





















The screenshot shows the Unity Editor interface. The top menu bar includes Archivo, Editar, Selección, Ver, Ir, Ejecutar, and others. The left sidebar displays the project structure under 'GAMERPG': Assets, Scripts, MonoBehaviours, Scriptable Objects, Sprites, and ESQUEMA. The 'Scriptable Objects' section contains 'HitPoints.cs', which is currently selected and open in the code editor. The code editor window shows the following C# script:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  [CreateAssetMenu(menuName = "HitPoints")]
6  public class HitPoints : ScriptableObject
7  {
8      public float value; //Valor para reflejar en la barra de salud
9  }
10 
```

Below the code editor are tabs for PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL, and PUERTOS. The bottom navigation bar includes buttons for Debug Any CPU, Live Share, Git Graph, and project status indicators (0 errors, 0 warnings). The bottom right corner shows file information: Lín. 10, col. 1, Espacios: 4, UTF-8, CRLF, and a timestamp: 01:56 p. m. 13/11/2024.

The screenshot shows the Unity Editor interface with the following details:

- Menu Bar:** Archivo, Editar, Selección, Ver, Ir, Ejecutar, ...
- Toolbar:** Includes icons for Project, Scene, Find, Undo, Redo, and others.
- Sidebar:** EXPLORADOR (Assets, Scripts, Sprites, etc.), EQUERA, LÍNEA DE TIEMPO, EXPLORADOR DE SOLUCIONES.
- Central Area:** A code editor window titled "Character.cs" (C#). The code is as follows:

```
Assets > Scripts > MonoBehaviours > Character.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  /*
6   * Clase genérica para todo tipo de personaje en el Juego.
7  */
8  public abstract class Character : MonoBehaviour
9 {
10    public HitPoints hitPoints; //Puntos de vida actuales
11    public int maxHitPoints; //Puntos de vida máximos
12 }
```

- Bottom Navigation:** PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL, PUERTOS.
- Bottom Status:** F5 C:\Users\Hp\Desktop\XD\desarrollo-videojuegos-camr\gameRPG>, powershell, and other system status indicators.

The screenshot shows the Unity Editor interface with the code editor open. The project browser on the left shows a folder structure for 'GAMERPG' containing 'Assets', 'Scenes', and 'Scripts'. The 'Scripts' folder contains 'MonoBehaviours' which includes 'Character.cs', 'Consumable.cs', 'HealthBar.cs', 'MovementController.cs', and 'Player.cs'. The code editor displays the 'Player.cs' script, which inherits from 'Character'. It includes methods for initializing a 'HealthBar' prefab and handling item collisions. The status bar at the bottom indicates the code is in C# and is saved.

```
Assets > Scripts > MonoBehaviours > C# Player.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  /*
6   * Clase Player que hereda de Character
7  */
8
9  public class Player : Character
10 {
11     public HealthBar healthBarPrefab; //Referencia HealthBar Prefab
12     private HealthBar healthBar; //Copia de referencia de HealthBar Prefab
13
14     void Start()
15     {
16         healthBar = Instantiate(healthBarPrefab); //Instanciar HealthBar
17         healthBar.character = this; //referencia del Player en HealthBar
18     }
19
20     public void OnTriggerEnter2D(Collider2D collision)
21     {
22         if (collision.gameObject.CompareTag("CanBePickedUp"))
23         {
24             Item hitObject = collision.gameObject.GetComponent<Consumable>().item;
25
26             if (hitObject != null)
27             {
28                 Debug.Log("Nombre: " + hitObject.objectName);
29                 bool shouldDisappear = false;
30
31                 switch (hitObject.itemType)
32                 {
33                     case Item.ItemType.HEALTH://Barra de Salud
34                         Debug.Log("Cantidad a Incrementar: " + hitObject.quantity);
35                         shouldDisappear = AdjustHitPoints(hitObject.quantity);
36                         break;
37
38                     if (shouldDisappear)
39                     {
40                         collision.gameObject.SetActive(false); //Desaparecer
41                     }
42
43                 }
44             }
45         }
46     }
47
48     private bool AdjustHitPoints(int amount)
49     {
50         if (hitPoints.value < maxHitPoints) // no se puede exceder el máximo de puntos
51         {
52             hitPoints.value = hitPoints.value + amount;
53             print("Ajustando Puntos: " + amount + ". Nuevo Valor: " + hitPoints.value);
54             return true; //Fue modificado
55         }
56         return false; //No se modifica entonces el Heart no desaparece
57     }
58
59 }
60
```

This screenshot shows the continuation of the 'Player.cs' script. It includes a method 'AdjustHitPoints' that increments the player's hit points up to a maximum. The code editor shows syntax highlighting for C# and the status bar indicates the code is in C# and is saved.

```
private bool AdjustHitPoints(int amount)
{
    if (hitPoints.value < maxHitPoints) // no se puede exceder el máximo de puntos
    {
        hitPoints.value = hitPoints.value + amount;
        print("Ajustando Puntos: " + amount + ". Nuevo Valor: " + hitPoints.value);
        return true; //Fue modificado
    }
    return false; //No se modifica entonces el Heart no desaparece
}
```

The screenshot shows the Visual Studio IDE interface. The title bar says "gameRPG". The left sidebar is the "EXPLORADOR" (Explorer) showing the project structure under "GAMERPG". The main editor window displays the "HealthBar.cs" script:

```
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI; //Componentes Interfaz Gráfica
5
6  public class HealthBar : MonoBehaviour
7  {
8      [HideInInspector]
9      public Player character; //Referencia al jugador
10     public Image meterImage; //Medidor Meter de la salud
11     public Text hpText; //Texto en barra de salud
12
13     void Start()
14     {
15         character.hitPoints.value = 0;
16     }
17
18     void Update()
19     {
20         if (character != null)
21         {
22             //Modifica barra de salud
23             meterImage.fillAmount = character.hitPoints.value / character.maxHitPoints;
24             //Texto a mostrar
25             hpText.text = "HP:" + (meterImage.fillAmount * 100);
26         }
27     }
28 }
```

The status bar at the bottom shows "Lin. 28, col. 2 Espacio: 4 UTF-8 CRLF C#".





