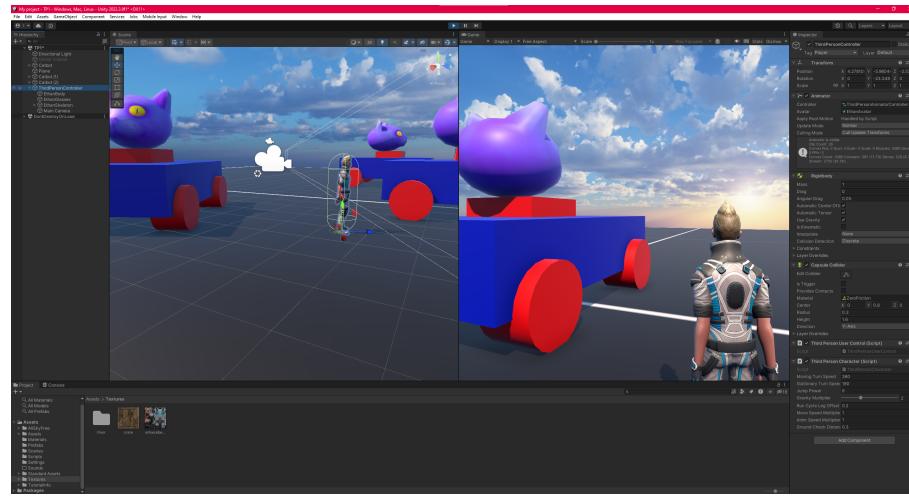


FIGURE 10 – TPC

Pour éviter que le personnage ne tombe à l'infini, nous créons un empty GameObject qui sera le point de spawn du personnage. Nous créons aussi une trigger box que nous plaçons en dessous du sol et qui doit être assez grande pour couvrir toute la zone où le joueur peut tomber. Nous plaçons un script sur cette trigger box qui nous permet de vérifier si le joueur rentre dans le trigger et, si c'est le cas, nous replaçons le joueur au point de spawn.

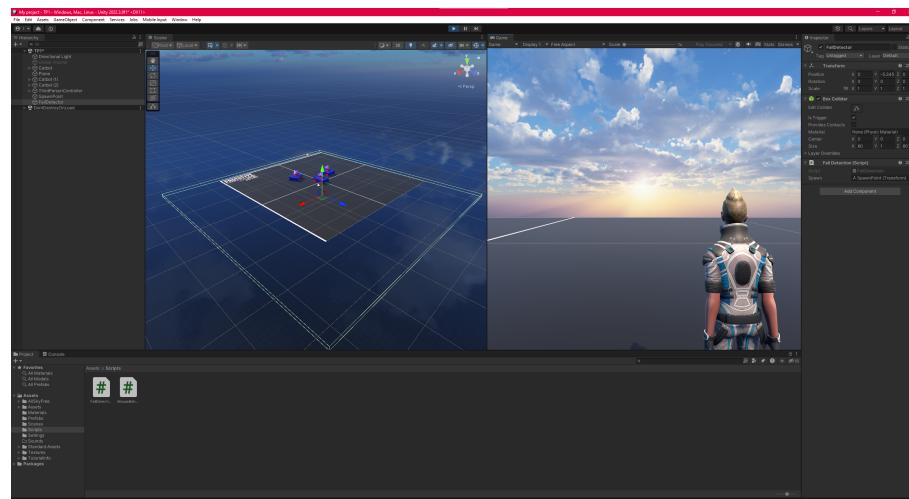


FIGURE 11 – Trigger box

## 7 Dragon

Nous avons ensuite placé un Dragon qui a été fournis. Ce dragon possède un animator et joue l'animation IDLE au lancement du jeu.

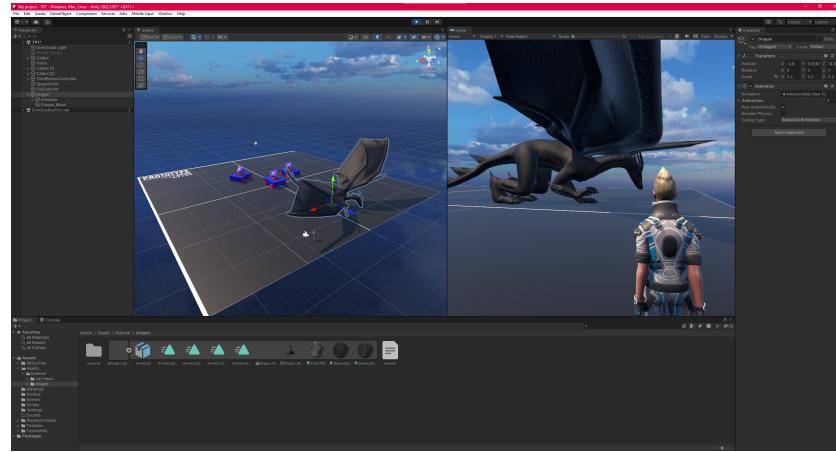


FIGURE 12 – Dragon en IDLE

## 8 Musique ambiante

Nous avons placé un AudioSource et lui avons donné la musique fournit. La musique est joué en boucle au moment où le jeu se lance.

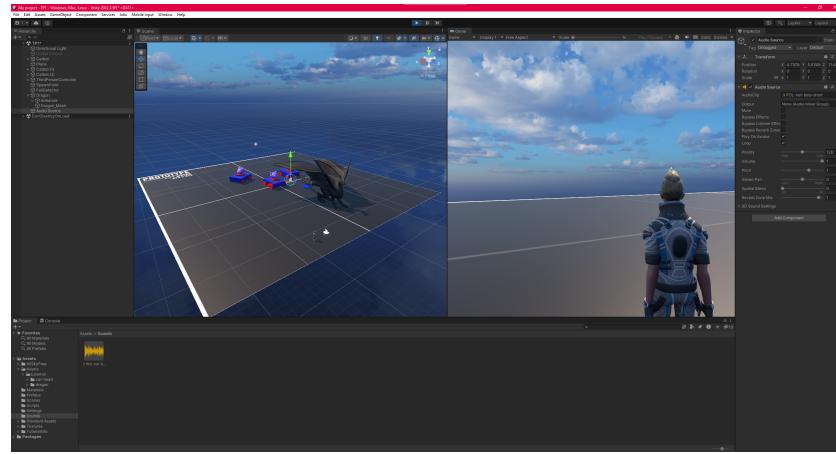


FIGURE 13 – AudioSource

## 9 Crédit de Catbots autonomes

Nous avons placé un empty gameobject permettant de faire spawn des prefabs à des positions aléatoires sur un plan. En lui donnant le prefab du Catvbot, nous pouvons faire spawn des Catbots. J'en ai fait spawn 5 autres sur le plan  $y = 0.89f$  pour qu'ils aient les roues sur le sol.

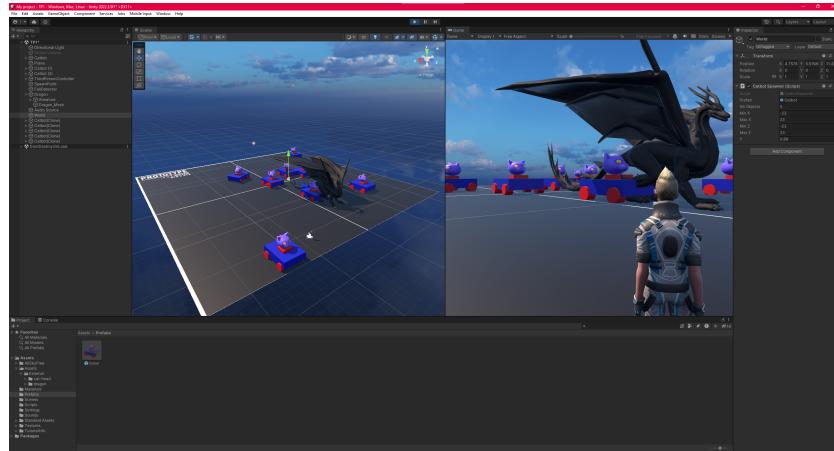


FIGURE 14 – 8 Catbots dont 5 Catbots qui ont spawn

Nous avons ensuite placé un navmesh pour permettre aux Catbots de se déplacer sur le sol de manière automatisée.

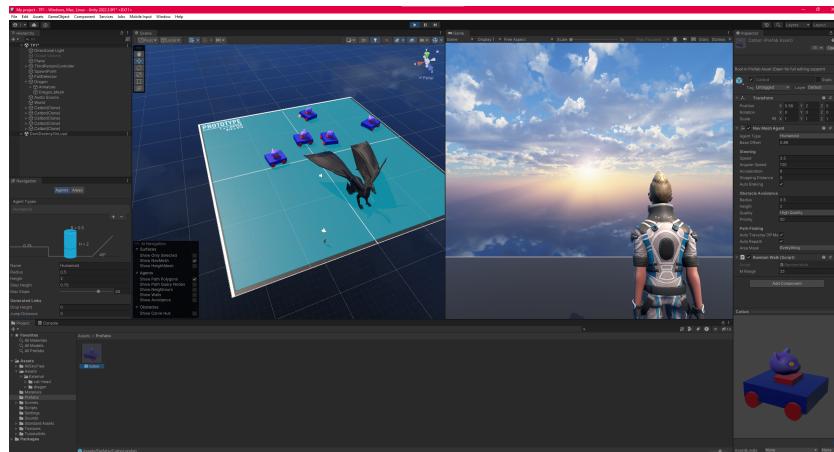


FIGURE 15 – Navmesh