

# Grafo de escena en Unity3D

## Sistema Solar

Depto. Sistemas Informáticos y Computación. UPV

# Objetivos de aprendizaje

Realizar transformaciones espaciales en Unity 3D

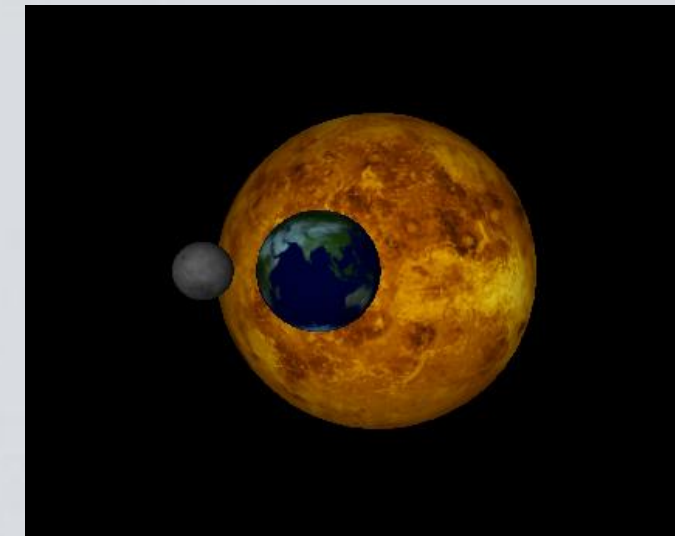
Conocer y emplear los grafos de escena en Unity 3D

Establecer jerarquías entre los objetos padres e hijos

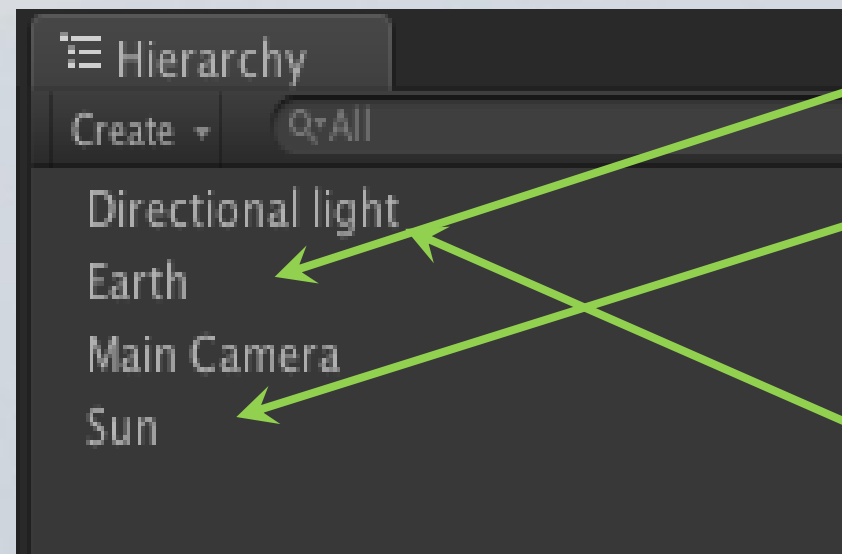
Tratar texturas

Crear y asignar materiales

# Unity. Sistema Solar (I)



Crear un proyecto nuevo, con una escena nueva llamada 'Sistema Solar'



Crear dos esferas denominadas 'Sun' y 'Earth'

Añadir una luz direccional

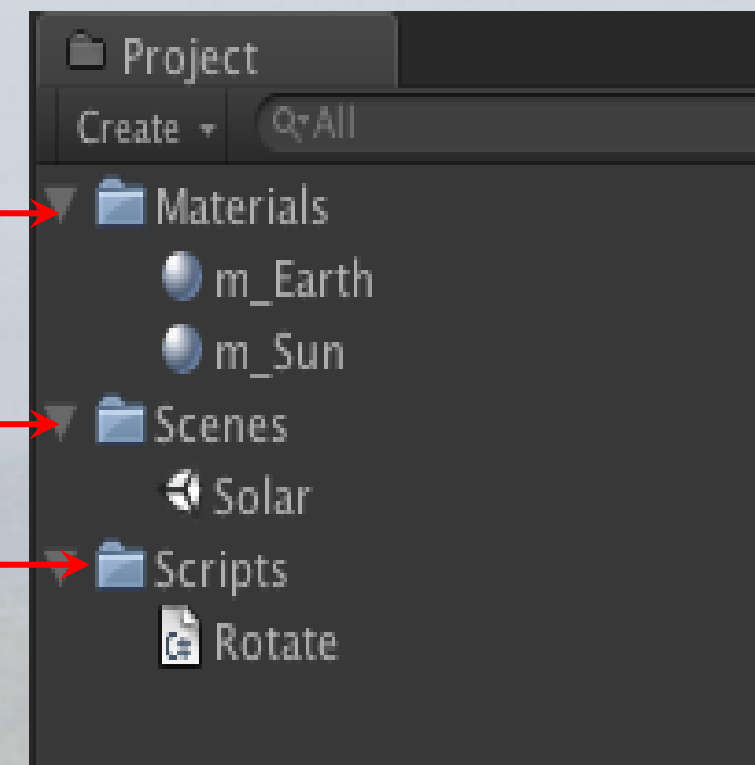
**Crea una carpeta**

*Materials* para guardar dentro todos los materiales empleados en la escena

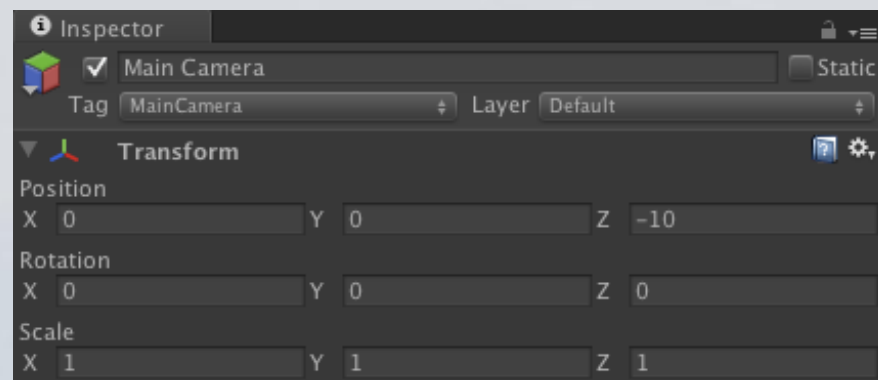
*Scenes* para almacenar allí un par de materiales.

Llámales *m\_Earth* y *m\_Sun*

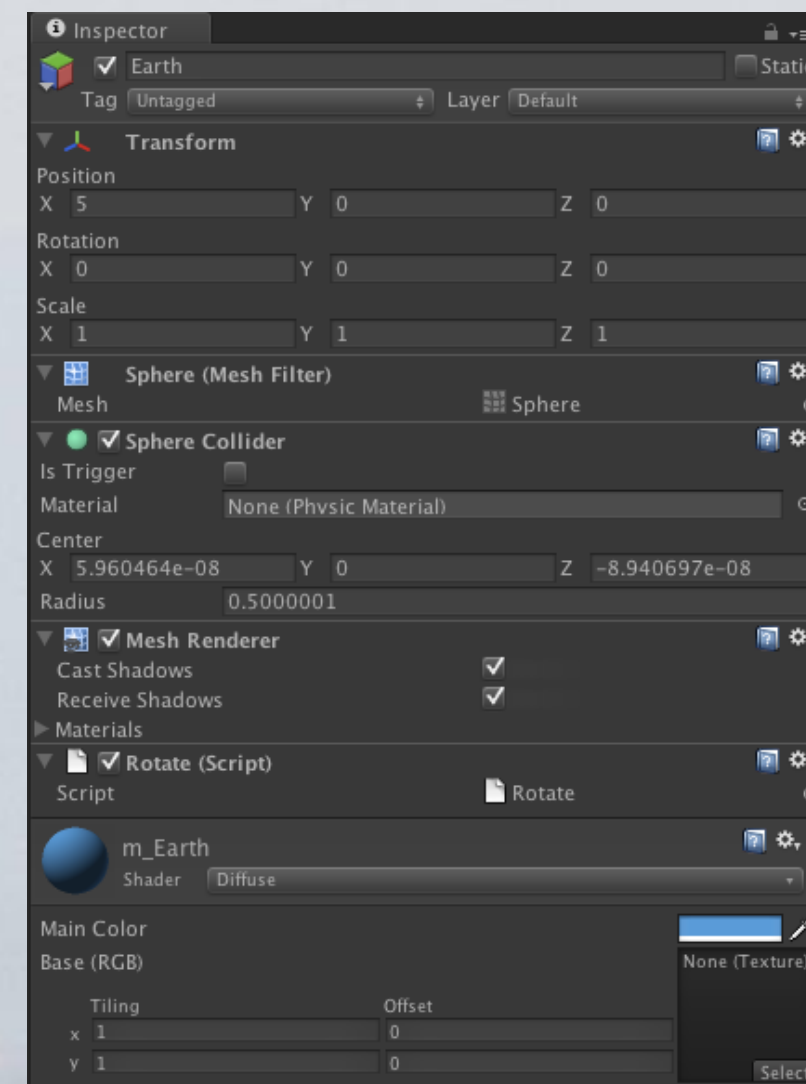
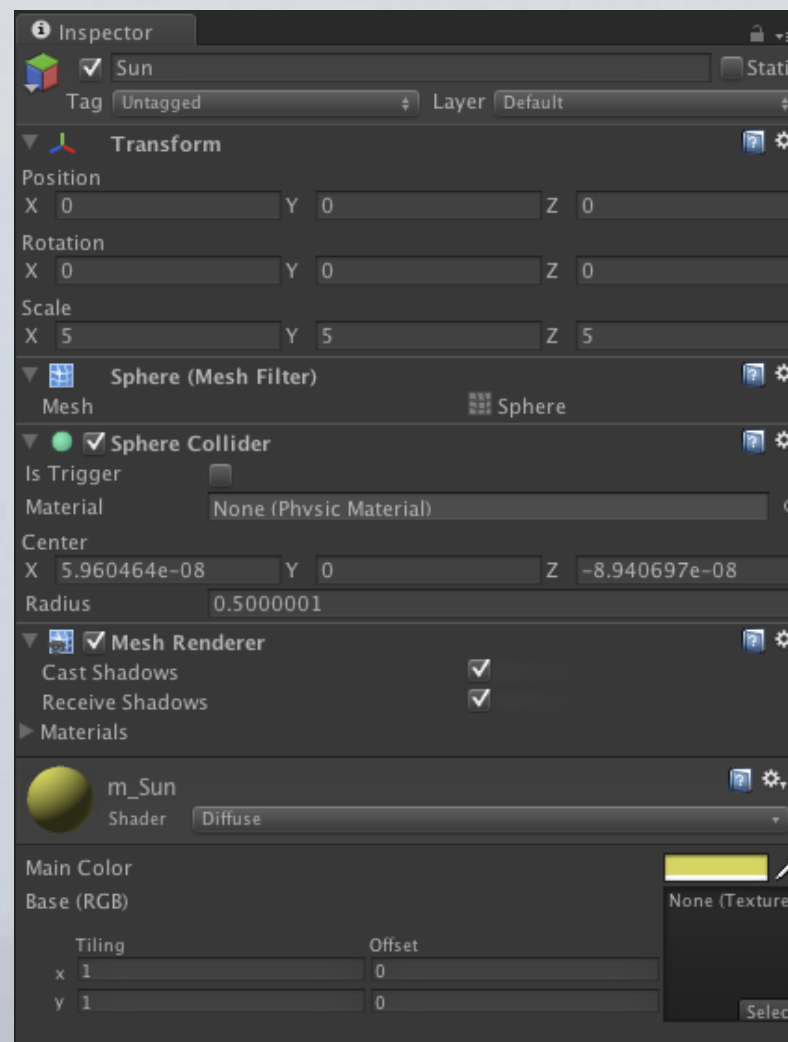
*Scripts* para guardar dentro los fragmentos de código



# Unity. Sistema Solar (II)



Modificar los parámetros de escena:



Sol: Position (0, 0, 0), Rotation (0, 0, 0), Scale (5, 5, 5)

Tierra: Position (6, 0, 0) , Rotation (0, 0, 0), Scale (1, 1, 1)



# Unity. Sistema Solar (III)

Arrastrar el material

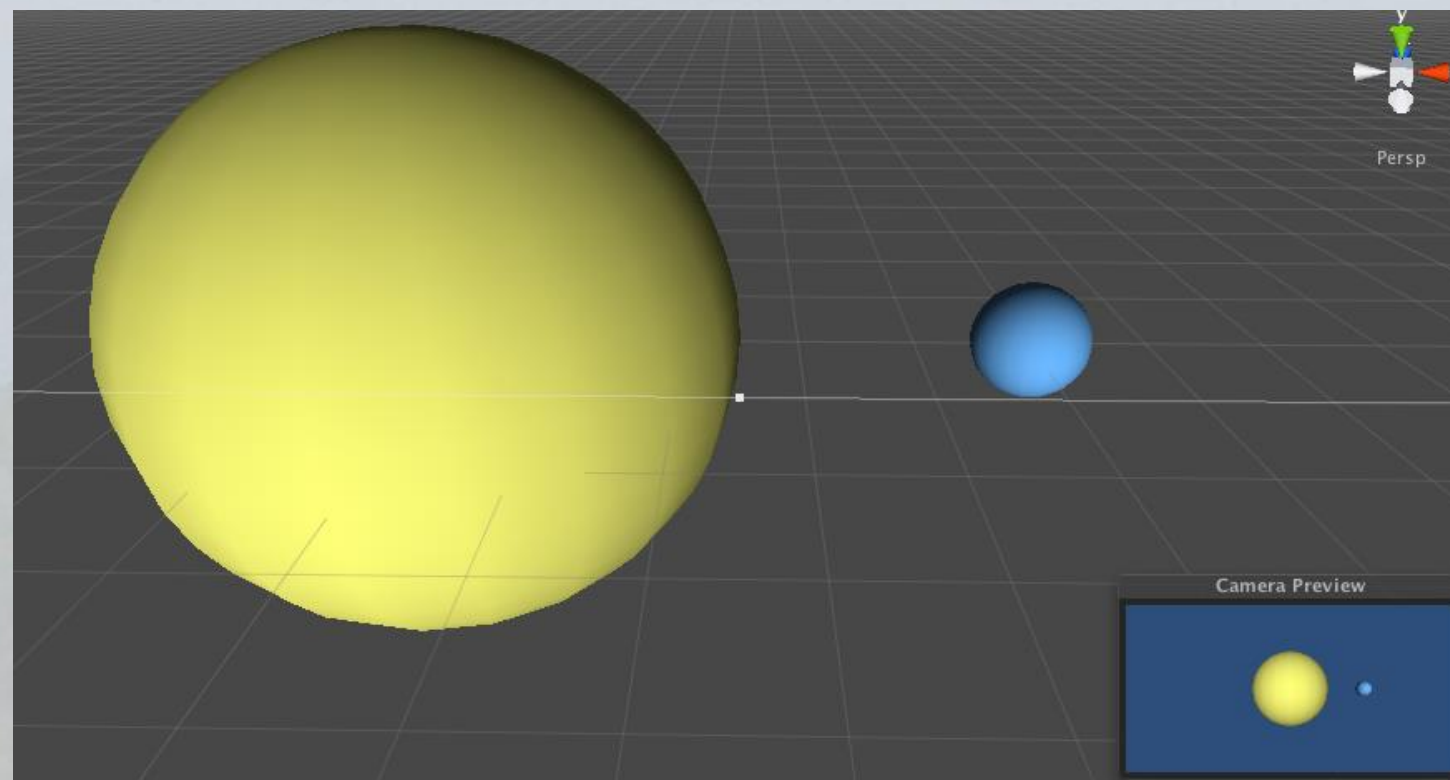
- *m\_Earth* sobre el objeto *Earth* y
- *m\_Sun* sobre el objeto *Sun*

Verificar el cambio de color de los objetos

Establece el fondo de la escena a negro (espacio exterior)

Para ello

- Selecciona la cámara principal
- Establece la propiedad “Clear Flags” a “Solid Color”
- Establece el color de fondo (background) a negro



# Unity. Sistema Solar (IV)

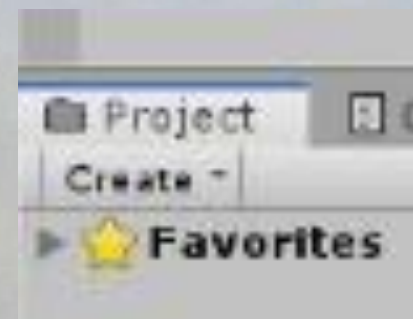
El guión *Rotate* permite aplicar una rotación alrededor del eje vertical Y

En la carpeta *Scripts* anterior, crear dentro un script en C# denominado *Rotate*

Para ello,

- Pulsar el RMB sobre ventana *Assets > Scripts* y seleccionar *Create > C# Script*
- O bien pulsar LMB sobre la opción *Create* debajo del nombre de la ventana *Project*

Añadir el código de la siguiente transparencia



```
public class Rotación : MonoBehaviour {  
    [SerializeField]  
    float    velRotacion;  
  
    // Start is called before the first frame update  
    void Start(){}  
  
    // Update is called once per frame  
    void Update(){  
        transform.Rotate      (Vector3.up * velRotacion);  
    }  
}
```

*Rotate* gira por defecto alrededor del sistema local

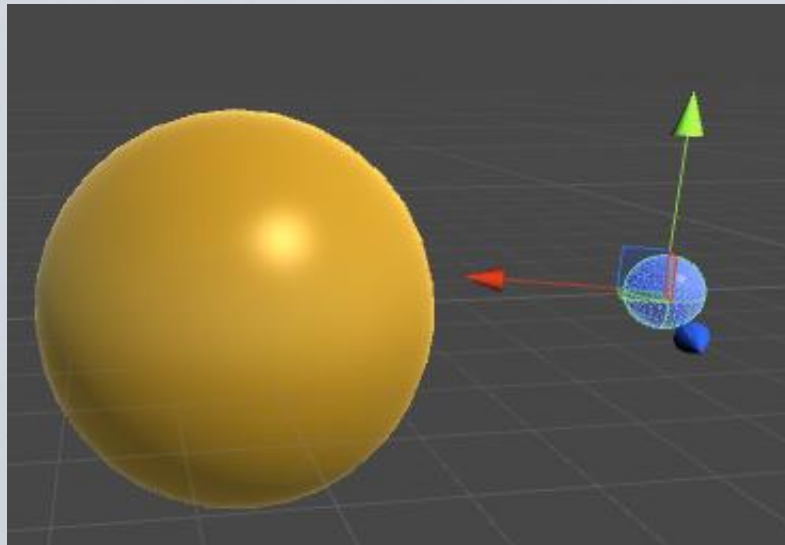
Asignar este script a 'Tierra' arrastrando el script sobre la esfera *Earth*

Darle al *play* y observar los resultados



# Unity. Sistema Solar (V)

La Tierra rota sobre sí misma



¿Cómo hacer que rote alrededor del Sol?

## Transform.RotateAround

SWITCH TO MANUAL

public void **RotateAround**([Vector3](#) point, [Vector3](#) axis, float angle);

### Parameters

### Description

Rotates the transform about `axis` passing through `point` in world coordinates by `angle` degrees.

This modifies both the position and the rotation of the transform.

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour {
    void Update() {
        transform.RotateAround(Vector3.zero, Vector3.up, 20 * Time.deltaTime);
    }
}
```



# Unity. Sistema Solar (VI)

## Rotación de la Tierra alrededor del Sol

Emplear método *RotateAround* del componente *Transform* para utilizar el sistema de coordenada global de la escena

Para que Tierra parezca que gira alrededor del Sol, hacer coincidir el centro del Sol con el centro de coordenadas universales (0,0,0) o *Vector3.zero* en Unity

```
public class Rotación : MonoBehaviour {
    [SerializeField]
    float    velRotacion,
            velOrbitacion;

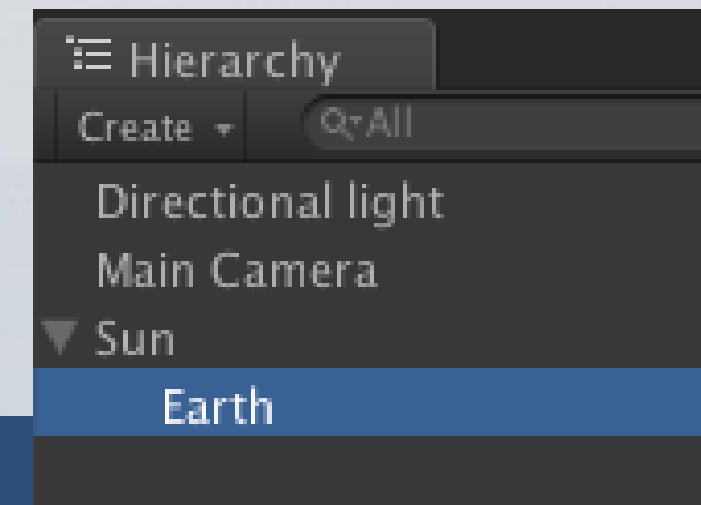
    // Update is called once per frame
    void Update(){
        transform.Rotate      (Vector3.up * velRotacion);
        transform.RotateAround (Vector3.zero, Vector3.up,
                               velOrbitacion * Time.deltaTime);
    }
}
```



# Unity. Sistema Solar (VII)

## Jerarquía

Arrastrar y soltar el objeto 'Tierra' encima del objeto 'Sol'. Ahora la 'Tierra' es hija del 'Sol'



A partir de ahora, cualquier transformación aplicada sobre el 'Sol' afectará a la 'Tierra'

Ahora la 'Tierra' rotará solidariamente alrededor del objeto 'Sol' sin necesidad de emplear *RotateAround*

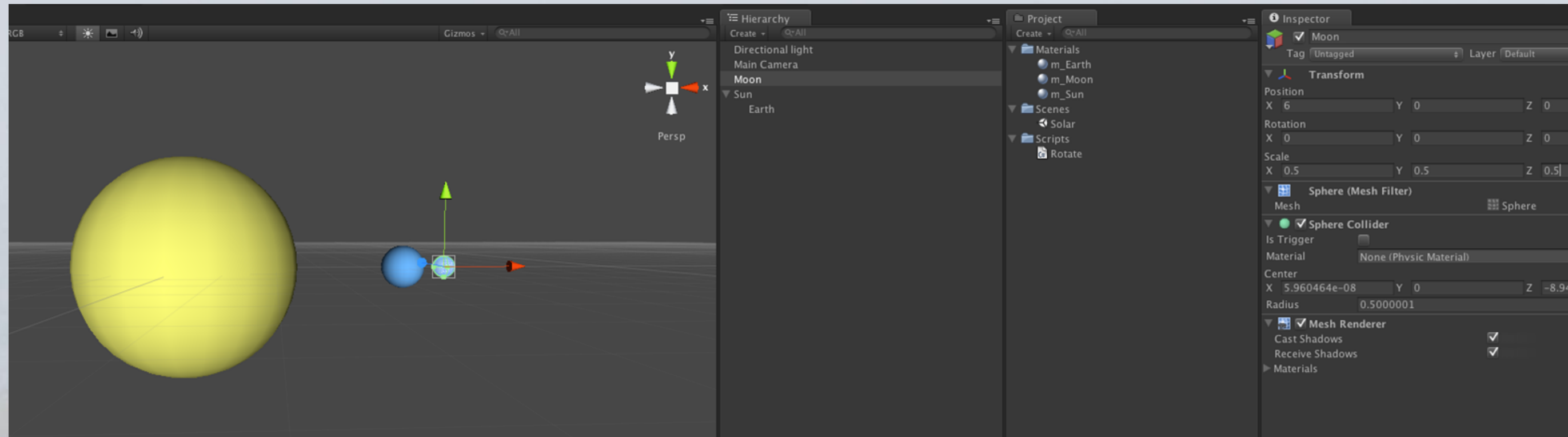
**Poner a cero la velocidad de orbitación**



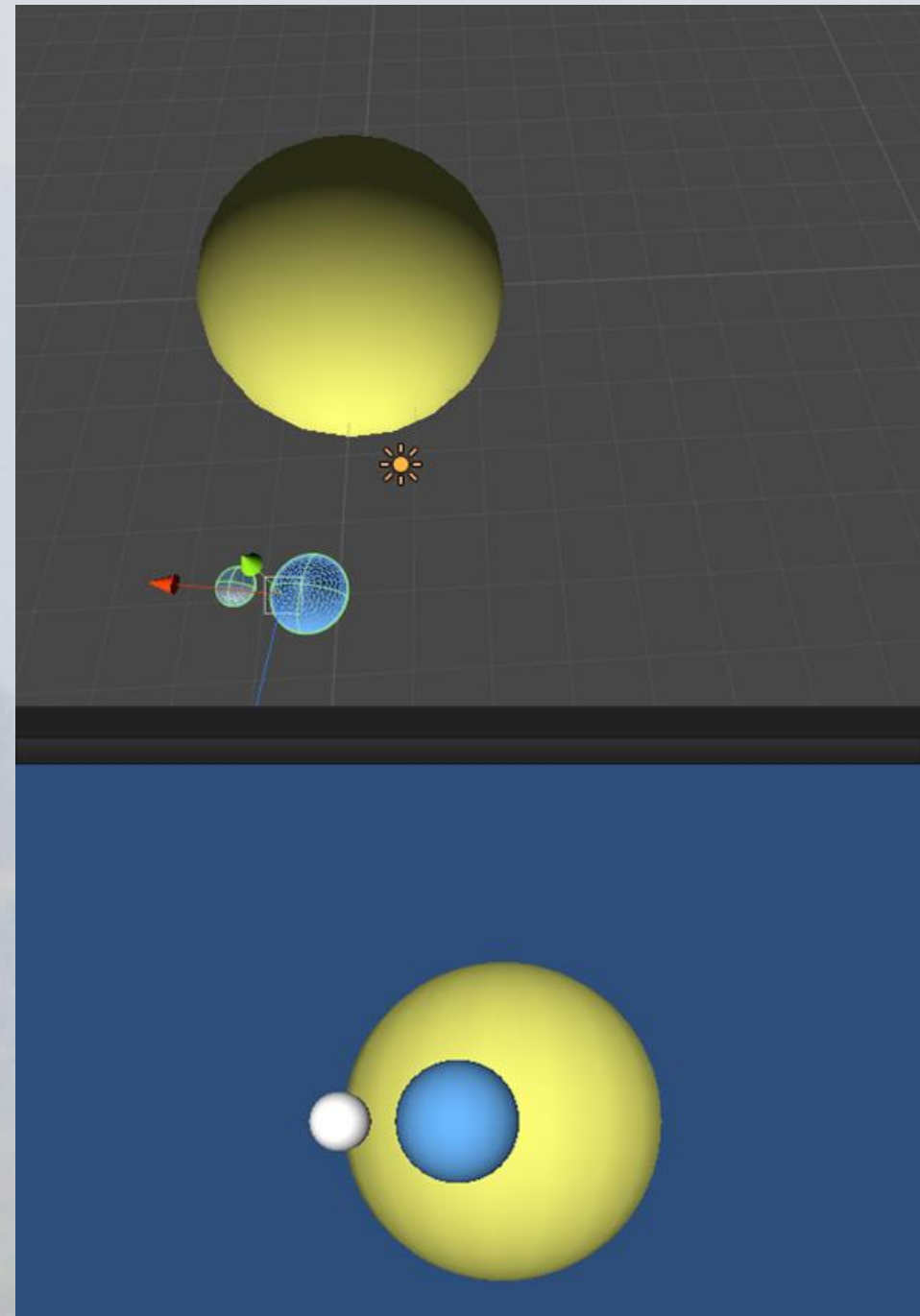
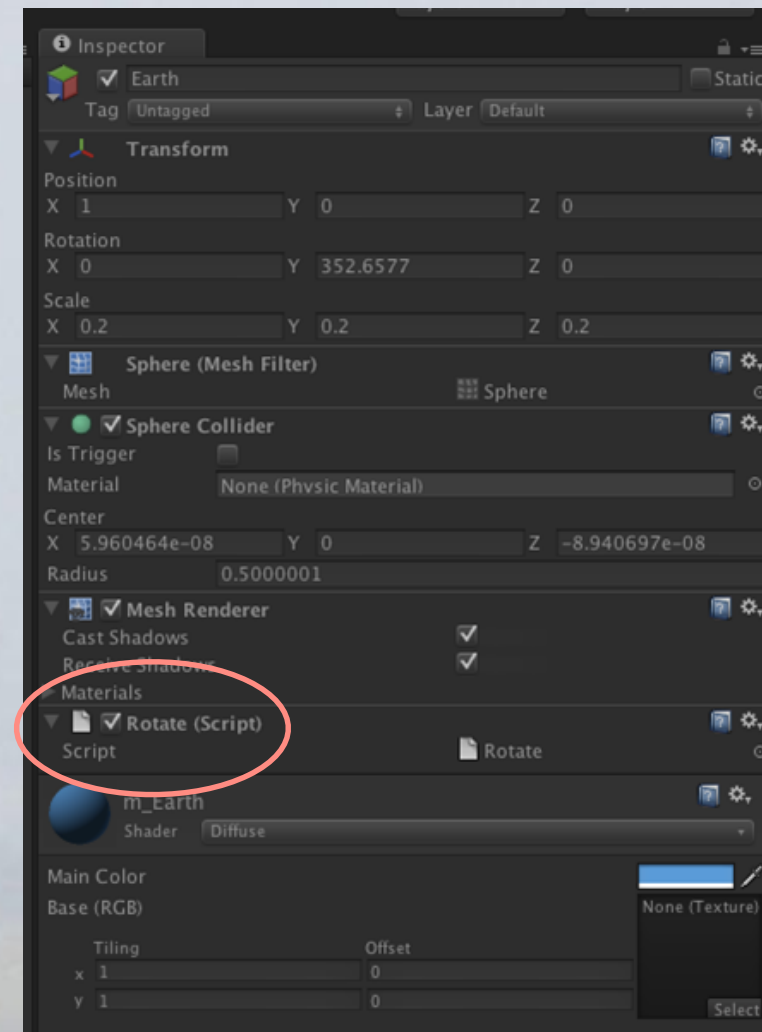
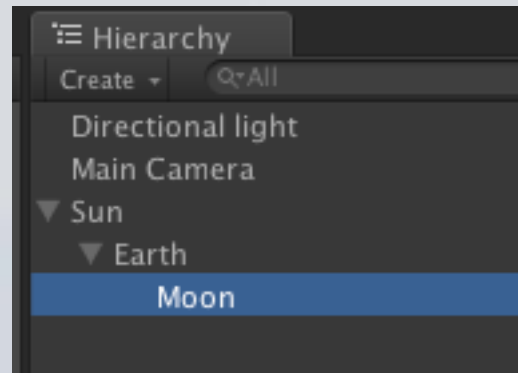
# Unity. Sistema Solar (VIII)

## Ejercicio

Añadir una Luna utilizando una esfera hija de Tierra y asignando el guión *Rotate* a 'Luna'



# Unity. Sistema Solar (IX)



# Unity. Sistema Solar (XII)

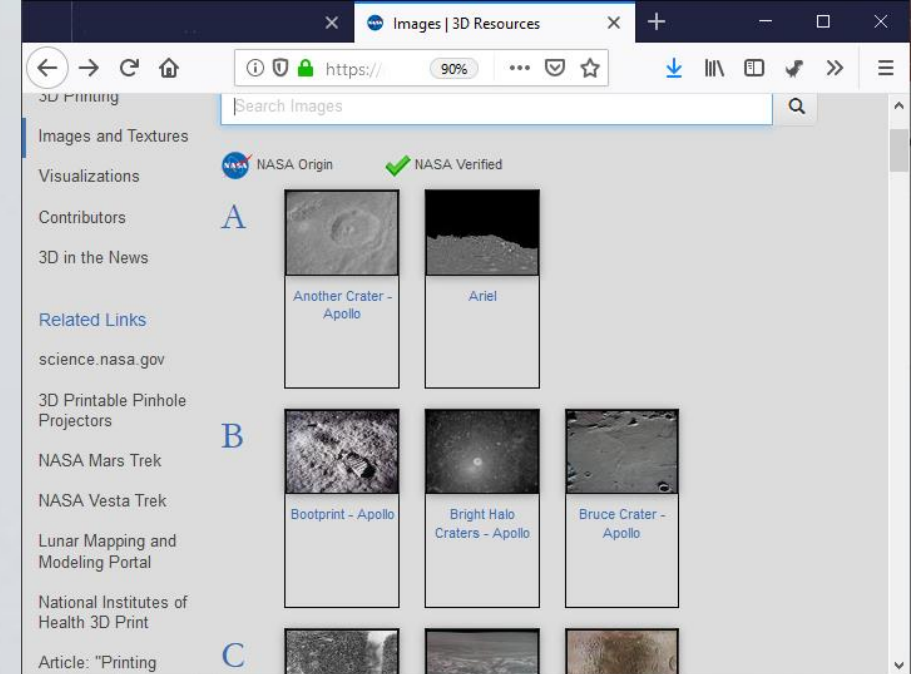
## Añadiendo texturas a los planetas

Conseguir texturas para planetas de, por ejemplo:

<https://nasa3d.arc.nasa.gov/images>

Descargarlas en la carpeta del proyecto llamada *Materials*

Puedes establecer algunos parámetros en el Inspector (p.e., tamaño máximo)



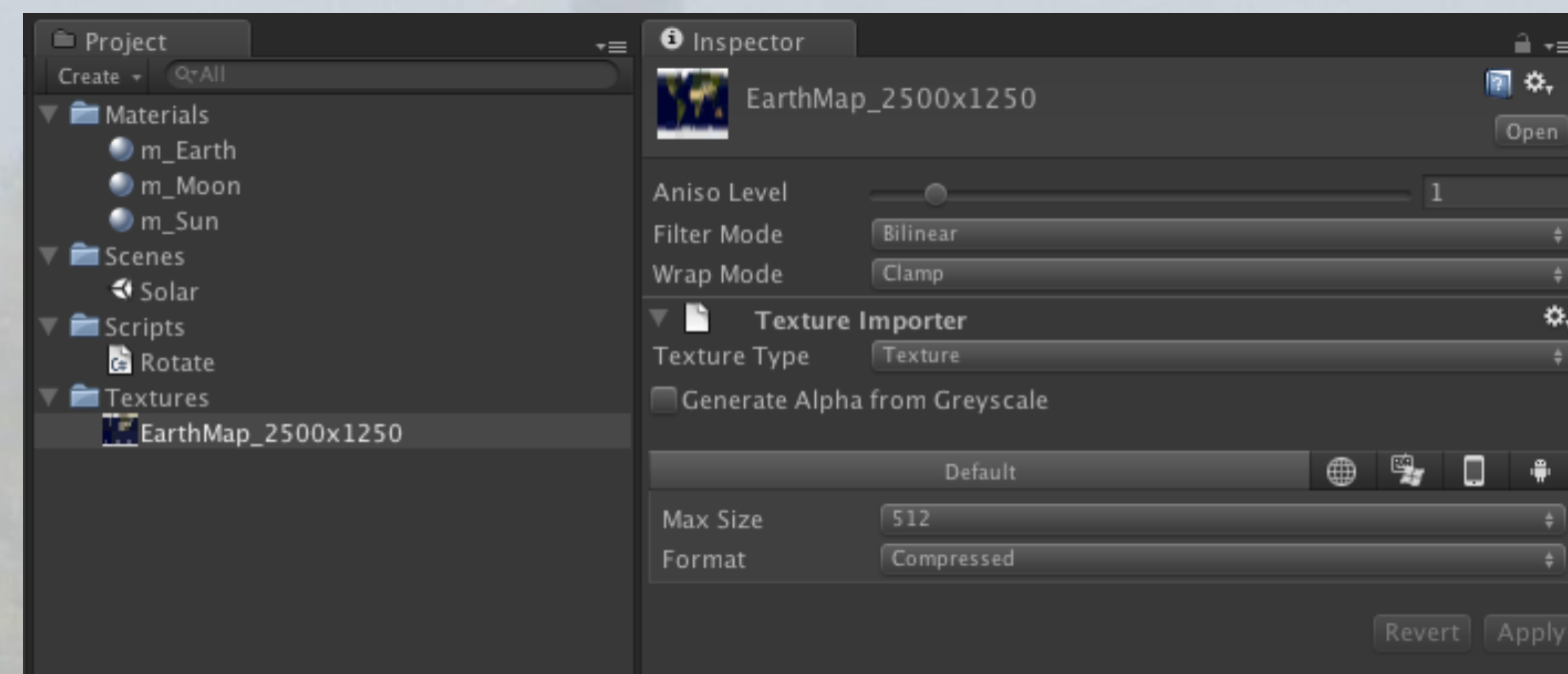


# Unity. Sistema Solar (XIII)

Si las descargas en otro sitio, deberás importarlas al proyecto

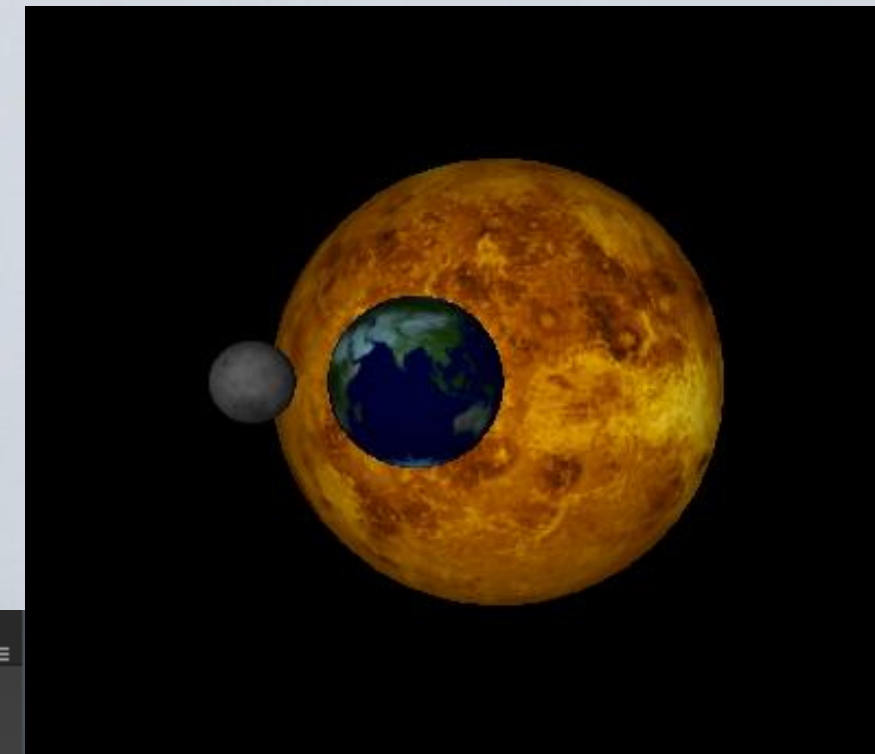
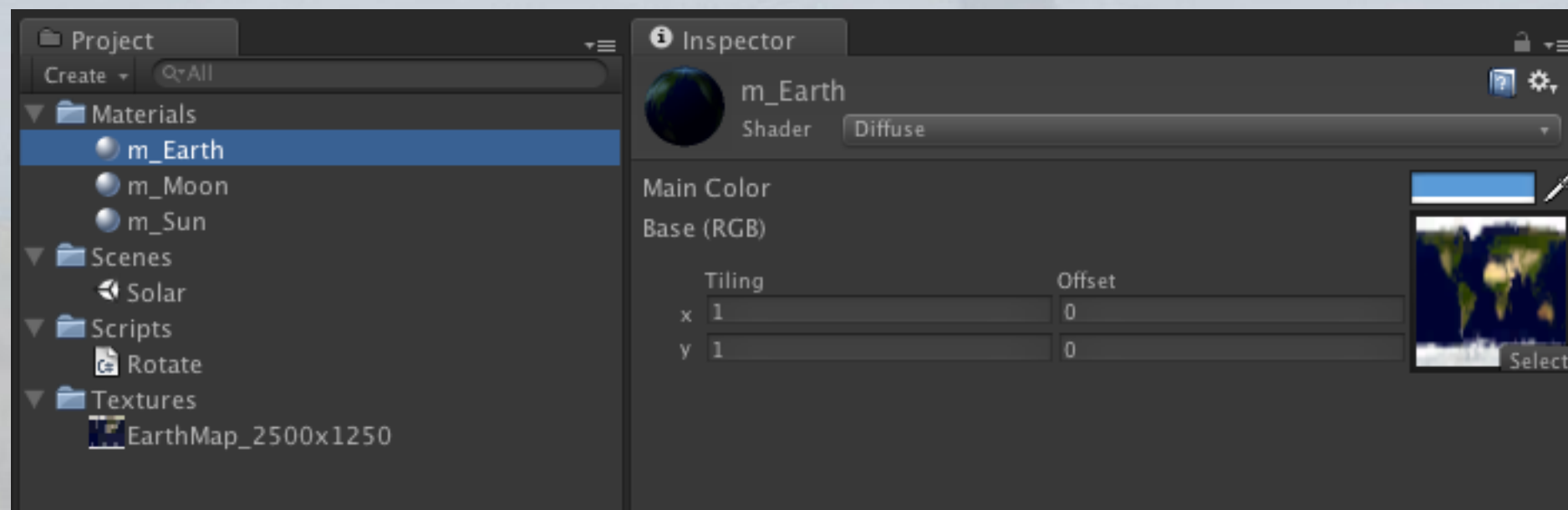
- Pulsando el LMB en el espacio en blanco y eligiendo la opción *Import New Asset...*
- Arrastrando las imágenes desde el explorador de ficheros al panel *Materials* dentro del entorno Unity3D

Seleccionando cada textura en el inspector, especificar 'Clamp' y una medida máxima de 512 para caso



# Unity. Sistema Solar (XIV)

1. Arrastrar y soltar cada textura sobre el material de cada planeta
2. Seleccionar el material para verlo en el panel Inspector
3. Arrastrar la textura al cuadrado del Albedo
4. Establece el color de Albedo a blanco



# Unity. Sistema Solar (XI)

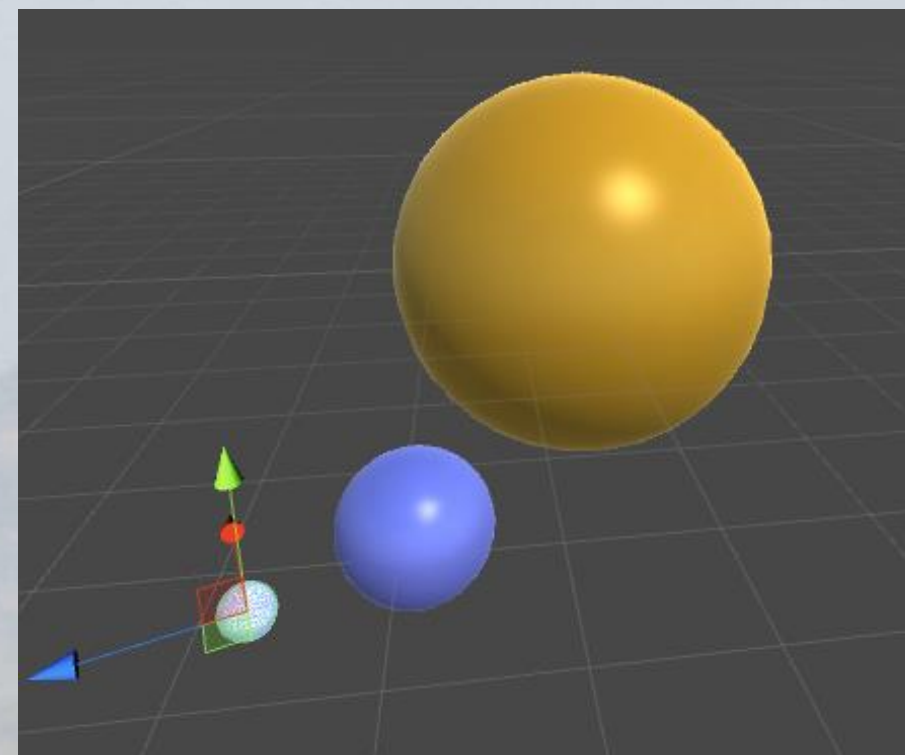
## Ejercicio

La velocidad de rotación de cada objeto espacial coincide con la de orbitación de cada padre

Independizar las velocidades de orbitación y rotación mediante el uso de empties y jerarquías

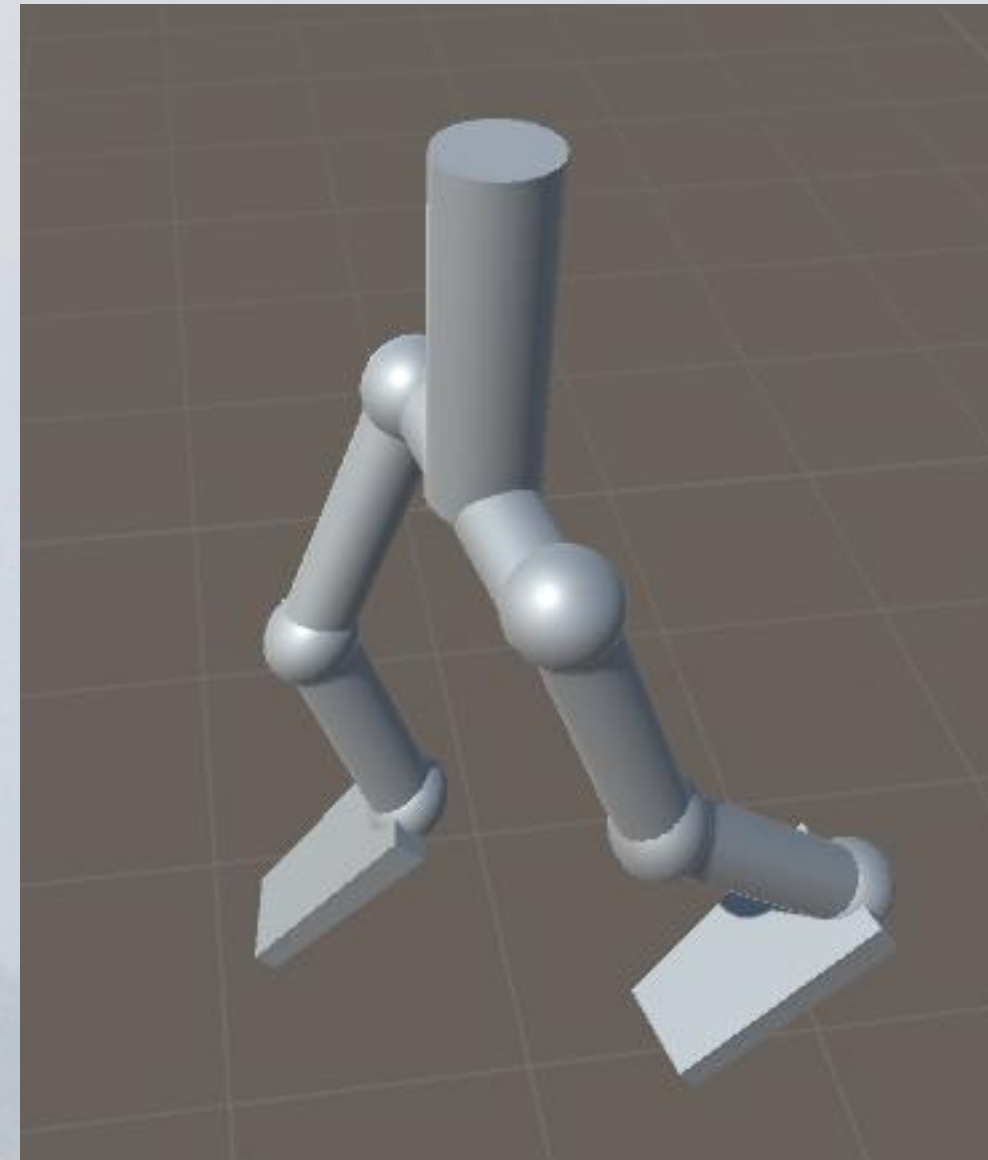
Ajustar las velocidades para hacer la animación más realista

La tierra está inclinada  $23.4^\circ$  con respecto a su plano orbital. Replica dicho ángulo en tu escena



# Unity. Sistema Solar (XV)

Crear un robot humanoide  
compuesto por una cintura, cadera  
izquierda y derecha, fémur, rodilla  
tibia, tobillo y pie de forma que  
puedan moverse jerárquicamente





# Bibliografía

Unity online manual:

<http://docs.unity3d.com/Manual/index.html>

S. J. Gortler. Foundations of 3D Computer Graphics.  
The MIT Press, 2012






Documentación generada por  
Dr. Ramón Mollá Vayá  
Sección de Informática Gráfica  
Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia

#### **Reconocimiento-NoComercial-CompartirIgual 2.5**

##### **Usted es libre de:**

copiar, distribuir y comunicar públicamente la obra  
hacer obras derivadas bajo las condiciones siguientes:

-  **Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador.
-  **No comercial.** No puede utilizar esta obra para fines comerciales.
-  **Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor

**Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.**