OLFY

SENTEZ L'EXPERIENCE

Documentation

1.0

Table des matières

Pour commencer

Cette documentation va vous accompagner sur la mise en place du dispositif olfactif avec Unity.

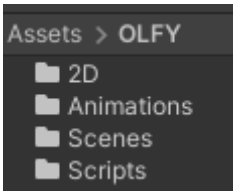
Dans le package que vous venez d'importer vous avez quatre dossiers.

2D : Contient des images/logos/… nécessaire pour la scène de démo.

Animation : Animation du logo de recherche présent dans la scène de démo.

Scenes : Dossier contenant la scène de demo

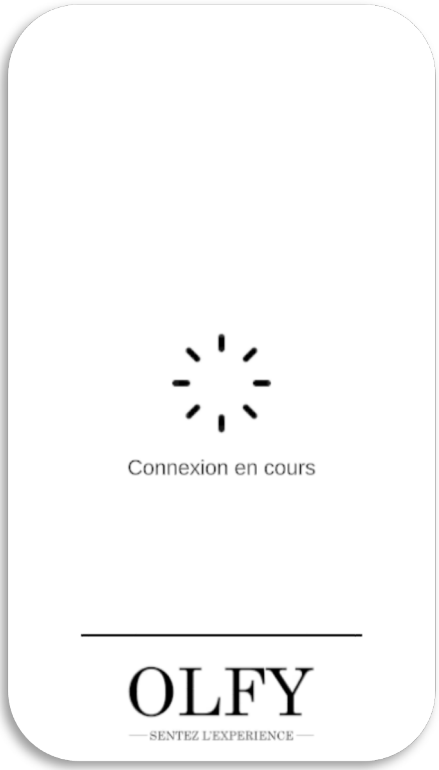
Scripts : Deux scripts dans ce dossier, Demo.cs et OlfyManager.

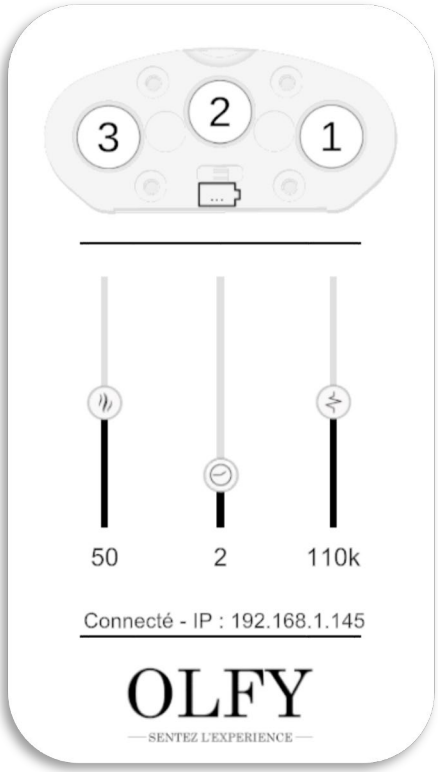


Scène de démo

Dans la scène de démo vous retrouvez une application vous permettant de tester le dispositif olfactif.

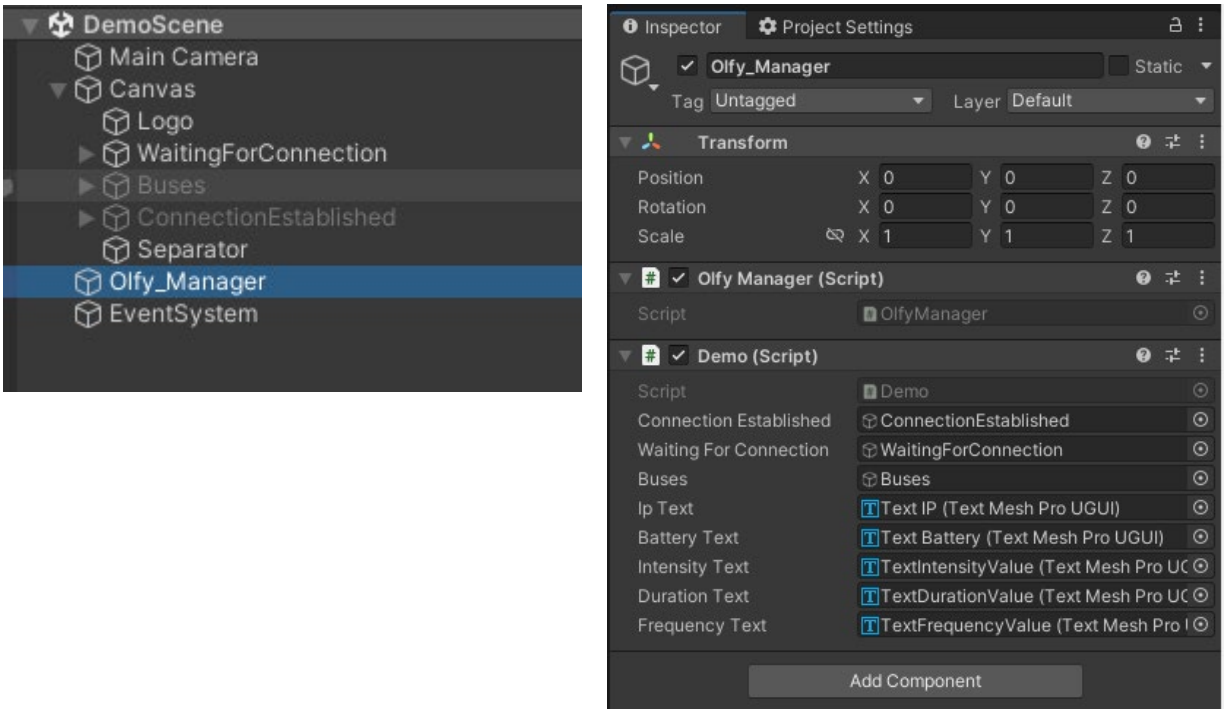
La première capture ci-dessous attend la connexion au dispositif et la seconde permet de tester le dispositif en appuyant sur les boutons 1, 2 ou 3 représentant les buses de diffusion. En dessous 3 « silders » permettant de gérer l'intensité de diffusion, sa durée et la fréquence.





Dans l'éditeur de Unity, vous retrouvez l'ensemble des éléments de la scène dans la hiérarchie (visuel de gauche).

A droite les scripts OlfyManager et Demo présent sur le GameObject Olfy_Manager.



Scripts

Un seul script vous sera utile pour la connexion et la diffusion avec le dispositif, il s'agit du script OlfyManager. Chaque fonction est expliquée directement dans le script.

Il s'occupe de la connexion au Start de l'application :

```
//Démarrage du scan au lancement de l'application
Message Unity | 0 références
private void Start()
{
    StartCoroutine(ScanNetwork());
}
```

Vous retrouvez également l'ensemble des fonctions pour échanger avec le dispositif :

```
//Déclenchement d'une odeur
1 référence
public void SendSmellToOlfy(int duration, string channel, int intensity, int freq, bool booster)
{
    StartCoroutine(SmellCoroutine(duration, channel, intensity, freq, booster));
}
//Déclenchement d'une odeur
1 référence
private IEnumerator SmellCoroutine(int duration, string channel, int intensity, int freq, bool booster)
{
    var request = new UnityWebRequest("http://" + address + "/olfy/diffuse", "POST");
    byte[] bodyRaw = System.Text.Encoding.UTF8.GetBytes("{\"duration\": " + duration + ", \"channel\": " + channel + ",\"intensity\"
    request.uploadHandler = (UploadHandler)new UploadHandlerRaw(bodyRaw);
    request.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();
    request.SetRequestHeader("Content-Type", "application/json");

    yield return request.SendWebRequest();

    Debug.Log("Status Code: " + request.responseCode);
}
```

```
//Mise en veille du dispositif
0 références
private IEnumerator DeepSleep()
{
    var request = new UnityWebRequest("http://" + address + "/olfy/deepsleep", "POST");
    request.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();
    request.SetRequestHeader("Content-Type", "application/json");

    yield return request.SendWebRequest();

    Debug.Log("Deep slepp Status Code: " + request.responseCode);
}
```

```
//Retourne les informations du Firmware
0 références
private IEnumerator GetFirmwareInformation()
{
    var request = new UnityWebRequest("http://" + address + "/olfy/firmware", "GET");
    byte[] bodyRaw = System.Text.Encoding.UTF8.GetBytes("{\"channel\": " + /*channel*/1 + ",\"duration\": " + /*duration */6000 + "}")
    request.uploadHandler = (UploadHandler)new UploadHandlerRaw(bodyRaw);
    request.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();
    request.SetRequestHeader("Content-Type", "application/json");

    yield return request.SendWebRequest();
    deviceInformation = request.responseCode.ToString();
    Debug.Log("Firmware: " + request.responseCode);
}
```

```
public IEnumerator GetBattery()
{
    var request = new UnityWebRequest("http://" + address + "/olfy/getbatt", "GET");

    request.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();

    yield return request.SendWebRequest();
    string data = JObject.Parse(request.downloadHandler.text)["battery"].ToString();
    batteryLevel = data;
    Debug.Log(request.downloadHandler.text);
}
```

```
private IEnumerator StopOlfy()
{
    var request = new UnityWebRequest("http://" + address + "/olfy/stop_all", "POST");
    byte[] bodyRaw = System.Text.Encoding.UTF8.GetBytes("{}");
    request.uploadHandler = (UploadHandler)new UploadHandlerRaw(bodyRaw);
    request.downloadHandler = (DownloadHandler)new DownloadHandlerBuffer();
    request.SetRequestHeader("Content-Type", "application/json");
    yield return request.SendWebRequest();
    Debug.Log("status: " + request.responseCode);
```