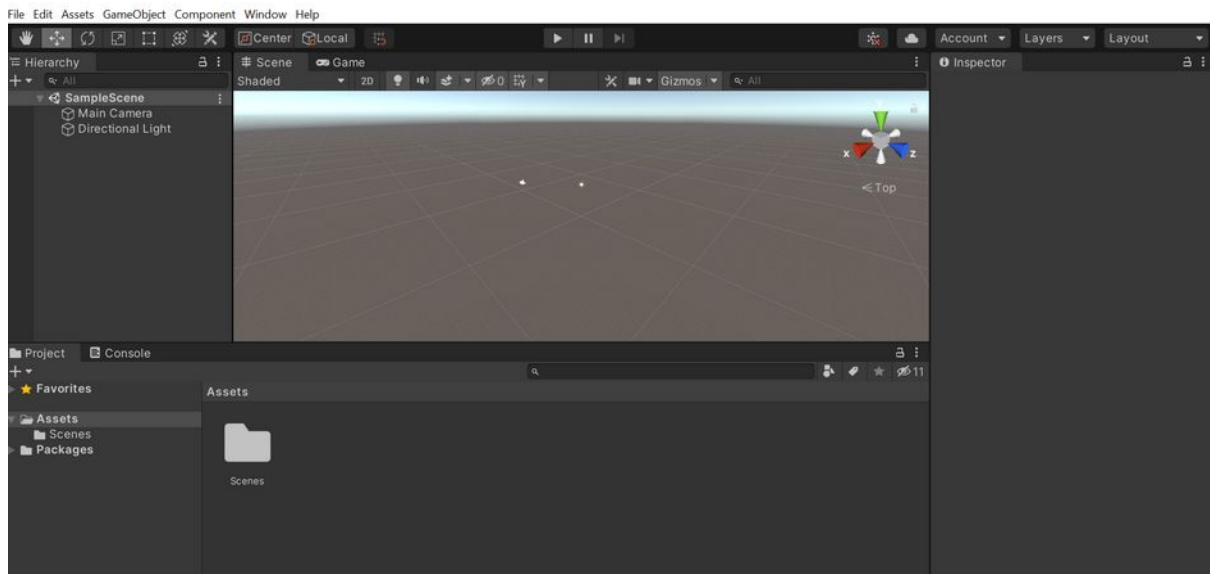


# UNITY 3D Introducció v2

Preparat per Sergi Gotarra Octubre 2022 (5-6 hores)

## UNITY



Hierarchy (jerarquia) Es veuen de manera visual les correspondències entre gameobjects

Project: La distribució d'elements per carpetes, habitualment es crea una d'assets i una de prefabs

Console: la sortida text

Scene: escena de treball

Game: sortida des de la càmera (habitualment)

Inspector: Components i opcions de cada gameobject

Tecles Play- pause - next frame (controlen l'execució del joc)

Atenció: en temps d'execució (amb el play apretat) es poden fer canvis a la scene, però aquests canvis desapareixeran quan parem el joc.

## Com mostrar missatges per la consola

(Crear un script nou i assignar-lo a un game object)

Debug.log o print

*Start s'executa quan es crea l'objecte*

*Update s'executa a cada frame*

```
using System.Collections;
```

```

using System.Collections.Generic;
using UnityEngine;

public class NewBehaviourScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        Debug.Log("hola carabola");
        Debug.LogWarning("advertencia");
        Debug.LogError("fallo");
    }

    // Update is called once per frame
    void Update()
    {

    }
}

```

Tot el que hi ha a una scene son game object  
 Podem pensar que un gameobject és un conjunt de components

Creem un nou gameobject (un cube)

Transform especifica les dimensions, posició o escala (tot gameobject té un component transform) aquests valors poden ser absoluts o referits al seu pare

Com canviar els valors transform per codi:(a través d'un element anomenat Vector3)

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class mou : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
    }
}

```

```

        float novay;
        novay = (float)(transform.position.y-0.01);
        transform.position = new
Vector3(transform.position.x,novay,transform.position.z);
    }
}

```

ROTACIÓ es canvia amb Quaternion:

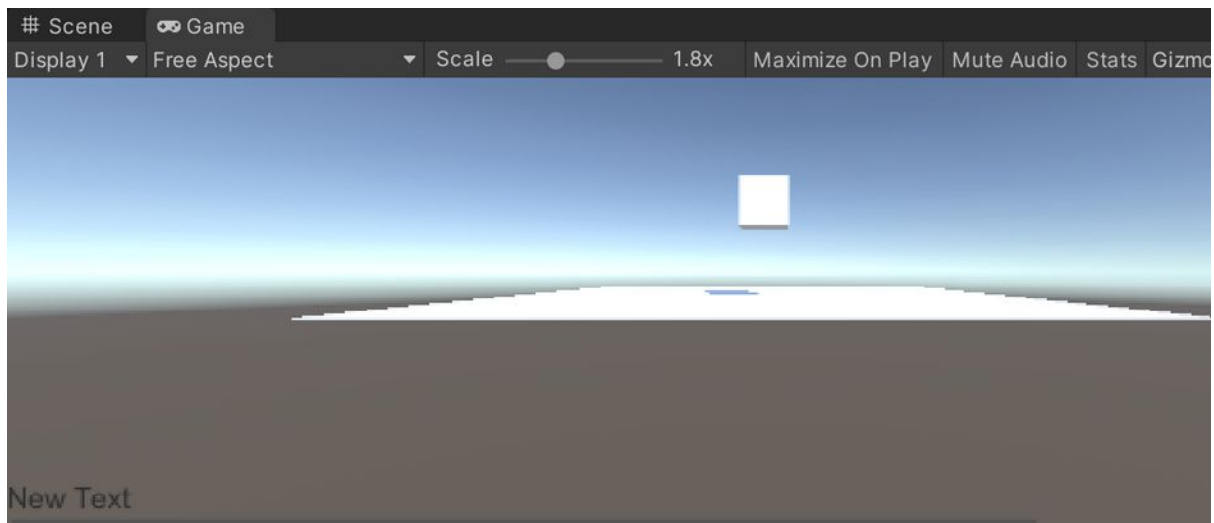
```
transform.rotation= Quaternion.Euler(0f,45f,0f);
```

**Prefabs:** Els gameobjects es guarden com a Prefabs, per a fer-ho n'hi ha prou amb arrossegar-los de hierachy a la carpeta prefab. Un cop allà per codi es poden instanciar fàcilment. També arrosegant-los a la scene.

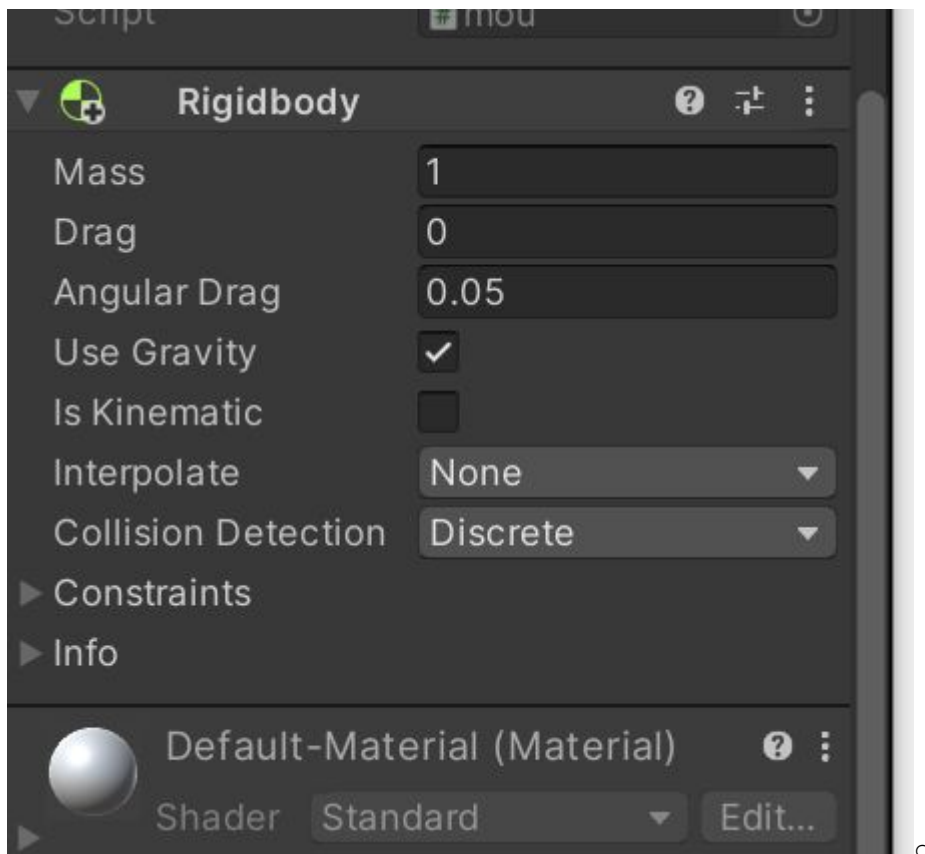
**Físiques:**

**Els gameobject poden tenir característiques físiques: solidesa, roçament, gravetat...**

Dins del scene creem un plane i el fem servir de terra.



Ara assignarem valors fisics al cube amb el component rigidbody. Per fer-ho anem a Inspector amb el cube seleccionat: addcomponent, rigidbody



Fent Play veiem com funciona la gravetat a l'escena

També podem afegir components en temps d'execució

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

public class Acceso_06_Addcomponent : MonoBehaviour
{
    // afagirem un component
    //tots els components tenen una referenecia a gameobject que es diu
    gameObject (en minuscules)
    //quan arrenqui el joc a aquest objecte se li afegira rigibodi

    void Start()
    {
        gameObject.AddComponent<Rigidbody>();
    }
}
```

## **GUI:**

**Pantalla a primer pla de la càmera on es poden situar butons, textos, etc.**

Farem servir aquests components:

*GUILayout.Label -> mostra text*

*GUILayout.TextField -> ens permet entrar una string*

*GUILayout.Button --> posa un botó*

## **Script a afegir al objecte que es vol modificar la posició per teclat**

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

public class mou : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }

    float newx, newy, newz;
    void OnGUI() {
        //s'executa després d'haver-se renderitzat tot
        // estem passant la string a una float, fem unes caixes per entrar
        text

        GUILayout.Label("x:");
        newx= float.Parse(GUILayout.TextField(newx.ToString(),4));

        GUILayout.Label("y:");
        newy= float.Parse(GUILayout.TextField(newy.ToString(),4));

        GUILayout.Label("z:");
        newz= float.Parse(GUILayout.TextField(newz.ToString(),4));

        if (GUILayout.Button("colocar"))
```

```

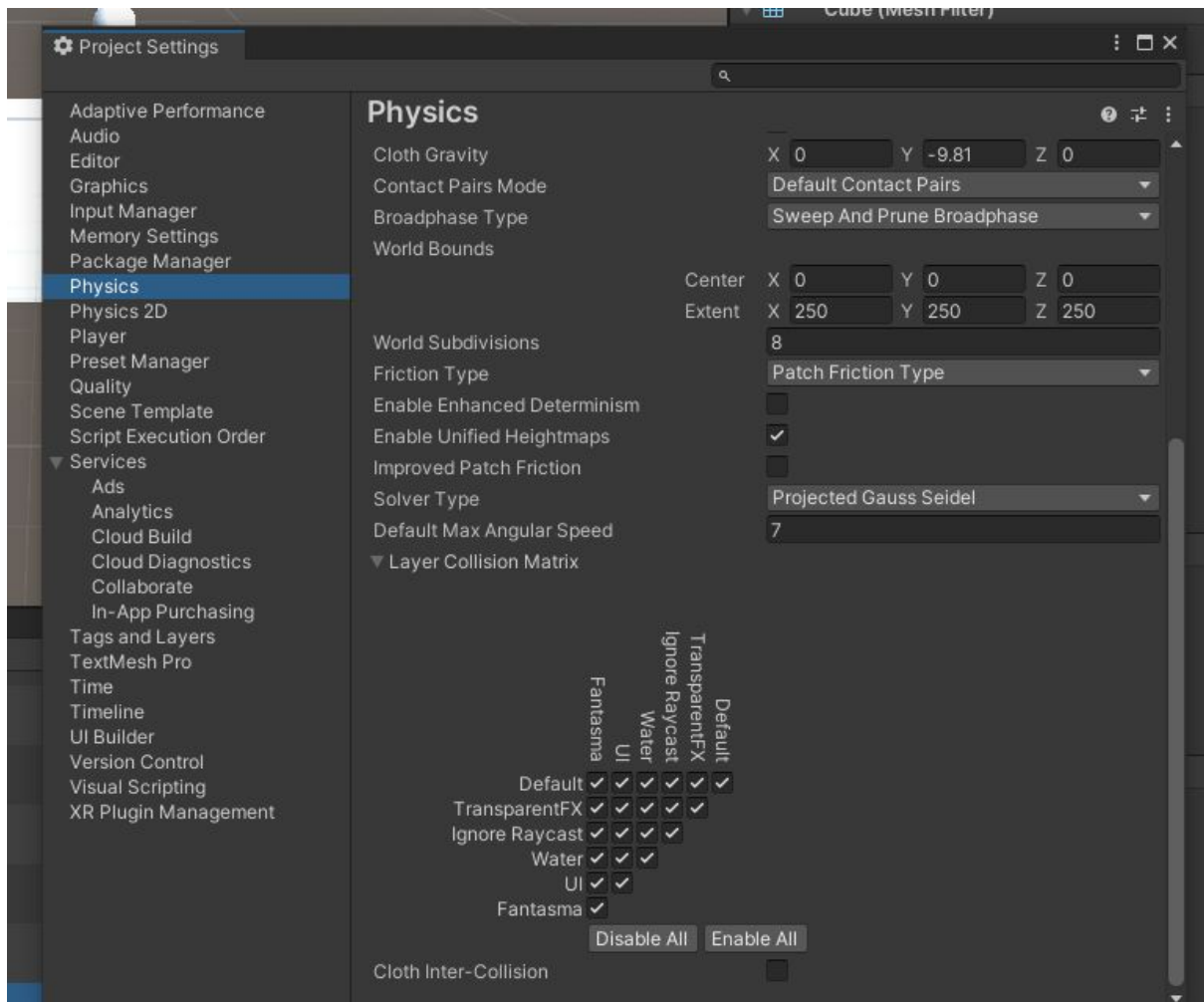
{
    print ("s'ha pulsat un botó");
    transform.position=new Vector3(newx,newy,newz);
}
}
}

```

## Detecció de col·lisions:

### Layers

A project settings, tags and layers podem crear noves capes. Al inspector de qualsevol element ens posicionem a label i allà escrivim el nom de la capa nova i endavant.

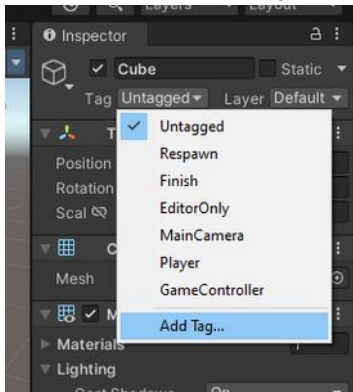


Hi ha una matriu de col·lisions on s'indica entre quines capes hi haurà col·lisions.

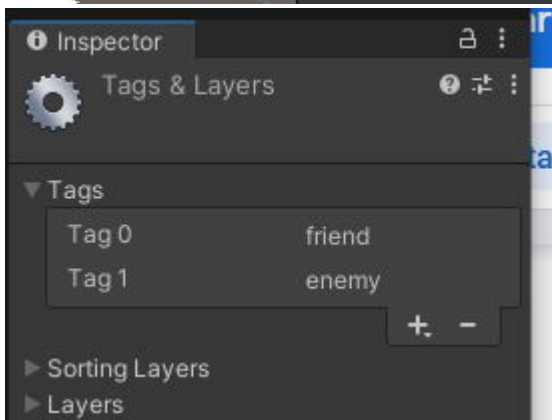
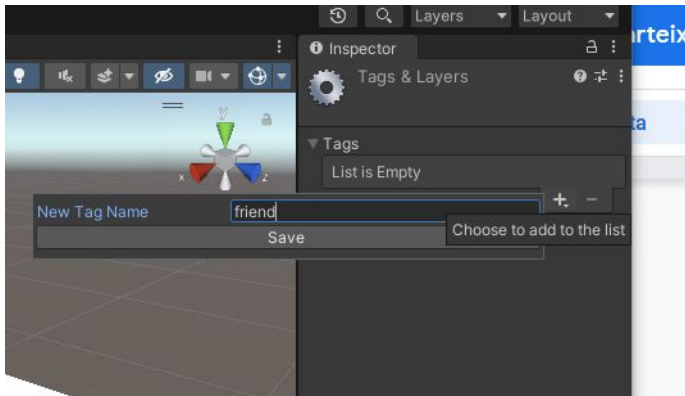
### TAGS

Etiquetes, podem localitzar un objecte per la seva etiqueta

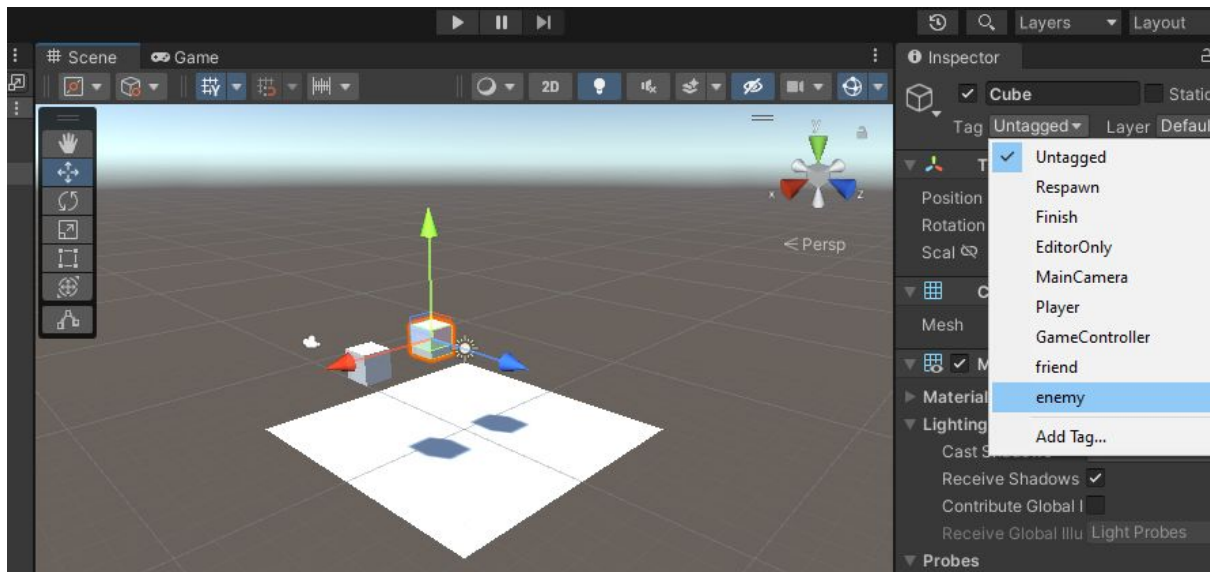
Creem dues etiquetes “friend” o “enemy”, fem una copia del cube i a un li assignem tag friend i a l’altre enemy



Crear 2 etiquetes noves, han de quedar així:



Ara assignem a cada cube un tag diferent



Script que assignarem al plane que fa de terra:

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

public class NewBehaviourScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        Debug.Log("hola carabola");
        Debug.LogWarning("adven");
        Debug.LogError("error");
    }

    // Update is called once per frame
    void Update()
    {
    }

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.CompareTag("enemy"))
        {
            Debug.Log ("Detectat enemy: "+collision.gameObject.name);
            Destroy(collision.gameObject);
            Debug.Log ("ENEMY A BORRAR");
        }
        if (collision.gameObject.CompareTag("friend"))
    }
}
```



```

        {

            Debug.Log ("Bravo FRIEND "+collision.gameObject.name);
        }

    }

}

```

*Quan hi hagi una col·lisió, detectarà el gameobject, i si és enemy, l'esborra.*

### **Mètodes d'entrada: Teclat, ratolí, tab...**

#### **TECLAT:**

//Canviant el valor de color d'un cube i veient el valor x a debug  
 //Utilitzem les tecles R,B i Z

```

void Update()
{
    if (Input.GetKeyDown(KeyCode.R))
    {
        GetComponent<Renderer>().material.color=Color.red;
    }
    if (Input.GetKeyDown(KeyCode.B))
    {
        GetComponent<Renderer>().material.color=Color.blue;
    }

    if (Input.GetKeyDown(KeyCode.Z))
    {
        print("posicion x :"+ transform.position.x);
    }
}

```

#### **UTILITZAR UN LA TOUCH SCREEN DEL MÒVIL:**

Veure tutorial: <https://cactiattack.wordpress.com/2015/08/25/tutorial-basic-input-for-androidiphone/>

```
void Update () {
    //Let's see if there are any touches
    if (Input.touchCount > 0)
    {
        Touch touch = Input.GetTouch(0);
        if (touch.phase == TouchPhase.Began)
        {
            Debug.Log("User Touched: " + touch.position);

            //Convert to worldSpace coordinates
            Vector3 worldSpace =
Camera.main.ScreenToWorldPoint(touch.position);
            Debug.Log("In WorldSpace this is: " + worldSpace);
        }
    }
}
```

**PROBLEMA:** Només es pot provar quan s'ha compilat o.... utilitzant un cable i **UNITY REMOTE:** Aplicació dissenyada per a ajudar en les aplicacions dissenyades per a android o ios.

- Primer: instal·lar unity remote al movil
- Segon: Un cop instal·lat connectar amb el ordinador per cable usb, Unity utilitzarà el movil com a generador de mètodes d'entrada

Deixem només un cube i el plane a la pantalla  
Ens assegurem que cube té el component rigidbody  
Al cube li afegim aquest script

Instanciar objectes des d'un script quan es pulsa 1

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

public class Api_08_instantiate : MonoBehaviour
{
    void Update()
```

```

{
    if (Input.GetKeyDown(KeyCode.Alpha1))
    {
        //s'instanciarà ell mateix
        Instantiate(gameObject);
    }
}
}

```

Afegim aquest Script al cube d'alçada random

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

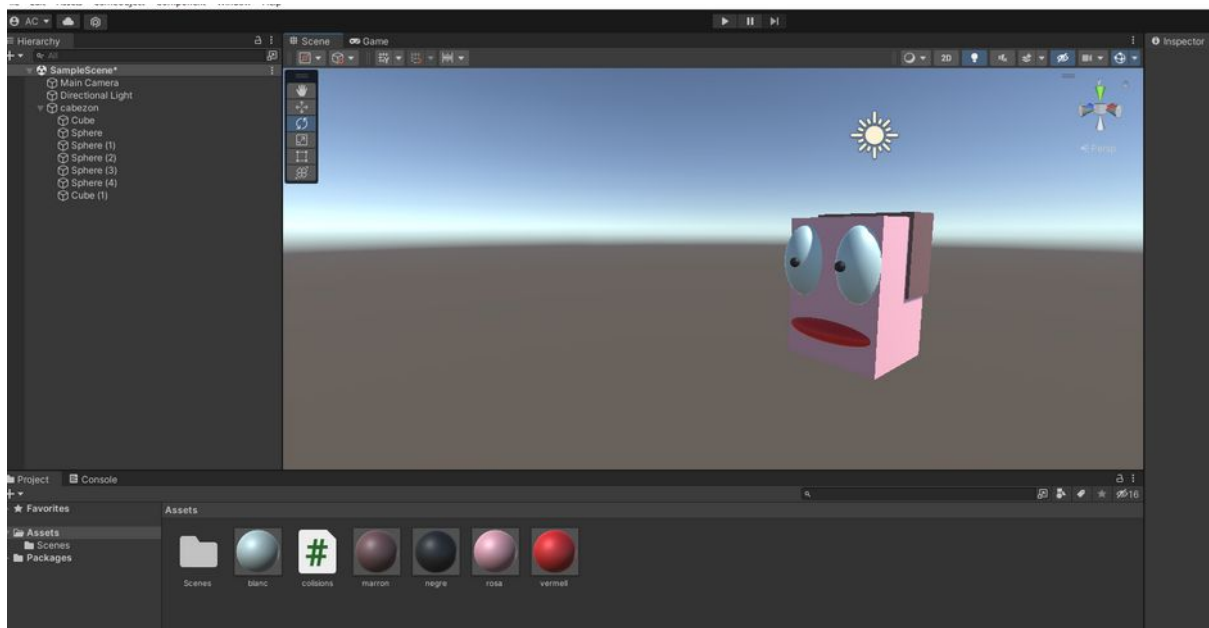
public class altura : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        float igriega=Random.Range(1.0F, 10.0F);
        transform.position = new
Vector3(transform.position.x,igriega,transform.position.z);
        Debug.Log(igriega.ToString());
    }

    // Update is called once per frame
    void Update()
    {
    }
}

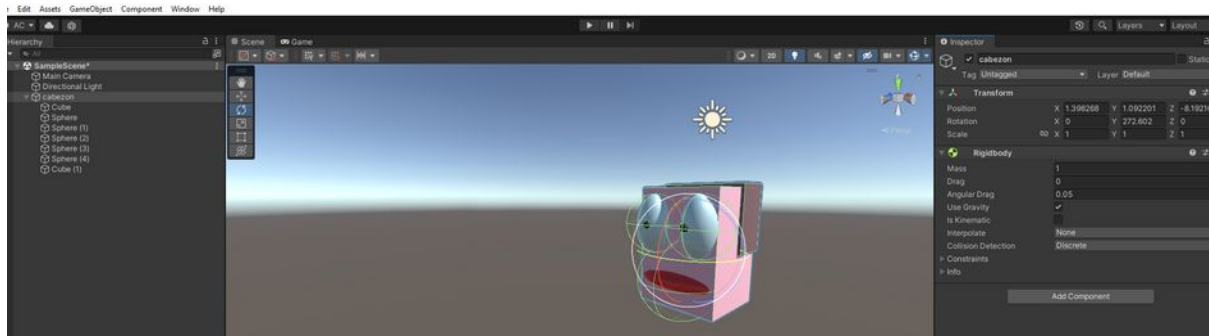
```

## **TREBALLAR AMB PREFABS**

Creem un prefab. Per exemple un cubo amb cara feta amb tres esferes. I el fem fill d'un objecte buit



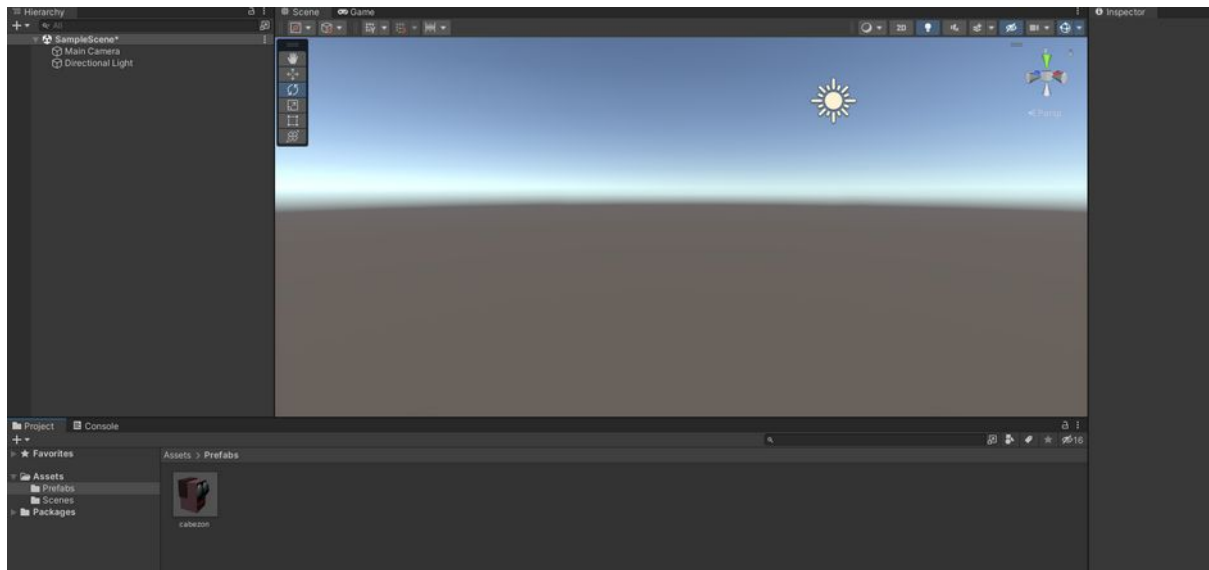
Afegim el component rigidbody al objecte pare



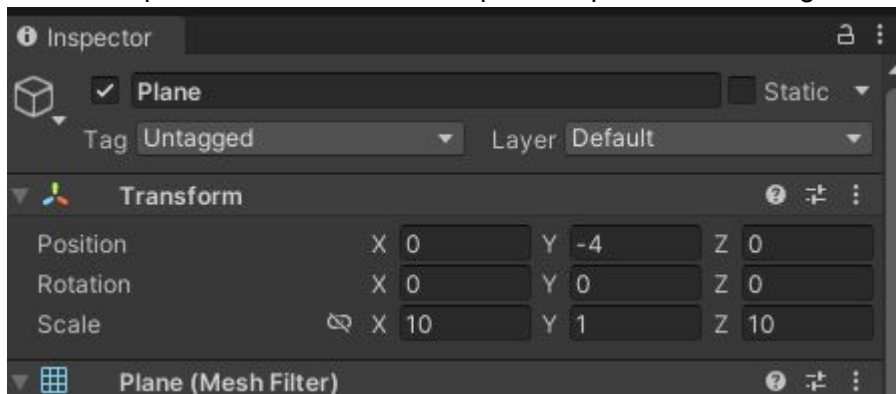
Creem una carpeta prefabs i arrosseguem l'objecte cabezon



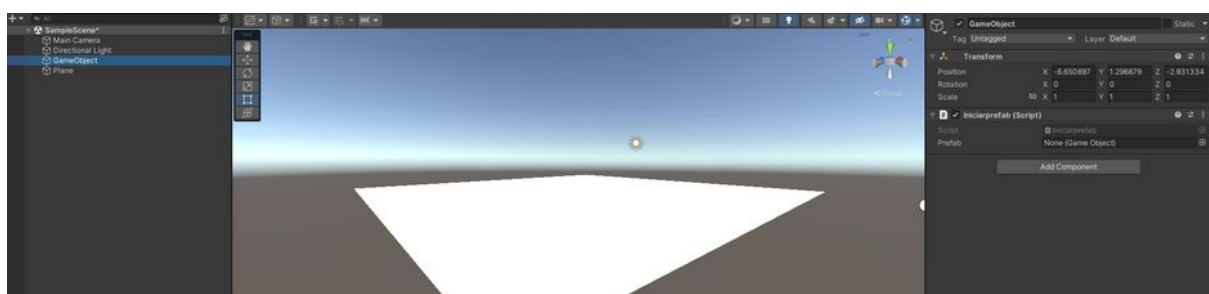
Ara ja es pot esborrar el objecte cabezon de la scene



Creem un plane com el de l'anterior pràctica però amb els següents paràmetres:



Creem un objecte buit a les coordenades 0,0,0  
i li assignem el codi Apiinstantiate



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

public class Api_08_instantiate : MonoBehaviour
{
    public GameObject prefab;
```

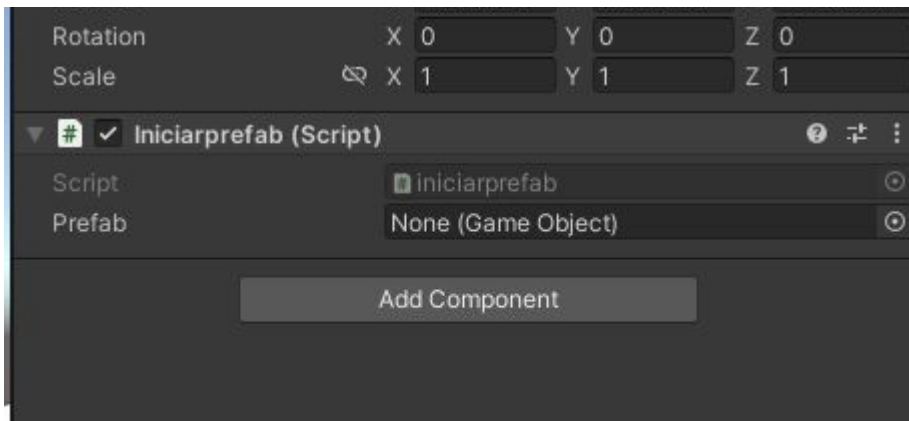
```

void Update()
{
    if (Input.GetKeyDown(KeyCode.Alpha1))
    {
        //s'instanciarà el prefab
        //assignem el script a un objecte buit

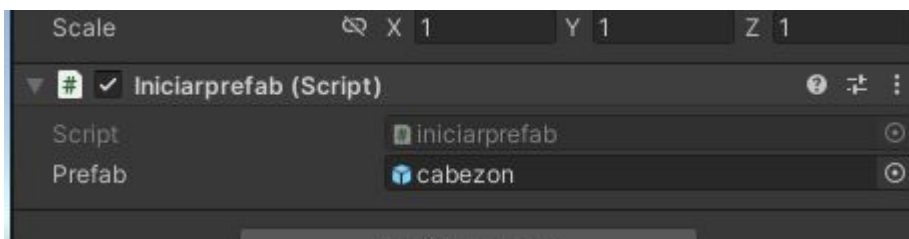
        Instantiate(prefab);
        //AFEGIM VALORS RANDOM PER A COL·LOCAR-LA A LA PANTALLA
        float xrandom=Random.Range (-3f,3f);
        float yrandom=Random.Range (-3f,3f);
        float zrandom=Random.Range (-3f,3f);
        Vector3 v=new Vector3(xrandom,yrandom,zrandom);
        prefab.transform.position=v;
    }
}
}

```

Ara a l'inspector del gameobject buit afegim el prefab a la variable que hem creat amb codi:



Arrossegant el prefab



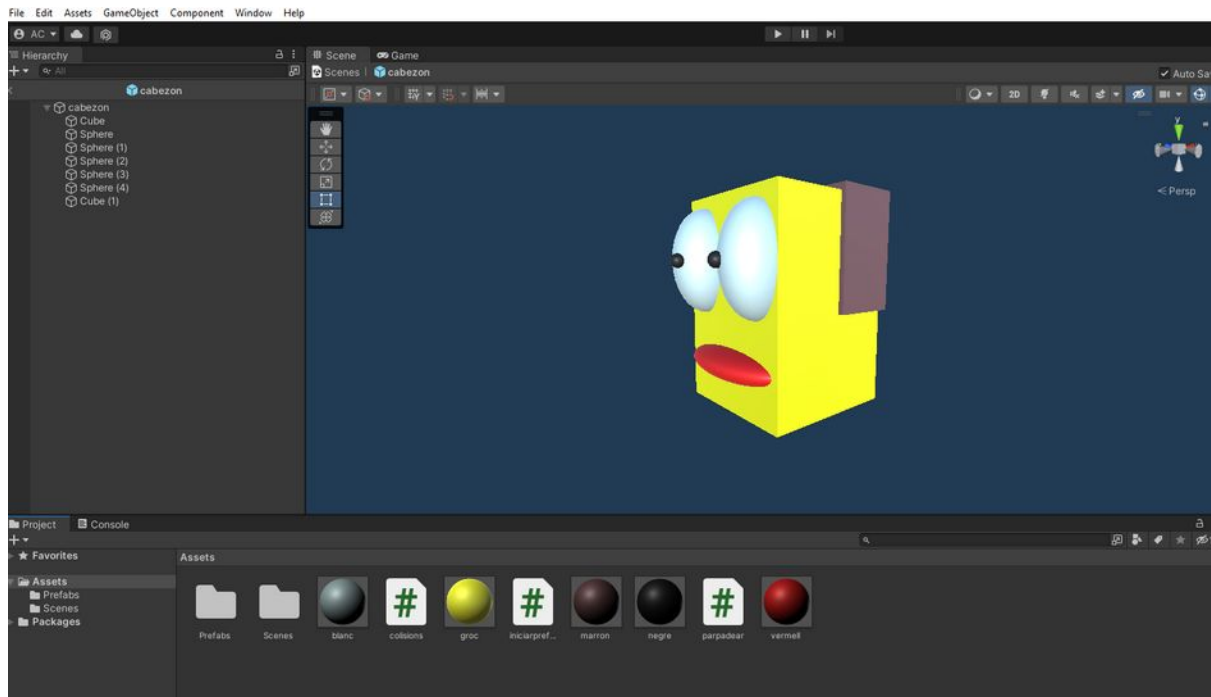
Executem el codi i cada vegada que pulsem 1 es crearà un nou gameobject copia del prefab

## COROUTINES:

### **Són rutines temporitzades**

Afegirem un parpadeix del nostre personatge cada 3 segons

Per fer-ho més fàcil obrim el prefab i canviem el color de la pell de rosa a groc



Tenint en compte que els fills de cabezon són:



Tal que cube és el fill 0, Sphere el fill 1, Sphere(1) el fill 2, etc  
Farem que sphere i Sphere 1 es tornin de color rosa cada 3 segons

Fem un script per al cabezon utilitzant coroutines:

```
using System.Collections;  
using System.Collections.Generic;
```

```

using UnityEngine;

public class parpadear : MonoBehaviour
{
    // Start is called before the first frame update

    void Start()
    {
        StartCoroutine("Coroutine");
    }
    IEnumerator Coroutine()
    {
        //s'espera un segon Debug.Log ("ara mesperare 3 segons");
        yield return new WaitForSeconds(3);
        //ja ha passat el segon Debug.Log ("YA");
        //ara parpadejem que consisteix pintar les ninetes
        //i pintar els globus oculars de groc

        GameObject p1 = this.transform.GetChild(1).gameObject;
        p1.GetComponent<Renderer>().material.color=Color.yellow;

        GameObject p2 = this.transform.GetChild(2).gameObject;
        p2.GetComponent<Renderer>().material.color=Color.yellow;

        GameObject p3 = this.transform.GetChild(4).gameObject;
        p3.GetComponent<Renderer>().material.color=Color.yellow;

        GameObject p4 = this.transform.GetChild(5).gameObject;
        p4.GetComponent<Renderer>().material.color=Color.yellow;

        yield return new WaitForSeconds(0.1f);

        p1.GetComponent<Renderer>().material.color=Color.white;
        p2.GetComponent<Renderer>().material.color=Color.white;
        p3.GetComponent<Renderer>().material.color=Color.black;
        p4.GetComponent<Renderer>().material.color=Color.black;

        StartCoroutine("Coroutine");
    }
}

```



```
// Update is called once per frame
void Update()
{

}
}
```

**Evidentment aquest script s'ha d'afegir al prefab!**

## SCENES a Build Settings

Hi ha un cuadro Scenes in Built on s'afegeixen les escenes que hi haurà a la compilació:

S'afegfeixena l'opció add open scenes arrossegant-les escenes des d'asserts.

Unity recorrerà totes les escenes i buscara els recursos que utilitza cadascuna i l'afegira al exe o apk final

**L'ORDRE NO TE IMPORTANCIA, salvat la 1a (NUMERO 0, que es la que carregarà quan arrenqui el joc**

El Timeline d'Unity és una característica que permet editar visualment música, reproduir animacions, mostrar/ocultar objectes, controlar partícules, etc. al llarg de l'eix del temps.

És similar a la capacitat d'afegir i editar música, vídeo, subtítols i direcció a una pista en el programari d'edició de vídeo.

Es pot utilitzar per crear una obra espacial on el moviment de múltiples objectes i el moment de la música es combinen per crear una història que es desenvolupa.

**Timeline asset:** és un fitxer amb la extensió .playable i el trobem al project view. Cada línia de temps no està lligada a cap objecte del projecte.

**Timeline instance:** és quan apliquem la timeline asset (similar a prefabs i instàncies de prefabs)

Els principals tipus de tracks són les següents:

Activation track: Una pista que controla l'activació o inactivitat de l'objecte objectiu.

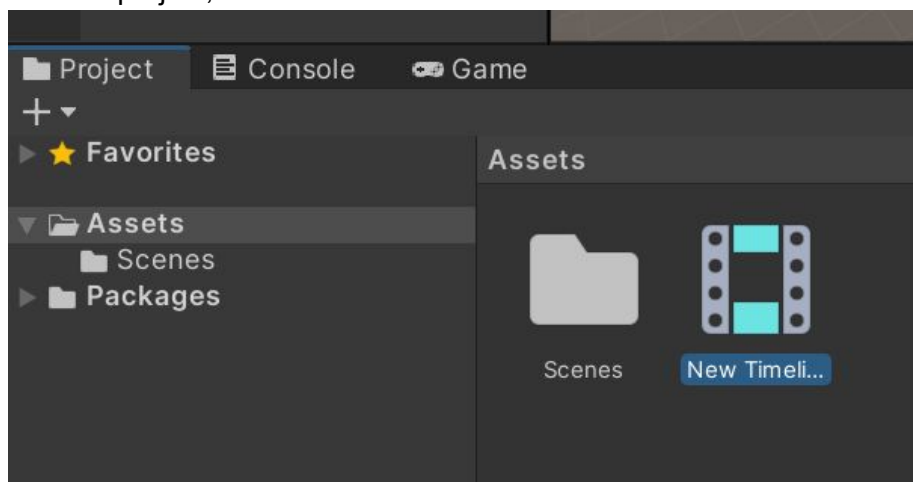
Animation track: Controla el moviment, la rotació i altres animacions de l'objecte objectiu.

Audio Track: Controla l'àudio.

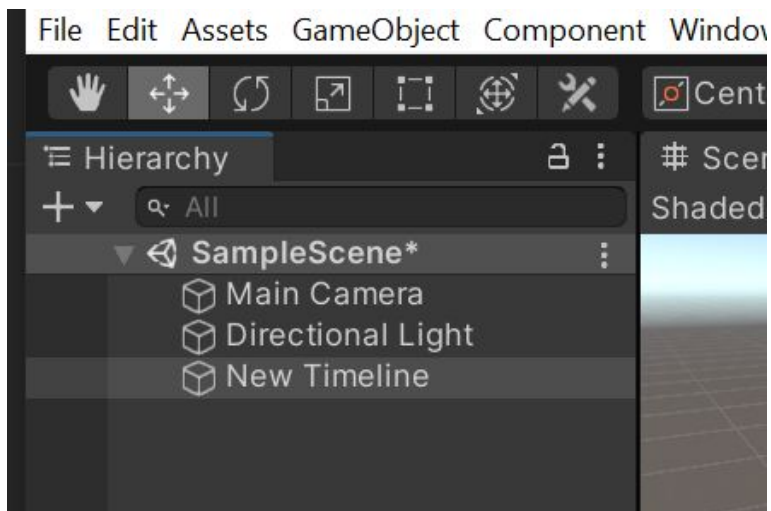
Control Track: Controls objectes, etc. Pista de control: Utilitzada per a la creació de scripts.

Prova:

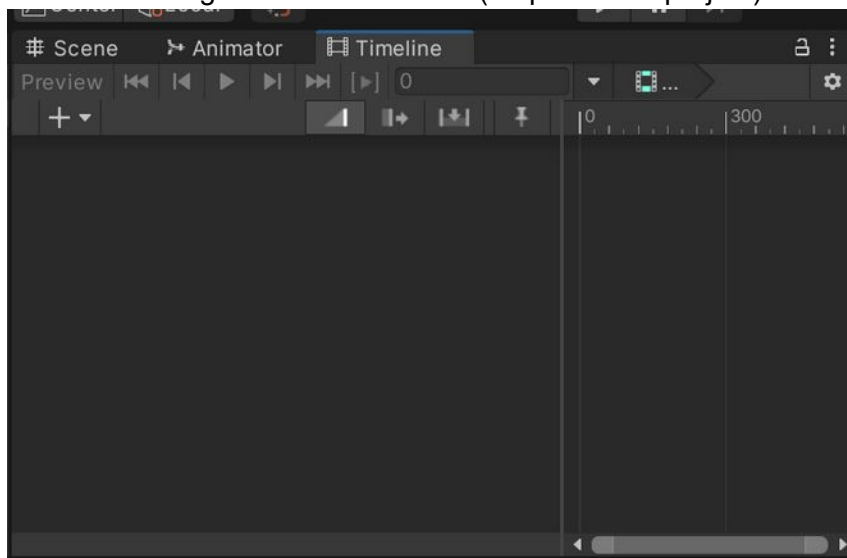
Anem a project, creem un timeline



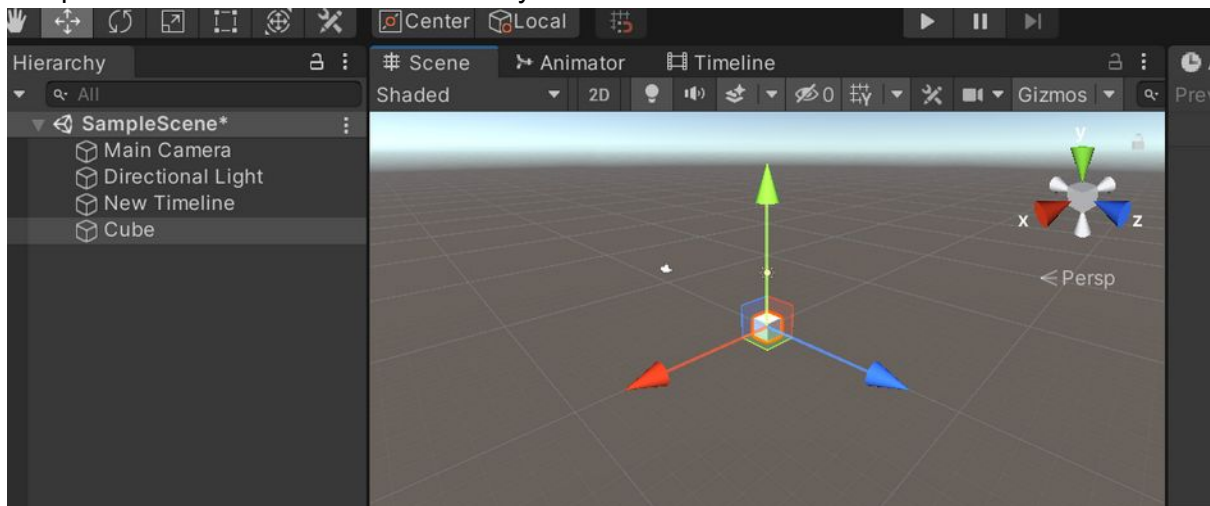
L'arroseguem a la scene



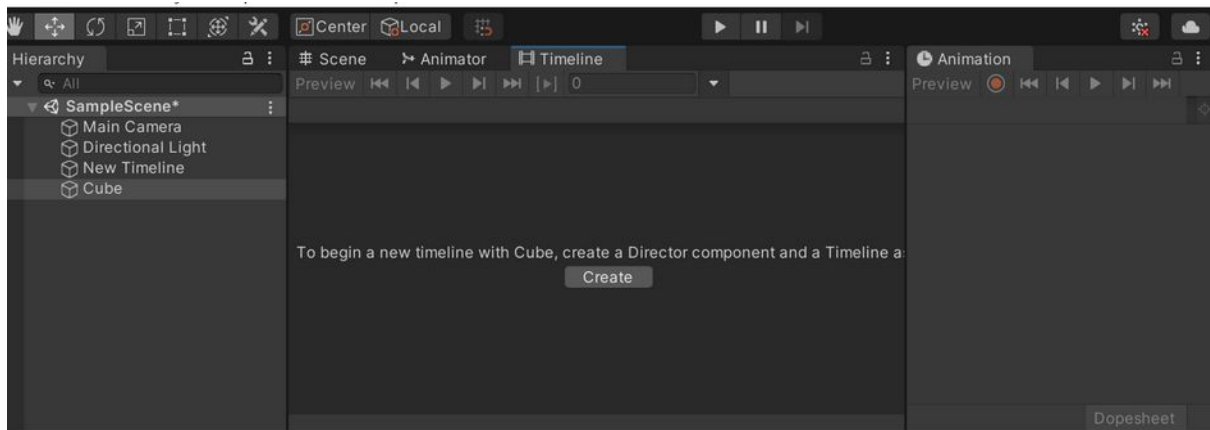
Picant dues vegades sobre timeline (el que està a project) s'obra la finestra del timeline



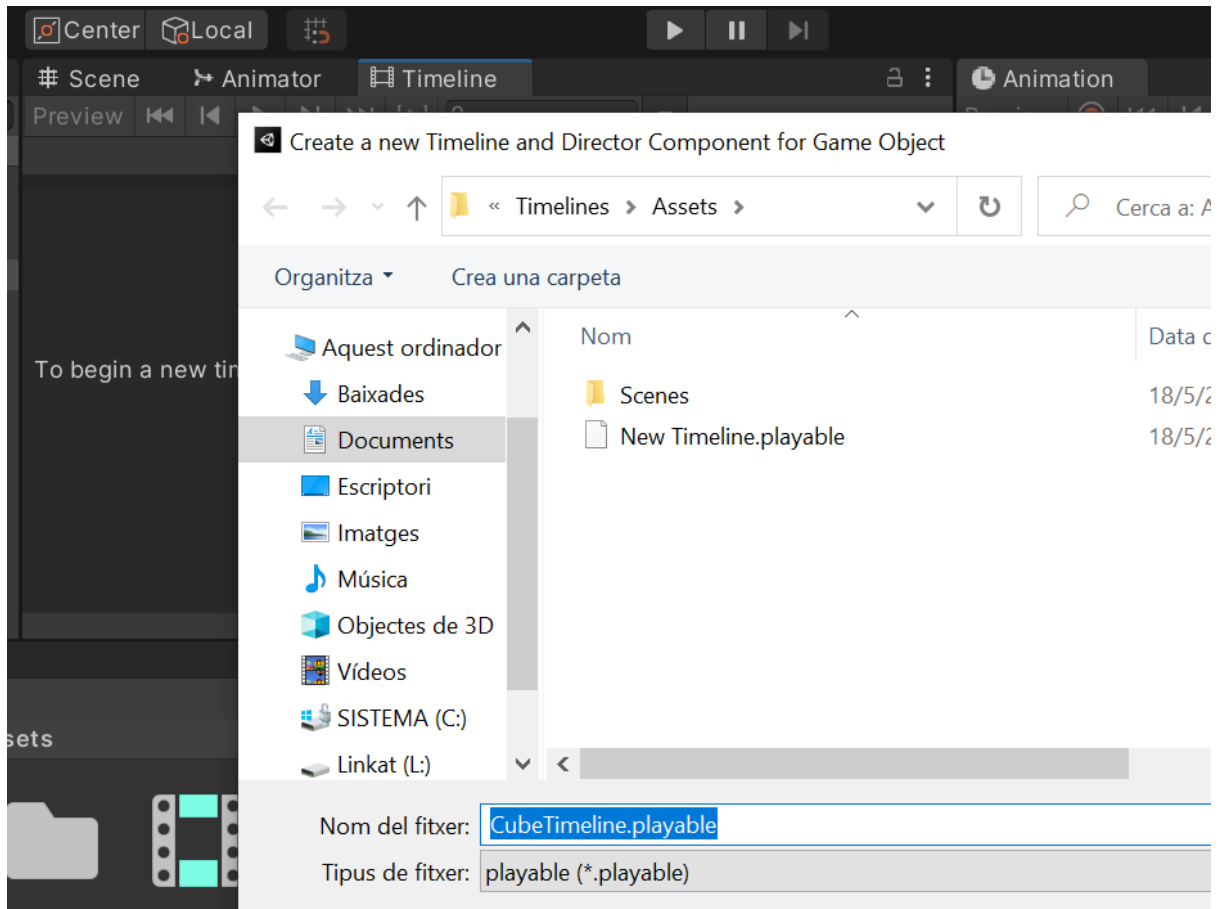
Per provar-ho creem un cube a hierachy

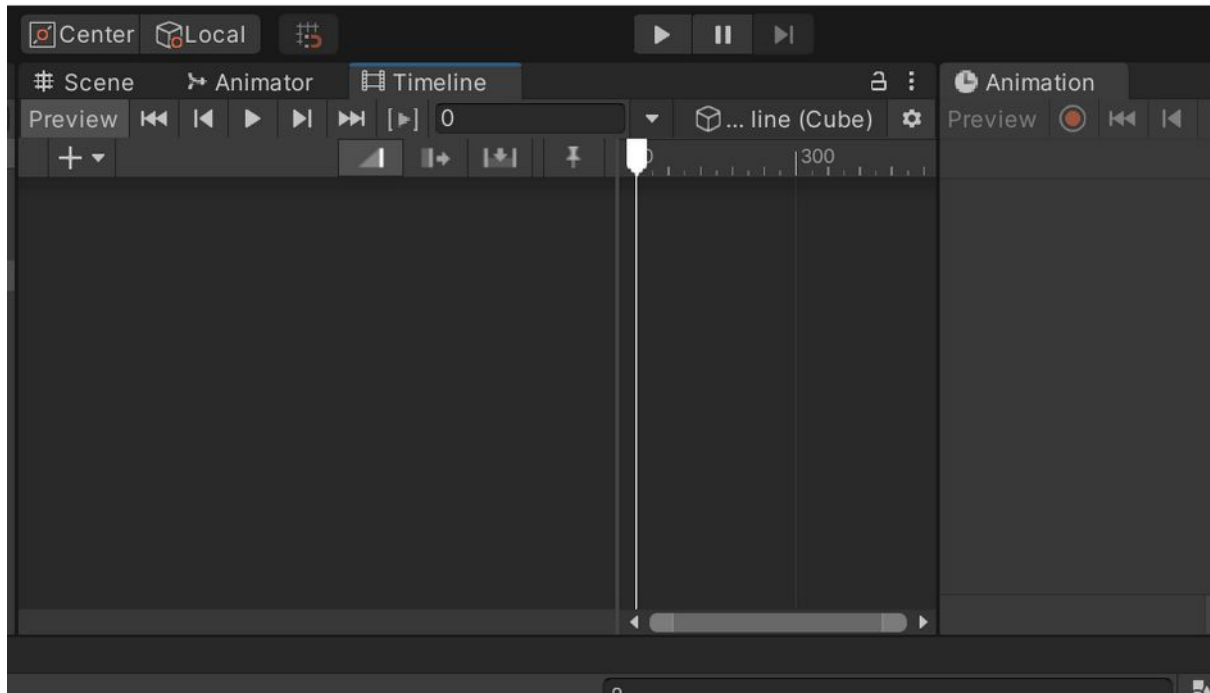


Ara anem a la pestanya de timeline



Li diem Create





A continuació en aquest video veiem com es grava l'animació

<https://drive.google.com/file/d/14YeTJqQVDwmM8o7myQQyumVJbqtX1IQR/view?usp=sharing>

Link a tutorial original: [https://styly.cc/tips/timeline\\_unity\\_kaki/](https://styly.cc/tips/timeline_unity_kaki/)

## TASCA ALUMNES

1.- Creem un nou projecte

2.- Posem un terrain 1000x1000 amb arbres

<https://docs.unity3d.com/es/530/Manual/script-Terrain.html>

Vídeo <https://www.youtube.com/watch?v=ddy12WHqt-M>

4.- Baixem un model d'un personatge i el col·loquem sobre el terrain

5.- Possem la càmera rere el personatge i afegim moviment en 1ª persona amb

CharacterController

Vídeo explicació: <https://www.youtube.com/watch?v=Zmh1FFywwx4>

6.- Baixem un model d'un enemic (fem una rutina de que quan l'enemic et veu gira sobre sí mateix i avança cap al jugador)

Video explicació: <https://www.youtube.com/watch?v=jxoY5kBtCRQ>