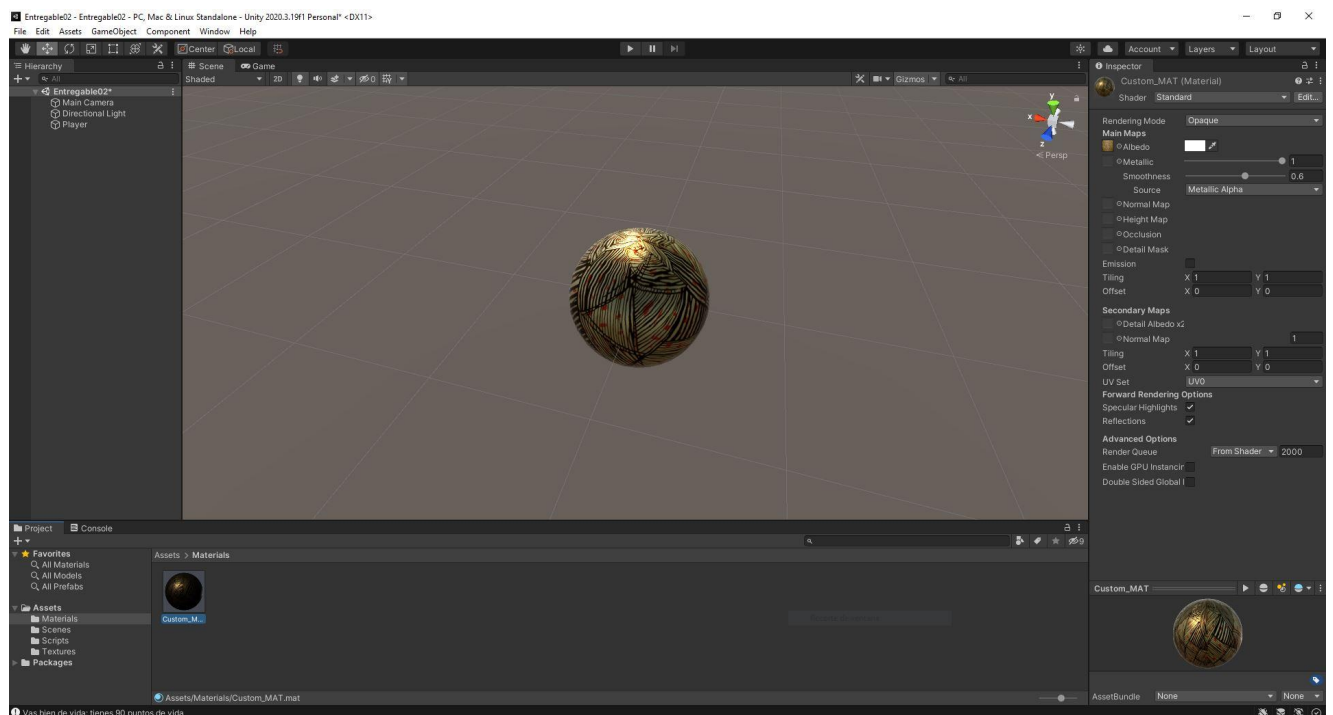


Entregable 02

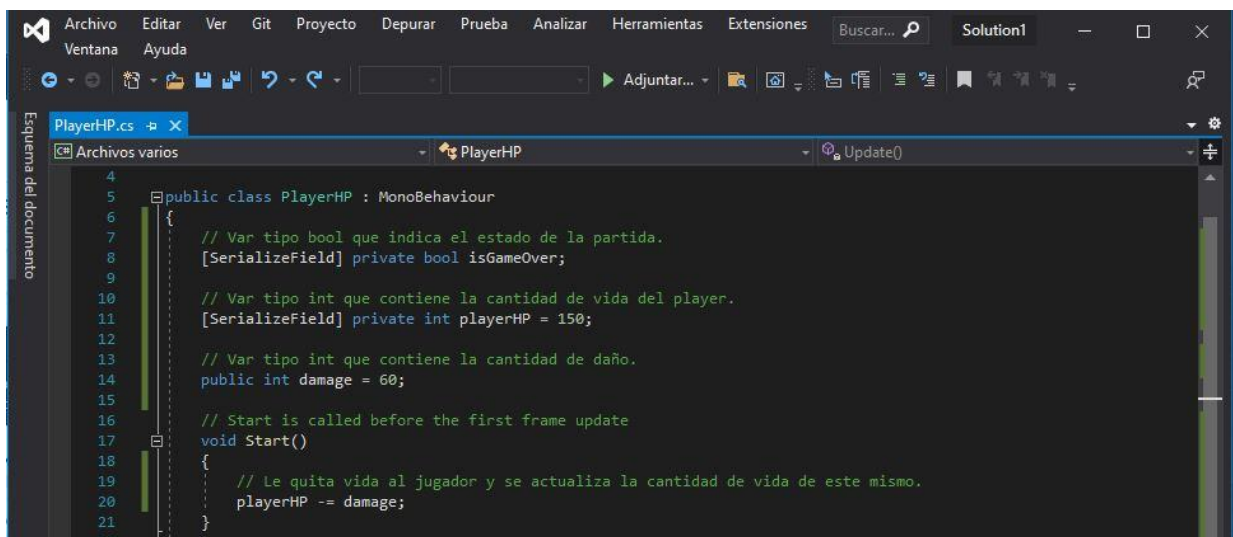
En la escena Entregable 2 debes crear un único Game Object que cumpla los siguientes requisitos:

- Debe ser un cubo, una esfera, una cápsula o un cilindro
- Llámalo Player
- Su posición debe ser (0, 0, 0)
- Su escala debe ser (1, 1, 1)
- Debe tener un material asignado. El material debe ser creado por ti y debe tener una textura asociada (en la propiedad Albedo). Llama al material Custom_Mat



Crea un script y llámalo PlayerHP.

- Asigna el script PlayerHP como componente del Game Object llamado Player dentro del script
- Declara la variable isGameOver, que es una variable privada serializada de tipo booleano sin valor por defecto.
- Declara la variable playerHP, que es una variable privada de tipo entero con valor por defecto 150
- Declara la variable damage, que es una variable pública de tipo entero con valor por defecto 60
- En el Start() debes restar el valor de la variable damage a la variable playerHP y guardar el resultado en la variable playerHP



The screenshot shows the Visual Studio IDE with the 'PlayerHP.cs' file open. The code defines a 'PlayerHP' class that inherits from 'MonoBehaviour'. It includes three private variables: 'isGameOver' (boolean, serialized), 'playerHP' (integer, initialized to 150, serialized), and 'damage' (public integer, initialized to 60). The 'Start()' method is implemented to subtract 'damage' from 'playerHP' at the beginning of the game.

```
4 public class PlayerHP : MonoBehaviour
5 {
6     // Var tipo bool que indica el estado de la partida.
7     [SerializeField] private bool isGameOver;
8
9     // Var tipo int que contiene la cantidad de vida del player.
10    [SerializeField] private int playerHP = 150;
11
12    // Var tipo int que contiene la cantidad de daño.
13    public int damage = 60;
14
15    // Start is called before the first frame update
16    void Start()
17    {
18        // Le quita vida al jugador y se actualiza la cantidad de vida de este mismo.
19        playerHP -= damage;
20    }
21 }
```

En el Update() debes hacer lo siguiente:

- Comprobar si isGameOver toma el valor booleano false. En ese caso: debes comprobar si el valor de la variable playerHP toma un valor menor o igual a 0. En ese caso deberás mostrar por consola el mensaje ***Te has quedado sin vida... GAME OVER***
- Debes usar un operador else if y comprobar si la variable playerHP toma un valor menor estricto a 30. En ese caso deberás mostrar por consola el mensaje ***!!!Tienes poca vida!!!***
- Si las dos condiciones anteriores han resultado ser falsas, debes mostrar por consola el mensaje ***Vas bien de vida: tienes {playerHP} puntos de vida***
- En caso de que la variable isGameOver no tome el valor booleano false, deberás mostrar por consola el mensaje ***GAME OVER***

```
22
23 // Update is called once per frame
24 void Update()
25 {
26
27     // Comprueba si GameOver es FALSE
28     if (!isGameOver)
29     {
30
31         // Comprueba si playerHP es menor o igual a 0
32         if (playerHP <= 0)
33         {
34             Debug.Log("Te has quedado sin vida... GAME OVER");
35         }
36
37         // Comprueba si playerHP es menor a 30
38         else if (playerHP < 30)
39         {
40             Debug.Log("!!!Tienes poca vida!!!");
41         }
42
43         // Muestra la cantidad de vida restante si las comprobaciones anteriores son falsas
44         else
45         {
46             Debug.Log($"Vas bien de vida: tienes {playerHP} puntos de vida");
47         }
48     }
49
50     // Muestra GAME OVER si la comprobación anterior es falsa
51     else
52     {
53         Debug.Log("GAME OVER");
54     }
55 }
56
57
```