

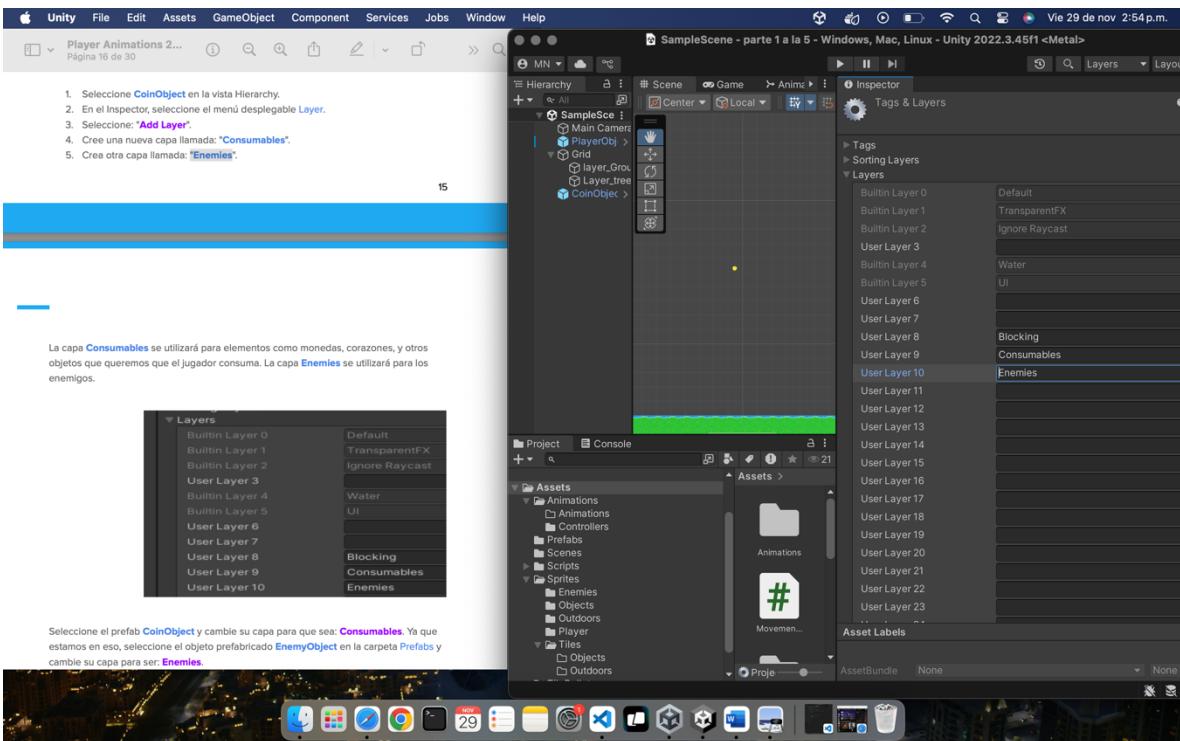
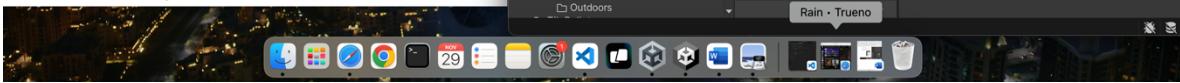
Detección de colisión basada en capas

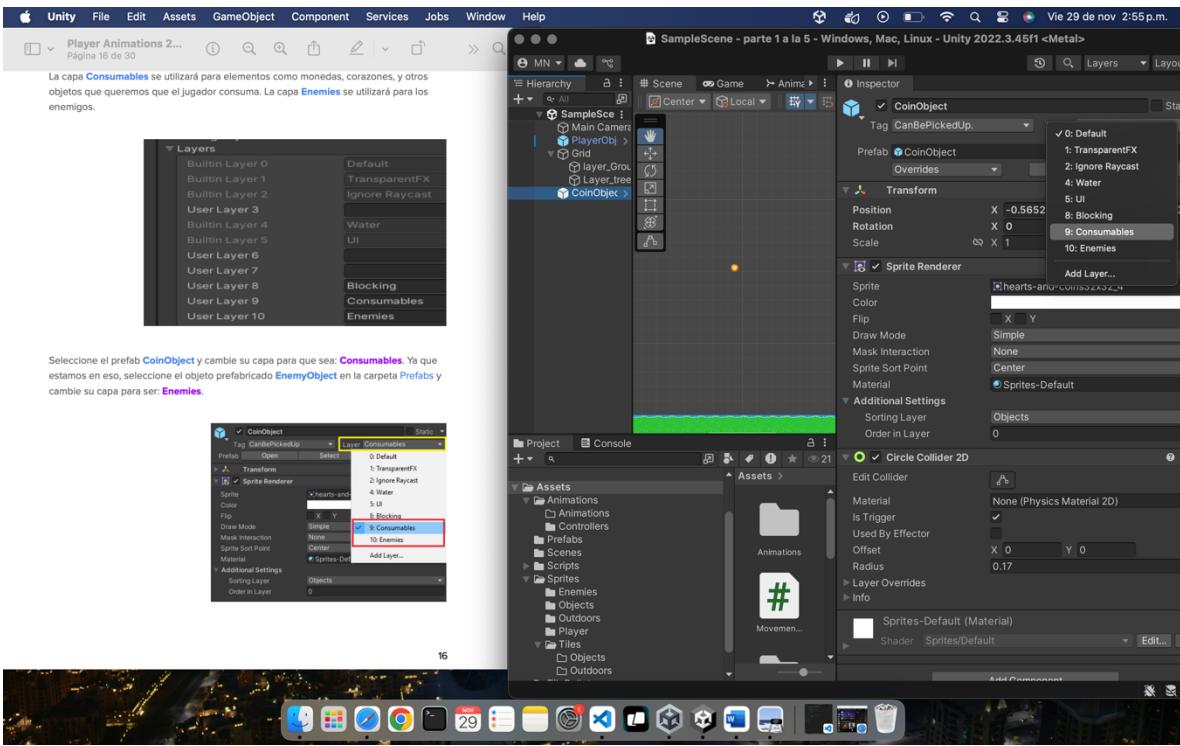
Queremos darle al jugador la capacidad de recoger monedas caminando sobre ellas. Nuestro juego también tendrá enemigos caminando por el mapa, pero queremos que los enemigos caminen a través de las monedas sin recogerlas.

Las capas se utilizan para definir colecciones de GameObjects. Los componentes de Collider que se adjuntan a GameObjects en la misma capa se conocerán entre sí y podrán interactuar. Podemos crear lógica basada en estas interacciones para hacer cosas como recoger objetos.

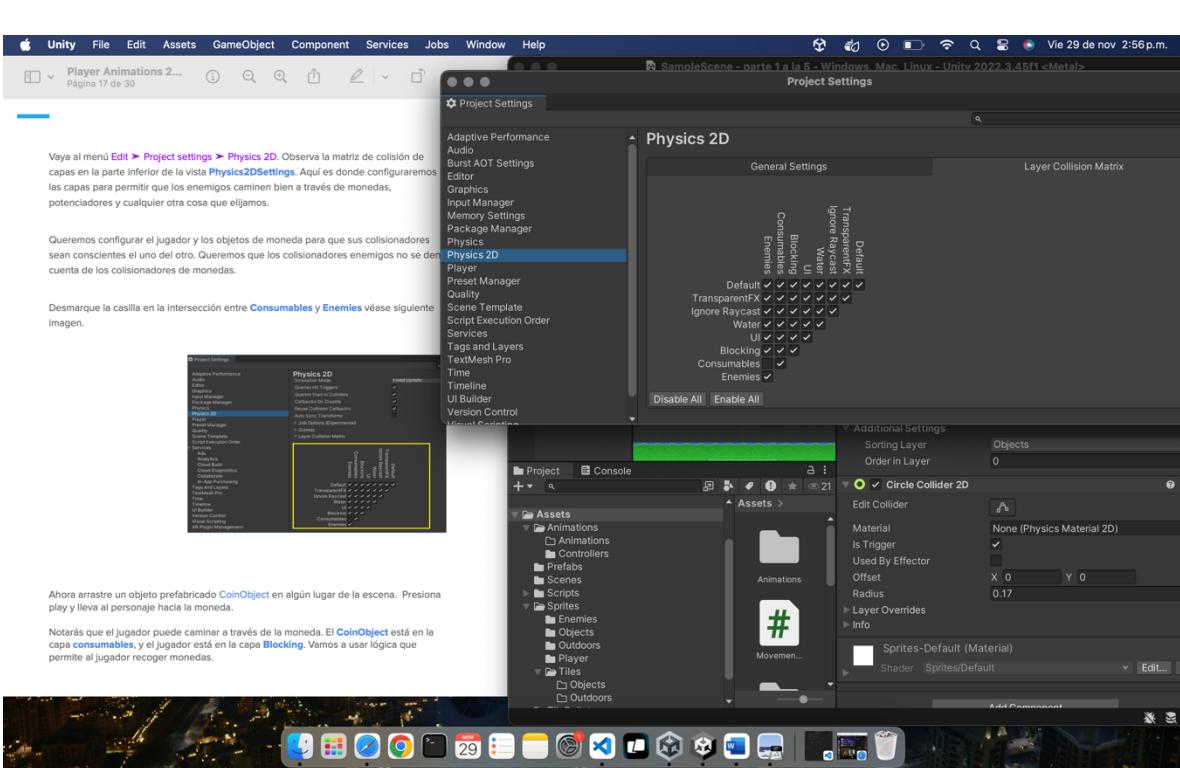
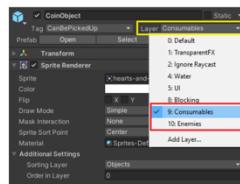
También existe una técnica para hacer componentes Collider en diferentes capas conscientes unas de otras. Este enfoque utiliza una función de Unity llamada **Detección de colisiones** basada en capas. Para ver esta función en acción, primero debemos crear y asignar capas a los GameObjects relevantes.

1. Seleccione CoinObject en la vista Hierarchy.
2. En el Inspector, seleccione el menú desplegable Layer.
3. Seleccione "Add Layer".
4. Cree una nueva capa llamada: "Consumables".





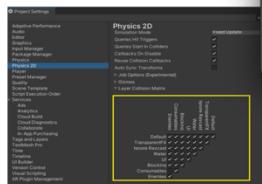
Selecione el prefab **CoinObject** y cambie su capa para que sea: **Consumables**. Ya que estamos en eso, seleccione el objeto prefabricado **EnemyObject** en la carpeta **Prefabs** y cambie su capa para ser: **Enemies**.



Vaya al menú **Edit > Project settings > Physics 2D**. Observa la matriz de colisión de capas en la parte inferior de la vista **Physics2DSettings**. Aquí es donde configuraremos las capas para permitir que los enemigos caminen bien a través de monedas, potenciadores y cualquier otra cosa que elijamos.

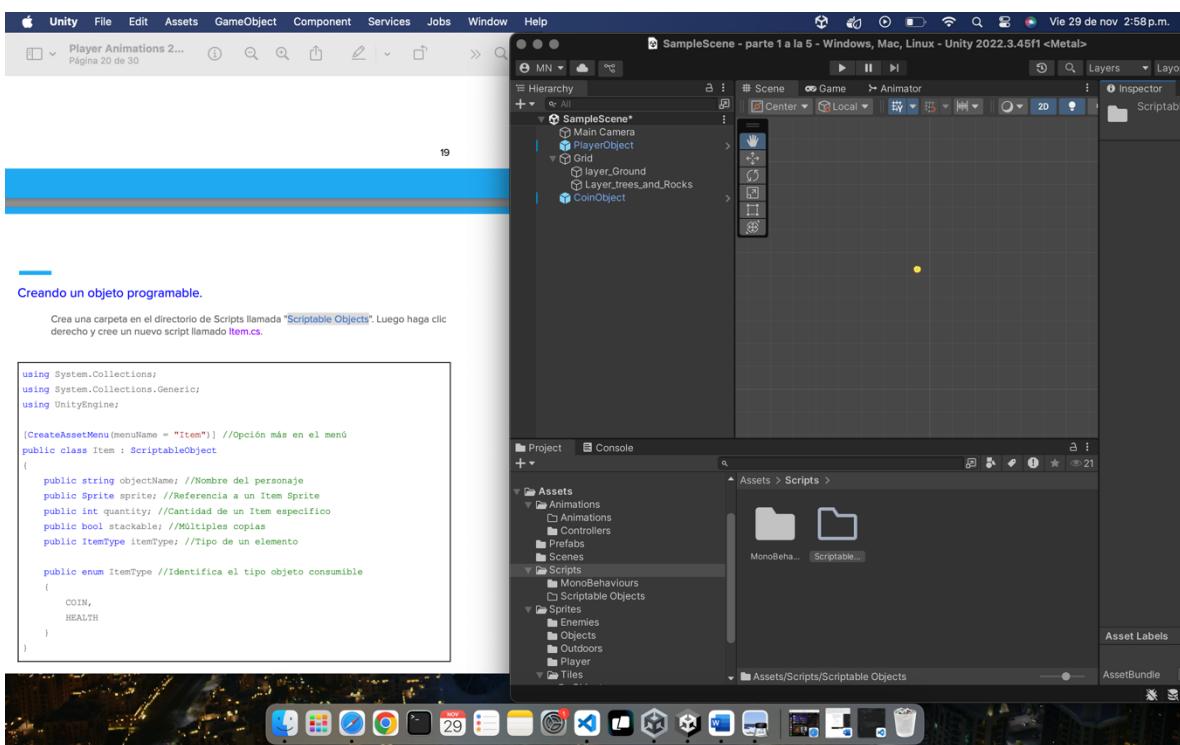
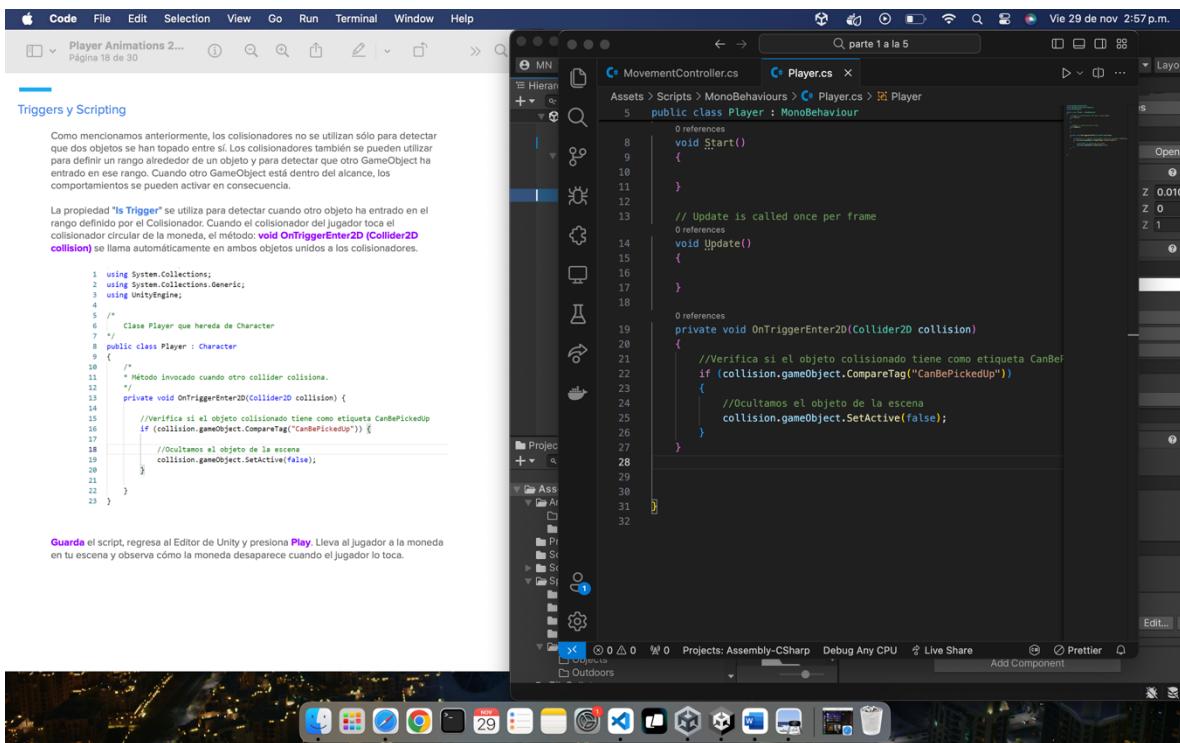
Queremos configurar el jugador y los objetos de moneda para que sus colisionadores sean conscientes el uno del otro. Queremos que los colisionadores enemigos no se den cuenta de los colisionadores de monedas.

Desmarque la casilla en la Intersección entre **Consumables** y **Enemies** véase siguiente imagen.



Ahora arrastra un objeto prefabricado **CoinObject** en algún lugar de la escena. Presiona play y lleva al personaje hacia la moneda.

Notarás que el jugador puede caminar a través de la moneda. El **CoinObject** está en la capa **consumables**, y el jugador está en la capa **Blocking**. Vamos a usar lógica que permite al jugador recoger monedas.



Creando un objeto programable.

Crea una carpeta en el directorio de Scripts llamada "Scriptable Objects". Luego hace clic derecho y crea un nuevo script llamado `Item.cs`.

```

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

[CreateAssetMenu(menuName = "Item")] //Opción más en el menú
public class Item : ScriptableObject
{
    public string objectName; //Nombre del personaje
    public Sprite sprite; //Referencia a un Item Sprite
    public int quantity; //Cantidad de un Item específico
    public bool stackable; //Múltiples copias
    public ItemType itemType; //Tipo de un elemento

    public enum ItemType //Identifica el tipo objeto consumible
    {
        COIN,
        HEALTH
    }
}

```

Se crea una entrada en el submenú Create

Vie 29 de nov 3:00 p.m.

Estas instancias de objetos programables se almacenan en el proyecto como archivos de activos separados y sus propiedades se pueden modificar a través del propio objeto vía la vista Inspector.

Construyendo un script `Consumable.cs`

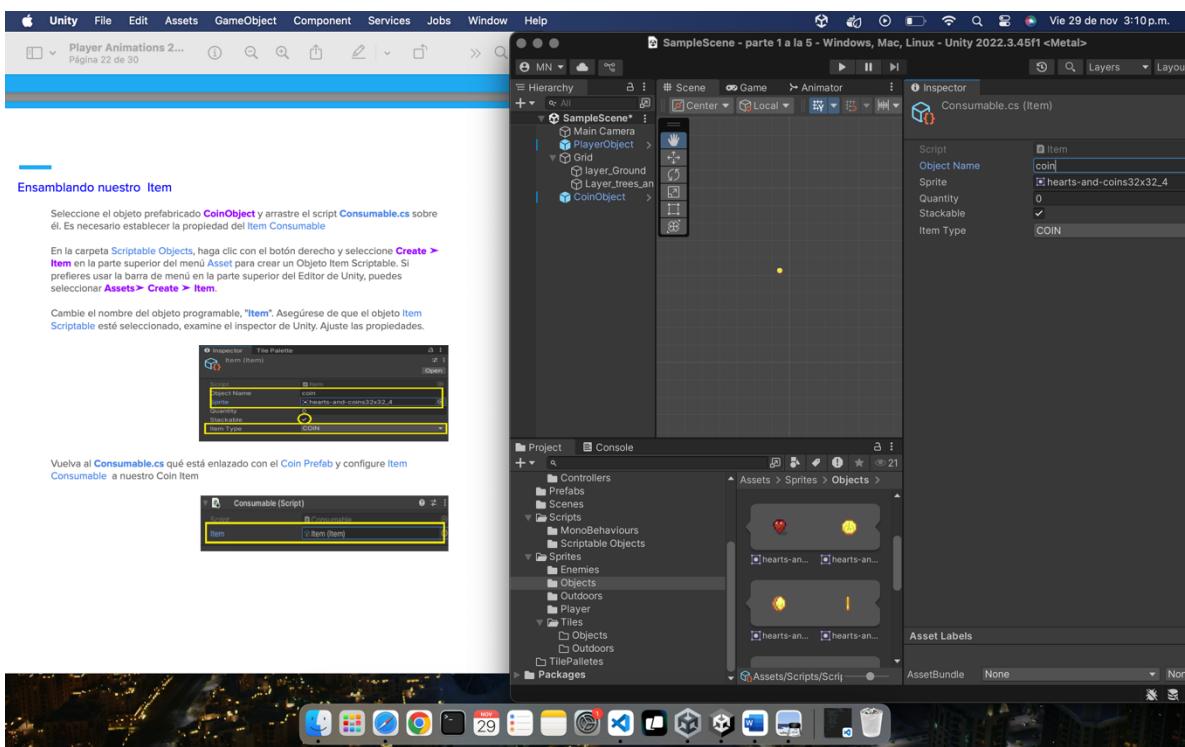
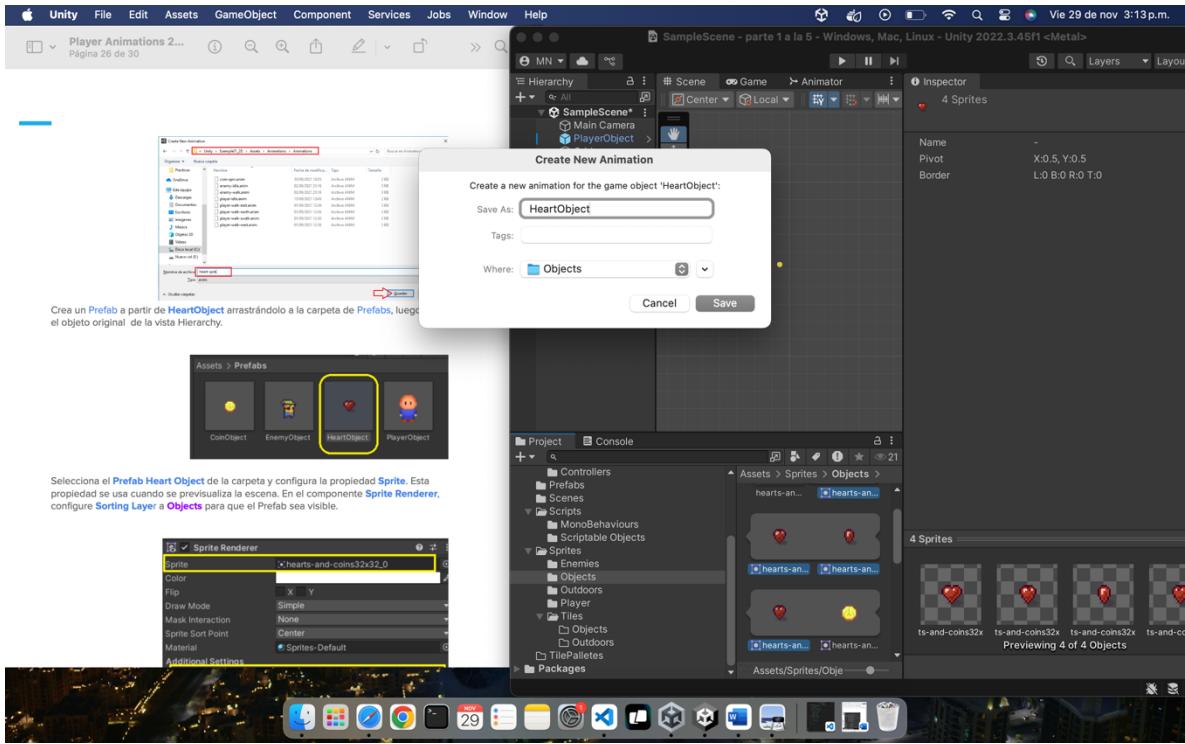
```

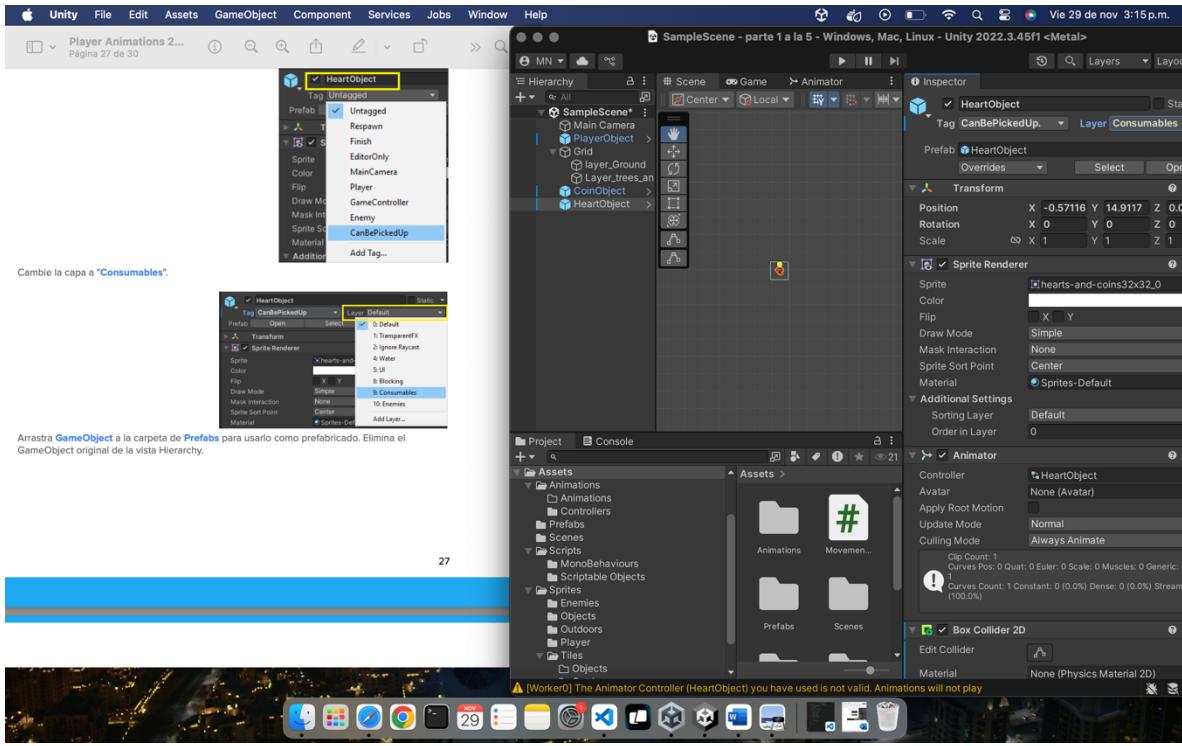
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

/*
 * Hereda de MonoBehaviour lo podemos enlazar a un GameObject.
 */
public class Consumable : MonoBehaviour
{
    //Referencia a un objeto Scriptable o programable
    public Item item;
}

```

Vie 29 de nov 3:08 p.m.





27