

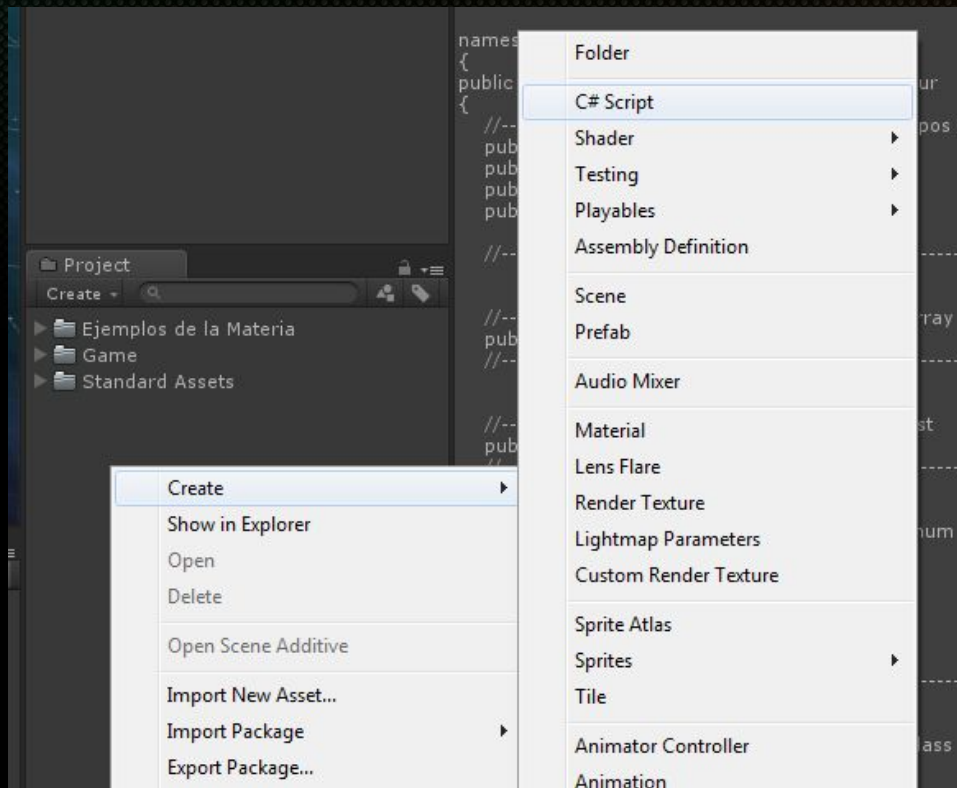


Clase 02

- C# Primer Script
- C# Anatomía de un Script (Monobehaviour)
- C# Consola de Unity
- C# Syntaxis básica y Tipos de datos básicos
- Unity: Componente Transform
- C# Manejo de Transform desde Script
- C# Time.deltaTime
- C# Input básico
- Unity: Prefabs Básicos
- Proyecto: StarMap
- Estructura de un Proyecto en Unity
- Links Útiles



Primer Script





C# Anatomía de un Script (MonoBehaviour)

- Awake()
- Start()

- Update()
- FixedUpdate() y LateUpdate()

- OnDisbled()
- OnEnabled()

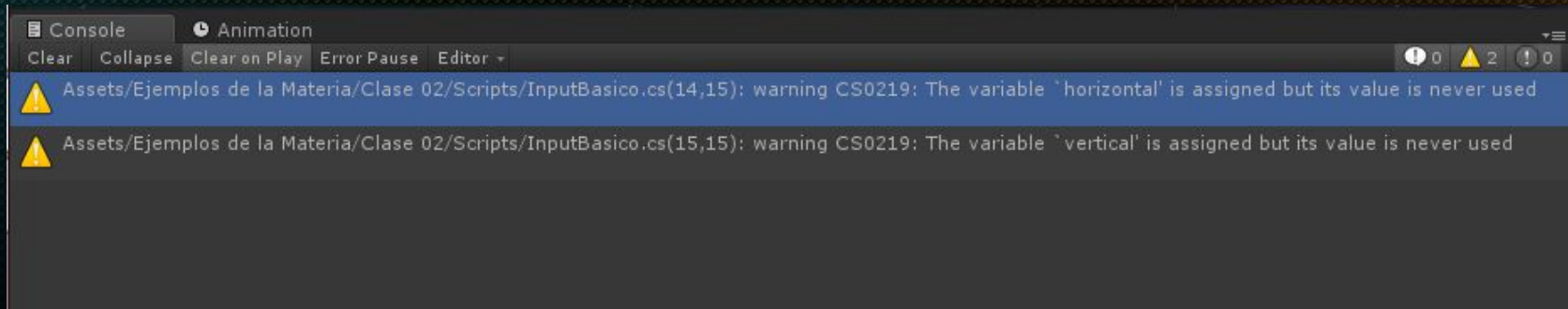
- OnGUI() (más adelante)



<https://docs.unity3d.com/ScriptReference/MonoBehaviour.html>



C# Consola de Unity



- Debug.Log()
- Debug.LogError()
- Debug.LogWarning()
- Debug.Break()



<https://docs.unity3d.com/ScriptReference/Debug.html>

C# Syntaxis básica y Tipos de datos básicos

```
TiposBasicosDeDatos.cs
Clase02.TiposBasicosDeDatos

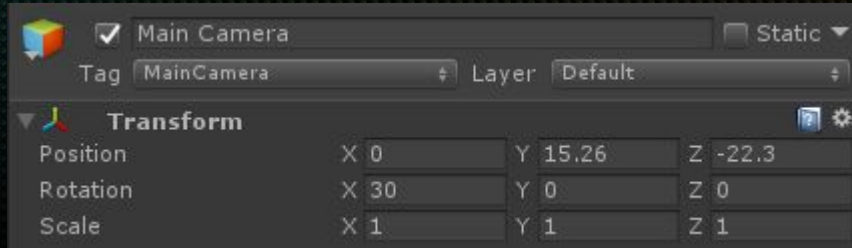
1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5
6 namespace Clase02
7 {
8     References
9     public class TiposBasicosDeDatos : MonoBehaviour
10     {
11         //----- Tipos abstractos
12         public int entero;
13         public float puntoFlotante;
14         public bool booleano;
15         public string texto;
16         //----- Array
17         public int[] arrayDeInts;
18         //----- List
19         public List<int> intList;
20         //----- Enum
21         public enum Enumerador
22         {
23             PrimerEnum,
24             SegundoEnum
25         }
26         public Enumerador enumerado;
27         //----- Class
28         [Serializable]
29         public class ClaseBasica
30         {
31             public int a;
32             public bool b;
33         }
34         public ClaseBasica ejemploClase;
35         //----- Method
36         public static void Method01() { }
37     }
38 }
```



<https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/types/>



Unity: Componente Transform





- Vector3 Position
- Vector3 Rotation (Euler Angles) (internamente Quaternion)
- Vector3 Scale



<https://docs.unity3d.com/ScriptReference/Transform.html>

C# Manejo de Transform desde Script



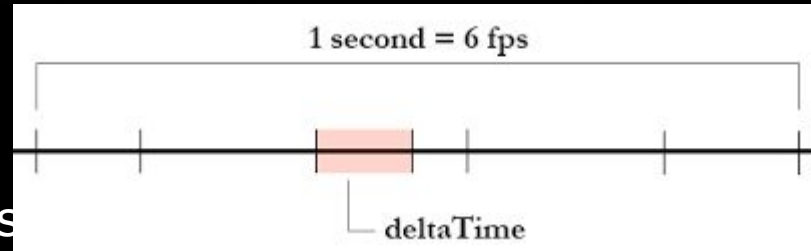
```
TransformCommons.cs X
TransformCommons
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class TransformCommons : MonoBehaviour
6 {
7     void Start ()
8     {
9         Transform t = this.transform;
10
11         Debug.Log("transform.position: " + transform.position);
12         Debug.Log("transform.localPosition: " + transform.localPosition);
13
14         Debug.Log("transform.rotation: " + transform.rotation);
15         Debug.Log("transform.localRotation: " + transform.localRotation);
16
17         Debug.Log("transform.lossyScale: " + transform.lossyScale);
18         Debug.Log("transform.localScale: " + transform.localScale);
19
20         //etc
21     }
22 }
23
```



<https://docs.unity3d.com/ScriptReference/Transform.html>

C# Time.deltaTime

El deltaTime es un concepto matemático que define el tiempo que pasó entre un momento y otro. En videojuegos se usa generalmente para definir el tiempo que pasó entre un frame y otro.



Unity nos
Time.deltaTime



<https://docs.unity3d.com/ScriptReference/Time-deltaTime.html>



C# Input Basics

```
if (Input.GetKeyDown(KeyCode.Alpha1))
    Debug.Log("Alpha1 presionado");

float horizontal = Input.GetAxis("Horizontal");
float vertical = Input.GetAxis("Vertical");

if (Input.GetKeyUp(KeyCode.Space))
{
    Debug.Log("Barra espaciadora soltada");
}

if (Input.GetKey(KeyCode.A))
{
    Debug.Log("Valor Horizontal al mantener A: " + horizontal);
    Debug.Log("Valor Vertical al mantener A: " + vertical);
}
```

- `bool Input.GetKeyDown(KeyCode c)`
Devuelve true si bien se presiona la tecla.
- `bool Input.GetKeyUp(KeyCode c)`
Devuelve true cuando la tecla es soltada.
- `bool Input.GetKey(KeyCode c)`
Devuelve true en todo momento que la tecla esté presionada.



<https://docs.unity3d.com/ScriptReference/Input.html>



Unity: Prefabs Básicos

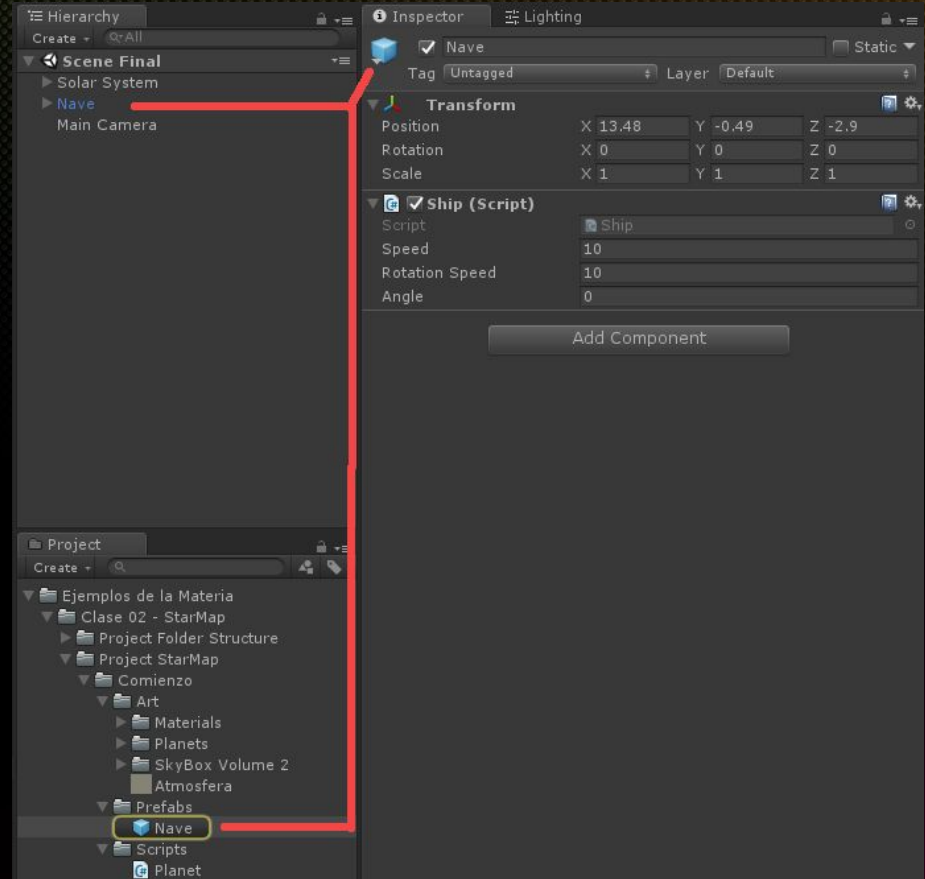
Los Prefabs es la forma que utiliza Unity para guardar una instancia de un GameObject para su reutilización luego.

Un Prefab puede contener una herencia de GameObjects con sus componentes.

Por arquitectura de Unity un Prefab **no** puede contener otro Prefab.



<https://docs.unity3d.com/es/current/Manual/Prefabs.html>





Proyecto: StarMap

1. Nave

- a. Moverla con las teclas
- b. Debe poder configurarse del inspector: velocidad de nave.

2. Script de Movimiento de Cámara.

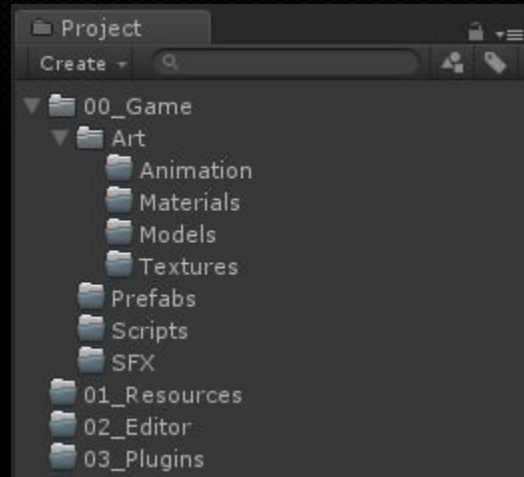
- a. Debe moverse de planeta a planeta con teclas 1 a 9. Con la tecla 0 apunta a la nave.
- b. Debe poder configurarse del inspector: tiempo de traslación, distancia a objetivo (en ejes Z e Y).

3. Chimi

- a. Reemplazar la horrible nave hecha en clase por la que está en "Ejemplos de Materia/Clase 02/Art/Nave"
- b. Hacerle una Atmósfera al Planeta Tierra.
Tip: Esfera, Standard Shader, Rendering Mode: Fade.
- c. Asignarle Trail Renderer a los propulsores de la nave.
Tip: "Add Component->Trail Renderer".
- d. Asignarle Trail Renderer a los planetas.
- e. Aceleración de la nave en base a tiempo.



Estructura Básica de Un Proyecto en Unity





Links Útiles



- Docs de Unity, Script Reference y Manual
<https://docs.unity3d.com/ScriptReference>
- Unity Learn
<https://unity3d.com/es/learn>
- YouTube!!!
- Google!!!