





This screenshot shows the Visual Studio Code interface with the `PlayerController.cs` script open. The Explorer on the left shows the project structure for `Driving3D`, including `Assets`, `Scripts`, and `Scenes`. The `PlayerController.cs` file is selected in the Explorer and the editor. The script is a `MonoBehaviour` class with the following code:

```
6 public class PlayerController : MonoBehaviour
15 void LateUpdate()
27 {
28     // transform.Rotate(Vector3.up, 5 * Time.deltaTime * 50); // Ajusta la velocidad de rotación con "50"
29     transform.Rotate(Vector3.up, 5); // Ajusta la velocidad de rotación con "50"
30 }
31
32 // Detectar colisión con DeathZone
33
34 0 referencias
35 void OnTriggerEnter(Collider collision)
36 {
37     if (collision.CompareTag("DeathZone"))
38     {
39         PlayerDeath();
40     }
41 }
42 // Reiniciar la escena en caso de muerte
43
44 1 referencia
45 void PlayerDeath()
46 {
47     SceneManager.LoadScene("Prototype 1");
48 }
```

The status bar at the bottom shows the project is `Driving3D.sln` and the current file is `PlayerController.cs`. The bottom right corner shows the time `7:38` and date `3/12/2024`.

This screenshot shows the Visual Studio Code interface with the `GarbageController.cs` script open. The Explorer on the left shows the project structure for `Driving3D`, including `Assets`, `Scripts`, and `Scenes`. The `GarbageController.cs` file is selected in the Explorer and the editor. The script is a `MonoBehaviour` class with the following code:

```
1 using UnityEngine;
2
3 0 referencias
4 public class GarbageController : MonoBehaviour
5 {
6     0 referencias
7     void OnTriggerEnter(Collider collision)
8     {
9         Destroy(collision.gameObject);
10 }
```

The status bar at the bottom shows the project is `Driving3D.sln` and the current file is `GarbageController.cs`. The bottom right corner shows the time `7:37` and date `3/12/2024`.

