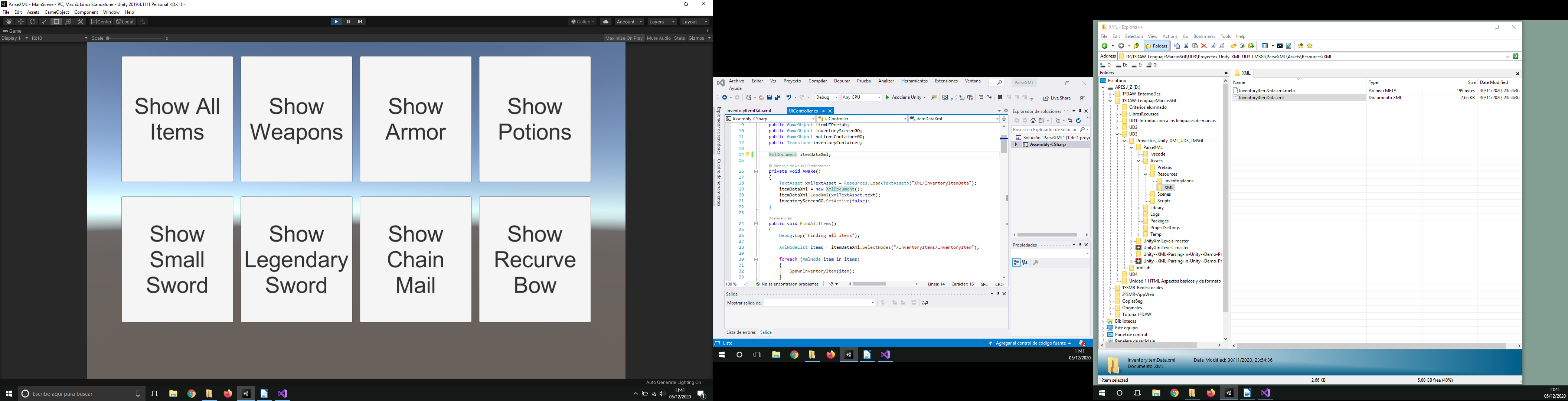
# Actividad 2. Inventario XML

Instrucciones:

1.Crear proyecto(NomApe\_Act2\_UD3), importar el paquete ‘Act2-UD3\_LMSGI.unitypackage’. Abrir la escena ‘MainScene’

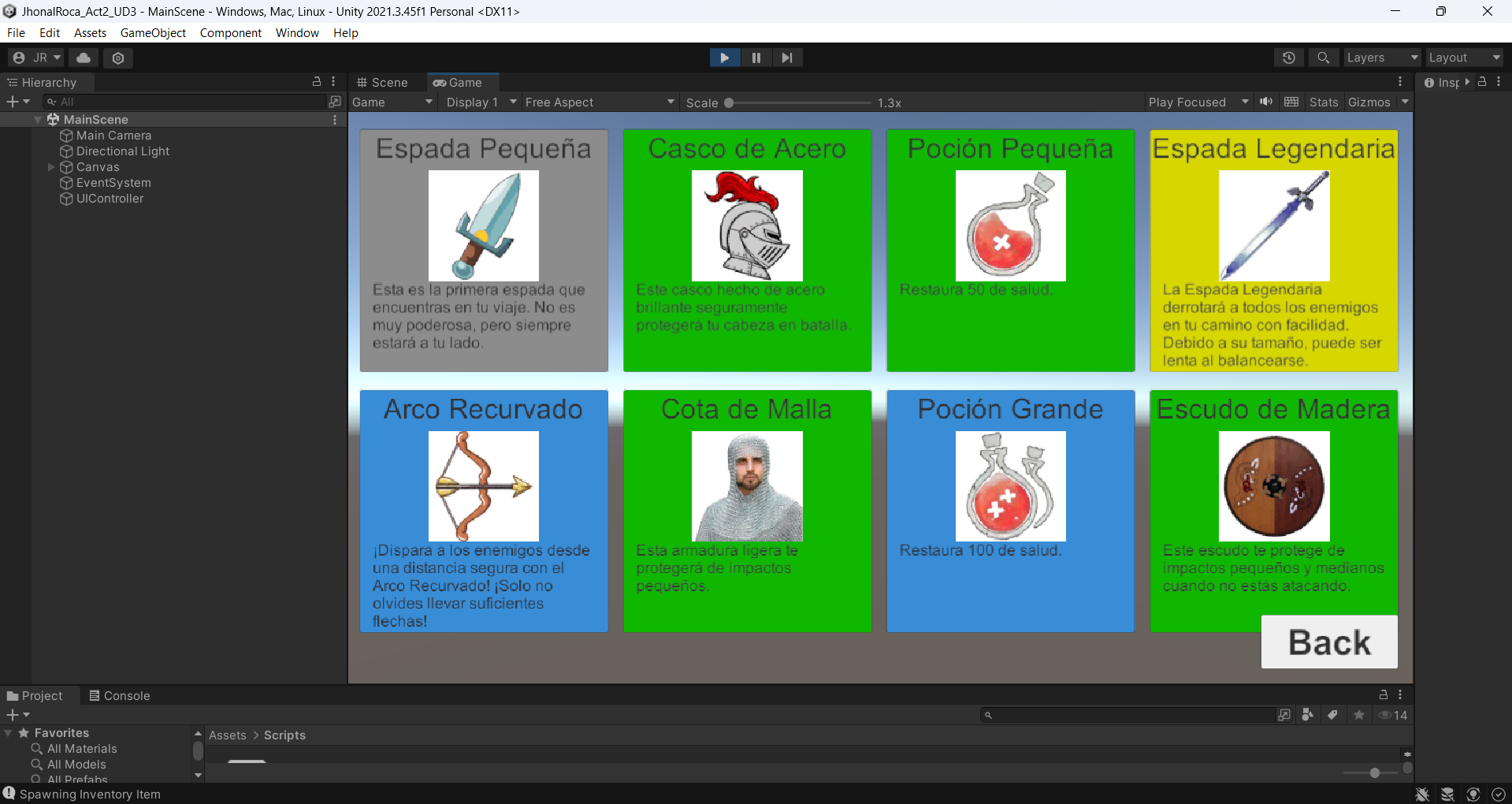
2. Corregir errores. Al ejecutar, debe mostrar la siguiente pantalla:

Cada panel da acceso a un nuevo panel, mostrando los items del inventario del tipo seleccionado.



3. Buscar XML del proyecto y traducir al castellano los datos a cargar de las siguientes etiquetas XML: <ItemTitle> y <ItemDesc>

Hacer click en el panel “Show All Items” y adjuntar una captura de pantalla que muestra todos los items con el nombre y descripción en castellano.



4.En el mismo XML, modificar los elementos XML atendiendo a la siguiente traducción:

<InventoryItems> –> <ItemsInventario>

</InventoryItems> –> </ItemsInventario>

<InventoryItem> –> <ItemInventario>

</InventoryItem> –> </ItemInventario>

<ItemTitle> –> <TituloItem>

</ItemTitle> –> </TituloItem>

<ItemDesc> –> <DescItem>

</ItemDesc> –> </DescItem>

5. Modificar el script ‘UIController.cs’ para que cargue correctamente el archivo XML según los nodos modificados.

Adjuntar a continuación el código del script ‘UIController.cs’

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<ItemsInventario>

  <ItemInventario ID = "xmltut\_small\_sword" Type ="weapon">

    <TituloItem>Small Sword</TituloItem>

    <DescItem>This is the first sword you find on your journey. Not very powerful, but it will always be by your side.</DescItem>

    <Color>

      <r>167</r>

      <g>167</g>

      <b>167</b>

      <a>255</a>

    </Color>

    <IMAGEN>smallsword</IMAGEN>

  </ItemInventario>

  <ItemInventario ID = "xmltut\_steel\_helmet" Type = "armor">

    <TituloItem>Steel Helmet</TituloItem>

    <DescItem>This helmet crafted from shiny steel will surely protect your noggin in battle.</DescItem>

    <Color>

      <r>19</r>

      <g>218</g>

      <b>0</b>

      <a>255</a>

    </Color>

    <IMAGEN>steelhelmet</IMAGEN>

  </ItemInventario>

  <ItemInventario ID = "xmltut\_health\_potion\_50" Type = "consumable">

    <TituloItem>Small Potion</TituloItem>

    <DescItem>Restores 50 health.</DescItem>

    <Color>

      <r>19</r>

      <g>218</g>

      <b>0</b>

      <a>255</a>

    </Color>

    <IMAGEN>50healthpot</IMAGEN>

  </ItemInventario>

  <ItemInventario ID = "xmltut\_big\_sword" Type = "weapon">

    <TituloItem>Legendary Sword</TituloItem>

    <DescItem>The Legendary Sword will slay all enemies in your path with ease. Due to it's size, it may be slow to swing.</DescItem>

    <Color>

      <r>255</r>

      <g>255</g>

      <b>0</b>

      <a>255</a>

    </Color>

    <IMAGEN>bigsword</IMAGEN>

  </ItemInventario>

  <ItemInventario ID = "xmltut\_bow" Type ="weapon">

    <TituloItem>Recurve Bow</TituloItem>

    <DescItem>Shoot enemies from a safe distance with the Recurve Bow! Just don't forget to bring enough arrows!</DescItem>

    <Color>

      <r>69</r>

      <g>169</g>

      <b>255</b>

      <a>255</a>

    </Color>

    <IMAGEN>recurvebow</IMAGEN>

  </ItemInventario>

  <ItemInventario ID = "xmltut\_chain\_mail" Type = "armor">

    <TituloItem>Chain Mail</TituloItem>

    <DescItem>This light-weight armor will protect you from small impacts.</DescItem>

    <Color>

      <r>19</r>

      <g>218</g>

      <b>0</b>

      <a>255</a>

    </Color>

    <IMAGEN>chainmail</IMAGEN>

  </ItemInventario>

  <ItemInventario ID = "xmltut\_health\_potion\_100" Type = "consumable">

    <TituloItem>Big Potion</TituloItem>

    <DescItem>Restores 100 health.</DescItem>

    <Color>

      <r>69</r>

      <g>169</g>

      <b>255</b>

      <a>255</a>

    </Color>

    <IMAGEN>100healthpot</IMAGEN>

  </ItemInventario>

  <ItemInventario ID = "xmltut\_wood\_shield" Type = "armor">

    <TituloItem>Wooden Shield</TituloItem>

    <DescItem>This shield protects you from small and medium strength impacts when you are not attacking.</DescItem>

    <Color>

      <r>19</r>

      <g>218</g>

      <b>0</b>

      <a>255</a>

    </Color>

    <IMAGEN>woodshield</IMAGEN>

  </ItemInventario>

</ItemsInventario>

6. En el mismo XML, modificar los elementos XML atendiendo a la siguiente traducción:

<Image> –> <IMAGEN>

</IMAGE> –> </IMAGEN>

7. Modificar el script ‘UIController.cs’ para que cargue correctamente el archivo XML según los nodos modificados.

Adjuntar a continuación el código MODIFICADO del script ‘UIController.cs’

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using System.Xml;

public class UIController : MonoBehaviour

{

    public GameObject itemUIPrefab;

    public GameObject inventoryScreenGO;

    public GameObject buttonsContainerGO;

    public Transform inventoryContainer;

    XmlDocument itemDataXml;

    private void Awake()

    {

        TextAsset xmlTextAsset = Resources.Load<TextAsset>("XML/InventoryItemData");

        itemDataXml = new XmlDocument();

        itemDataXml.LoadXml(xmlTextAsset.text);

        inventoryScreenGO.SetActive(false);

    }

    public void FindAllItems()

    {

        Debug.Log("Finding all items");

        XmlNodeList items = itemDataXml.SelectNodes("/ItemsInventario/ItemInventario");

        foreach (XmlNode item in items)

        {

            SpawnInventoryItem(item);

        }

        ShowInventoryItems();

    }

    public void FindItemsOfType(string itemType)

    {

        Debug.Log("Finding all items of type: " + itemType);

        XmlNodeList items = itemDataXml.SelectNodes($"/ItemsInventario/ItemInventario[@Type='{itemType}']");

        foreach (XmlNode item in items)

        {

            SpawnInventoryItem(item);

        }

        ShowInventoryItems();

    }

    public void FindItemsWithID(string itemID)

    {

        XmlNode curNode = itemDataXml.SelectSingleNode($"/ItemsInventario/ItemInventario[@ID='{itemID}']");

        if (curNode == null)

        {

            Debug.LogError($"Error: could not find Inventrio Item with ID: {itemID} in InventoryItemData.xml");

            return;

        }

        SpawnInventoryItem(curNode);

        ShowInventoryItems();

    }

    void ShowInventoryItems()

    {

        buttonsContainerGO.SetActive(false);

        inventoryScreenGO.SetActive(true);

    }

    void SpawnInventoryItem(XmlNode item)

    {

        Debug.Log("Spawning Inventory Item");

        GameObject newItemUI = GameObject.Instantiate(itemUIPrefab, inventoryContainer);

        InventoryItem newInventoryItem = new InventoryItem(item);

        newInventoryItem.UpdateInventoryUI(newItemUI);

    }

    public void OnButtonBack()

    {

        foreach (Transform t in inventoryContainer)

        {

            Destroy(t.gameObject);

        }

        inventoryScreenGO.SetActive(false);

        buttonsContainerGO.SetActive(true);

    }

    class InventoryItem

    {

        public string itemID { get; private set; }

        public string itemType { get; private set; }

        public string itemTitle { get; private set; }

        public string itemDescription { get; private set; }

        public Color bgColor { get; private set; }

        public Texture itemImage { get; private set; }

        public InventoryItem(XmlNode curItemNode)

        {

            itemID = curItemNode.Attributes["ID"]?.Value ?? "Unknown";

            itemType = curItemNode.Attributes["Type"]?.Value ?? "Unknown";

            itemTitle = curItemNode["TituloItem"]?.InnerText ?? "Unknown";

            itemDescription = curItemNode["DescItem"]?.InnerText ?? "No description";

            XmlNode colorNode = curItemNode.SelectSingleNode("Color");

            if (colorNode != null)

            {

                float.TryParse(colorNode["r"]?.InnerText, out float bgR);

                float.TryParse(colorNode["g"]?.InnerText, out float bgG);

                float.TryParse(colorNode["b"]?.InnerText, out float bgB);

                float.TryParse(colorNode["a"]?.InnerText, out float bgA);

                bgColor = new Color(NormalizeColorValue(bgR), NormalizeColorValue(bgG), NormalizeColorValue(bgB), NormalizeColorValue(bgA));

            }

            else

            {

                bgColor = Color.white;  // Default color

            }

            string pathToImage = "InventoryIcons/" + curItemNode["IMAGEN"]?.InnerText ?? "defaultImage";

            itemImage = Resources.Load<Texture2D>(pathToImage);

        }

        float NormalizeColorValue(float value)

        {

            return value / 255f;

        }

        public void UpdateInventoryUI(GameObject inventoryUI)

        {

            Transform inventoryUITransform = inventoryUI.transform;

            Image itemBGPanel = inventoryUITransform.GetComponent<Image>();

            if (itemBGPanel == null)

            {

                Debug.LogError($"Error: could not find itemBGPanel on inventoryUITransform: {inventoryUITransform.gameObject.name}");

                return;

            }

            Transform itemBGPanelTransform = itemBGPanel.GetComponent<Transform>();

            RawImage itemRawImage = itemBGPanelTransform.Find("ItemRawImage")?.GetComponent<RawImage>();

            Text itemTitleText = itemBGPanelTransform.Find("ItemTitleText")?.GetComponent<Text>();

            Text itemDescriptionText = itemBGPanelTransform.Find("ItemDescriptionText")?.GetComponent<Text>();

            if (itemRawImage == null)

            {

                Debug.LogError($"Error: could not find itemRawImage on inventoryUITransform: {inventoryUITransform.gameObject.name}");

            }

            if (itemTitleText == null)

            {

                Debug.LogError($"Error: could not find itemTitleText on inventoryUITransform: {inventoryUITransform.gameObject.name}");

            }

            if (itemDescriptionText == null)

            {

                Debug.LogError($"Error: could not find itemDescriptionText on inventoryUITransform: {inventoryUITransform.gameObject.name}");

            }

            itemBGPanel.color = bgColor;

            itemRawImage.texture = itemImage;

            itemTitleText.text = itemTitle;

            itemDescriptionText.text = itemDescription;

        }

    }

}

8. En el panel de jerarquía de la escena en Unity, modificar los paneles para que el texto que se muestre sea en castellano.

