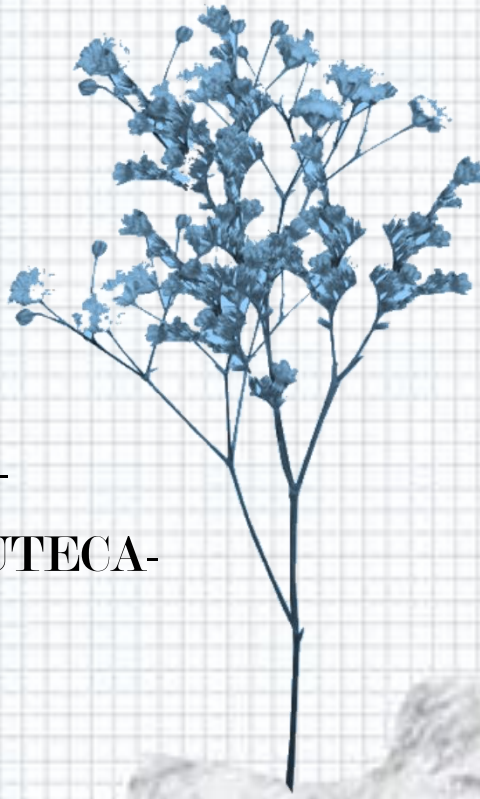
Realidad Botánica

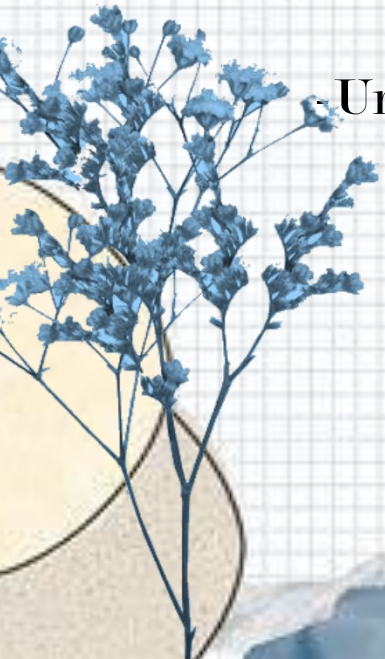
-Unity 3D-



-Nadia González-

-Realidad Aumentada y Virtual-

-Universidad Tecnológica Americana UTECA-



índice

Contexto

Es una aplicación de realidad virtual que te ayudara a visualizar un catalogo de plantas para ver como quedan en tu casa.

Para esta aplicación se utilizó la herramienta Google ArCore además de sus demás herramientas integradas de XR Origins y XR Session.

También se utilizan los Canvas, Raw Image y el Text Mesh Pro para las diferentes interfaces.

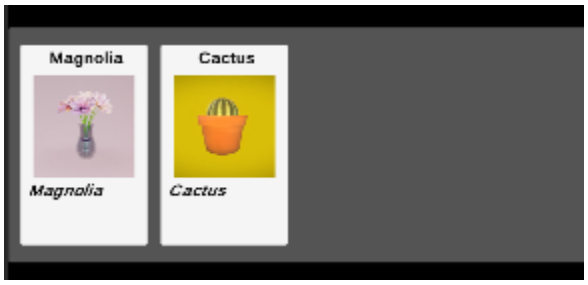
Esta herramienta proporciona un sistema de rotación, movimiento y fijación de los modelos 3D que se desean visualizar.

Para controlar la aplicación simplemente debes darle clic a los botones correspondientes:

 Este Botón es para abrir el menú de ítems.

 Este botón es para cerrar la aplicación.

 Este botón es para cerrar el menú de ítems.



Interfaz que se puede mover hacia la derecha para visualizar el número de ítems que uno guste, además de seleccionar el modelo deseado.

 Este botón es para fijar el modelo en el lugar deseado del escenario.

Nota: Al darle clic al modelo ya fijado, se podrá mover nuevamente o eliminar.

 Este botón es para eliminar el modelo de la escena.

La rotación se lleva a cabo usando dos dedos, mientras uno se queda estático otro realiza el movimiento para rotar el modelo.

En pocas palabras esta aplicación hará lo siguiente:

- ✓ Te otorgara una interfaz con la cual podrás cerrar la aplicación.
- ✓ Interfaz para abrir un menú de ítems
- ✓ Explorar el menú de ítems.
- ✓ Cerrar el menú de ítems
- ✓ Colocar, rotar, mover y fijar modelos 3d en tu escena de realidad aumentada.
- ✓ Y podrás eliminar los modelos creados de esta misma escena.

Consideraciones:

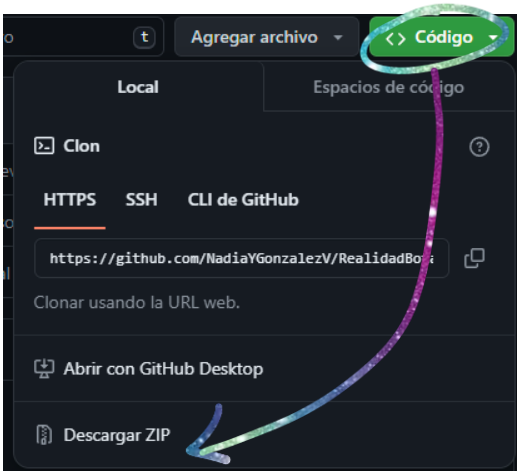
- ✓ Este Proyecto trabaja con ArCore, así que deberás instalarlo en el unity.
- ✓ Revisa la compatibilidad de tu celular con los parámetros escritos en Unity, para ver si soportara el programa.
- ✓ Tener instalado Text Mesh Pro
- ✓ Recuerda agregar los diferentes elementos correspondientes en cada una de las casillas que proporcionan los scripts, además de los diferentes eventos y el Tag.
- ✓ Todos los modelos fueron descargados de SketchFab.

Implementación:

Deberás descargar todo el proyecto para que funcione correctamente en Unity desde GitHub, el repositorio con el sistema está en el siguiente enlace:

https://github.com/NadiaYGonzalezV/RealidadBotanica.git

Una vez que te encuentres en el repositorio, le darás clic al botón de "Código" y posterior mente al de "Descargar ZIP"



Una vez descargado el archivo lo ubicas en tu dispositivo (normalmente en la carpeta de descargas)

✓ En la carpeta donde se encuentre el ZIP descargado, crearas una nueva carpeta para mover el Zip dentro de ella (esto para tener orden, opcional).

✓ Una vez dentro de la nueva carpeta, extraerás los archivos de la carpeta dando clic derecho y en “Extraer aquí”

✓ Listo, tendrás acceso a todos y cada uno de los archivos del proyecto, junto al Unity package file.

Si quieres probar la aplicación los APK estarán en la carpeta “BuildApk” se la última versión (actualmente 2024) “Realidad Botanica_07”.

Contenido:

Este documento te ayudara a personalizar la aplicación a tu gusto, para que se visualicen los ítems que gustes en tu escena.

En este sistema se proporcionan varios elementos:

- ✓ 6 scripts que serán los responsables de que nuestra aplicación funcione a la perfección:

Los siguientes scripts se encargan de la buena funcionalidad de la interfaz y de los Canvas:

- ✓ UIManager
- ✓ GameManager
- ✓ ButtonItemMnager

Los siguientes scripts ayudan a la asignación de los elementos que se mostrarán en pantalla, así como el movimiento, rotación y fijación de los ítems deseados.

- ✓ Ítem
- ✓ DataManager
- ✓ ARInteractionManager

A continuación, se explicarán fragmentos de código, de algunos de los scripts:

```
[SerializeField] private Camera aRCamera; Cámara utilizada
para raycasting en AR.
private GameObject aRPointer; Puntero AR usado para indicar la
posición en el mundo AR.
private GameObject item3DModel; Modelo 3D que se va a
posicionar en el entorno AR.

public GameObject Item3DModel
 {
     set
     {
         item3DModel = value;

item3DModel.transform.position=aRPointer.transform.position;
         item3DModel.transform.parent = aRPointer.transform;
         isInitialPosition = true;
      }
 }
(…)
```

Establece el modelo 3D y lo coloca en la posición del puntero AR.

```
void Start()
 {
     transform.GetChild(0).GetComponent<TMP_Text>().text =
itemName;
     transform.GetChild(1).GetComponent<RawImage>().texture =
itemimage.texture;
```

```
     transform.GetChild(2).GetComponent<TMP_Text>().text =
itemDescription;

     var button = GetComponent<Button>();

button.onClick.AddListener(GameManager.instance.ARPosition);
     button.onClick.AddListener(Create3DModel);

     interactionManager =
FindObjectOfType<ARInteractionManager>();
 } (…)
```

Este script gestiona un botón en la UI que, al ser presionado, muestra la información del ítem (nombre, descripción, imagen) y crea una instancia de su modelo 3D en el entorno de AR, permitiendo al usuario interactuar con el modelo a través de ARInteractionManager.

```
[SerializeField] private List<Item> items = new List<Item>();
[SerializeField] private GameObject buttonContainer;
[SerializeField] private ButtonItemMnager itemButtonManager;

private void Start()
{
    GameManager.instance.OnitemsMenu += CreateButtons;
}

private void CreateButtons()
{
    foreach (var item in items)
    {
        ButtonItemMnager buttonItem;
        buttonItem = Instantiate(itemButtonManager
(…)
```

Este script DataManager gestiona la creación dinámica de botones en la UI para cada ítem en una lista. Cuando se dispara el evento OnitemsMenu del GameManager, el método CreateButtons es llamado, el cual instancia botones basados en un prefab ButtonItemMnager y los configura con la información del ítem correspondiente. Esto permite una fácil y dinámica actualización de la UI basada en los ítems disponibles, y garantiza que cada botón está correctamente configurado para interactuar con su ítem correspondiente en el entorno de AR.

```
[CreateAssetMenu]
public class Item : ScriptableObject
{
    public string ItemName;
    public Sprite ItemImage;
    public string ItemDescription;
    public GameObject Item3DModel;

}
```

[**CreateAssetMenu**] Este atributo permite crear instancias de Item desde el menú de Unity. Facilita la creación de activos ScriptableObject desde el editor sin necesidad de escribir código adicional para su instanciación.

public string ItemName : Almacena el nombre del ítem.

public Sprite ItemImage : Almacena la imagen del ítem que se puede utilizar en la interfaz de usuario.

public string Item: Almacena la descripción del ítem.

public GameObject Item3DModel: Almacena el modelo 3D del ítem que se puede utilizar en el entorno de juego o AR.

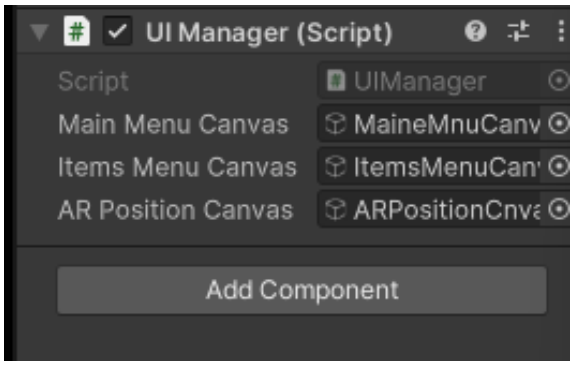
El script Item define un conjunto de datos que describe un ítem. Al usar ScriptableObject, se puede crear, almacenar y gestionar datos de ítems en Unity de una manera muy eficiente. Los ScriptableObject se utilizan comúnmente en Unity para almacenar datos que pueden ser fácilmente reutilizados y compartidos entre diferentes objetos y escenas.

¿Como modificarlo en unity?

Este proyecto se puede modificar y personalizar de distintas maneras:

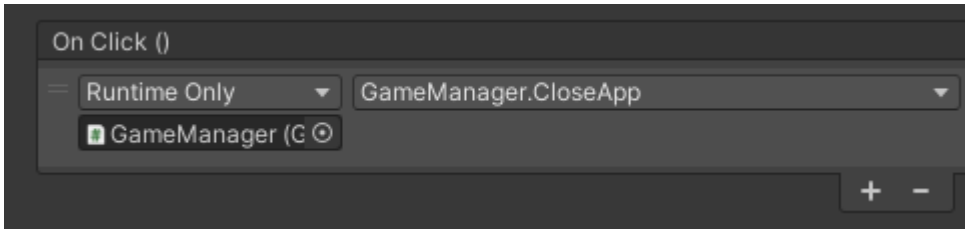
❖ UI MANAGER

Aquí se tendrán que asignar todos los canvas que aparecerán en nuestra aplicación, así que debemos otorgarle el script correspondiente.

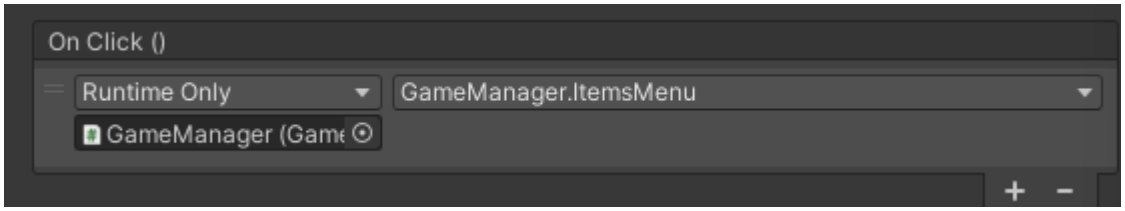


Main Menu Canvas se deberán agregar dos botones y sus respectivos eventos.

Cerrar aplicación:

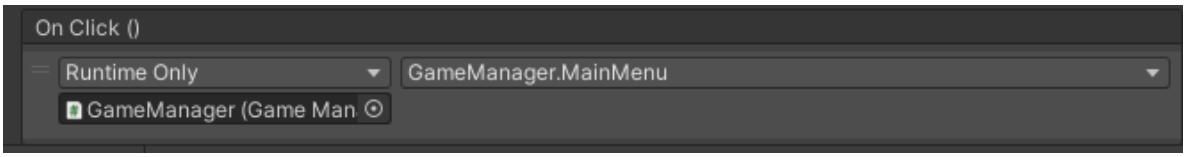


Ver ítems:



❖ ItemsMenuCanva:

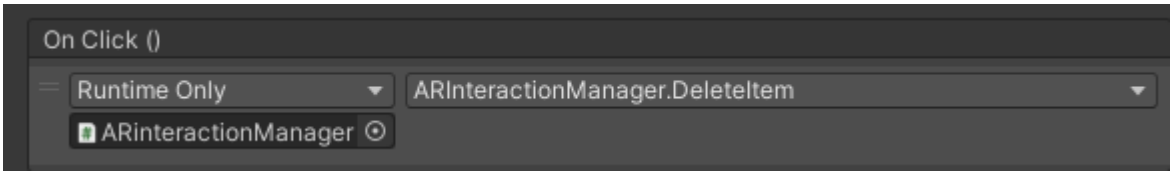
ItemsOpen:



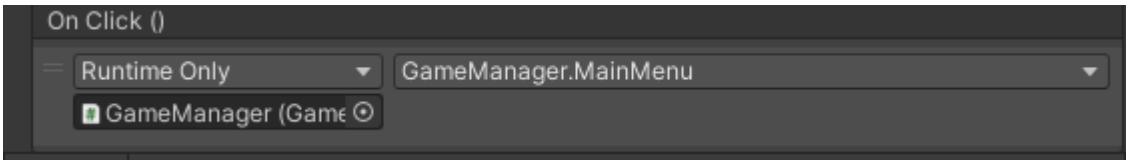
Además de agregar un ScrollView el cual se encargará de mostrar en orden nuestros ítems.

❖ ARPositionCanvas:

Botón Delete:

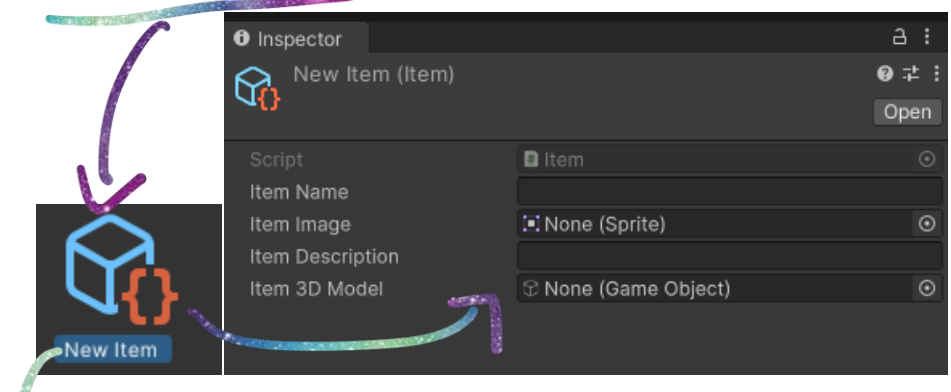


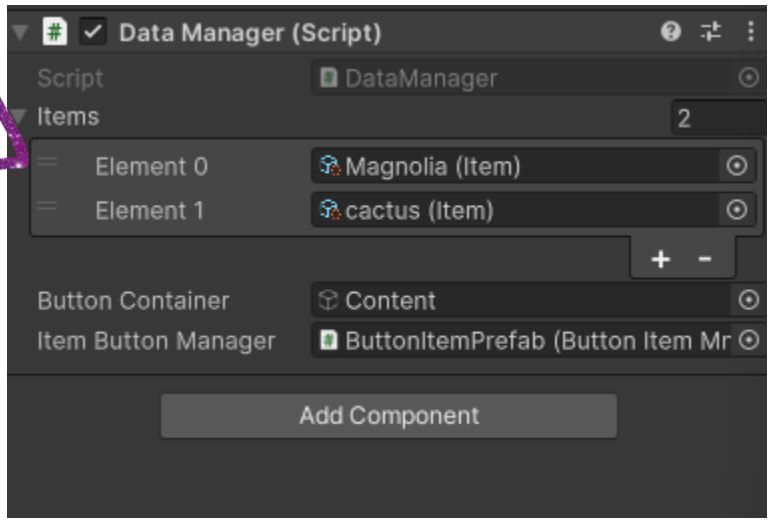
Botón de Aprove:



❖ Data manager

Aquí agregaremos el “content” del Scroll View además de nuestros ítems creados gracias a nuestra variable que aparece en el script “Data Manager” y los cuales para tener orden estarán creados en la carpeta “ScriptibleObject”

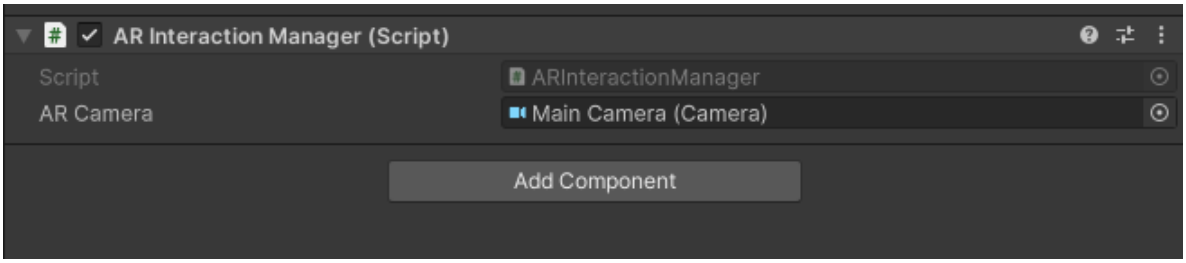




Aquí podrás agregar todos y cada uno de los prefabs de tus modelos que quieras mostrar.

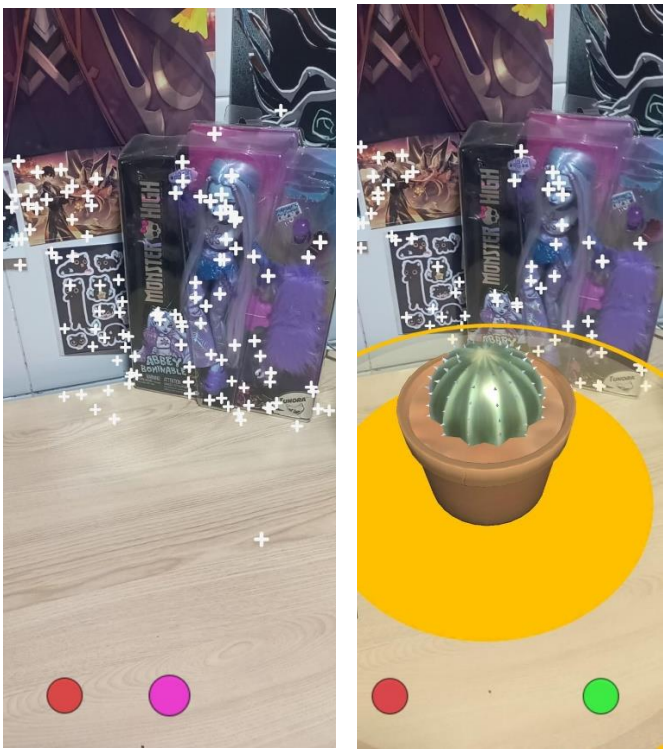
❖ ARInteractionManager:

Asignar la cámara que se utilizara la cual viene en XR Origin.



Por último, a los modelos que están convertidos en prefabs, se deberán colocar como hijos de un objeto vacío y a este mismo agregarles el Tag de “Ítem” y un “Box Collider”.

Muestra de la aplicación.



Espero que esta documentación te haya sido de ayuda y que puedas disfrutar al máximo la aplicación de “Realidad Botánica”

Contacto: yologonzalezv@gmail.com