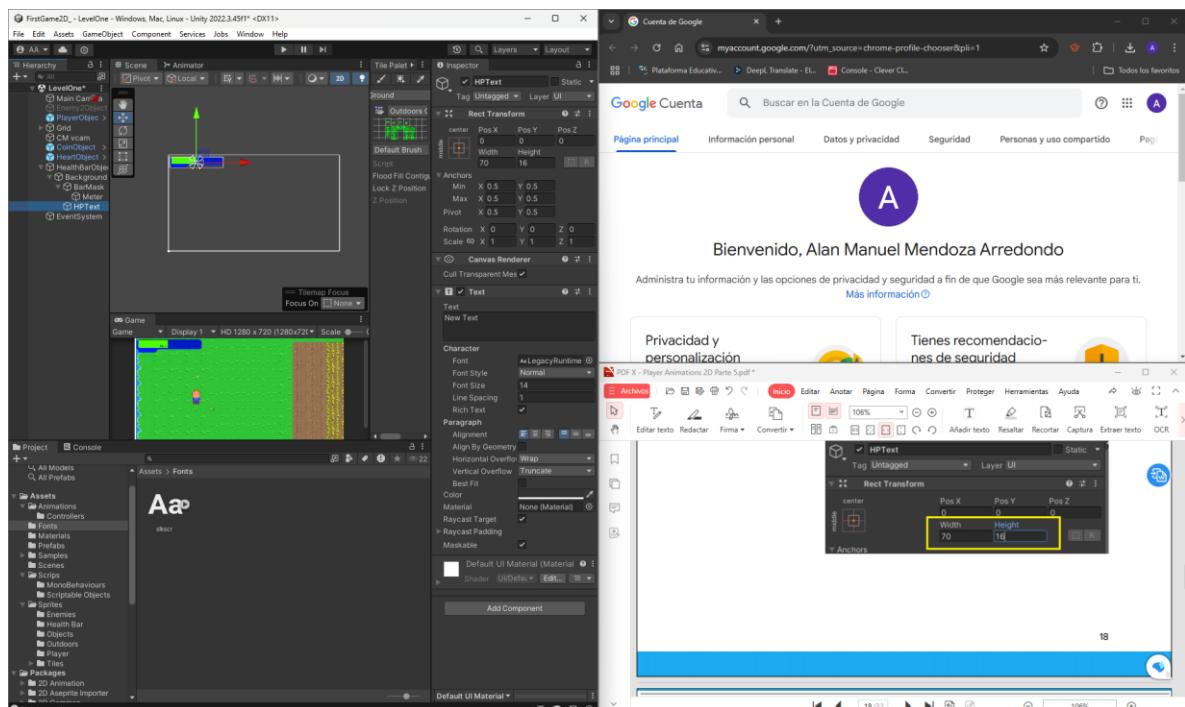


Bienvenido, Alan Manuel Mendoza Arredondo

Administra tu información y las opciones de privacidad y seguridad a fin de que Google sea más relevante para ti.

Más información

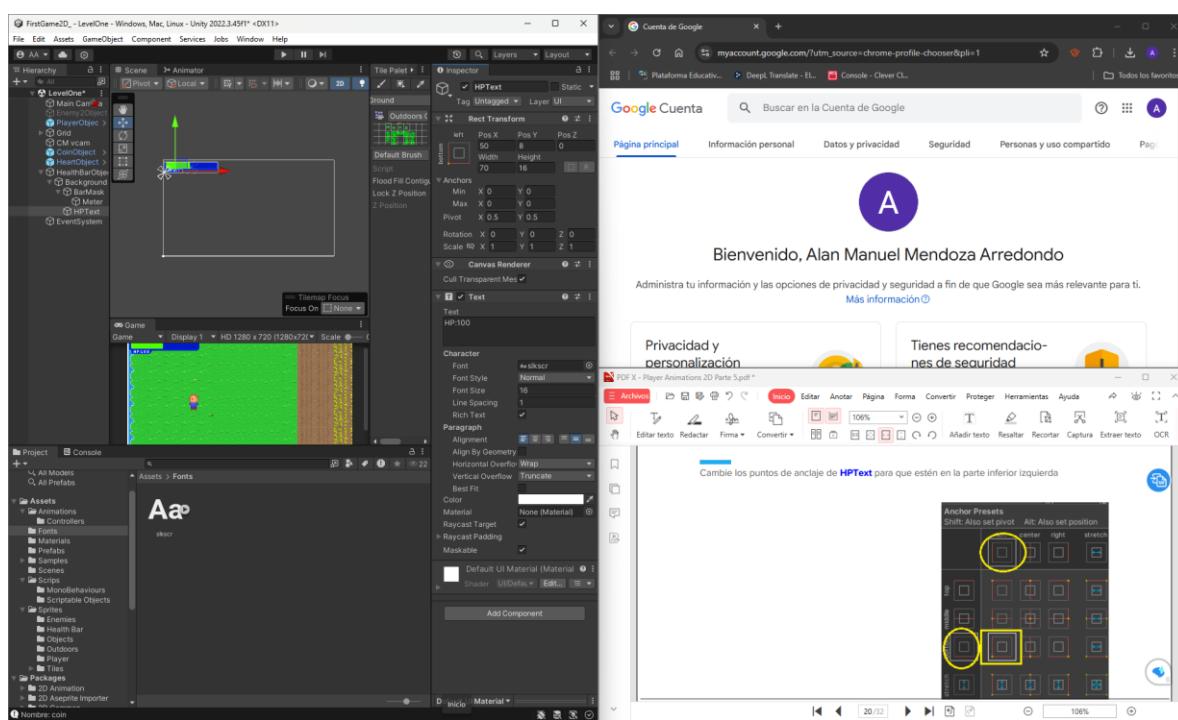
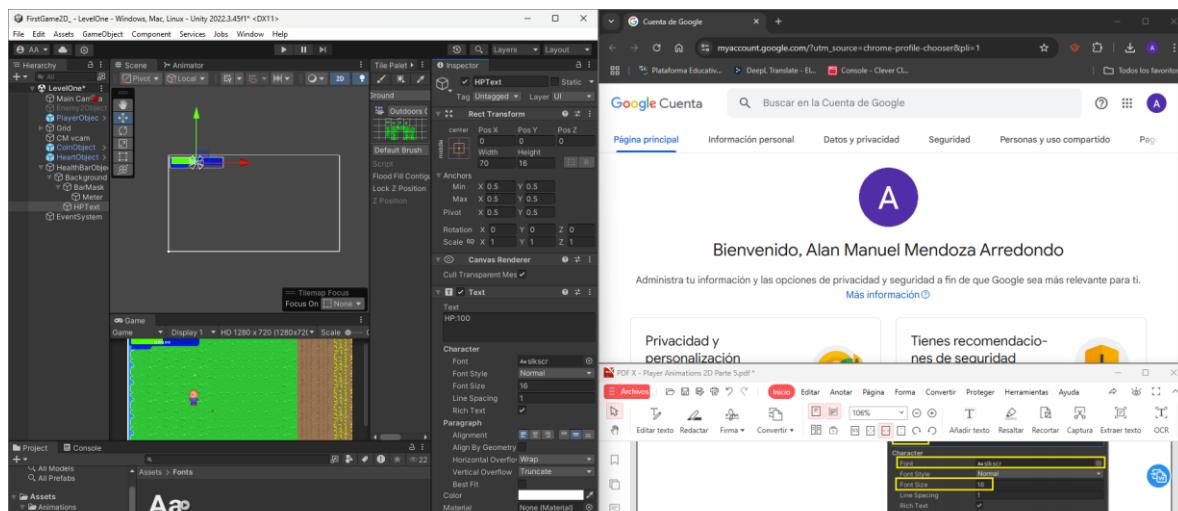
En el componente Rect Transform de HPTText, establece el Ancho en: 70, y Altura a: 16. En el componente Texto de HPTText, cambia el Tamaño de fuente a 16 y el Color a blanco. Cambia la fuente a "silscr", que es la fuente personalizada que acabamos de importar. Establecer el párrafo horizontal y vertical Alineación a la izquierda y al centro, respectivamente.

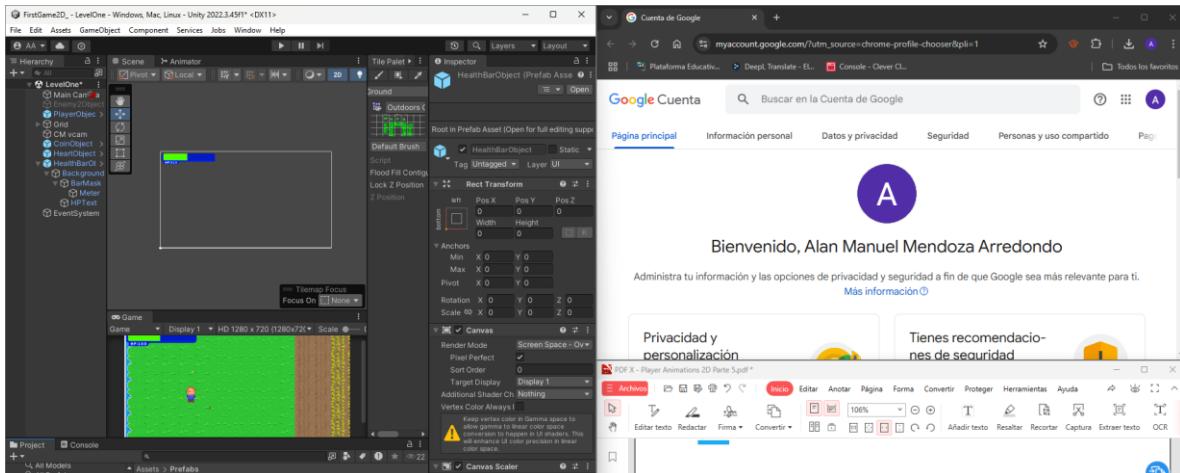


Bienvenido, Alan Manuel Mendoza Arredondo

Administra tu información y las opciones de privacidad y seguridad a fin de que Google sea más relevante para ti.

Más información

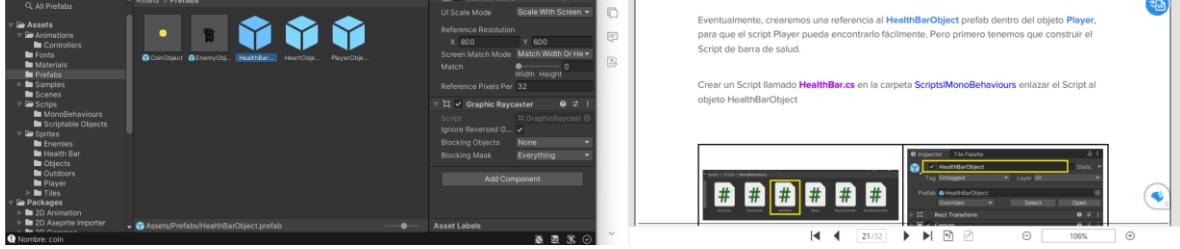




Bienvenido, Alan Manuel Mendoza Arredondo

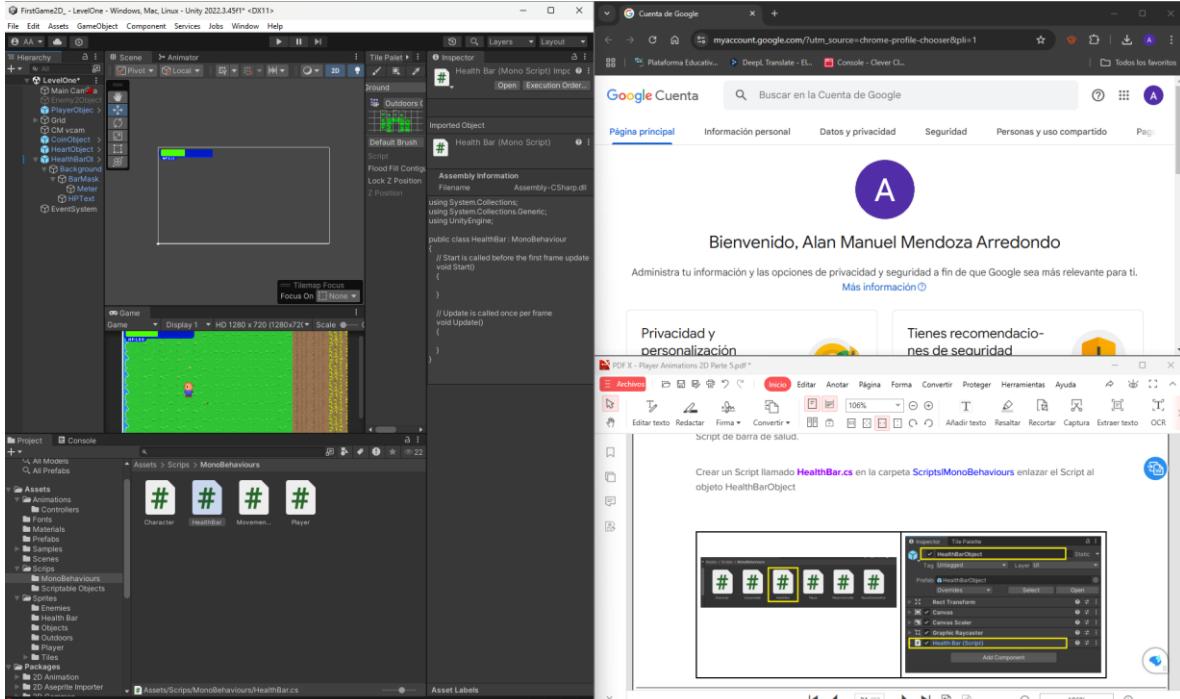
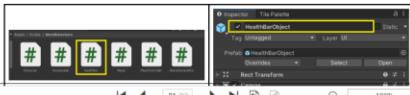
Administra tu información y las opciones de privacidad y seguridad a fin de que Google sea más relevante para ti.

Más información



Eventualmente, crearemos una referencia al **HealthBarObject** prefab dentro del objeto **Player**, para que el script Player pueda encontrarlo fácilmente. Pero primero tenemos que construir el Script de barra de salud.

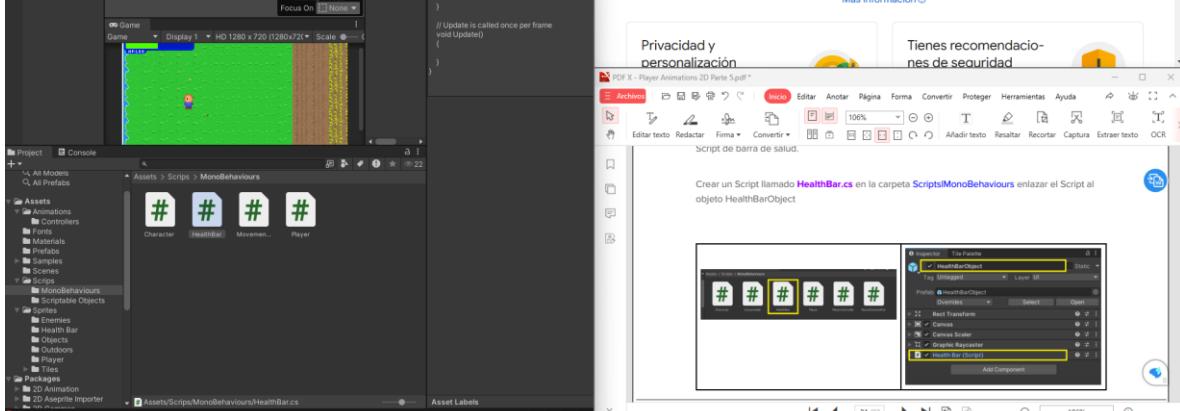
Crear un Script llamado **HealthBar.cs** en la carpeta **Scripts/MonoBehaviours** enlazar el Script al objeto **HealthBarObject**



Bienvenido, Alan Manuel Mendoza Arredondo

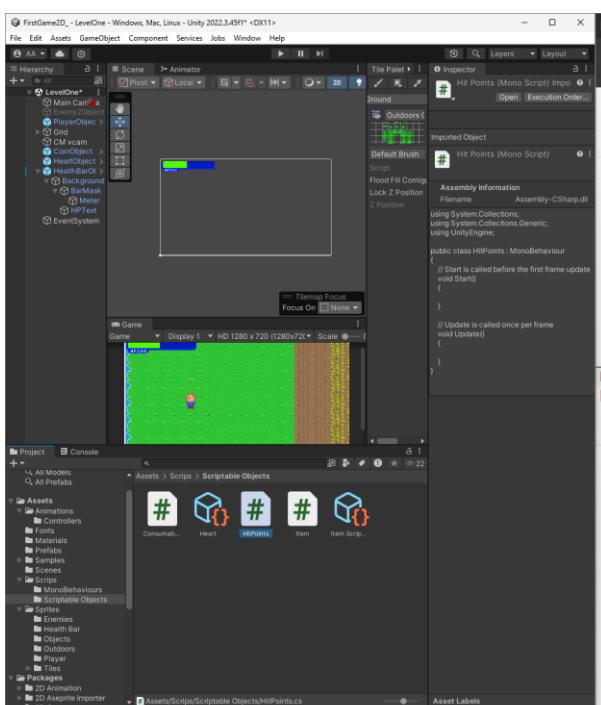
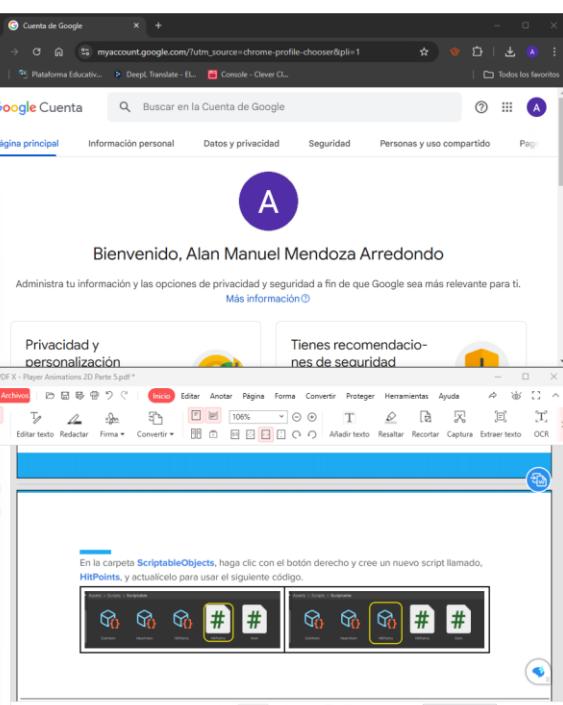
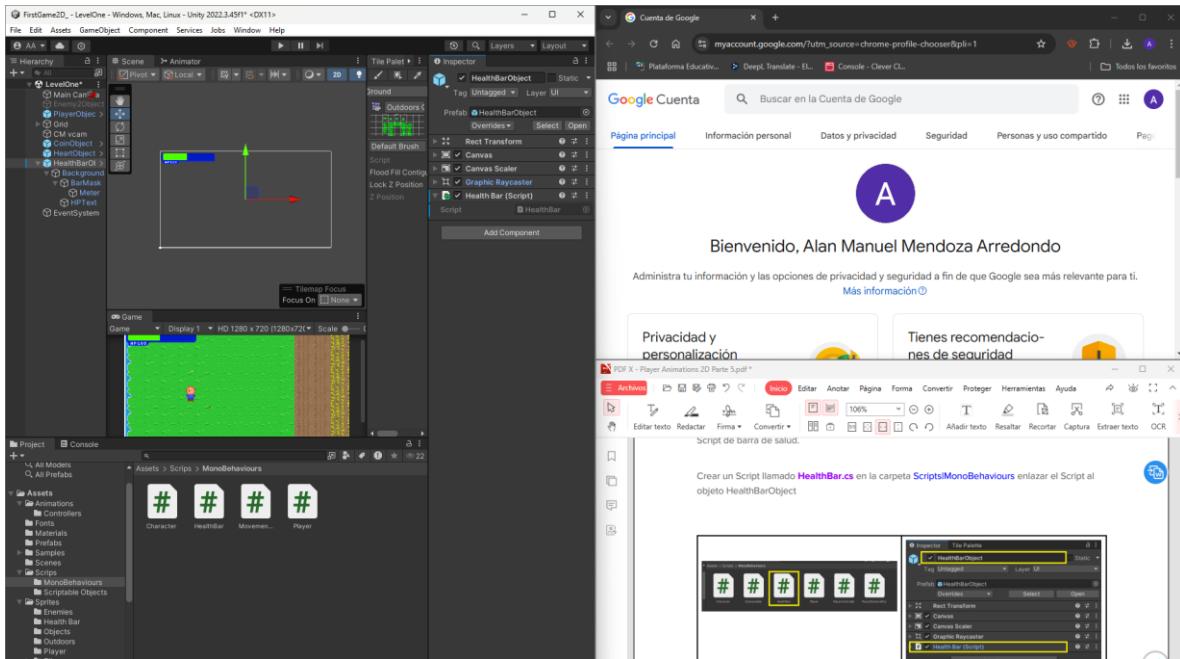
Administra tu información y las opciones de privacidad y seguridad a fin de que Google sea más relevante para ti.

Más información

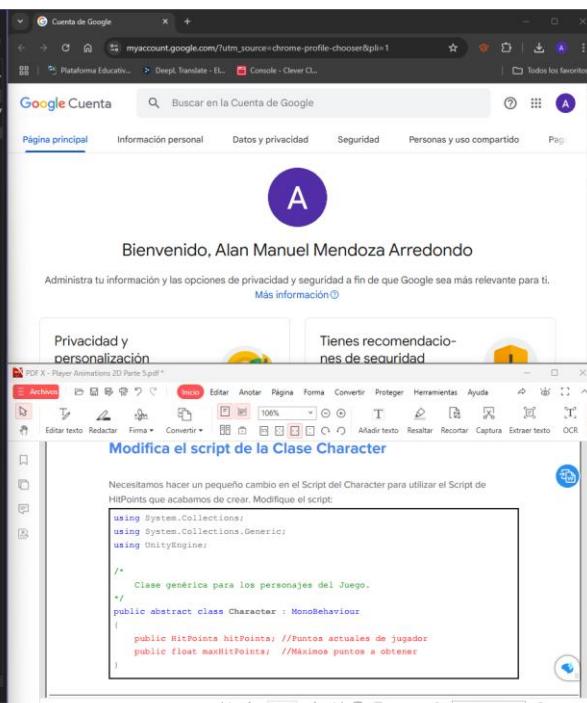


Crear un Script llamado **HealthBar.cs** en la carpeta **Scripts/MonoBehaviours** enlazar el Script al objeto **HealthBarObject**





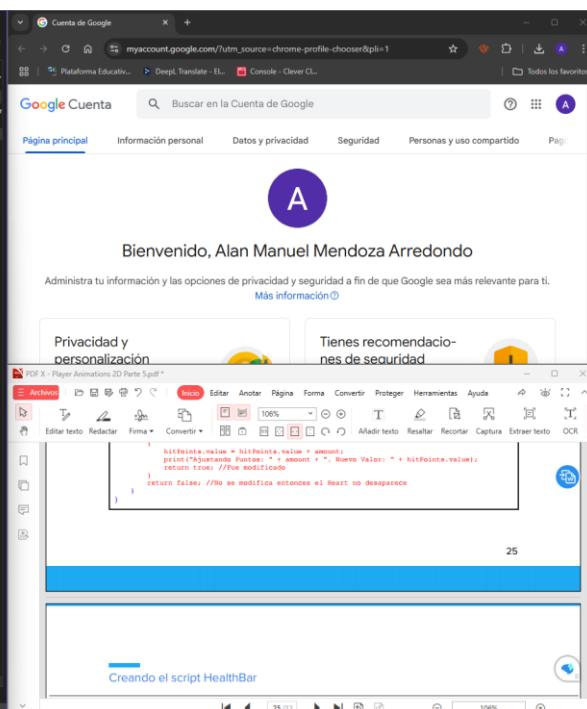
```
HPPoints.cs Characters
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  // Clase genérica para los personajes del Juego.
6
7  public abstract class Character : MonoBehaviour
8  {
9      public HitPoints hitPoints; //Puntos actuales de jugador
10     public float maxHitPoints; //Máximos puntos a obtener
11 }
12
```

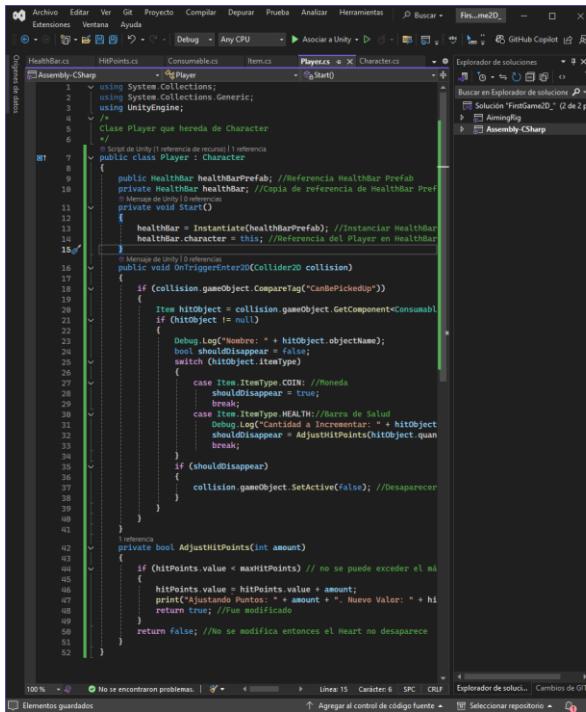


The screenshot shows the Unity Editor interface with the following details:

- Top Bar:** Archivo, Editor, Ver, Gt, Proyecto, Compilar, Depurar, Prueba, Analizar, Herramientas, Buscar, Fins...ms2D, GitHub Copilot.
- Left Sidebar:** Extensiones, Ventana, Ayuda.
- Central Area:** Assembly Sharp tab selected. The code for the `Player` class is displayed:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  ...
5  Classe Player que hereda de Character
6  ...
7  public class Player : Character
8  {
9      public HealthBar healthBarPrefab; //Referencia HealthBar Prefab
10     private HealthBar healthBar; //Copia de referencia de HealthBar Prefab
11     ...
12     void Start()
13     {
14         healthBar = Instantiate(healthBarPrefab); //Instanciar HealthBar
15         healthBar.character = this; //Referencia del Player en HealthBar
16     }
17     ...
18     void OnTriggerEnter2D(Collider2D collision)
19     {
20         if (collision.gameObject.CompareTag("CanBePickedUp"))
21         {
22             Item hitObject = collision.gameObject.GetComponent<Consumable>();
23             if (hitObject != null)
24             {
25                 Debug.Log("Nombre: " + hitObject.gameObject.name);
26                 bool shouldDisappear = false;
27                 switch (hitObject.itemType)
28                 {
29                     case Item.ItemType.COIN: //Moneda
30                         shouldDisappear = true;
31                         break;
32                     case Item.ItemType.HEALTH://Barra de Salud
33                         Debug.Log("Cantidad a Incrementar: " + hitObject.value);
34                         shouldDisappear = AdjustHitPoints(hitObject.value);
35                         break;
36                     case Item.ItemType.GOLD://Moneda de Oro
37                         shouldDisappear = true;
38                         break;
39                 }
40             }
41         }
42         ...
43         Tolerancia
44         private bool AdjustHitPoints(int amount)
45         {
46             if (hitPoints.value < maxHitPoints) // no se puede exceder el má
47             {
48                 hitPoints.value = hitPoints.value + amount;
49                 print("Ajustando Puntos: " + amount + ". Nuevo Valor: " + hi
50             }
51         }
52     }
53 }
```
- Bottom Status Bar:** 100%, 0, 1, 0, 0, 0, 0, Linea: S2, Carácter: 2, SPC, CR/LF.
- Bottom Right:** Explorador de soluciones, Explorador de sol., Cambios de...





```

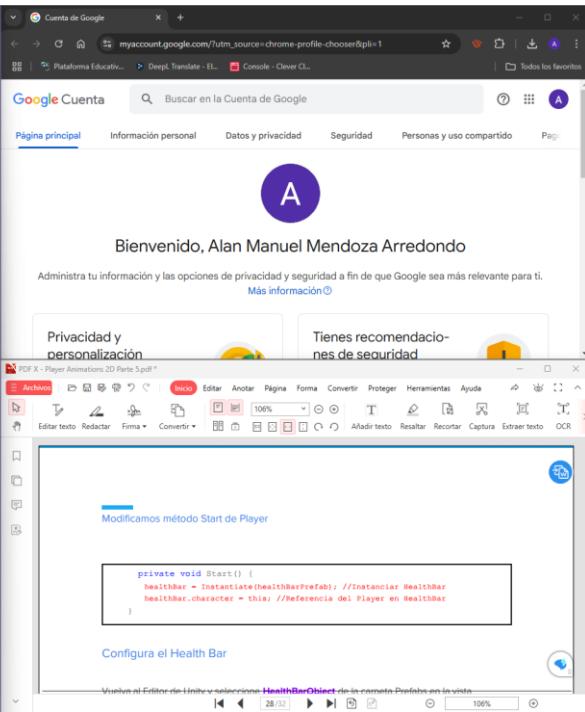
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : Character
{
    public HealthBar healthBarPrefab; //Referencia HealthBar Prefab
    private HealthBar healthBar; //Copia de referencia de HealthBar Prefab
    private void Start()
    {
        healthBar = Instantiate(healthBarPrefab); //Instanciar healthbar
        healthBar.character = this; //Referencia del Player en healthbar
    }

    void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.gameObject.CompareTag("CanDiePickedUp"))
        {
            HitObject hitObject = collision.gameObject.GetComponent<HitObject>();
            if (hitObject != null)
            {
                Debug.Log("Nombre: " + hitObject.objectName);
                bool shouldDisappear = false;
                switch (hitObject.itemType)
                {
                    case ItemType.COIN: //Moneda
                        shouldDisappear = true;
                        break;
                    case ItemType.HEALTH://Barra de Salud
                        Debug.Log("Cantidad a Incrementar: " + hitObject.quantity);
                        shouldDisappear = AdjustHitPoints(hitObject.quantity);
                        break;
                }
                if (shouldDisappear)
                {
                    collision.gameObject.SetActive(false); //Desaparecer
                }
            }
        }
    }

    private bool AdjustHitPoints(int amount)
    {
        if (hitPoints.value > maxHitPoints) //No se puede exceder el má
        {
            hitPoints.value = hitPoints.value + amount;
            print("Ajustando Puntos: " + amount + ". Nuevo Valor: " + hi
            return true; //Se modificó
        }
        return false; //No se modifica entonces el Heart no desaparece
    }
}

```



Bienvenido, Alan Manuel Mendoza Arredondo

Administrá tu información y las opciones de privacidad y seguridad a fin de que Google sea más relevante para ti.

Más información

Privacidad y personalización

Tienes recomendaciones de seguridad.

PDF X - Player Animations 2D Parte 5.pdf\*

Archivo Editar Anotar Página Forma Convertir Proteger Herramientas Ayuda

Modificamos método Start de Player

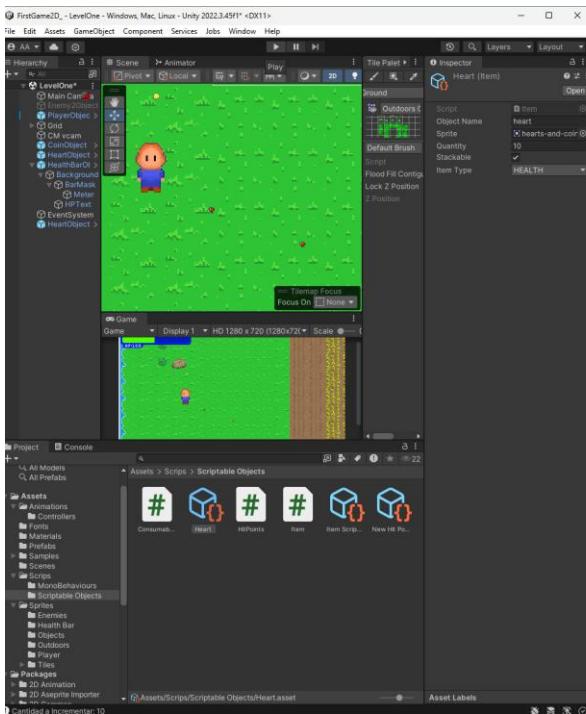
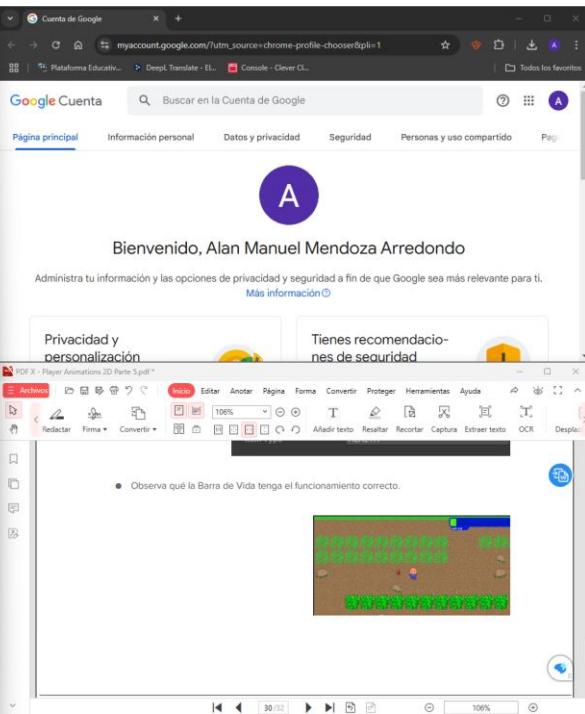
```

private void Start()
{
    healthbar = Instantiate(healthBarPrefab); //Instanciar Healthbar
    healthbar.character = this; //Referencia del Player en Healthbar
}

```

Configura el Health Bar

Volvemos al Editor de Unity y seleccionamos HealthBarObject de la lista Prefabs en la vista.

Bienvenido, Alan Manuel Mendoza Arredondo

Administrá tu información y las opciones de privacidad y seguridad a fin de que Google sea más relevante para ti.

Más información

Privacidad y personalización

Tienes recomendaciones de seguridad.

PDF X - Player Animations 2D Parte 5.pdf\*

Archivo Editar Anotar Página Forma Convertir Proteger Herramientas Ayuda

- Observa qué la Barra de Vida tenga el funcionamiento correcto.

● Observa qué la Barra de Vida tenga el funcionamiento correcto.

100%