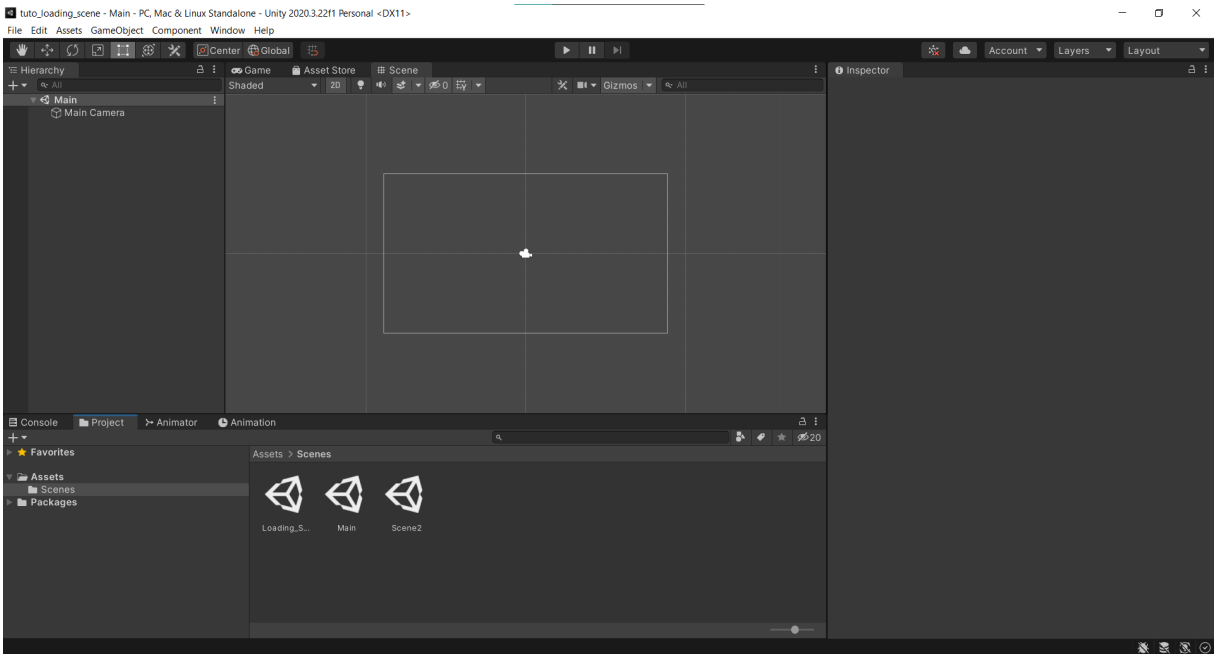
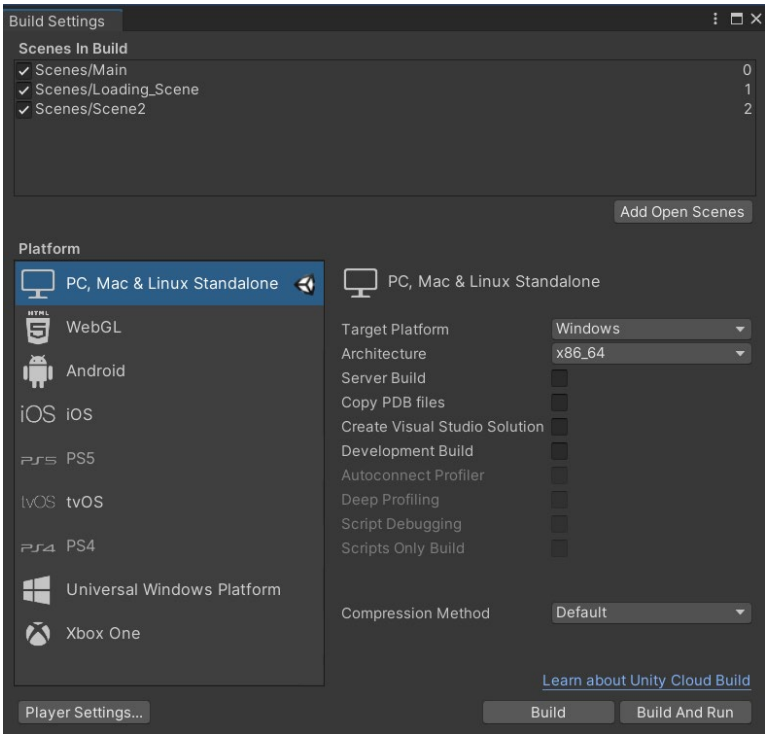
Tuto loading screen Unity :

(Il s'agit d'une version améliorée et simplifiée du système de « WTRB ? » ; le projet Unity du tutoriel est téléchargeable ici : https://github.com/Lamaxelle/tuto_loading_screen_with_animation)

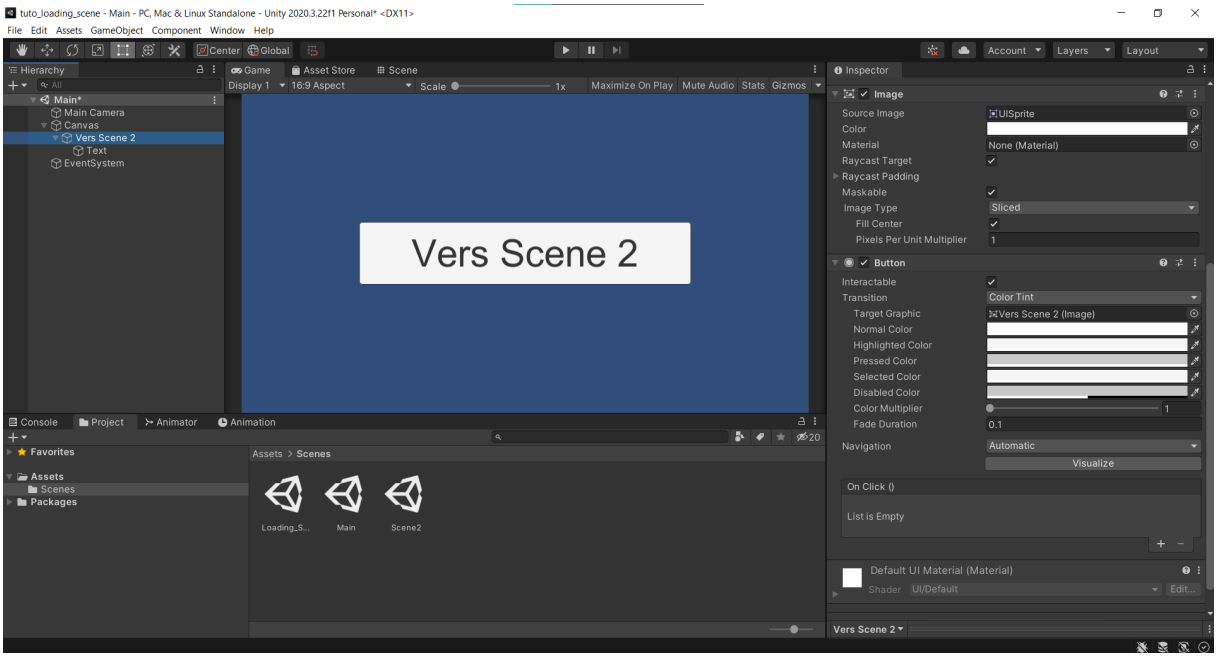
1- Créer plusieurs scènes, au moins 3 :
   - Celle de départ
   - Celle de chargement
   - Celle d'arrivée

   Puis veiller à toutes las ajouter aux « build settings »

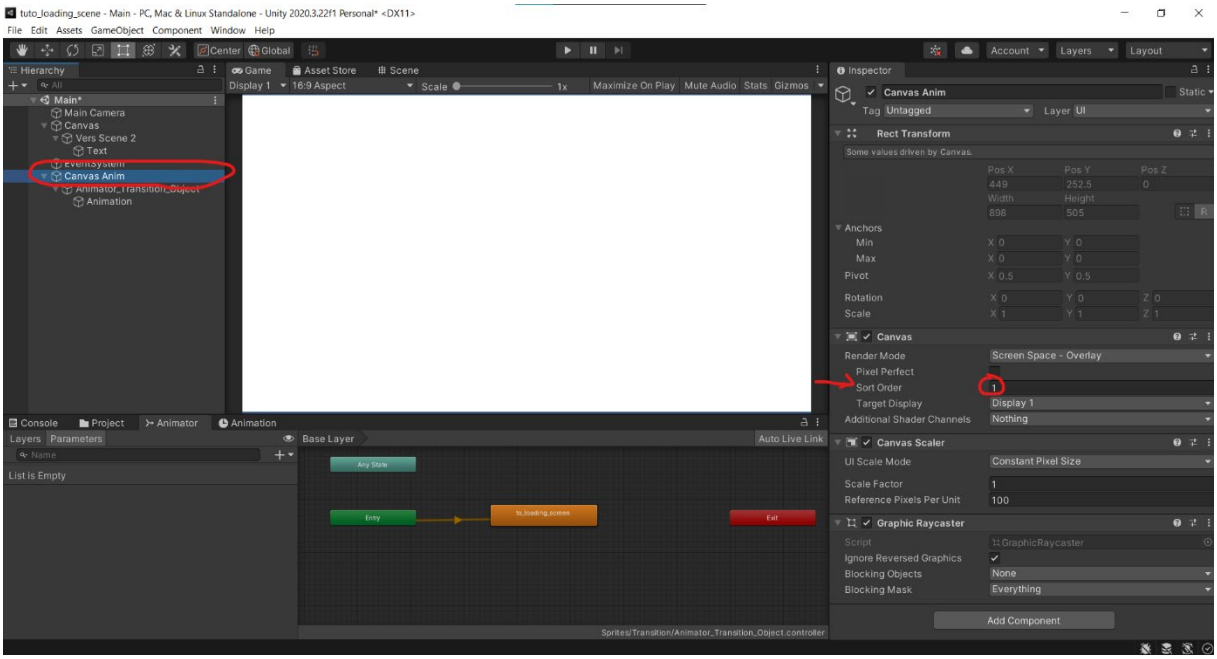


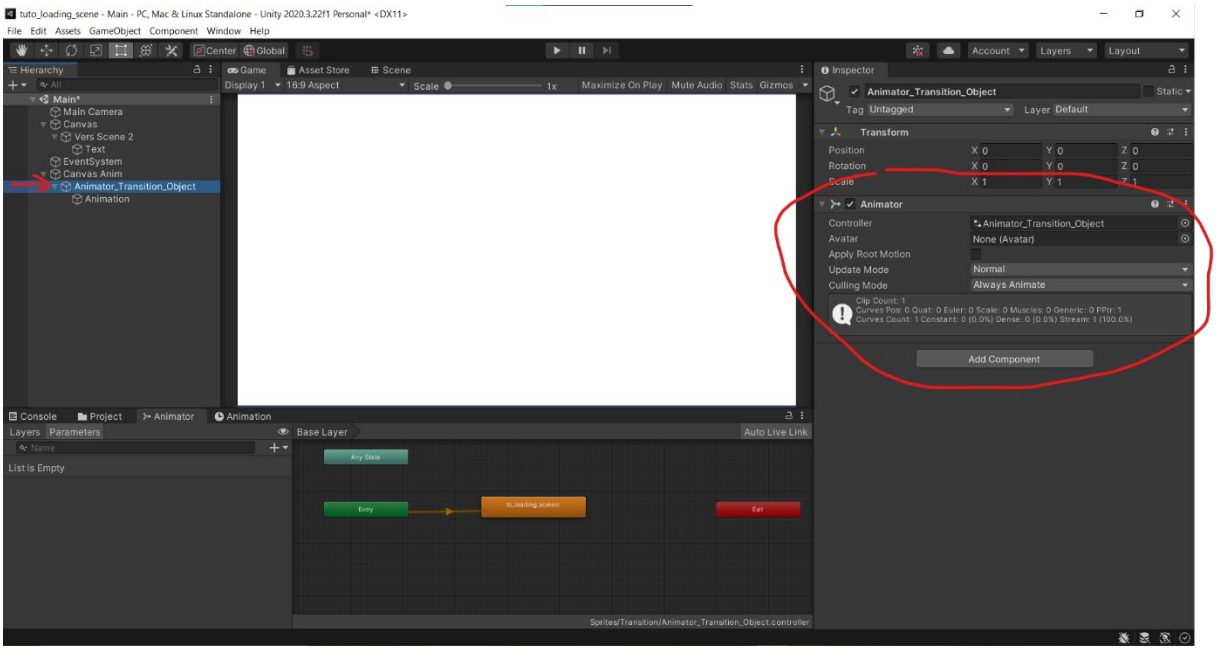


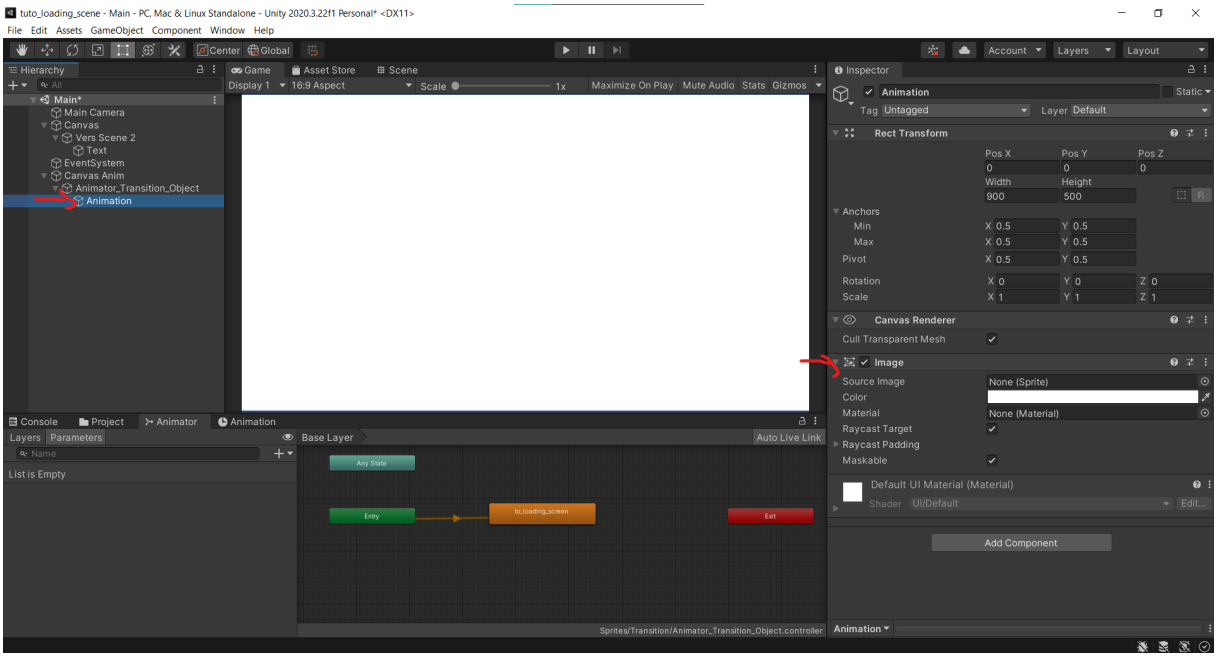
2- Créer un bouton (ou une autre interaction) dans la scène de départ pour aller dans la scène d'arrivée



3- Créer l'animation de transition :
   - Créer un gameobject vide dans un canvas autre que celui du bouton avec un « Sort Order» supérieur
   - Ajouter un animator à cet objet vide
   - Mettre en enfant de cet objet un autre objet contenant l'animation/l'objet à animer (ça peut être un objet 3D, un sprite avec des bones comme Rose, ou une succession de frame qui créée une animation), dans mon cas, il s'agit d'une animation en frame par frame, donc je mets un objet avec le component « image » en enfant et je fais l'animation en changeant la « source image ».

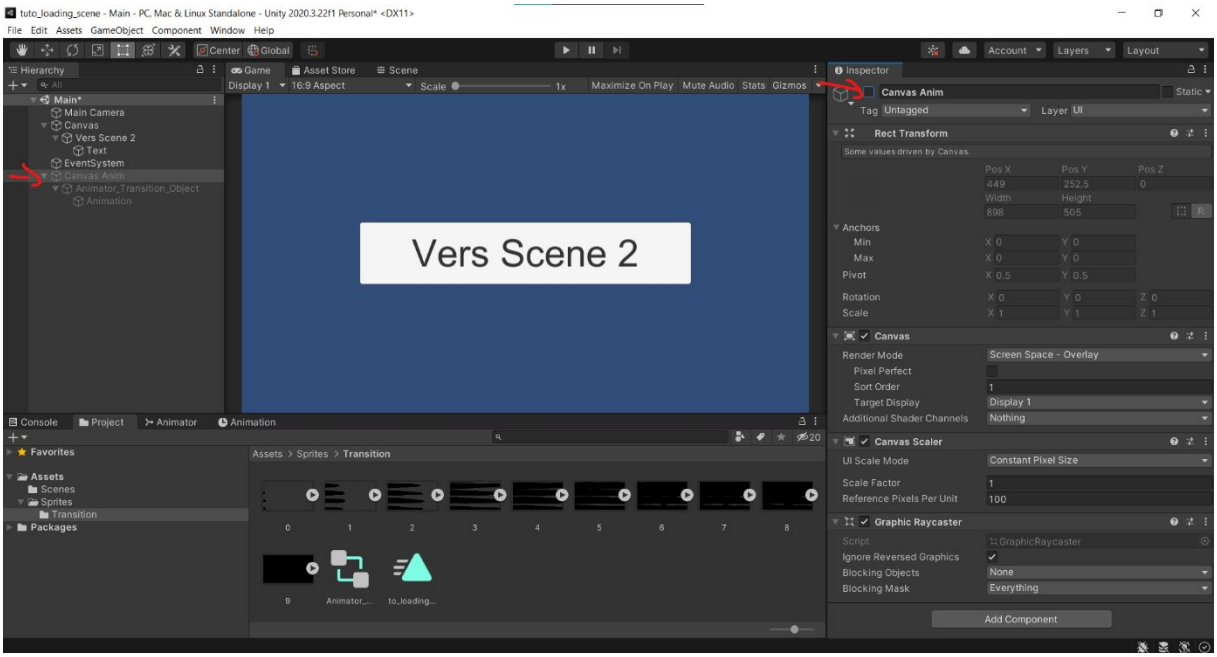


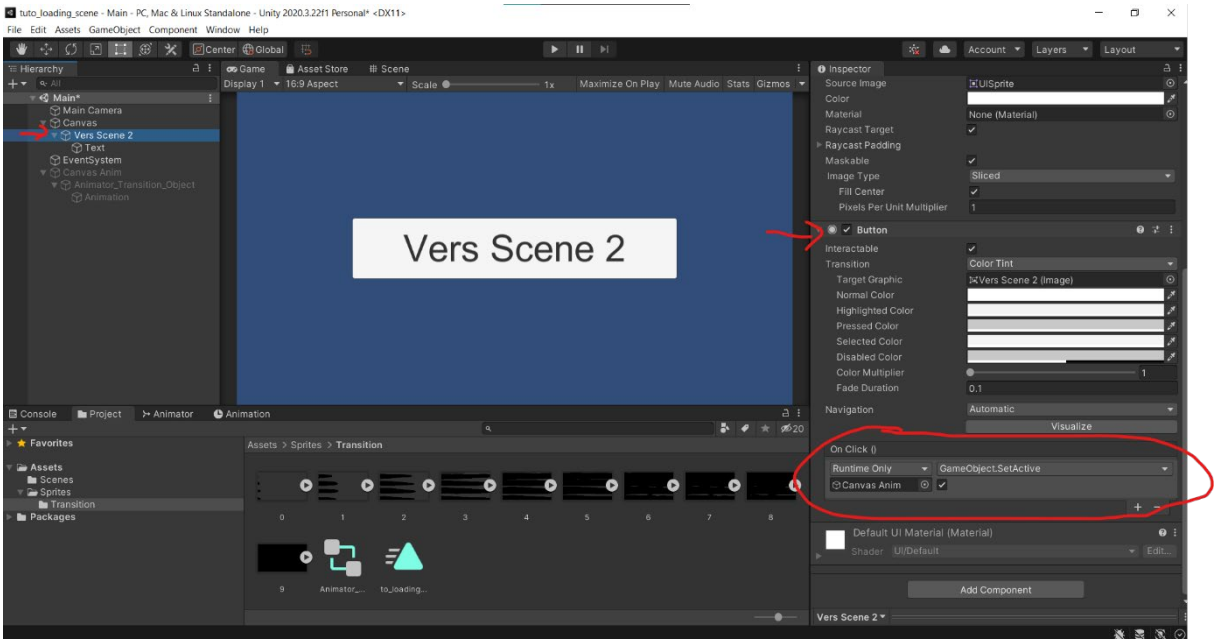




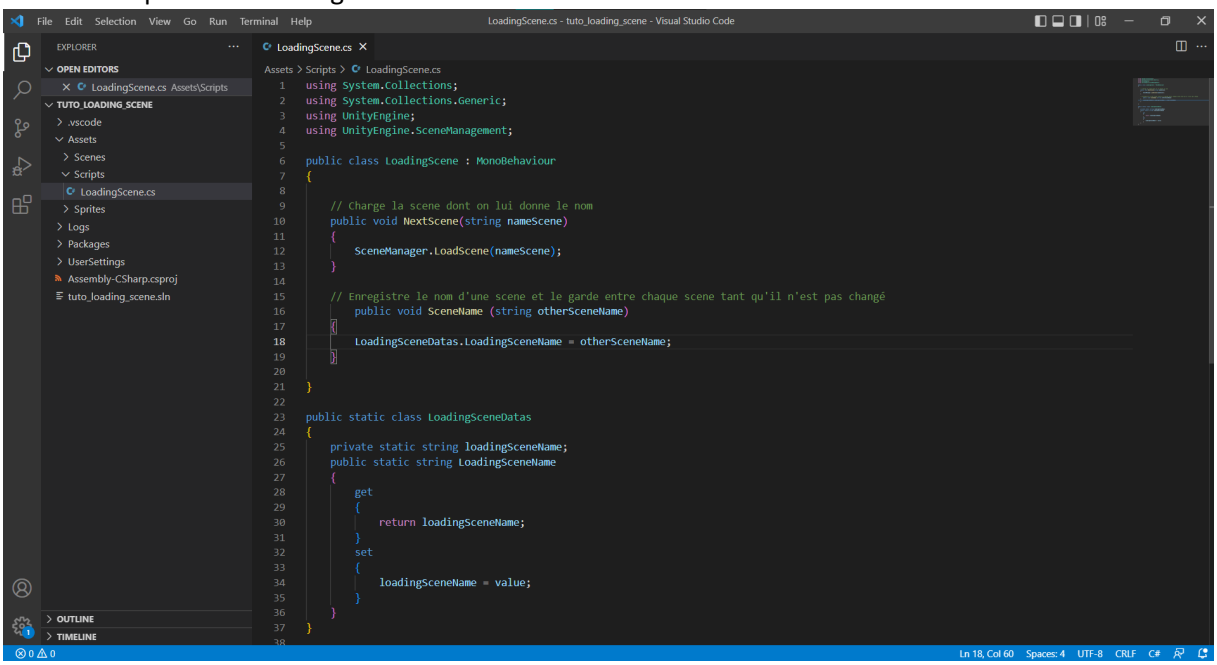
4- Configuration du bouton, partie 1 :
   - Désactiver le gameobject « Canvas Anim »
   - Ajouter « Canvas Anim » dans le component « Button » de « Vers Scene 2 »

- Chercher « GameObject.SetActive » dans le menu déroulant et cocher la case. L'animation apparait maintenant quand on appuie sur le bouton.



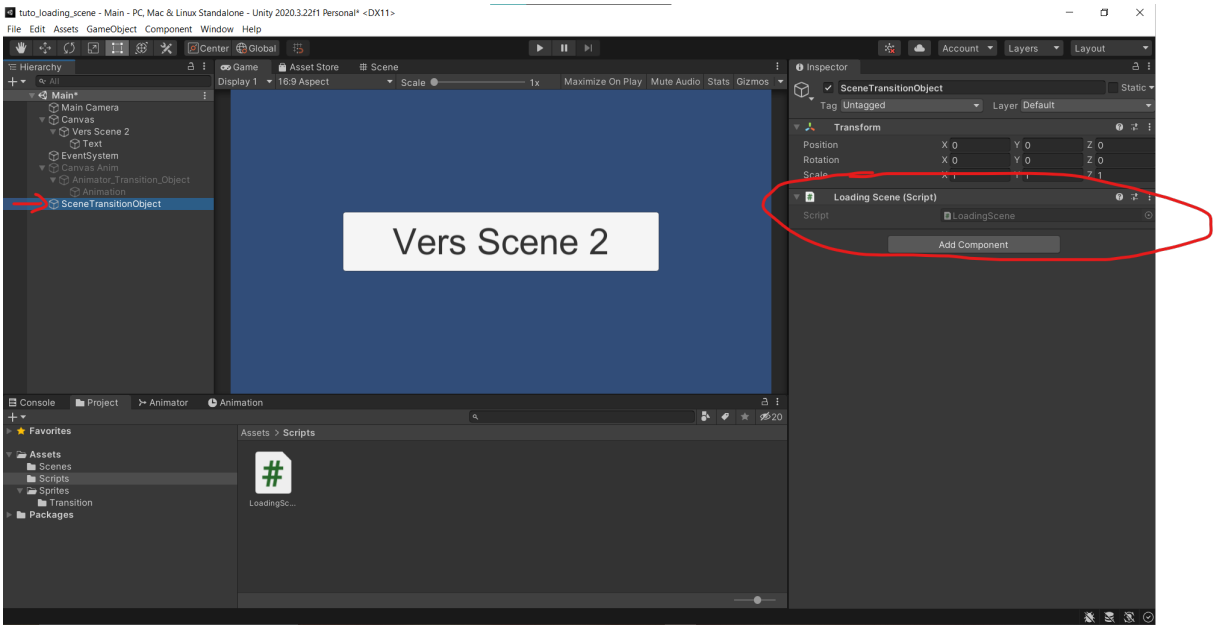


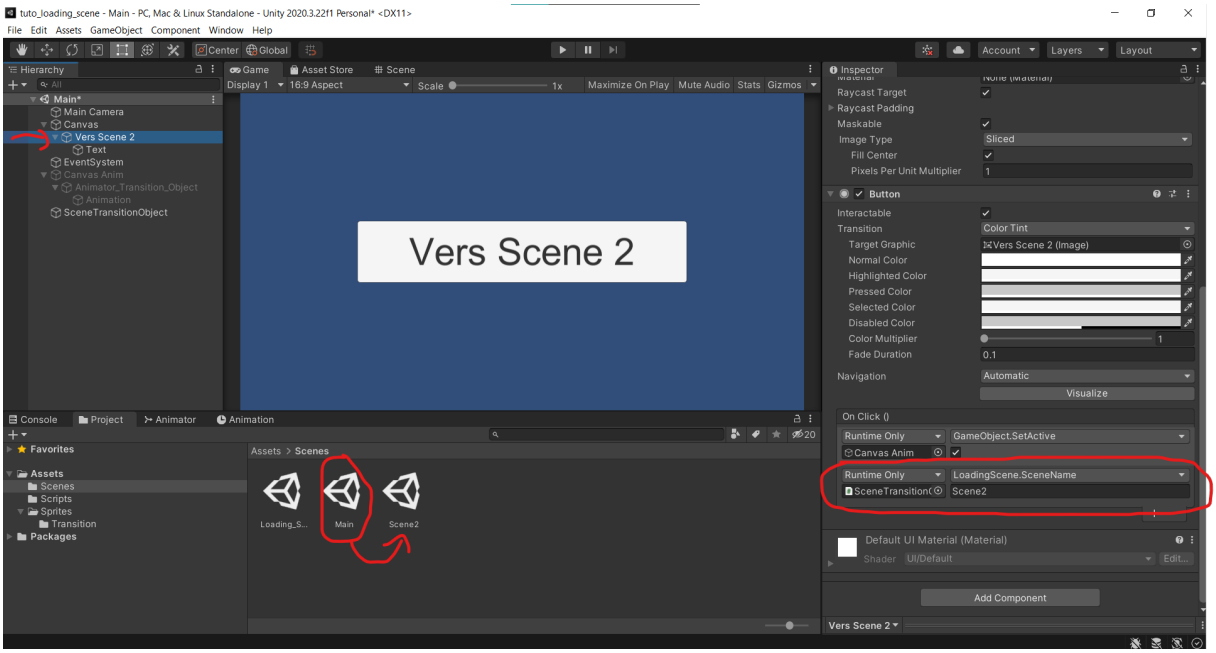
5- Création du script de chargement de scène :
   - Créer un nouveau script
   - Créer le script comme l'image suivante :



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

public class LoadingScene : MonoBehaviour
{

    // Charge la scene dont on lui donne le nom
    public void NextScene(string nameScene)
    {
        SceneManager.LoadScene(nameScene);
    }

    // Enregistre le nom d'une scene et le garde entre chaque scene tant qu'il n'est pas changé
        public void SceneName (string otherSceneName)
    {
        LoadingSceneDatas.LoadingSceneName = otherSceneName;
    }

}

public static class LoadingSceneDatas
{
    private static string loadingSceneName;
    public static string LoadingSceneName
    {
        get
        {
            return loadingSceneName;
        }
        set
        {
            loadingSceneName = value;
        }
    }
}
```

6- Lier le script à la scène :
   - Créer un gameobject vide dans la scène et y glisser le script que l'on vient de créer
   - Dans le component « Button » de l'objet « Vers Scene 2 », ajouter une ligne supplémentaire dans le « On Click () » et y faire glisser le nouveau gameobject créé. Chercher dans le menu déroulant « LoadingScene.SceneName », puis écrire le nom de la scène de destination souhaitée (pas la scène de chargement).
   - Le bouton indique maintenant au script de chargement de scène le nom de la prochaine scène souhaitée.





7- Revenir sur le script « LoadingScene » pour y ajouter les modifications suivantes :

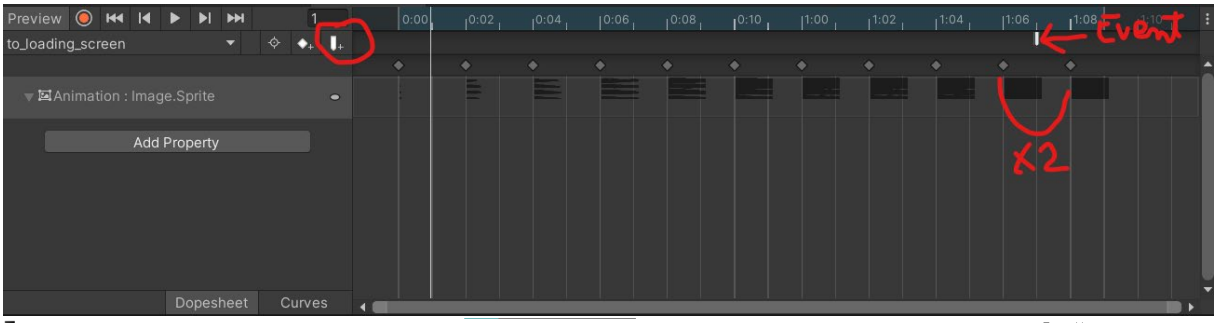
```
void Start()
{
    // Permet de charger une scène uniquement lorque l'on est dans la scène de chargement
    if (SceneManager.GetActiveScene().name=="Loading_Scene")
    {
        StartCoroutine (LoadAsyncOperation());
    }
}
```

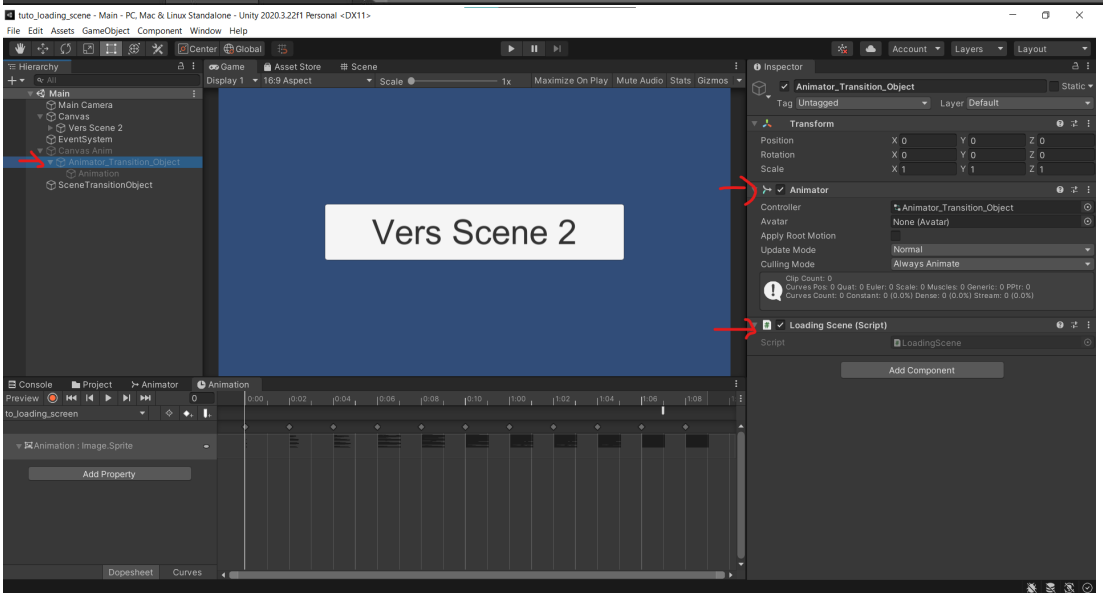
```
// Permet de charger une scène uniquement lorque l'on est dans la scène de chargement
IEnumerator LoadAsyncOperation ()
{
    if (SceneManager.GetActiveScene().name=="Loading_Scene")
    {
        AsyncOperation asyncOperation = SceneManager.LoadSceneAsync(LoadingSceneDatas.LoadingSceneName);
        while (asyncOperation.progress<0.9f)
        {
            asyncOperation.allowSceneActivation = false;
            yield return null;
        }
        if (asyncOperation.progress>=0.9f)
        {
            asyncOperation.allowSceneActivation = true;
            yield return null;
        }
    }
}
```

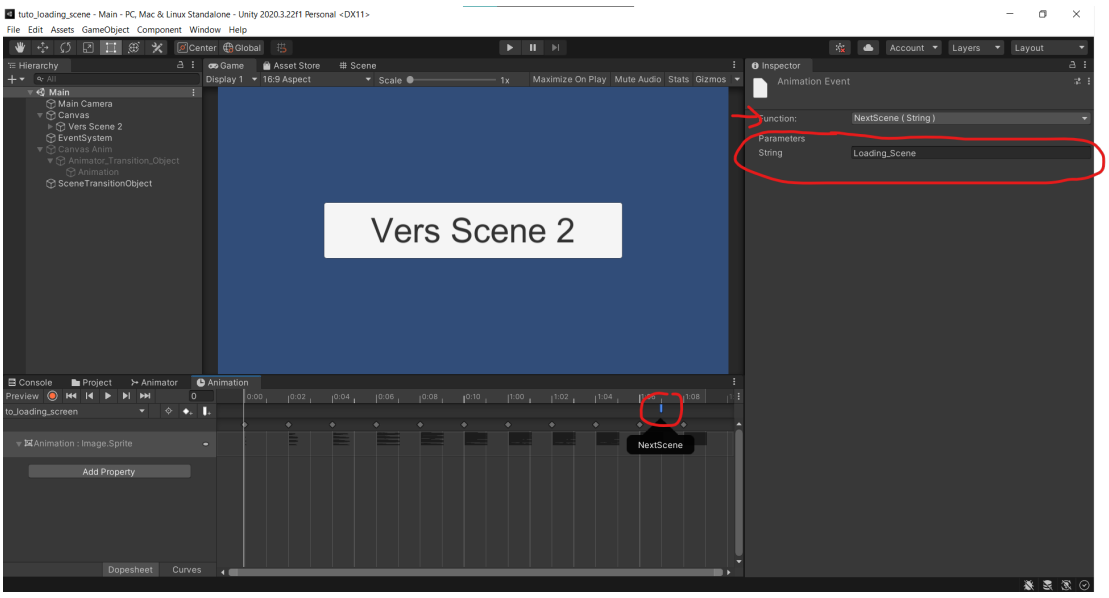
Remarque : le nom « Loading_Scene » peut être remplacé par le nom de scène que tu as donné à ta scène de chargement.

8- Derniers préparatifs sur la première scène :

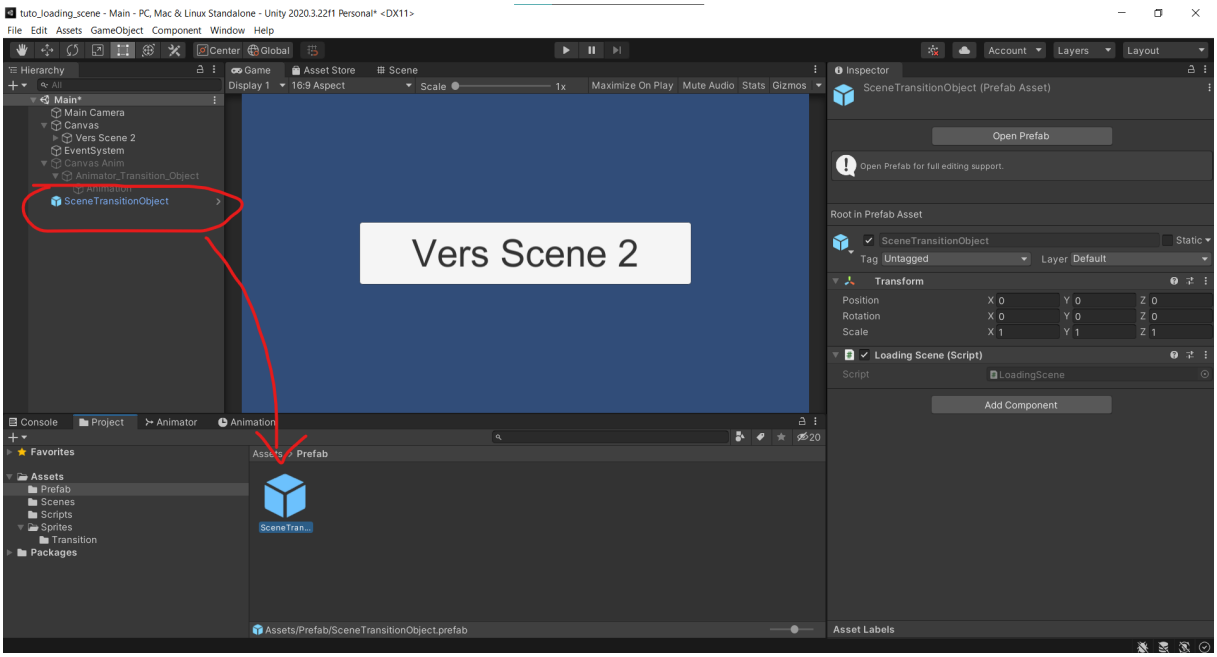
- Modifier l'animation de disparition de la scène pour y ajouter un event (à la fin de l'animation, de préférence répéter la dernière frame et mettre l'event entre les deux frames qui se répètent).
- Glisser le script « LoadingScene » sur l'objet avec l'animator
- Cliquer sur l'event dans l'animation, puis dans l'inspector sélectionner « NextScene (string) » dans le menu déroulant « Function ».
- Ecrire le nom de la scène de chargement dans l'emplacement vide (« Loading_Scene » dans mon cas).



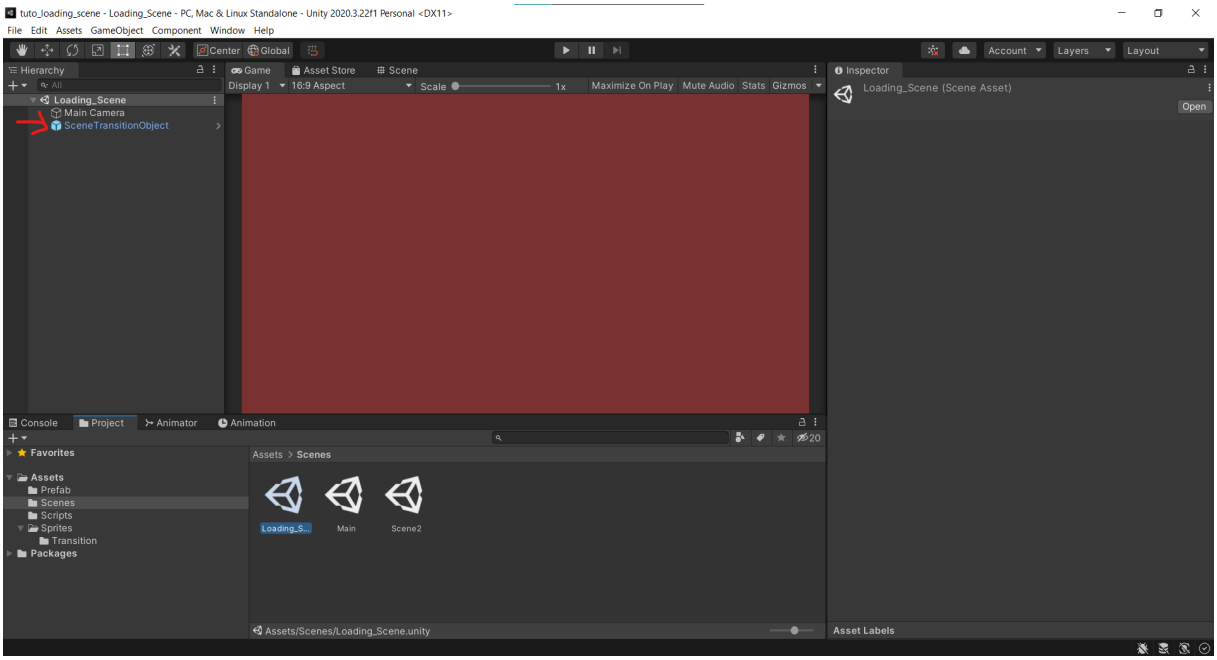




9- Faire un préfab avec le gameobject « SceneTransitionObject » (le glisser dans un fichier)



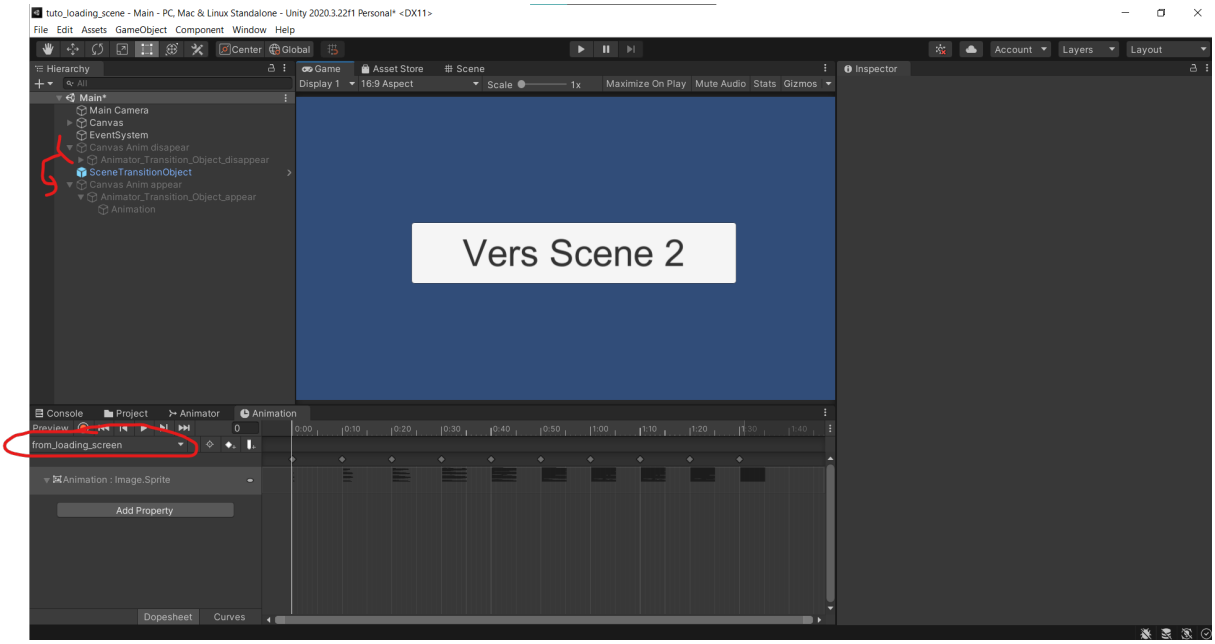
10- Dans la scène « Loading_Scene » ajouter le prefab « SceneTransitionObject »



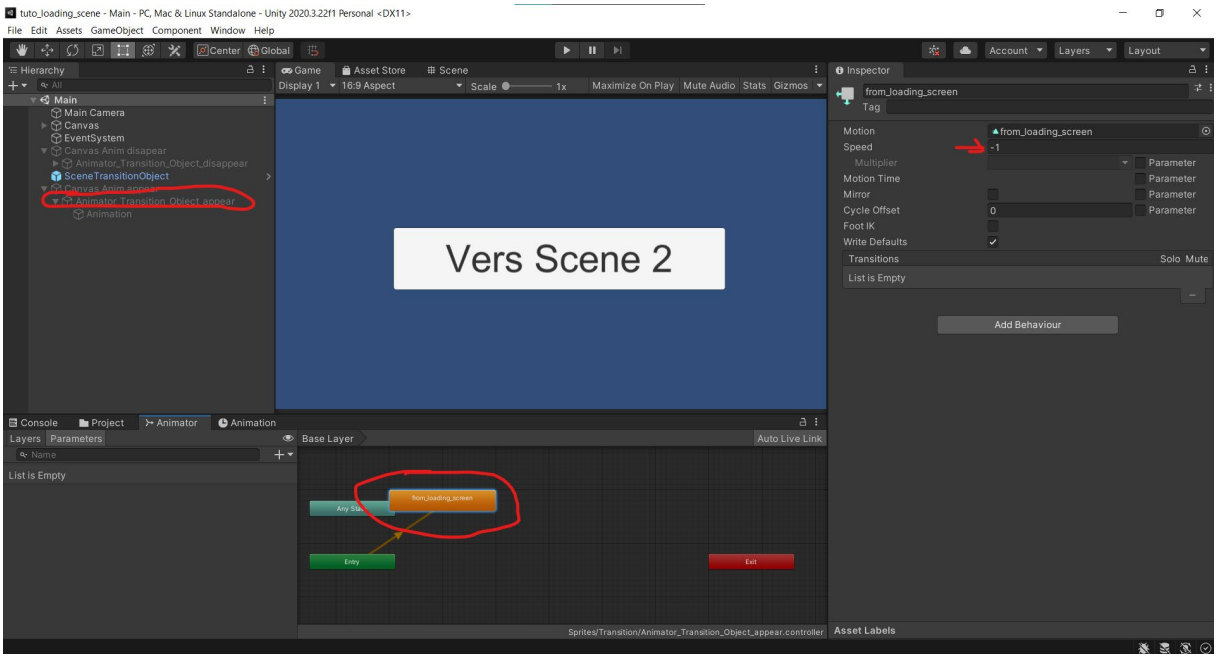
11- La scène de transition s'affiche maintenant tant que la scène suivante n'a pas chargé ! On peut maintenant pousser plus loin en ajoutant également une animation d'apparition.

Fin du système de « WTRB ? » ; Ce qui suit permet d'avoir une transition encore plus fluide.

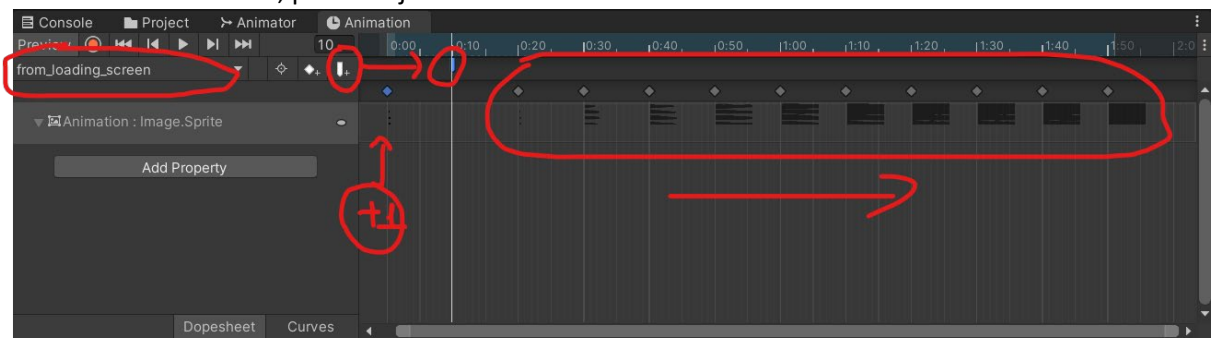
12- Dans un premier temps, on duplique l'objet « Canvas Anim » dans la première scène. On supprime l'animator de l'objet « Animator_Transition_Object », puis on en recréé un. On créé une nouvelle animation dans cet animator, et on copie-colle les frames de l'animation précédente, sans la dernière frame et l'event. J'ai également ajouter « appear » et « disappear » à la fin de chaque canvas et chaque animator pour pouvoir les différencier plus facilement.



13- Dans le nouvel animator, on ajuste la vitesse de l'animation à -1 pour qu'elle défile à l'envers.



14- On décale les frames de l'animation « from_loading_screen » vers la droite, on ajoute une autre frame 0 au début, puis on ajoute un event entre les deux frame 0.



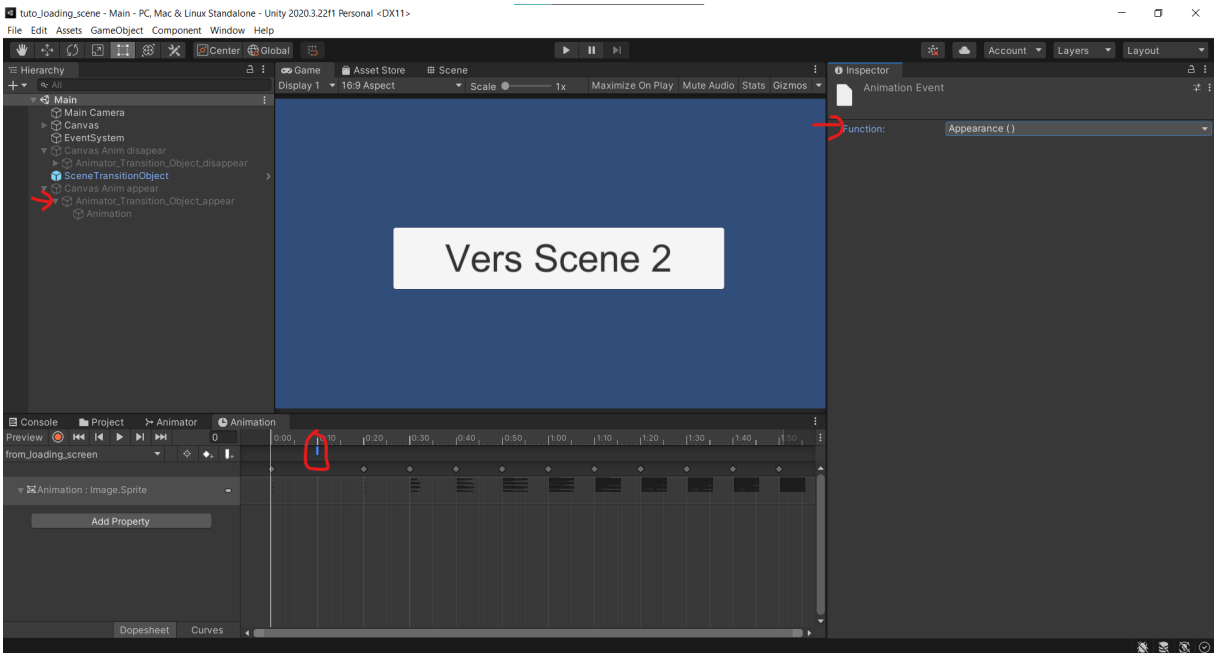
15- On modifie une dernière fois le script en ajoutant les lignes suivantes :

```
// Boolean qui enregistre si une transition vient d'être faite
public static bool fromLoadingScene = false;

// Canvas de l'animation d'apparition
public GameObject appearanceTransitionObject;
```

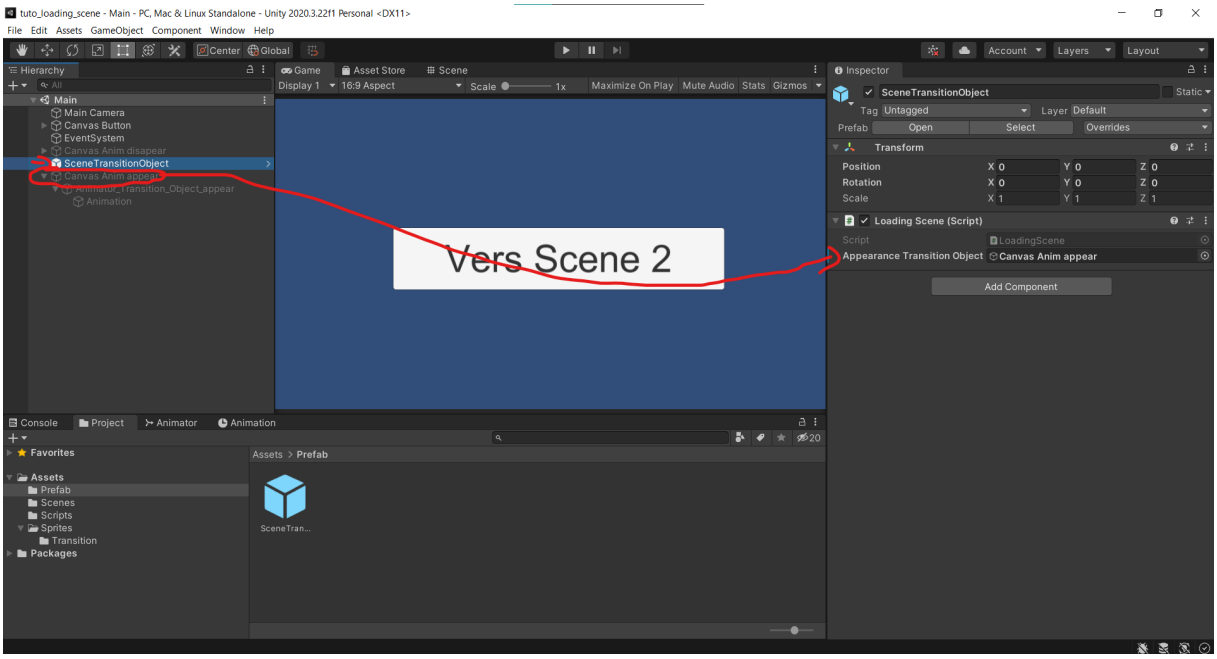
```
void Update ()
{
    // Active l'animation d'apparition
    if (fromLoadingScene && appearanceTransitionObject!=null)
    {
        if (!appearanceTransitionObject.activeInHierarchy)
        {
            appearanceTransitionObject.SetActive(true);
        }
    }
}

// Dit ce qu'il faut faire à la fin de l'animation d'apparition
public void Appearance ()
{
    fromLoadingScene = false;
    if (appearanceTransitionObject!=null)
    {
        appearanceTransitionObject.SetActive(false);
    }

}

// Enregistre le fait que l'on va avoir besoin d'une animation d'apparition à la prochaine scène
public void Disappearance ()
{
    fromLoadingScene = true;
}
```

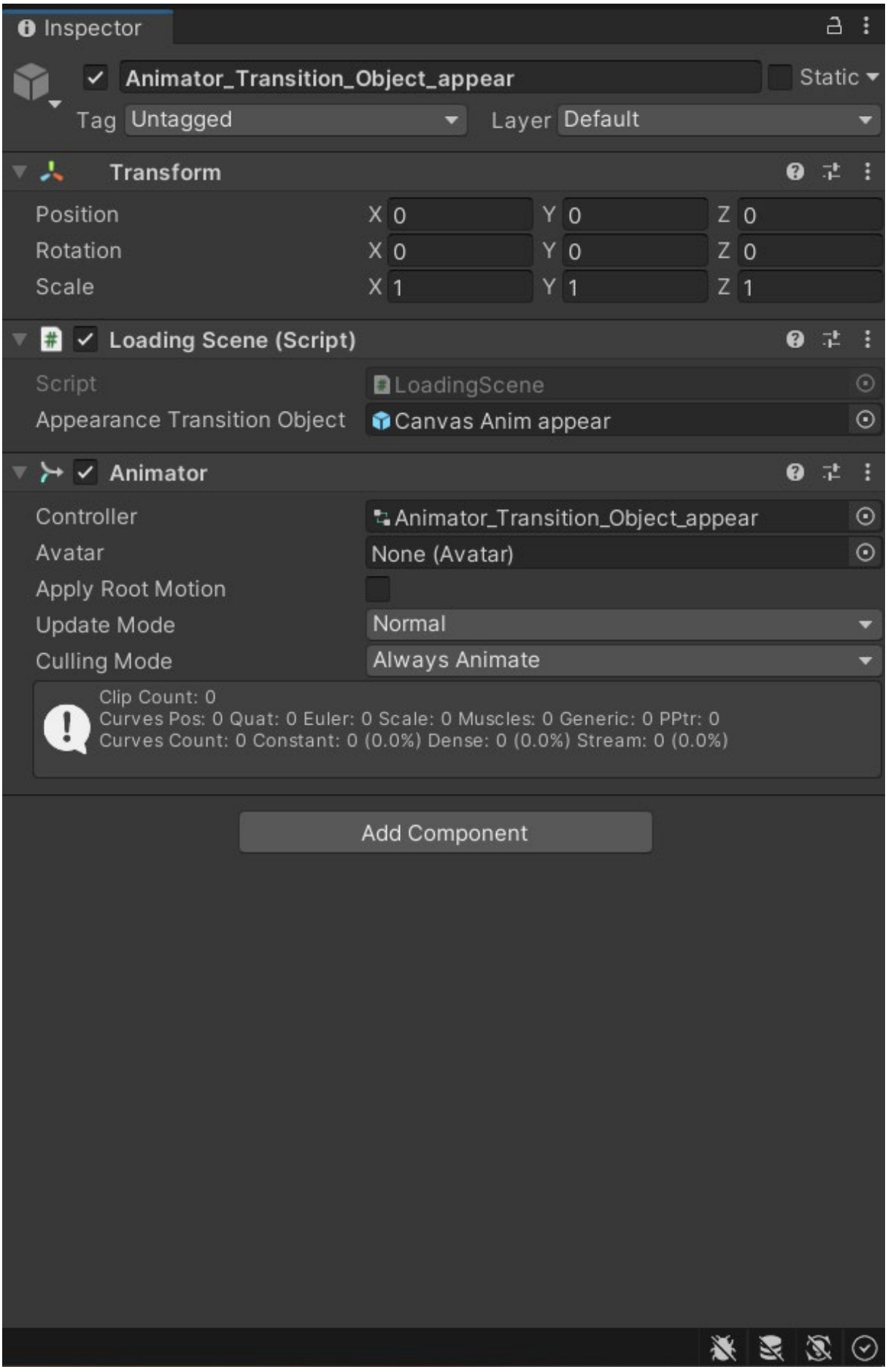
```
// Charge la scene dont on lui donne le nom
public void NextScene(string nameScene)
{
    SceneManager.LoadScene(nameScene);
    Disappearance ();
}
```

16- Dans l'event de la nouvelle animation, chercher dans le menu déroulant « Appearance () »

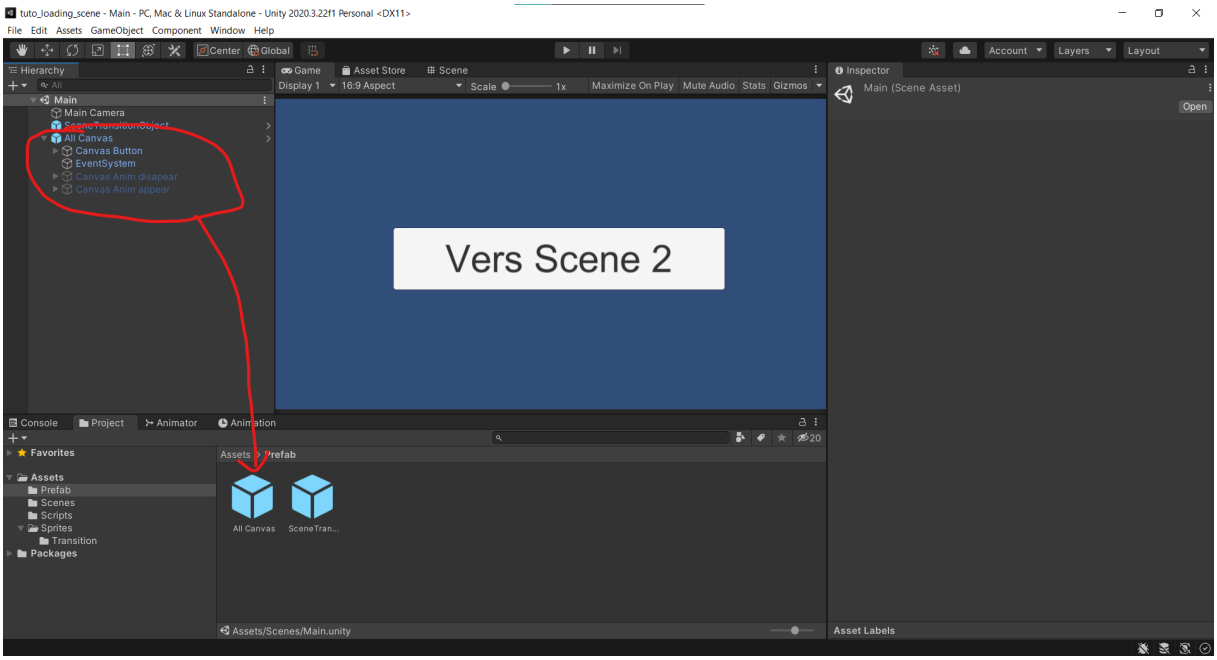


17- Sélectionner « SceneTransitionObject », puis glisser le canvas « Canvas Anim appear » dans la case « Appearance Transition Object ». Répéter la même opération pour l'objet « Animator_Transition_Object_appear »

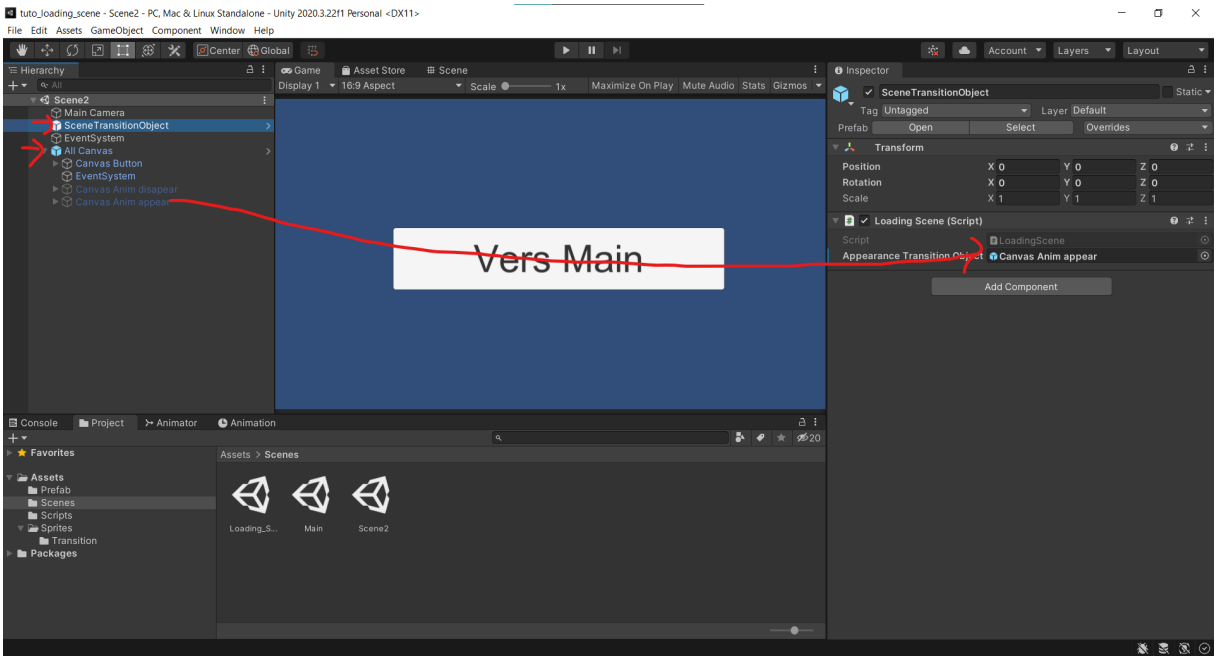



Inspector
Animator_Transition_Object_appear
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Loading Scene (Script)
Script LoadingScene
Appearance Transition Object Canvas Anim appear
Animator
Controller Animator_Transition_Object_appear
Avatar None (Avatar)
Apply Root Motion
Update Mode Normal
Culling Mode Always Animate
Clip Count: 0
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 0 (0.0%)
Add Component

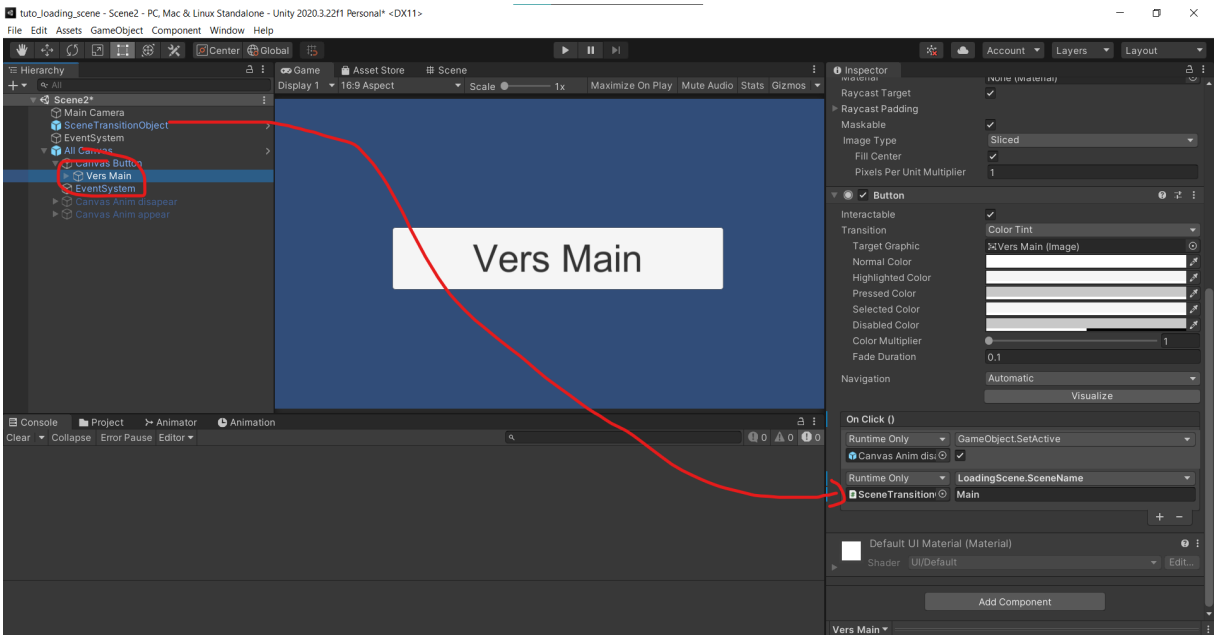
18- Mettre les trois canvas et l'event system dans un objet vide, puis en faire un prefab



19- Dans la scène d'arrivée, mettre les 2 prefab et répéter l'étape 17 (changer également l'apparence et le texte du bouton si besoin)



20- Dans le bouton de la scène 2, glisser l'objet « SceneTransitionObject » dans la 2ème ligne du « On Click () », puis chercher « LoadingScene.SceneName » dans le menu déroulant et y écrire le nom d'une autre scène (la scène main dans mon cas)



21- C'est finit ! (remarque : la scène étant vide, le chargement est très rapide. L'écran de chargement a été coloré en rouge dans l'exemple du tuto pour qu'on voit la transition).

Autre remarque : On n'a que la transition de disparition dans « WTRB ? », j'ai ajouté le reste pour rendre le tout plus fluide.