



RÉALITÉ AUGMENTÉE : AR + GPS

1. Objectif :

Afficher la position GPS du téléphone en AR.

2. Sources :

Tuto youtube : Unity Mobile GPS par N3K

Autre tuto à voir : Markless Augmented Reality - How to - Unity 3D par N3K

3. Mise en oeuvre

En suivant le tuto tout fonctionne bien. Attention le mode 'AR' de N3K nécessite de suivre un autre tuto afin de créer une caméra AR.

C'est pour cela que nous utiliserons une caméra 'Vuforia' au lieu de la 'main camera'. Cela permet de ne pas créer la caméra proposée dans le tuto de N3K.

Ci dessous mes codes et copies d'écran.

3.1 Méthode

Créer un nouveau projet

Effacer la 'main camera'.

Placer un 'ARcamera' de vuforia

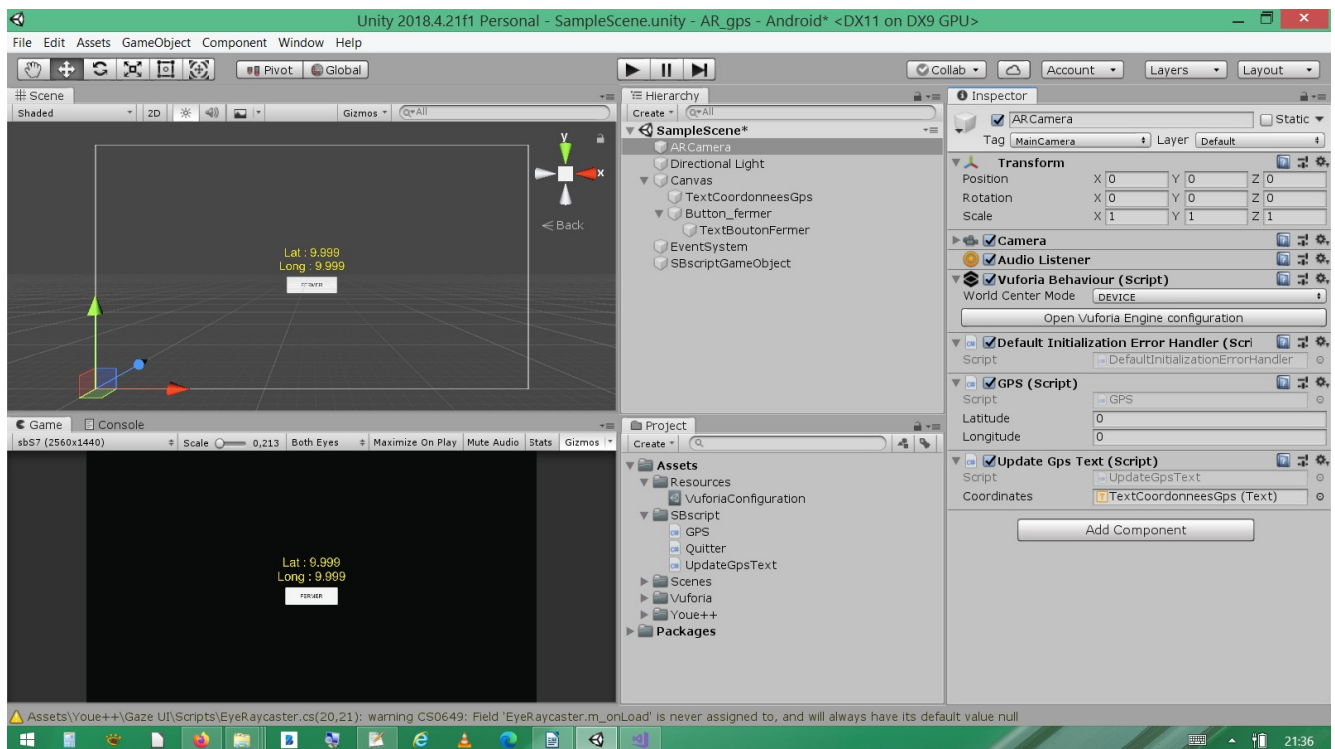
Placer un canvas avec un 'text' afin d'afficher les coordonnées : latitude et longitude.

Ajouter les scripts ci-dessous et les associer aux différents éléments.

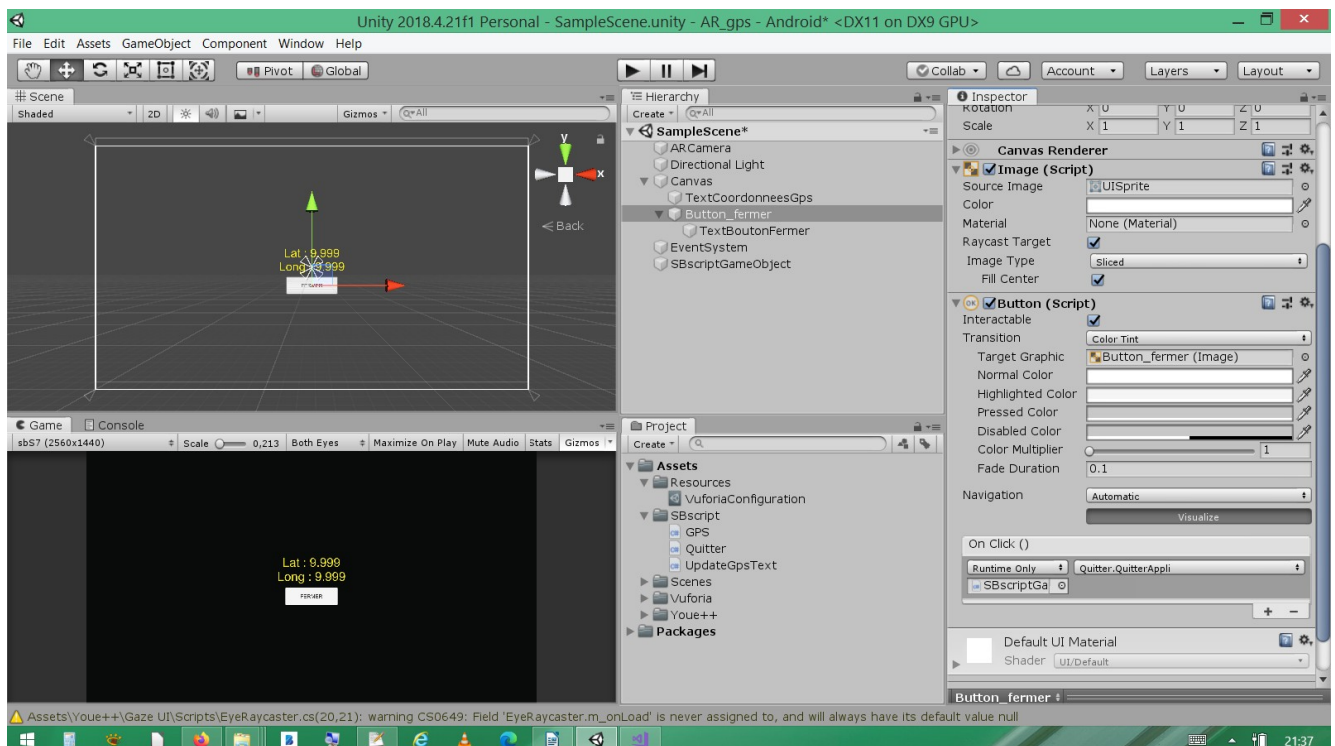
Compiler (build).

3.2 Organisation sous Unity

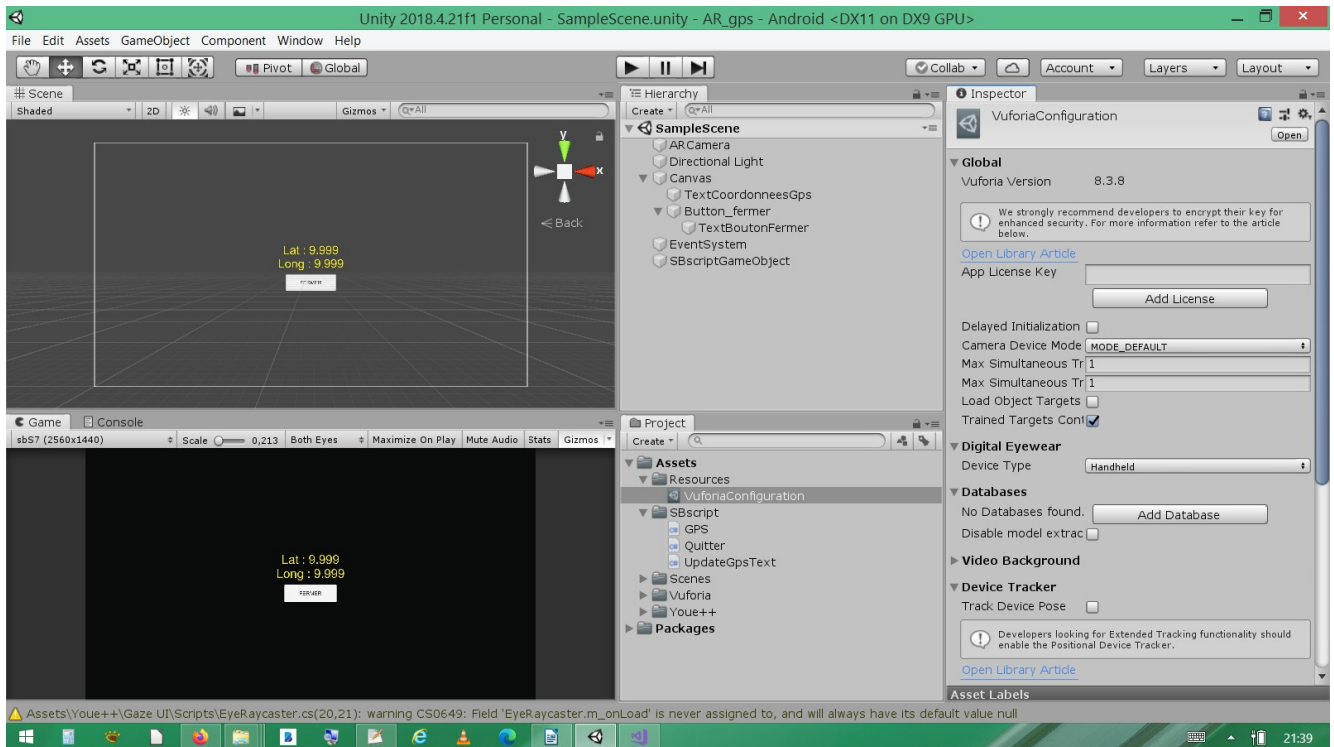
Configuration : 'ARcamera' de Vuforia



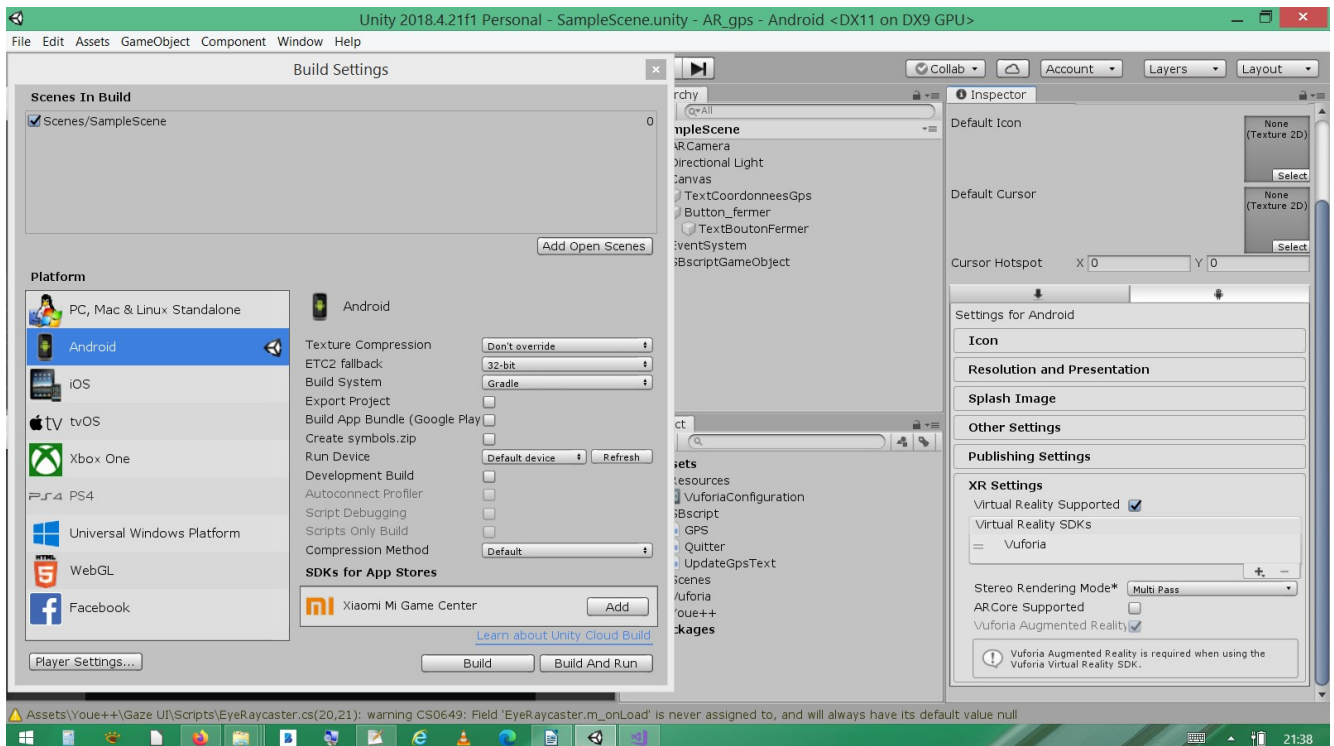
Bouton : 'fermer' : pour fermer l'application (optionnel)



Configuration de 'Arcamera' Vuforia :



Buid :



3.3 Code des scripts

3.3.1 GPS.cs (lecture des coordonnées en provenance du téléphone)

Le script est à placer dans l'Arcamera.

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class GPS : MonoBehaviour
{
    public static GPS Instance { set; get; }

    public float latitude;
    public float longitude;

    // Start is called before the first frame update
    void Start()
    {
        Instance = this;
        DontDestroyOnLoad(gameObject);
        StartCoroutine(StartLocationService());
    }

    private IEnumerator StartLocationService()
    {
        if (!Input.location.isEnabledByUser)
        {
            Debug.Log("GPS non autorisé par l'utilisateur");
        }
        Input.location.Start();
        int maxWait = 20;
        while (Input.location.status == LocationServiceStatus.Initializing && maxWait > 0)
        {
            yield return new WaitForSeconds(1);
            maxWait--;
        }
        if (maxWait <= 0)
        {
            Debug.Log("Timed out");
            yield break;
        }
        if (Input.location.status == LocationServiceStatus.Failed)
```

```
{
    Debug.Log("Impossible de trouver le device");
    yield break;
}
latitude = Input.location.lastData.latitude;
longitude = Input.location.lastData.longitude;
}
```

3.3.2 UpdateGpsText.cs (mise à jour du texte affichant les coordonnées)

Le script est à placer dans l'Arcamera :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class UpdateGpsText : MonoBehaviour
{
    public Text coordinates;

    // Update is called once per frame
    void Update()
    {
        coordinates.text = "Lat : " + GPS.Instance.latitude.ToString() + " Long : " +
GPS.Instance.longitude.ToString();
    }
}
```

3.3.3 Quitter.cs (bouton pour fermer l'application)

Le script est à placer dans un gameobject 'SbgameObject' puis dans 'onclick' du bouton.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

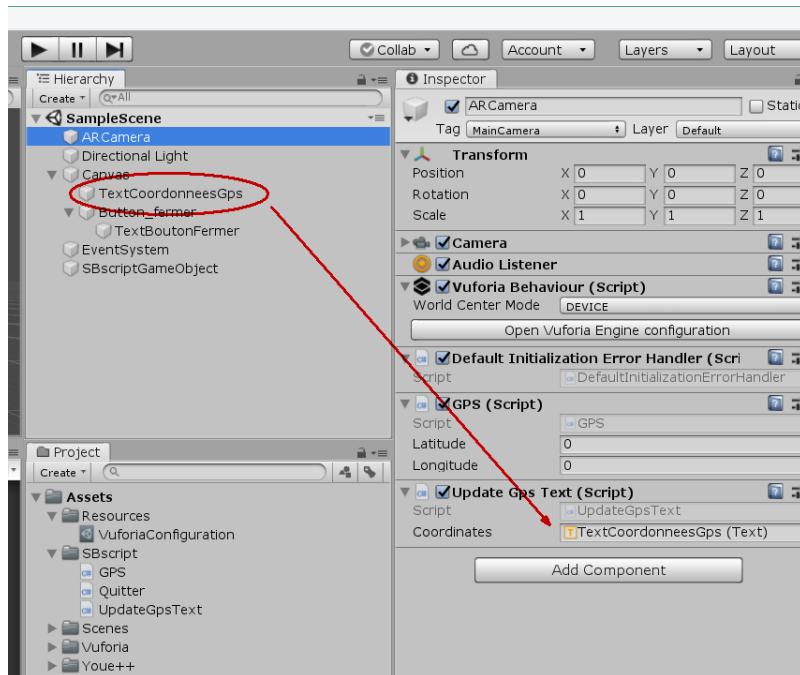
public class Quitter : MonoBehaviour
{
    public void QuitterAppli()
    {
        Application.Quit();
    }
}
```

```
}  
}
```

3.3.4 Pour rappel :

Les variables déclarées publiques dans les scripts apparaissent dans Unity comme les paramètres des scripts.

Il suffit de faire un clic-déposer du paramètre vers le script :



4. Conclusion

Ce tuto permet d'inclure la position de votre téléphone dans une application AR.