



IES AUGUSTO GONZALES DE LINARES
CFGS DAM2º

INVESTIGACIÓN SOBRE PROGRAMACIÓN MULTIMEDIA

PROGRAMACIÓN MULTIMEDIA Y
DISPOSITIVOS MÓVILES

Gutiérrez Valverde, Ramiro
Espinosa García, Daniel
Diez de Paulino, Albano



2023/2024

Índice

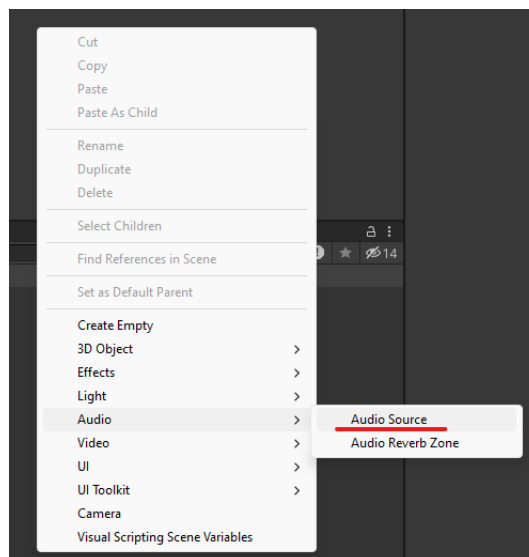
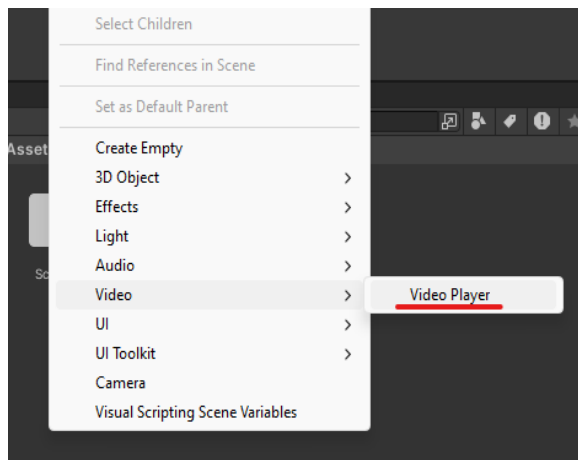
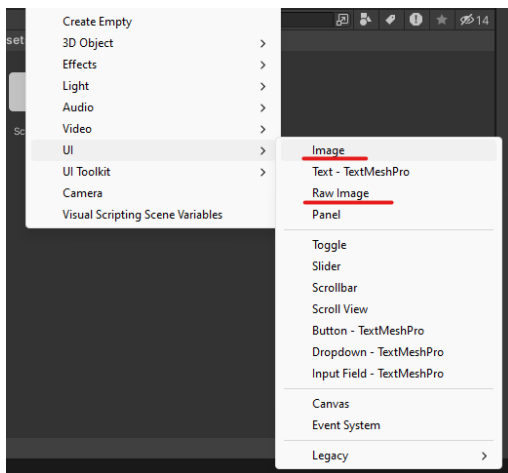
b) Se han reconocido las clases que permiten la captura, procesamiento y almacenamiento de datos multimedia.	2
c) Se han utilizado clases para la conversión de datos multimedia de un formato a otro.	3
d) Se han utilizado clases para construir procesadores para la transformación de las fuentes de datos multimedia.	4
e) Se han utilizado clases para el control de eventos, tipos de media y excepciones, entre otros.	5
f) Se han utilizado clases para la creación y control de animaciones.	6
g) Se han utilizado clases para construir reproductores de contenidos multimedia.....	7
h) Se han depurado y documentado los programas desarrollados.	8

b) Se han reconocido las clases que permiten la captura, procesamiento y almacenamiento de datos multimedia.

Unity dispone de Clases Genéricas para guardar la información de archivos multimedia e introducirlos en un Canvas.

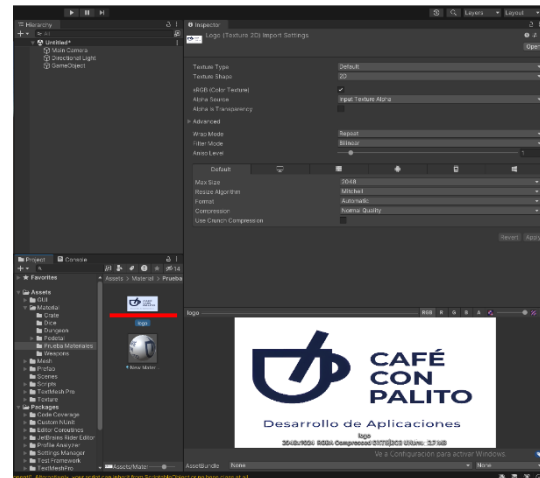
A continuación, se muestra una lista de las clases según el tipo de archivo:

- Imágenes → Image o RAW Image
- Videos → VideoPlayer
- Audio → AudioSource

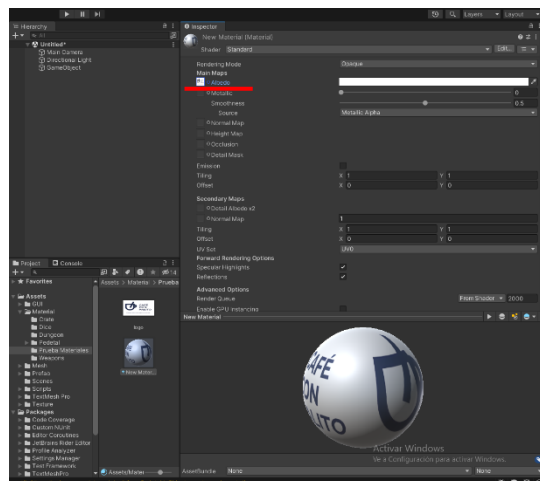


c) Se han utilizado clases para la conversión de datos multimedia de un formato a otro.

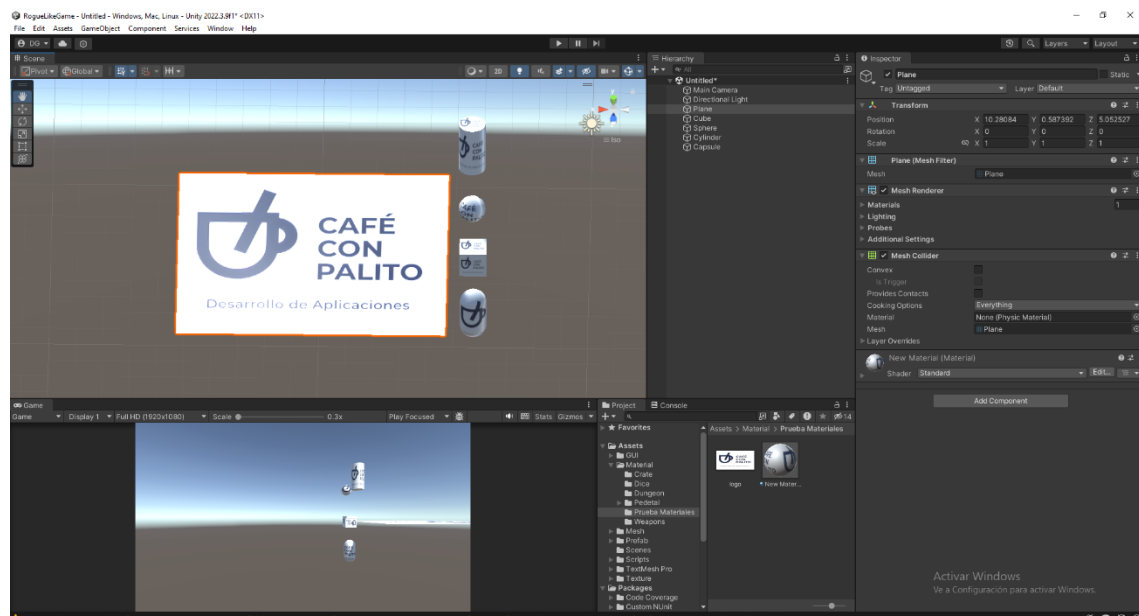
Cargo una imagen PNG a nuestro proyecto de Unity, dentro Assets -> Materials, para mantener el orden del proyecto, en este caso cargo el Logo de nuestra empresa Café Con Palito.



Dentro de los Materials creo un nuevo material standard, al cual le cargo el Logo de la empresa. Este nuevo material lo podre añadir a todos los Objetos que desee.



Ahora puedo utilizar este nuevo material con la imagen para añadirlo a los objetos que tenga en la escena o prefabs. Esto permite reutilizar y modificar de manera más fácil y rápida distintos materiales.

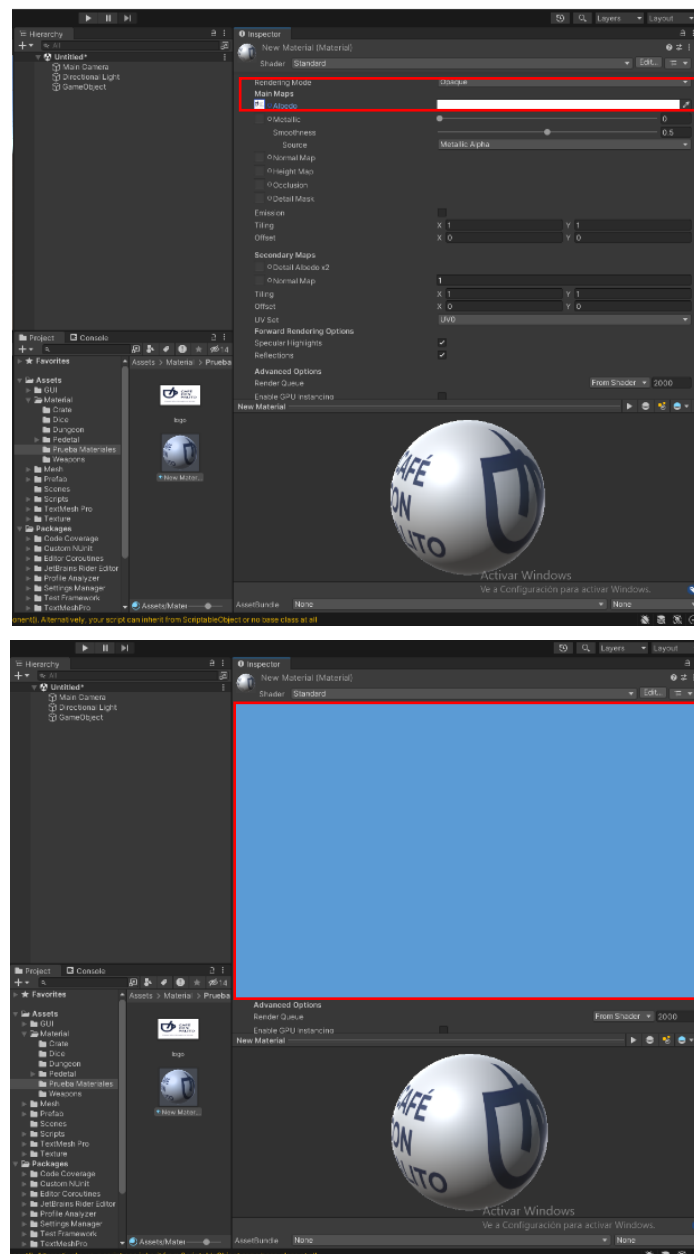


d) Se han utilizado clases para construir procesadores para la transformación de las fuentes de datos multimedia.

Unity tiene integrado en el inspector varias clases que permiten manipular diferentes tipos de archivos multimedia como se ha mencionado en el apartado “c) Se han utilizado clases para la conversión de datos multimedia de un formato a otro”.

Esto nos permite cambiar las propiedades de los archivos de forma sencilla sin tener que recurrir a “picar” código.

Por ejemplo si cargamos una imagen como textura, hay que marcar si se va a aplicar sobre un objeto 2D o 3D, para que se acople perfectamente, pero si dicha imagen la cargamos como material ese problema desaparece porque Unity se encarga de hacerlo, y te añade nuevas funcionalidades.

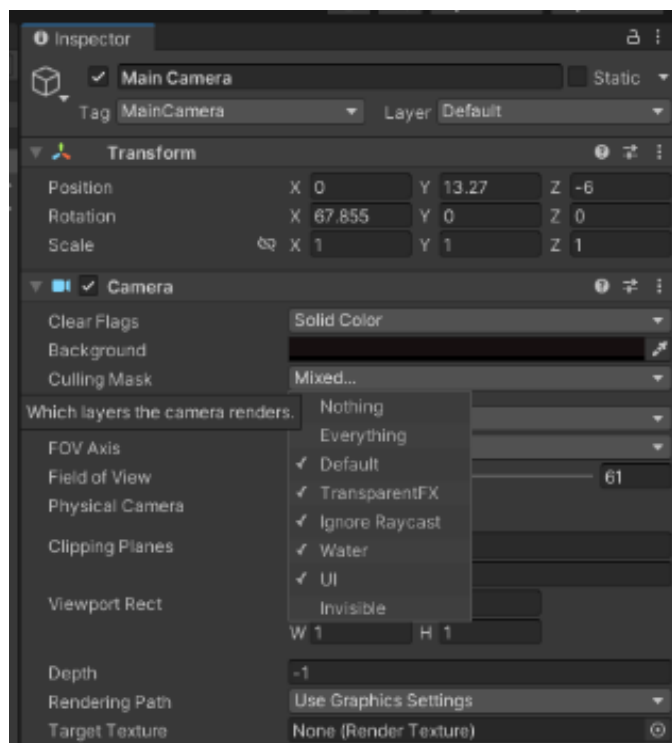
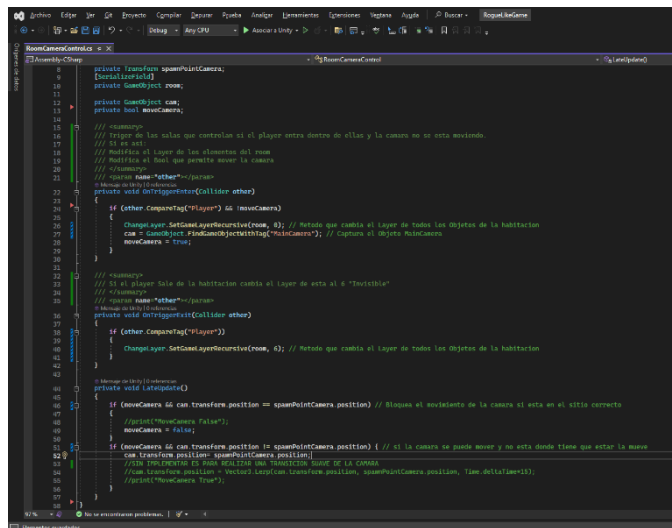


e) Se han utilizado clases para el control de eventos, tipos de media y excepciones, entre otros.

En el Script RoomCameraControl utilizamos OnTriggerEnter y OnTriggerExit para capturar la entrada del Jugador “Player” y controlar los eventos que suceden si esto ocurre.

En este caso controla la posición de la cámara y si esta se puede mover. Además de controlar si el jugador entra o sale de la habitación lo cual llama a un método estático que recibe como parámetros un objeto y un entero que determinara el Layer al que se quiere poner dicho Objeto.

Este método es recursivo por lo cual cambiara de Layer a todos los hijos del Objeto pasado. En función del Layer en el cual se encuentre 0 “Default” o 6 “Invisible”, creado por nosotros, la Main Camara dejara de renderizar utilizando las Culling Mask.



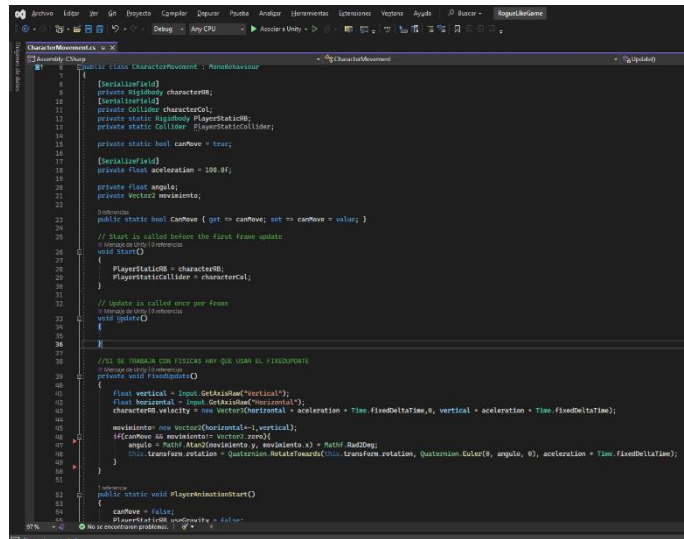
f) Se han utilizado clases para la creación y control de animaciones.

El Script CharacterMovement se ocupa de controlar el movimiento del personaje utilizando los Input Axis de Unity.

Al mover al personaje sobre el espacio esta ira rotando de forma progresiva hasta que apunte en la dirección en la que se mueve.

Unity cuenta con su gestor de animaciones el cual permite montar secuencias de animación de Objetos.

En lo que llevamos realizado de proyecto no hemos podido implementar aun las animaciones para el personaje principal, pero llegaremos a ello en el futuro.



```

using UnityEngine;

public class CharacterMovement : MonoBehaviour
{
    [SerializeField]
    private Rigidbody characterRB;
    [SerializeField]
    private Collider characterCol;
    private static Rigidbody PlayerStaticRB;
    private static Collider PlayerStaticCollider;
    private static bool canMove = true;

    [SerializeField]
    private float acceleration = 100f;
    private float angular;
    private Vector3 movimiento;

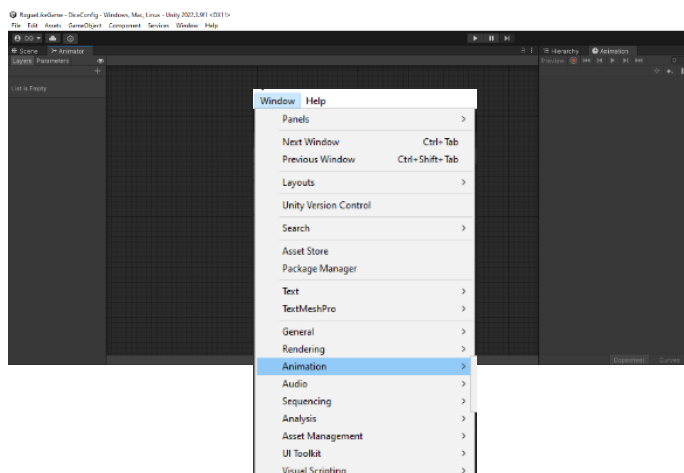
    // Start is called before the first frame update
    void Start()
    {
        PlayerStaticRB = characterRB;
        PlayerStaticCollider = characterCol;
    }

    // Update is called once per frame
    void Update()
    {
        // Si se mueve con físicas hay que usar el FixedUpdate
        // Update en Unity
    }

    // FixedUpdate
    void FixedUpdate()
    {
        float vertical = Input.GetAxisRaw("Vertical");
        float horizontal = Input.GetAxisRaw("Horizontal");
        characterRB.velocity = new Vector3(horizontal, vertical, acceleration * Time.fixedDeltaTime);
        movimiento = new Vector3(horizontal, vertical, 0);
        if (canMove && movimiento.magnitude > 0.1f)
        {
            angular = Mathf.Rad2Deg * Mathf.Atan2(movimiento.x, movimiento.z);
            this.transform.rotation = Quaternion.Euler(0, angular, 0);
        }
    }

    // Inspector
    public static void PlayerMovementStart()
    {
        canMove = false;
        PlayerStaticRB.isKinematic = false;
    }
}

```



g) Se han utilizado clases para construir reproductores de contenidos multimedia.

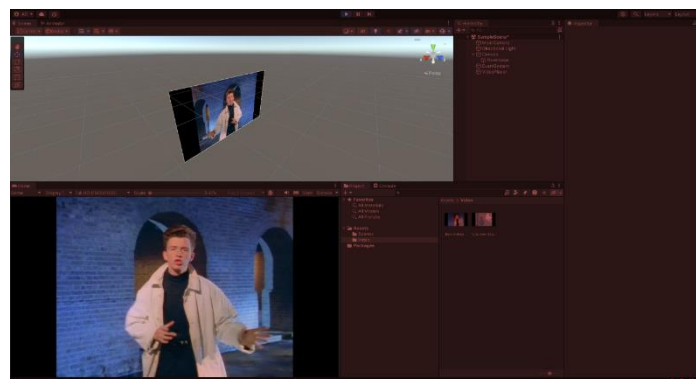
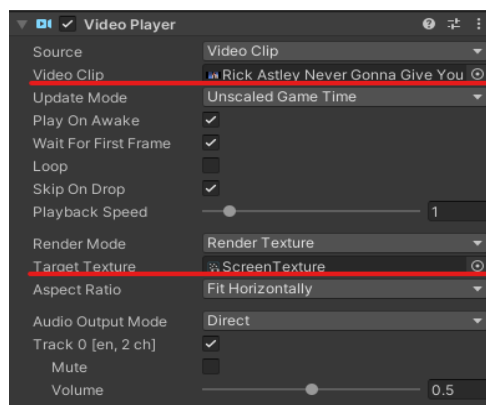
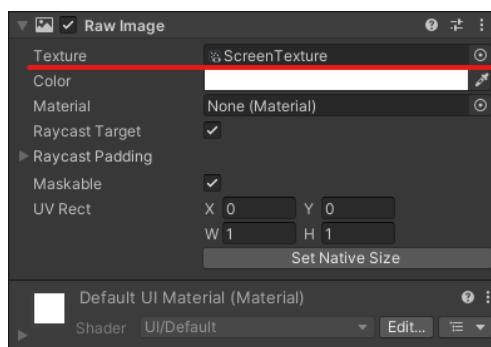
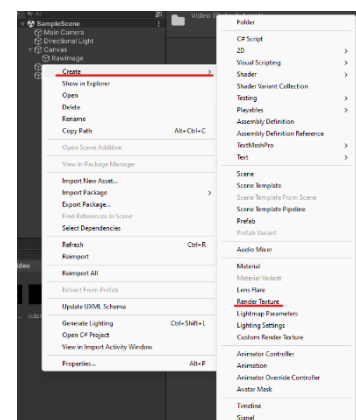
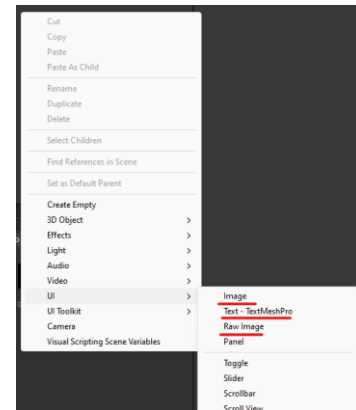
En Unity se puede introducir en el Canvas imágenes, texto o videos, para los dos primeros Unity tiene unos tipos de objetos nativos que permite su manipulación.

Pero para cargar videos, no hay un objeto nativo que lo permita.

Para lograr reproducir un video hay que seguir estos pasos:

- Crear un objeto Raw Image con la dimensión que se desea
- Crear un archivo Render Texture
- Cargar el archivo Render Texture al Raw Image
- Añadir el componente VideoPlayer al Raw Image
- Cargar el video desde local o desde URL al VideoPlayer y el Render Texture.

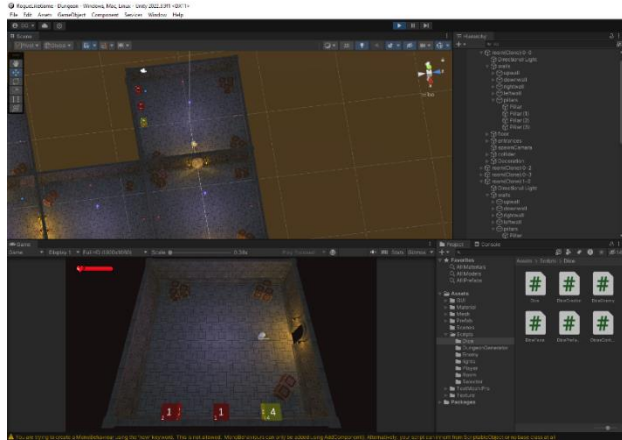
Cada vez que el componente Raw Image se haga visible empezara a reproducir el video cargado.



h) Se han depurado y documentado los programas desarrollados.

En los cambios de habitación cada "Room" tiene un trigger que esta dentro de la habitación contigua.

En principio teníamos un cambio fluido de la cámara y del Jugador, pero a lo largo de ese movimiento el personaje tocaba el trigger de la habitación que abandonaba y la Cámara se quedaba atascada entre las dos habitaciones intentando llegar a su posición, determinada por un SpawnPointCamera, que es un punto prefijado en el espacio que todas las habitaciones "Room".



Spawn Point Camara

Triggers Entradas



Para solucionar este problema optamos por eliminar el movimiento fluido de la cámara y el Jugador. ya que al cambiar de sala todo se queda a oscuras y se vuelve visible al entrar en la nueva habitación. De esta manera solventamos el problema de que la cámara se quede bloqueada.