




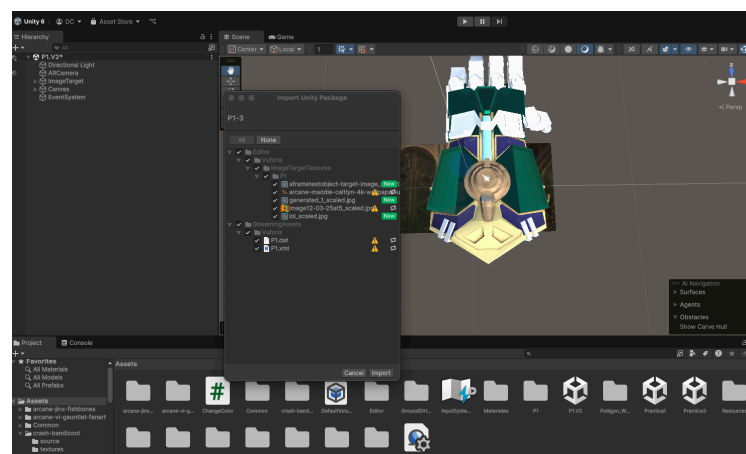


Práctica 4

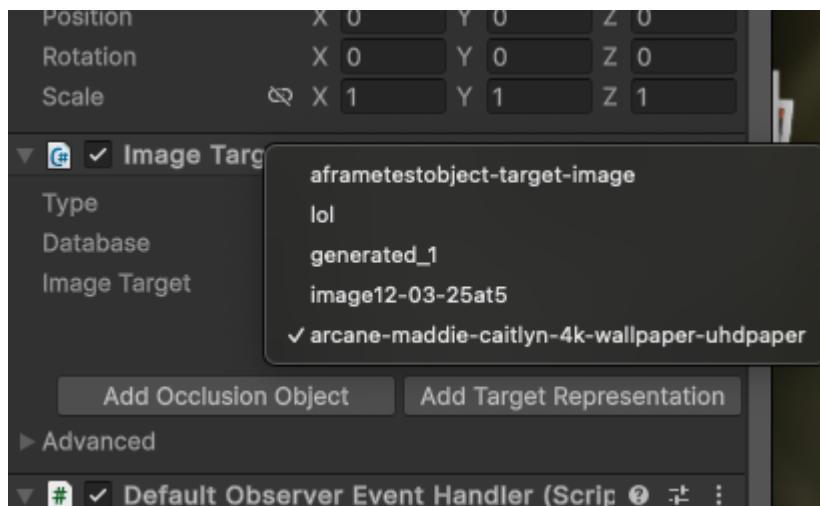
Para empezar la practica debemos tener nuestra escena configurada , asi como nuestra database de vuforia lista con nuestras 5 imagenes:

P1 Edit Name						
Type: Device						
Targets (5)						
Add Target Download Database (All)						
<input type="checkbox"/>	Image	Target Name	Type	Rating ^①	Status [▼]	Date Modified
<input type="checkbox"/>		aframetestobject-target-image	Image	★★★★★	Active	Mar 20, 2025
<input type="checkbox"/>		lol	Image	★★★★★	Active	Mar 20, 2025
<input type="checkbox"/>		generated_1	Image	★★★★★	Active	Mar 20, 2025
<input type="checkbox"/>		image12-03-25at5	Image	★★★★★	Active	Mar 12, 2025
<input type="checkbox"/>		arcane-maddie-caitlyn-4k-wallpaper-uhdpaper	Image	★★★★☆	Active	Feb 26, 2025

5 imágenes que usaremos

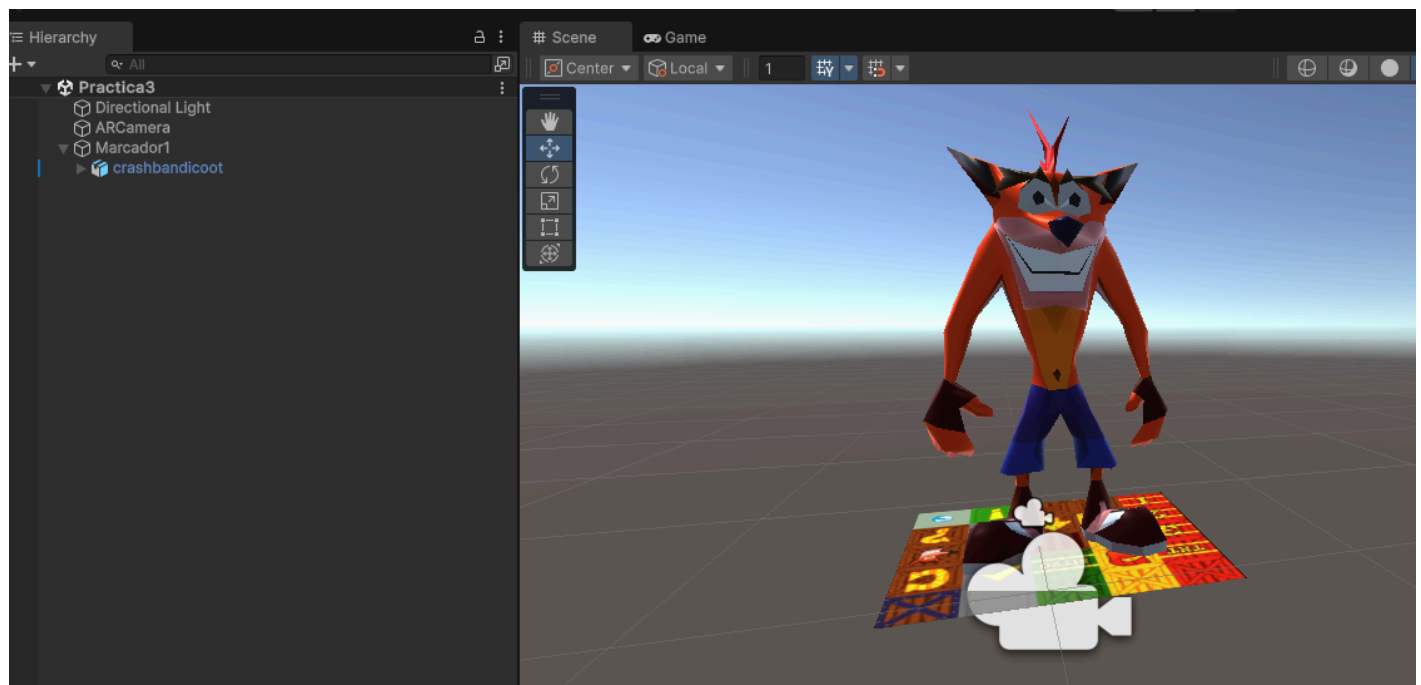


Importamos nuestra Database

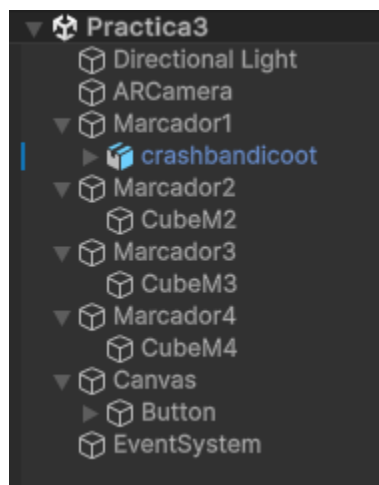
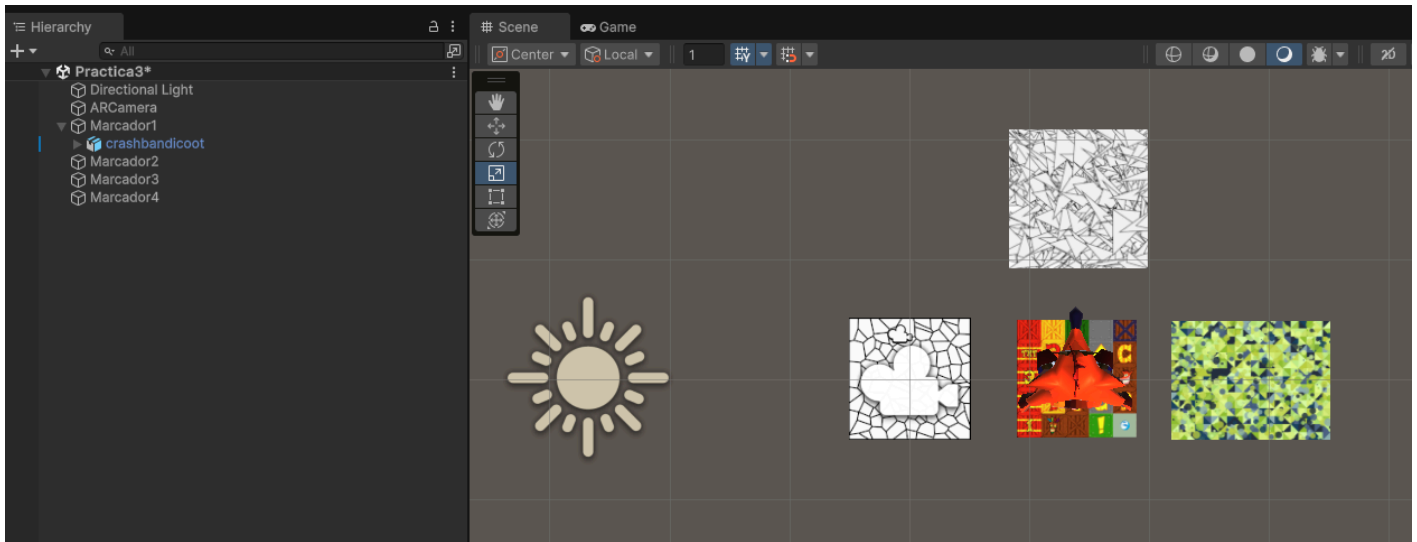


Una vez importada ya nos aparecen nuestras nuevas imagenes para el image target

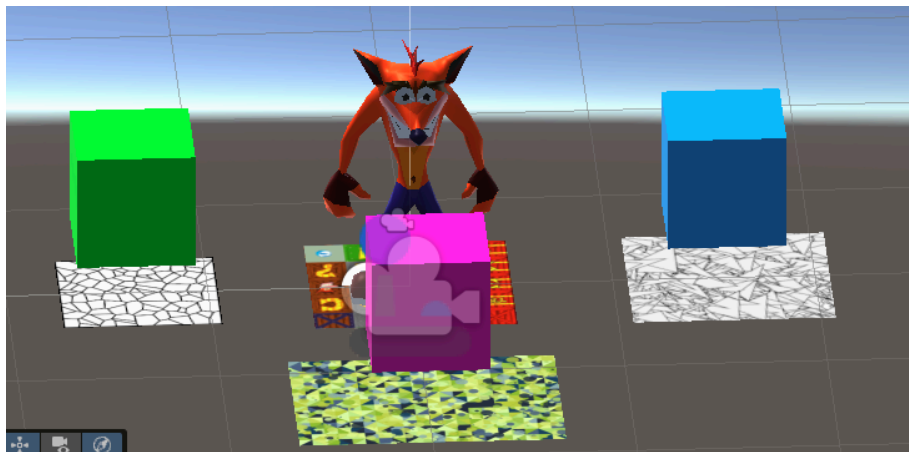
Para esta practica creamos una nueva escena , en la cuál vamos a tener un nuevo modelo , esta vez quise poner a crash , en ella vamos a ir creando nuestros marcadores .



Tenemos nuestros cuatro image targets donde ponemos al simagenes que habiamos seleccionado en nuestra base de datos de vuforia



Aparte de nuestros marcadores vamos a duplicar un cubo con el cuál vamos a tener un punto de referencia hacia donde nuestro modelo inicial se va a desplazar.



Vamos ahora a crear nuestro script para definir el movimiento , el script se llamara Move,

Este script hace que un objeto en este caso va a ser `model` se mueva entre distintos marcadores de Vuforia.

- **`moveToNextMarker()`** se encarga de empezar el movimiento si el objeto no está en movimiento.

```
0 references
public void moveToNextMarker()
{
    if (!isMoving){
        StartCoroutine(MoveModel());
    }
}
```

- **`MoveModel()`** toma el siguiente marcador detectado y mueve el objeto hacia él poco a poco.

```
1 reference
private IEnumerator MoveModel() //para ver el movimiento y su trayectoria
{
    isMoving = true;
    ObserverBehaviour target = GetNextDetectedTarget();
    if(target == null){
        isMoving=false;
        yield break;
    }

    Vector3 startPosition=model.transform.position;
    Vector3 endPosition=target.transform.position;

    //trazamos trayectoria
    float journey =0;

    while (journey <= 1f)
    {
        journey+=Time.deltaTime * speed;
        model.transform.position=Vector3.Lerp(startPosition,endPosition,journey); //journey es la velocidad de traslacion
        yield return null;
    }

    currentTarget=(currentTarget+1)%ImageTargets.Length;
    isMoving=false;
}
}
```

- **GetNextDetectedTarget()** busca el próximo marcador visible y lo devuelve.

Básicamente, el objeto se va moviendo de un marcador a otro, pero solo si hay uno visible y si no está ya en movimiento.

```
1 reference
private ObserverBehaviour GetNextDetectedTarget()
{
    foreach(ObserverBehaviour target in ImageTargets)
    {
        if (target != null && (target.TargetStatus.Status==Status.TRACKED || target.TargetStatus.Status==Status.EXTENDED_TRACKED))
        {
            return target;
        }
    }
    return null;
}
```