Códigos

Este repositorio fue creado con el objetivo de hacer un sistema de economía el cual sea capaz de vender objetos.

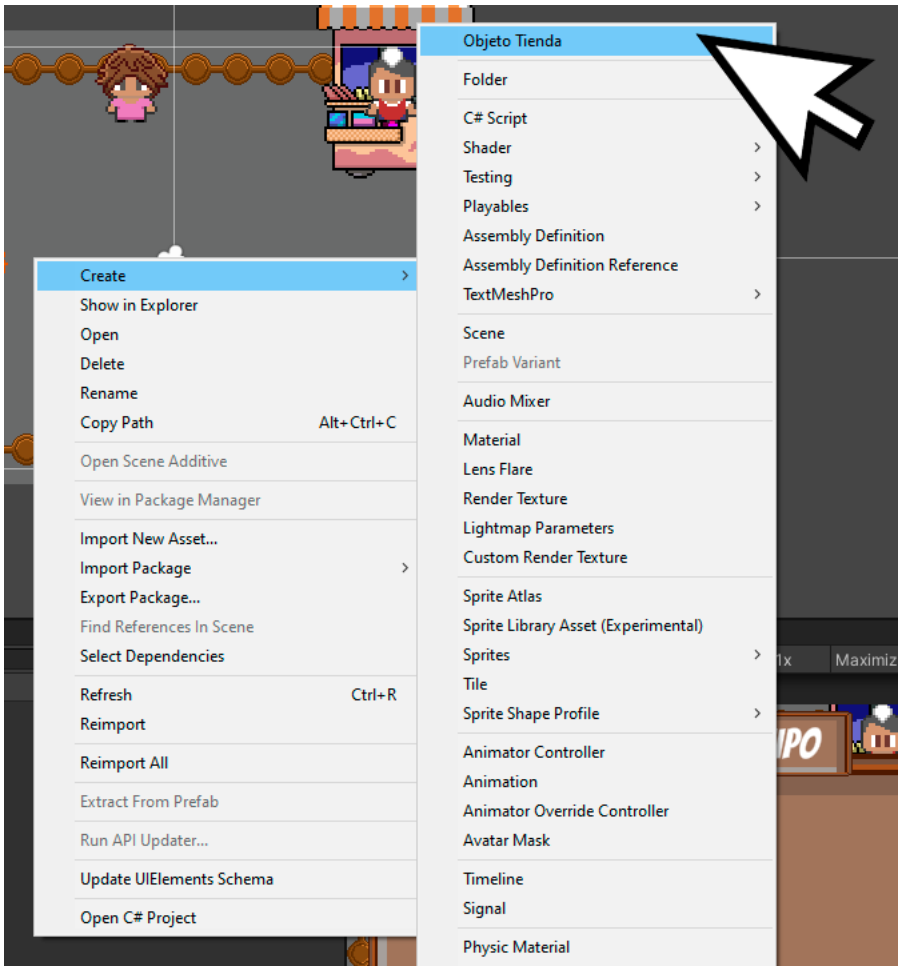
A continuación podrás leer los códigos usados para que este juego funcione.

Plantilla objeto:

1. Se creo un script con el nombre “PlantillaObjeto”
2. A este se le convirtió en un objeto scriptable y se le agrego 3 variables.

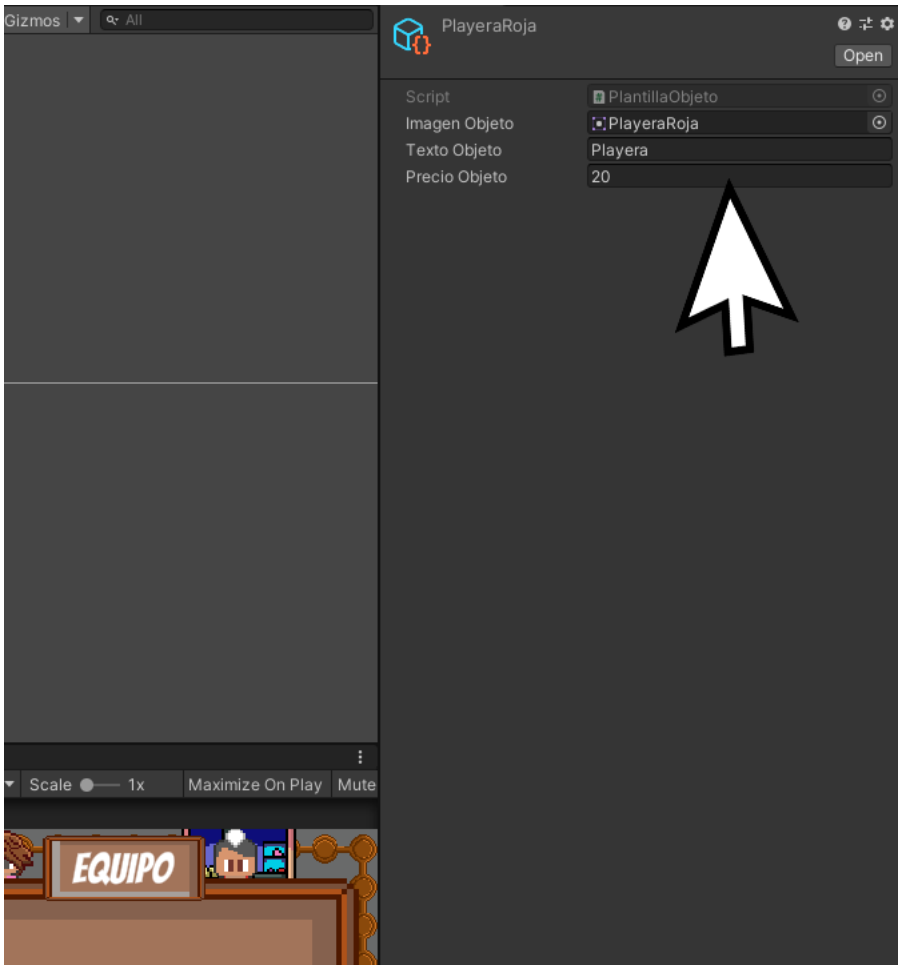
```
[CreateAssetMenu (fileName = "Objeto", menuName = "Objeto Tienda")] //Aqui es donde se va a crear el scriptable
public class PlantillaObjeto : ScriptableObject
{
    public Sprite imagenObjeto;//Esta va a ser la imagen del objeto.
    public string textoObjeto;//Esta va a ser el nombre del objeto.
    public int precioObjeto; // Este va a ser el precio del objeto.
}
```

Este logro hacer que ahora exista un nuevo elemento el cual se va a colocar en una carpeta para los items que se desean vender.



A este nuevo objeto se le pondrá la imagen del objeto que se desea vender, su nombre y su precio.

(Crea tantos items como quieras)



El código completo se vería así.

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

[CreateAssetMenu (fileName = "Objeto", menuName = "Objeto Tienda")] //Aqui es donde se va a crear el scriptable
public class PlantillaObjeto : ScriptableObject
{
    public Sprite imagenObjeto;//Esta va a ser la imagen del objeto.
    public string textoObjeto;//Esta va a ser el nombre del objeto.
    public int precioObjeto; // Este va a ser el precio del objeto.
}
```

Objeto

Este script es el encargado de gestionar los elementos que se pasan del objeto scriptable (el anterior elemento creado) al prefab del botón.

1. Se creo el script llamado “Objeto”
2. Este script se pone sobre el BotonObjeto.
3. A este se le agregaron las siguientes variables.

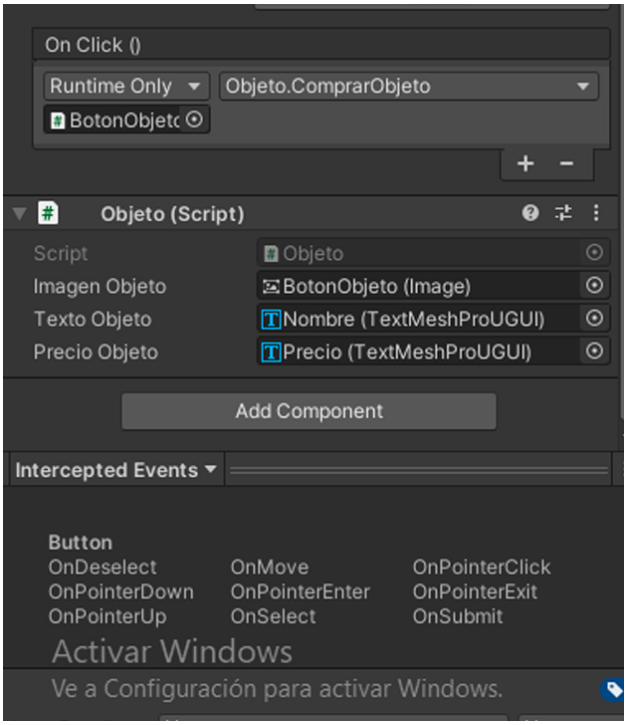
```
[SerializeField] Image imagenObjeto; //La imagen del objeto.
    [SerializeField] TextMeshProUGUI textoObjeto; //Nombre del objeto.
```

```
    [SerializeField] TextMeshProUGUI precioObjeto;//Precio del objeto.

    private int precio;
    private Equipo equipo;
```

4. Se creo un método el cual se encarga de pasar los datos de objeto scriptable a las variables del prefab.

```
public void CrearObjeto(PlantillaObjeto datosObjeto)//Metodo que va a pasar la información del objeto a las variables del prefab.
    {
        precio = datosObjeto.precioObjeto; //Se va a actualizar el dinero con el del scriptable.
        imagenObjeto.sprite = datosObjeto.imagenObjeto;//Aqui se define la imagen del objeto.
        textoObjeto.text = datosObjeto.textoObjeto;//Aqui se define el nombre del objeto.
        precioObjeto.text = datosObjeto.precioObjeto.ToString();//Aqui se va a definir el precio del objeto.
    }
```

En el inspector agrega la imagen del botónObjeto, texto dentro del botón y el precio, una vez realizado conviertió el botonObjeto en un prefab y se borro la que seguía en la escena y se le referencio con el método Objeto.ComprarObjeto y se le agrego el botonObjeto.



El código completo se vería así.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class Objeto : MonoBehaviour
{
    [SerializeField] Image imagenObjeto; //La imagen del objeto.
    [SerializeField] TextMeshProUGUI textoObjeto; //Nombre del objeto.
    [SerializeField] TextMeshProUGUI precioObjeto;//Precio del objeto.

    private int precio;
    private Equipo equipo;

    private void Awake()
    {
```

```
        equipo = FindObjectOfType<Equipo>();
    }

    public void CrearObjeto(PlantillaObjeto datosObjeto)//Metodo que va a pasar la información del objeto a las variables del prefab.
    {
        precio = datosObjeto.precioObjeto; //Se va a actualizar el dinero con el del scriptable.
        imagenObjeto.sprite = datosObjeto.imagenObjeto;//Aqui se define la imagen del objeto.
        textoObjeto.text = datosObjeto.textoObjeto;//Aqui se define el nombre del objeto.
        precioObjeto.text = datosObjeto.precioObjeto.ToString();//Aqui se va a definir el precio del objeto.
    }

    public void ComprarObjeto()
    {
        equipo.IncluirEquipo(precio, imagenObjeto); //Este se va a ejecutar cuando se presione el boton.
    }
}
```

Tienda:

Esta va a ser la encargada de gestionar la aparición del objetos creados de manera aleatoria

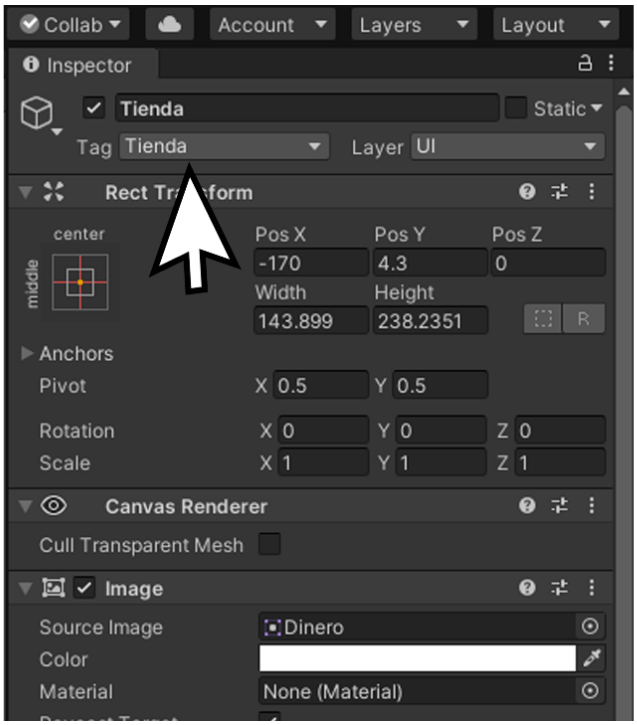
1. Se creo el script llamado “Tienda”
2. Se le agrego las siguientes variables.

```
    [SerializeField] GameObject prefabObjetoTienda; //Este va a usar el prefab de la tienda.
    [SerializeField] int numeroMaximoObjetosTienda;//El numero maximo de objetos que va a tener la tienda.
    [SerializeField] PlantillaObjeto[] listaTienda;//Va a contener todos los objetos scriptables.
    [SerializeField] List<PlantillaObjeto> listaProvisionalTienda;//Se va a crear una lista de objetos que no esten dentro de la tienda de

    private Objeto objeto; //Una referencia al script de objeto.
```

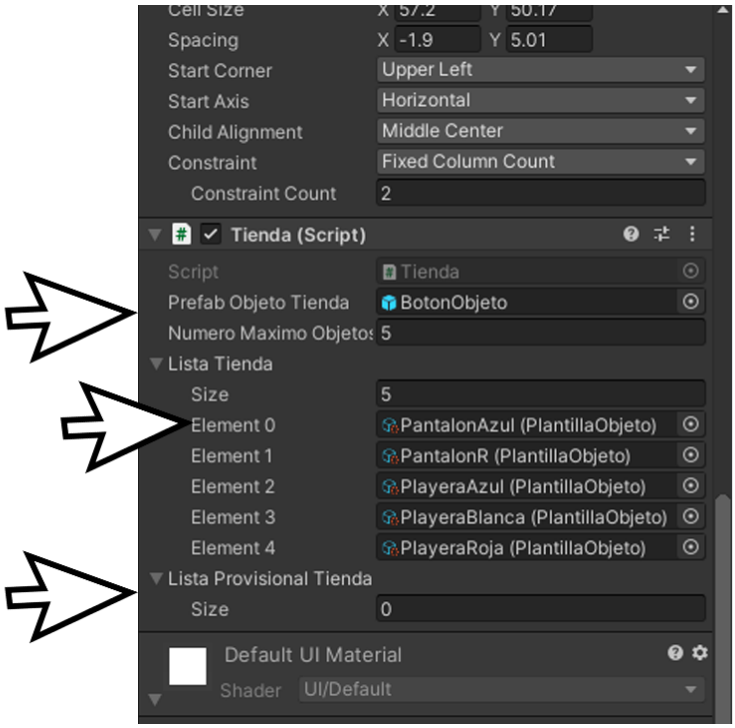
3. En el método start se agrego lo siguiente.

```
private void Start()
    {
        listaProvisionalTienda.AddRange(listaTienda);//Este va a incluir a todos lo elementos en la lista.

        for(int i = 0; i <= numeroMaximoObjetosTienda - 1; i++)//Se va a encargar de colocar cada uno de los objetos en la tienda.
        {
            GameObject tienda = GameObject.Instantiate(prefabObjetoTienda, Vector2.zero, Quaternion.identity, GameObject.FindGameObjectWith
                int indice = Random.Range(0, listaProvisionalTienda.Count);//Los va a colocar de manera aleatoria y que sea el numero de ob
            objeto = tienda.GetComponent<Objeto>();//Aqui se va a buscar el metodo que se creo en el script objeto.
            objeto.CrearObjeto(listaProvisionalTienda[indice]);//Se va a llamar al metodo.
            listaProvisionalTienda.Remove(listaProvisionalTienda[indice]);//Le vamos a decir que elimine el elemento de la lista para que n
        }
    }
```

Recuerda que la tienda debe tener el Tag “Tienda” para que este código funcione.



A esta tienda se le agrego el script y todos los objetos scriptables se movieron a este código, el numero máximo de objetos, es el numero de items que van a salir en la tienda, al Prefab Objeto Tienda se le agrego el prefab del botonObjeto y la lista provisional Tienda, van a salir todos los items que no fueron capaces de salir en la tienda, esto sirve para cuando tengas demasiados items, y un espacio limitado de espacio, los items restantes se guarden.



Así se ve el código completo:

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

public class Tienda : MonoBehaviour
{
    [SerializeField] GameObject prefabObjetoTienda; //Este va a usar el prefab de la tienda.
    [SerializeField] int numeroMaximoObjetosTienda;//El numero maximo de objetos que va a tener la tienda.
    [SerializeField] PlantillaObjeto[] listaTienda;//Va a contener todos los objetos scriptables.
    [SerializeField] List<PlantillaObjeto> listaProvisionalTienda;//Se va a crear una lista de objetos que no esten dentro de la tienda de

    private Objeto objeto; //Una referencia al script de objeto.

    private void Start()
    {
        listaProvisionalTienda.AddRange(listaTienda);//Este va a incluir a todos lo elementos en la lista.

        for(int i = 0; i <= numeroMaximoObjetosTienda - 1; i++)//Se va a encargar de colocar cada uno de los objetos en la tienda.
        {
            GameObject tienda = GameObject.Instantiate(prefabObjetoTienda, Vector2.zero, Quaternion.identity, GameObject.FindGameObjectWith
                int indice = Random.Range(0, listaProvisionalTienda.Count);//Los va a colocar de manera aleatoria y que sea el numero de ob
            objeto = tienda.GetComponent<Objeto>();//Aqui se va a buscar el metodo que se creo en el script objeto.
            objeto.CrearObjeto(listaProvisionalTienda[indice]);//Se va a llamar al metodo.
            listaProvisionalTienda.Remove(listaProvisionalTienda[indice]);//Le vamos a decir que elimine el elemento de la lista para que n
        }
    }
}
```

Equipo:

Este es el encargado de guardar los objetos y modificar la cantidad de dinero que se tiene.

1. Se creo el script llamado “Equipo”.
2. Se agregaron las variables.

```
[SerializeField] private int dineroTotal = 50;//Una cantidad predefinida de dinero que va a tener el jugador.
    [SerializeField] TextMeshProUGUI textoDinero;// Va a mostrar el dinero en pantalla.
    [SerializeField] GameObject objetoDeEquipo; // Va ser una referencia a unuestros objetos de equipo.

    private int numeroMaximoObjetos = 0;//Es la referencia de la cantidad de objetos maximos que podemos tener en nuestro equipo.
```

3. Se referencio el dinero que tenemos en cada momento en el método start.

```
private void Start()
    {
        textoDinero.text = dineroTotal.ToString();// Vamos a mostrar el dinero que tenemos en cada momento.
    }
```

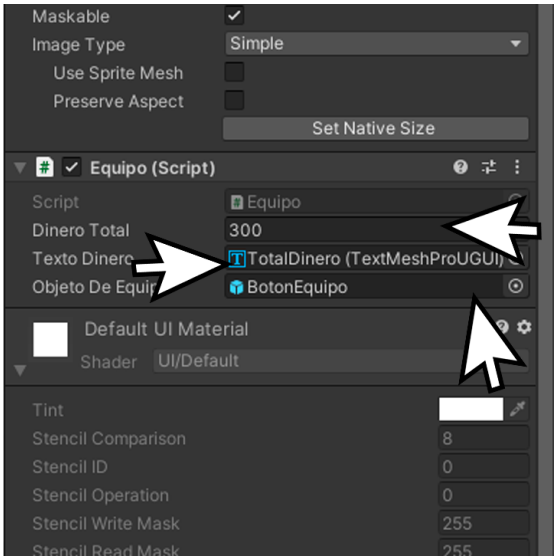
4. Se creo el método el cual fue encargado de crear el elemento de equipo

```
public void IncluirEquipo(int dinero, Image imagenEquipo) //Este va a crear los elementos del equipo.
    {
        if(dinero <= dineroTotal && numeroMaximoObjetos <= 11) //Si tenemos el suficiente dinero y el numero de objetos no supera el maximo
        {
            dineroTotal -= dinero; //se va a restar la cantidad de dinero total.
            numeroMaximoObjetos++; // se va a sumar el numero maximo de objetos, para no exceder el numero maximo de objetos.
            GameObject equipo = GameObject.Instantiate(objetoDeEquipo, Vector2.zero, Quaternion.identity, GameObject.FindGameObjectWithTag(
            Image imagen = equipo.GetComponent<Image>();// Va a tomar como referencia nuestra imagen de objeto de nuestro equipo.
            imagen.sprite = imagenEquipo.sprite; // A essta imagen se le va a cambiar el sprite por la variable en el metodo start (textoDi
            textoDinero.text = dineroTotal.ToString();// Se va a actualzar el texto de dinero para dar la nueva cantidad.
        }
    }
```

Al equipo con el script se le modifica lo siguiente:

1. Dinero total: La cantidad de dinero que va a tener el jugador

2. Texto dinero: A este se le agrega el texto que nos va a indicar cuanto dinero tenemos.
3. Se le agrego el botónEquipo.



Así se ve el código completo:

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

public class Tienda : MonoBehaviour
{
    [SerializeField] GameObject prefabObjetoTienda; //Este va a usar el prefab de la tienda.
    [SerializeField] int numeroMaximoObjetosTienda;//El numero maximo de objetos que va a tener la tienda.
    [SerializeField] PlantillaObjeto[] listaTienda;//Va a contener todos los objetos scriptables.
    [SerializeField] List<PlantillaObjeto> listaProvisionalTienda;//Se va a crear una lista de objetos que no esten dentro de la tienda de

    private Objeto objeto; //Una referencia al script de objeto.

    private void Start()
    {
        listaProvisionalTienda.AddRange(listaTienda);//Este va a incluir a todos lo elementos en la lista.

        for(int i = 0; i <= numeroMaximoObjetosTienda - 1; i++)//Se va a encargar de colocar cada uno de los objetos en la tienda.
        {
            GameObject tienda = GameObject.Instantiate(prefabObjetoTienda, Vector2.zero, Quaternion.identity, GameObject.FindGameObjectWith
                int indice = Random.Range(0, listaProvisionalTienda.Count);//Los va a colocar de manera aleatoria y que sea el numero de ob
            objeto = tienda.GetComponent<Objeto>();//Aqui se va a buscar el metodo que se creo en el script objeto.
            objeto.CrearObjeto(listaProvisionalTienda[indice]);//Se va a llamar al metodo.
            listaProvisionalTienda.Remove(listaProvisionalTienda[indice]);//Le vamos a decir que elimine el elemento de la lista para que n
        }
    }
}
```