
Diseño de Sistemas Interactivos

Curso 2023 - 2024

Profesor: José Manuel Velasco


Despacho 309, 3ª planta. Facultad de Informática.

Laboratorio 7: Introducción a los UI Toolkit Project samples.

Índice

- Project Samples
 - Dragon Crashers
 - Custom Controls
 - Tabbed menú
 - Radial progress bar
 - Quizu
 - Demos
 - CustomControls
 - Manipulators
 - Quiz
 - Data
 - Scripts
 - Coroutines
 - ScriptablesObjects
 - Actions
- Proyecto

Dragon Crashers



Search for assets

3D 2D Add-Ons Audio AI Decentralization Essentials Templates Tools VFX **Sale** Sell Assets


Over 11,000 five-star assets

Rated by 85,000+ customers









Supported by 100,000+ forum members

Every asset moderated by Unity

Home > Essentials > Tutorial Projects > UI Toolkit Sample – Dragon Crashers



1/16





UI Toolkit Sample – Dragon Crashers

Unity Technologies ★★★★☆ (20) ♥ (979)

FREE

533 views in the past week

Add to My Assets 

 YuzuTamaki

★★★★★ a year ago

Good for dissecting to learn

Great sample to see how UI Toolkit can be used for a complex project (though warning: it's super complicated!). A working knowledge of UI Toolkit is n...

[Read more reviews](#)

License agreement

[Standard Unity Asset Store EULA](#)

License type

[Extension Asset](#)

File size

212.3 MB

Latest version

1.40

Latest release date

Nov 24, 2023

Original Unity version

2021.3.7 or higher


Support

[Visit site](#)

Overview

Package ContentReleasesReviewsPublisher infoAsset Quality

QuizU

**Unity Asset Store**

3D2DAdd-OnsAudioAIDecentralizationEssentialsTemplatesToolsVFXSaleSell Assets


Over 11,000 five-star assets

Rated by 85,000+ customers

Supported by 100,000+ forum members

Every asset moderated by Unity

Home > Essentials > Tutorial Projects > QuizU - A UI toolkit sample



**LEARN UI TOOLKIT
AND DESIGN PATTERNS**

1/20

QuizU

QuizU - A UI toolkit sample

Unity Technologies ★★★★☆ (3) | ♥ (124)

FREE

403 views in the past week

[Add to My Assets](#)

Mj-Kkaya
★★★★★ 3 months ago

For UI developers

This example is so useful! It includes many important advantages such as Design Pattern, ScribableObjects and other useful features.

[Read more reviews](#)

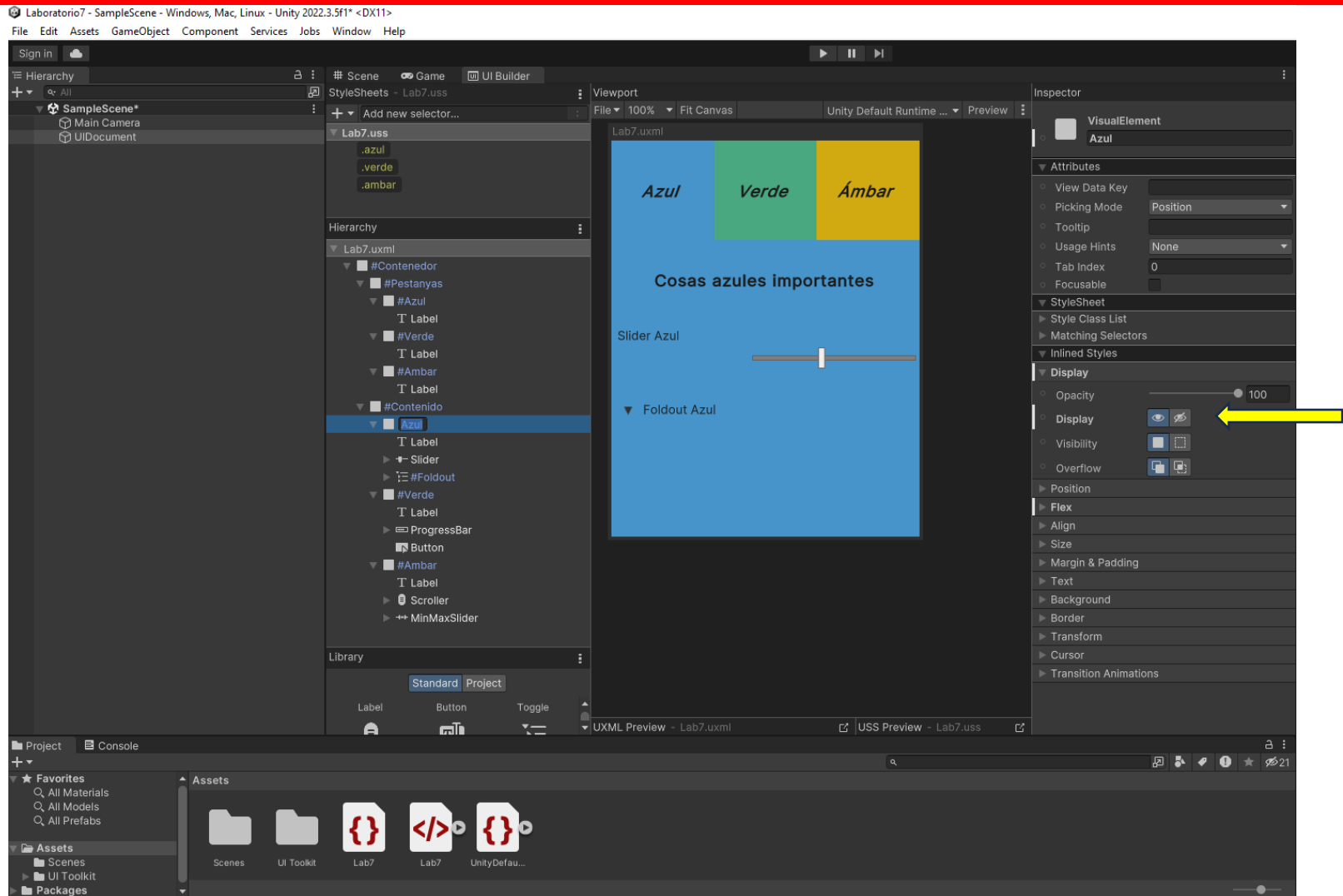
License agreement	Standard Unity Asset Store EULA
License type	Extension Asset
File size	12.9 MB
Latest version	1.1.0
Latest release date	Dec 8, 2023
Original Unity version	2022.3.14 or higher
Support	Visit site

[Overview](#)[Package Content](#)[Releases](#)[Reviews](#)[Publisher info](#)[Asset Quality](#)

Índice

- Project Samples
 - **Dragon Crashers**
 - **Custom Controls**
 - **Tabbed menú**
 - **Radial progress bar**
 - Quizu
 - Demos
 - CustomControls
 - Manipulators
 - Quiz
 - Data
 - Scripts
 - Coroutines
 - ScriptablesObjects
 - Actions
- Proyecto

Menú con pestañas muy sencillo



Menú con pestañas

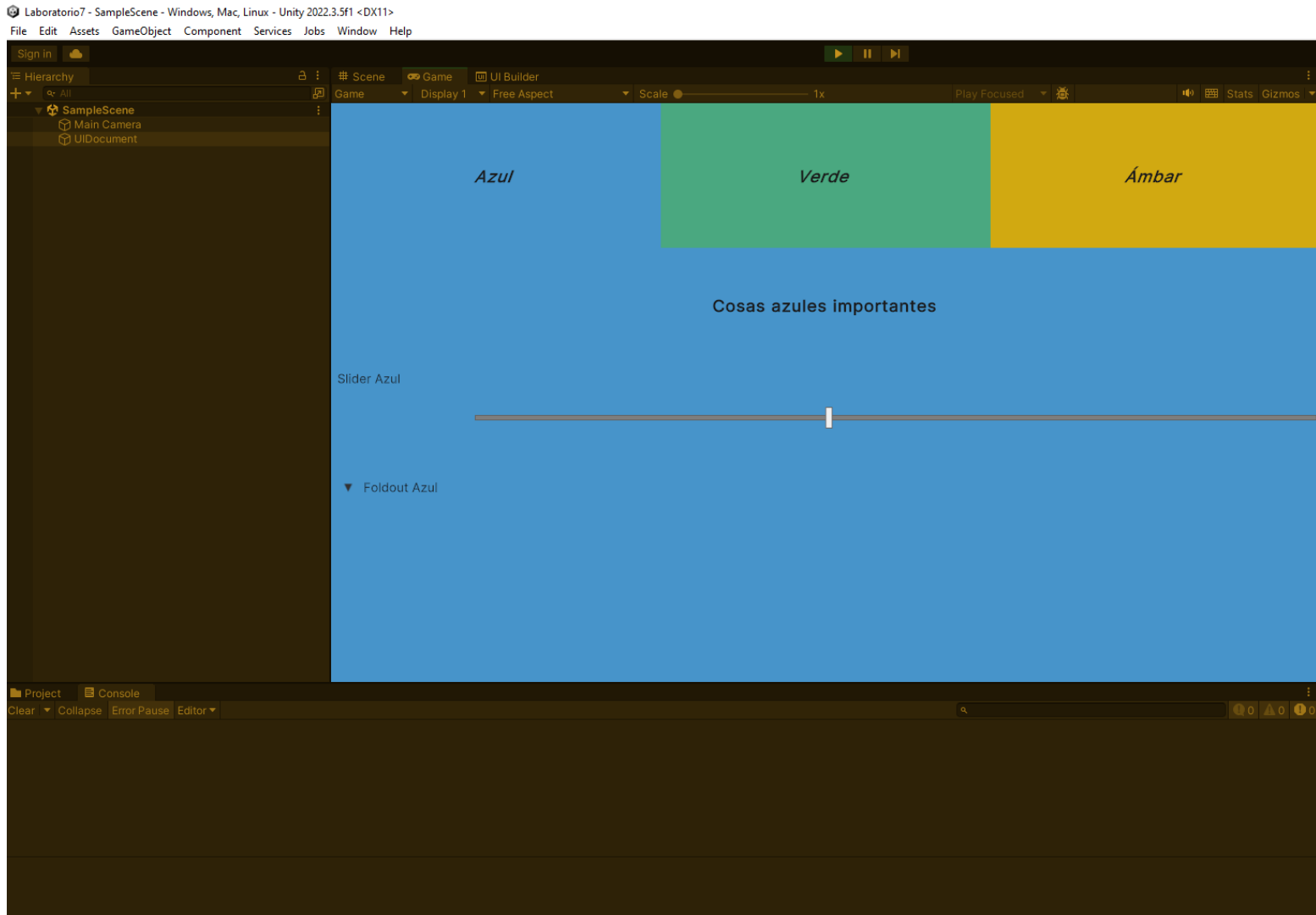
```
Assets > Lab7.cs > Lab7 > OnEnable
1  using UnityEngine;
2  using UnityEngine.UIElements;
3
4  0 references
   public class Lab7 : MonoBehaviour
5  {
6      3 references
       VisualElement contenido_azul;
7      3 references
       VisualElement contenido_verde;
8      3 references
       VisualElement contenido_ambar;
9      2 references
       VisualElement pestanya_azul;
10     2 references
       VisualElement pestanya_verde;
11     2 references
       VisualElement pestanya_ambar;
12     3 references
       private void NoContenido(){
13         contenido_azul.style.display = DisplayStyle.None;
14         contenido_verde.style.display = DisplayStyle.None;
15         contenido_ambar.style.display = DisplayStyle.None;
16     }
17
```

Menú con pestañas

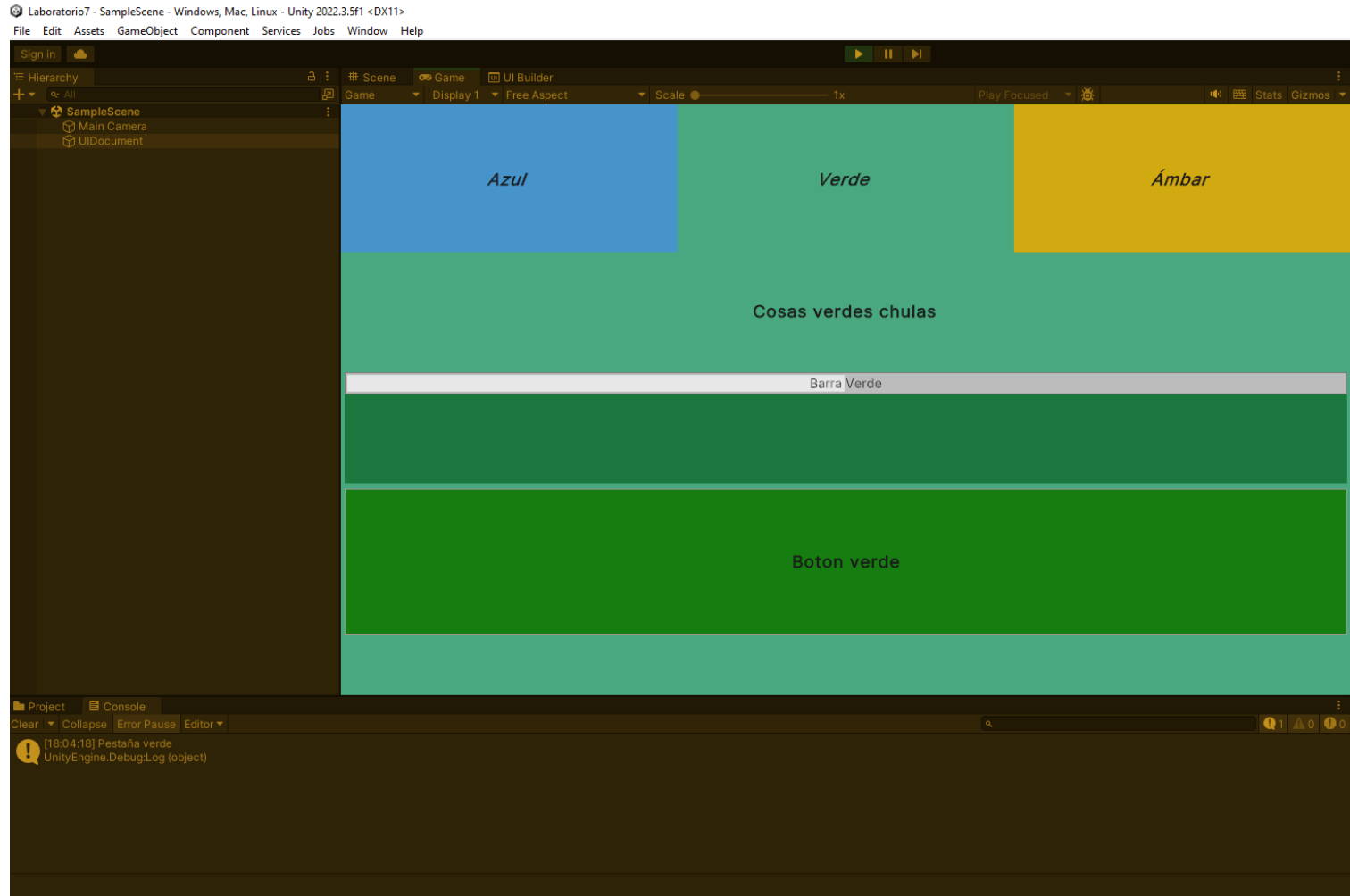
```
0 references
18 private void OnEnable() {
19     UIDocument uidoc = GetComponent<UIDocument>();
20     VisualElement rootve = uidoc.rootVisualElement;
21
22     VisualElement contenido = rootve.Q("Contenido");
23     VisualElement pestanyas = rootve.Q("Pestanyas");
24
25     pestanya_azul = pestanyas.Q("Azul");
26     pestanya_verde = pestanyas.Q("Verde");
27     pestanya_ambar = pestanyas.Q("Ambar");
28
29     contenido_azul = contenido.Q("Azul");
30     contenido_verde = contenido.Q("Verde");
31     contenido_ambar = contenido.Q("Ambar");
32
33
34     pestanya_azul.RegisterCallback<MouseDownEvent>(evt => {
35         Debug.Log("Pestaña azul");
36         NoContenido();
37         contenido_azul.style.display = DisplayStyle.Flex;
38     });
39
40     pestanya_verde.RegisterCallback<MouseDownEvent>(evt => {
41         Debug.Log("Pestaña verde");
42         NoContenido();
43         contenido_verde.style.display = DisplayStyle.Flex;
44     });
45
46     pestanya_ambar.RegisterCallback<MouseDownEvent>(evt => {
47         Debug.Log("Pestaña ambar");
48         NoContenido();
49         contenido_ambar.style.display = DisplayStyle.Flex;
50     });
51 }
52 }
```



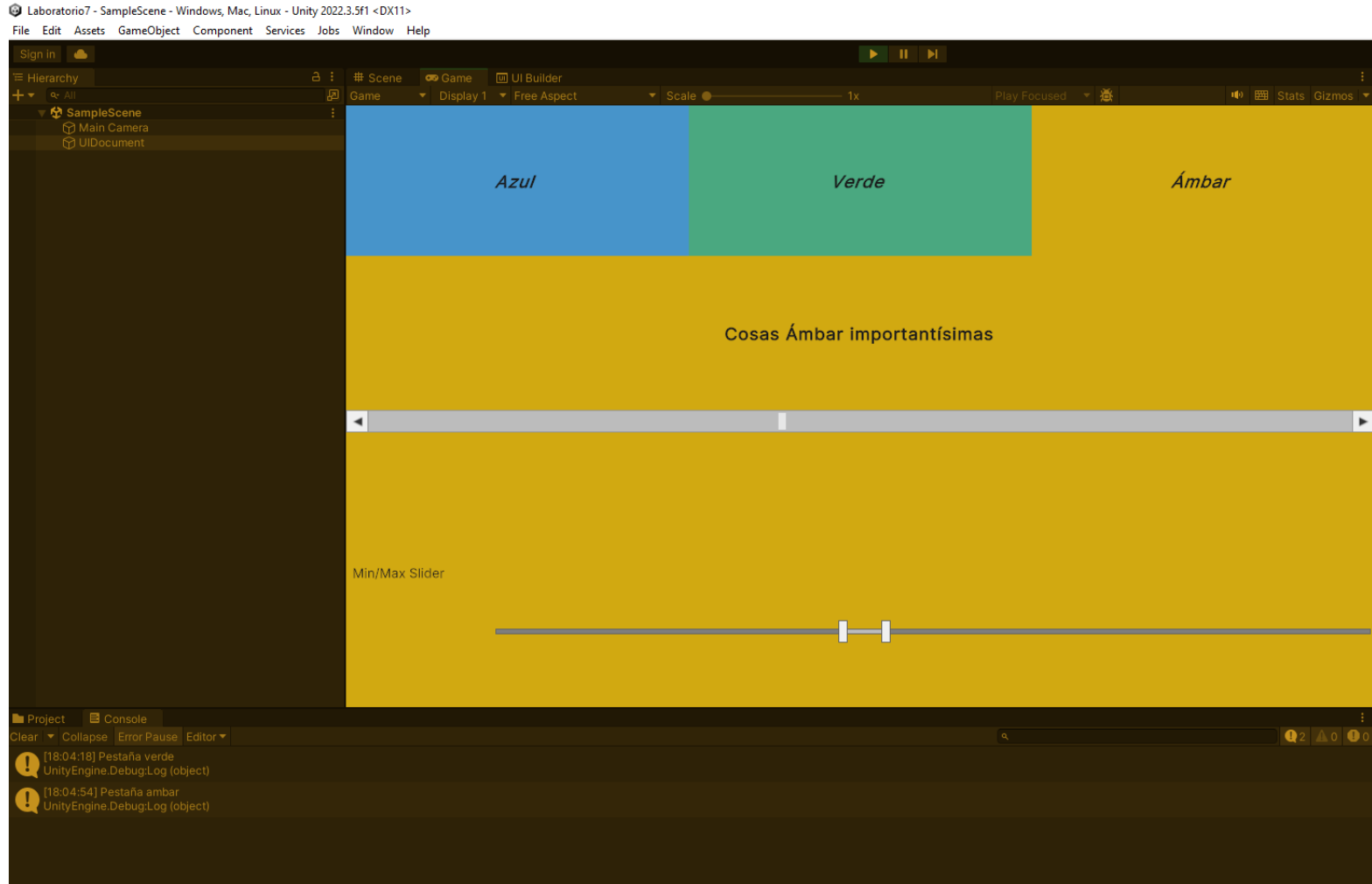
Menú con pestañas



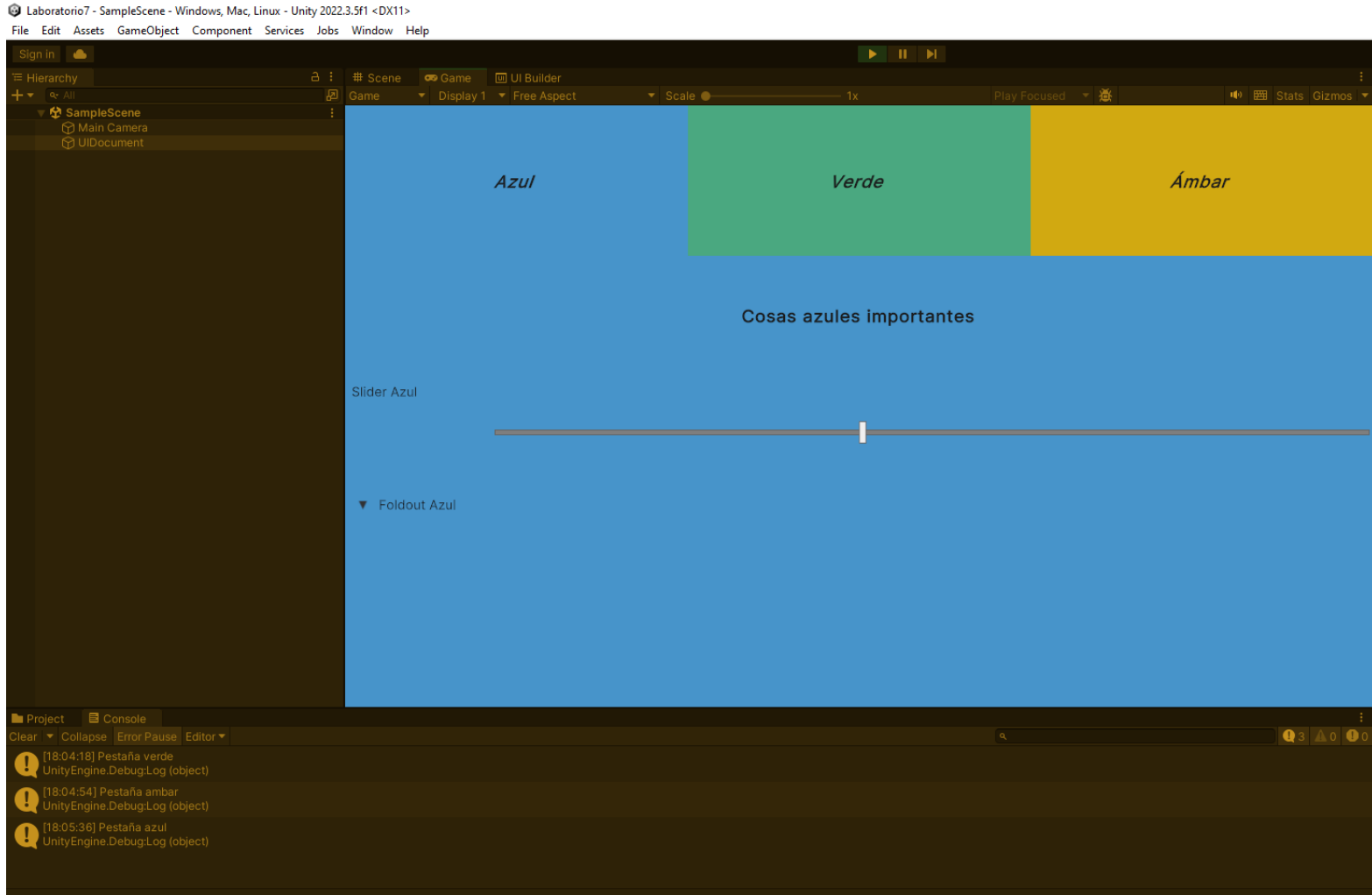
Menú con pestañas



Menú con pestañas



Menú con pestañas



Dragon Crashers → Tabbed menú



Dragon Crashers → TabbedMenu (campus virtual)

```
TabbedMenuController.cs  TabbedMenu.cs X
E: > PROYECTOS > UNITY > sample2 > Assets > Scripts > UI > Components > TabbedMenu.cs > ...
1  using UnityEngine;
2  using UnityEngine.UIElements;
3
4  namespace UIToolkitDemo
5  {
6      // adapted from: https://docs.unity3d.com/2021.2/Documentation/Manual/UIE-create-tabbed-menu-for-runtime.html
7
8      // This establishes a tabbed menu system for the UI document. This modifies the original to support multiple
9      // tabbed menus within the same document.
10
11      [System.Serializable]
12      1 reference
13      public struct TabbedMenuIDs
14      {
15          // UXML selector for a clickable tab
16          2 references
17          public string tabClassName; //= "tab";
18
19          // UXML selector for currently selected tab
20          2 references
21          public string selectedTabClassName; //= "selected-tab";
22
23          // UXML selector for content to hide
24          2 references
25          public string unselectedContentClassName; //= "unselected-content";
26
27          // use a basename to pair a tab with its content, e.g. 'name1-tab' matches 'name1-content'
28
29          // suffix naming convention for tabs
30          2 references
31          public string tabNameSuffix; //= "-tab";
32
33          // suffix naming convention for content
34          2 references
35          public string contentNameSuffix; //= "-content";
36      }
37      0 references
```


Dragon Crashers → TabbedMenu (campus virtual)

```
32 public class TabbedMenu : MonoBehaviour
    0 references
47 private void OnEnable()
48 {
49     if (m_Document == null)
50         m_Document = GetComponent<UIDocument>();
51
52     VisualElement root = m_Document.rootVisualElement;
53     m_MenuElement = root.Q(m_MenuElementName);
54
55     // create a new TabbedMenuController for a specific element (fall back to the entire tree if unspecified)
56     m_Controller = (string.IsNullOrEmpty(m_MenuElementName) || m_MenuElement == null) ?
57         new TabbedMenuController(root, m_TabbedMenuStrings) : new TabbedMenuController(m_MenuElement, m_TabbedMenuStrings);
58
59     // set up the click events on the tab
60     m_Controller.RegisterTabCallbacks();
61 }
62
63 // fill in default names for convenience - make these unique for each tabbed menu/UI
    0 references
64 void OnValidate()
65 {
66     if (string.IsNullOrEmpty(m_TabbedMenuStrings.tabClassName))
67     {
68         m_TabbedMenuStrings.tabClassName = "tab";
69     }
70
71     if (string.IsNullOrEmpty(m_TabbedMenuStrings.selectedTabClassName))
72     {
73         m_TabbedMenuStrings.selectedTabClassName = "selected-tab";
74     }
75
76     if (string.IsNullOrEmpty(m_TabbedMenuStrings.unselectedContentClassName))
77     {
78         m_TabbedMenuStrings.unselectedContentClassName = "unselected-content";
79     }
80
81     if (string.IsNullOrEmpty(m_TabbedMenuStrings.tabNameSuffix))
82     {
83         m_TabbedMenuStrings.tabNameSuffix = "-tab";
84     }
85
86     if (string.IsNullOrEmpty(m_TabbedMenuStrings.contentNameSuffix))
87     {
88         m_TabbedMenuStrings.contentNameSuffix = "-content";
89     }
90 }
```

Dragon Crashers → TabbedMenuController (campus virtual)

```
TabbedMenuController.cs X
E: > PROYECTOS > UNITY > sample2 > Assets > Scripts > UI > Controllers > TabbedMenuController.cs > TabbedMenuController

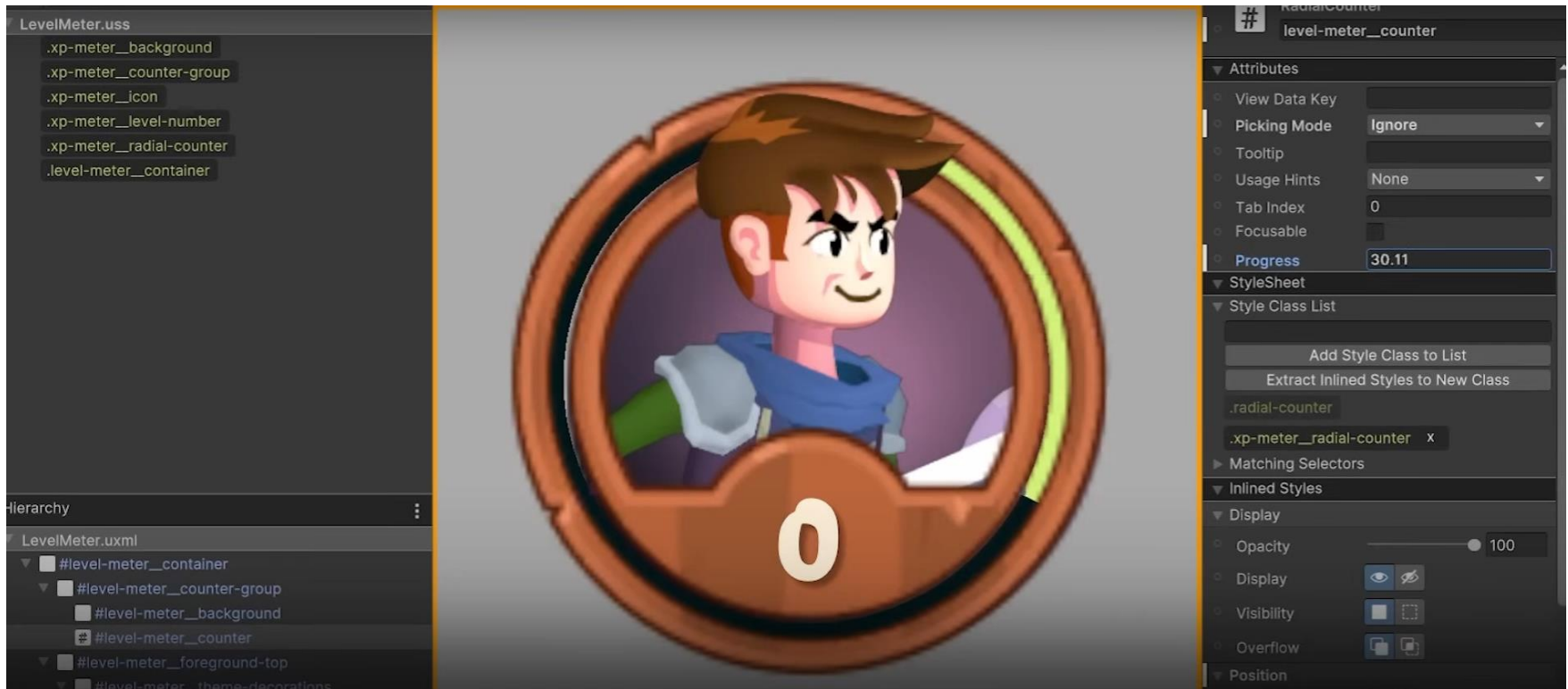
1 using UnityEngine;
2 using UnityEngine.UIElements;
3 using System;
4
5 namespace UIToolkitDemo
6 {
7
8     // adapted from: https://docs.unity3d.com/2021.2/Documentation/Manual/UIE-create-tabbed-menu-for-runtime.html
9
10    1 reference
11    public class TabbedMenuController
12    {
13        // event to notify other objects
14        1 reference
15        public static event Action TabSelected;
16
17        // base VisualElement of the UI (e.g. MailScreen, CharScreen, ShopScreen)
18        5 references
19        readonly VisualElement m_Root;
20
21        // strings to query for VisualElements
22        12 references
23        readonly TabbedMenuIDs m_IDs;
24
25        // initialize the root Visual Element for reuse (constructor for non-Monobehaviour)
26        0 references
27        public TabbedMenuController(VisualElement root, TabbedMenuIDs ids)
28        {
29            m_Root = root;
30            m_IDs = ids;
31        }
32
33        // set up click events for tab buttons
34        0 references
35        public void RegisterTabCallbacks()
36        {
37            // identify each tab
38            UQueryBuilder<VisualElement> tabs = GetAllTabs();
39
40            // register the ClickTab event handler for each Visual Element
41            tabs.ForEach(
42                (t) =>
43                {
44                    t.RegisterCallback<ClickEvent>(OnTabClick);
45                });
46        }
47    }
48
49    // locate all VisualElements that have the tab class name
50    4 references
51    UQueryBuilder<VisualElement> GetAllTabs()
52    {
53        return m_Root.Query<VisualElement>(className: m_IDs.tabClassName);
54    }
55}
```



Dragon Crashers → TabbedMenuController

```
41
42 // process a click
43 // 1 reference
44 void OnTabClick(ClickEvent evt)
45 {
46     VisualElement clickedTab = evt.currentTarget as VisualElement;
47
48     // if the clicked tab is not already selected, select the correct one
49     if (!IsTabSelected(clickedTab))
50     {
51         // de-select any other tabs that are currently active
52         GetAllTabs().Where(
53             (tab) => tab != clickedTab && IsTabSelected(tab)
54         ).ForEach(UnselectTab);
55
56         // select the clicked tab
57         SelectTab(clickedTab);
58         AudioManager.PlayDefaultButtonSound();
59     }
60 }
```

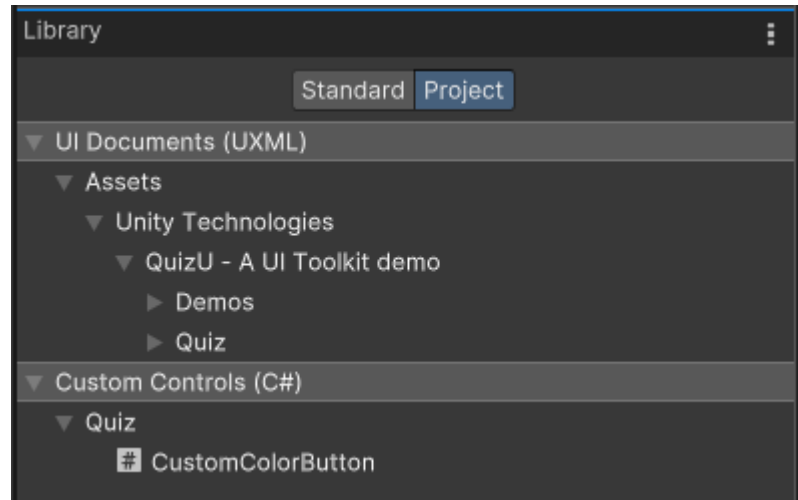
Dragon Crashers → radial progress bar



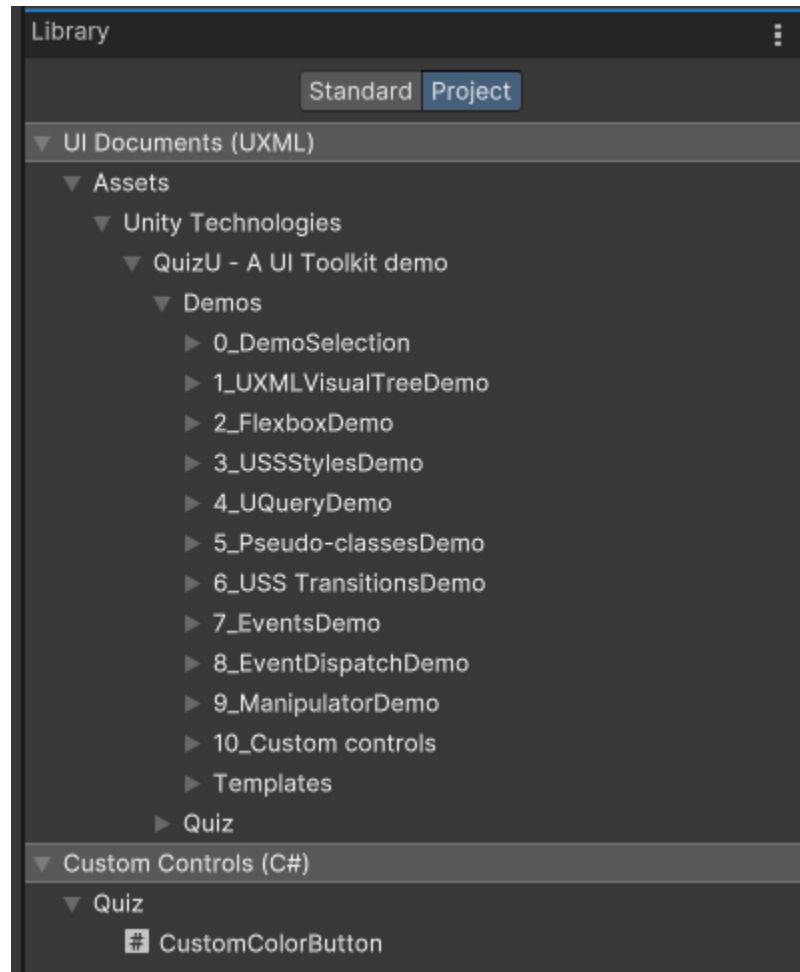
Índice

- Project Samples
 - Dragon Crashers
 - Custom Controls
 - Tabbed menú
 - Radial progress bar
 - Quizu
 - Demos
 - CustomControls
 - Manipulators
 - Quiz
 - Data
 - Scripts
 - Coroutines
 - ScriptablesObjects
 - Actions
- Proyecto

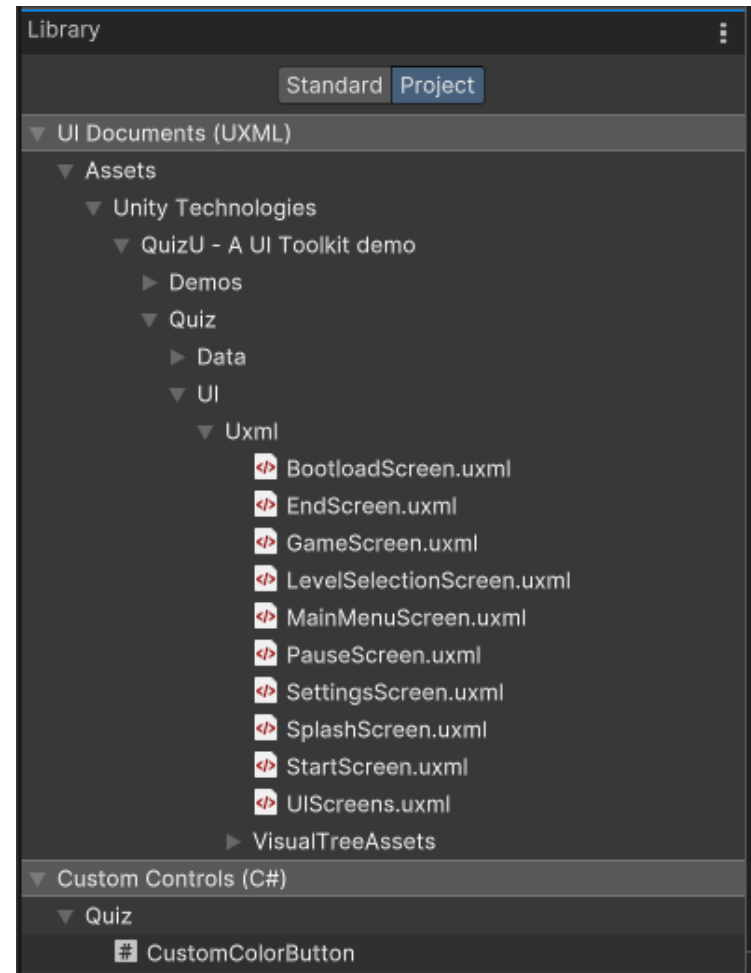
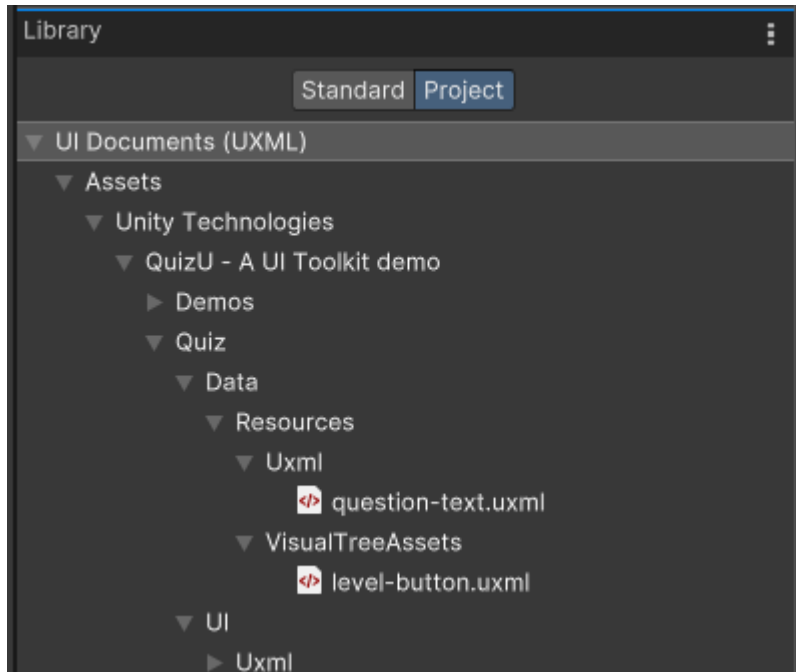
QuizU



QuizU - Demos



QuizU - Quiz



CustomControl → C# - USS transition

```
1  using UnityEngine;
2  using UnityEngine.UIElements;
3
4  namespace Quiz
5  {
6      /// <summary>
7      /// This button toggles between an active and inactive color, showing a simple custom control
8      /// available in the UI Builder.
9      /// </summary>
10     3 references
11     public class CustomColorButton : Button
12     {
13         // Properties to hold the active and inactive colors
14         2 references
15         Color activeColor = new Color(0f, 0.5f, 0.85f, 1f);
16         3 references
17         Color inactiveColor = new Color(0.5f, 0.5f, 0.5f, 1f);
18
19         3 references
20         bool m_IsActive = false;
21
22         1 reference
23         public new class UxmlFactory : UxmlFactory<CustomColorButton, UxmlTraits> { }
24
25         // This exposes attributes in the UI Builder
26         1 reference
27         public new class UxmlTraits : Button.UxmlTraits
28         {
29             1 reference
30             UxmlColorAttributeDescription m_AttrActiveColor =
31                 new UxmlColorAttributeDescription
32                 {
33                     name = "active-color",
34                     defaultValue = new Color(0f, 0.5f, 0.85f, 1f)
35                 };
36
37             1 reference
38             UxmlColorAttributeDescription m_AttrInactiveColor =
39                 new UxmlColorAttributeDescription
40                 {
41                     name = "inactive-color",
42                     defaultValue = new Color(0.5f, 0.5f, 0.5f, 1f)
43                 };
44         }
45     }
46 }
```

CustomControl → C# - USS transition

```
57
58 0 references
59 public CustomColorButton()
60 {
61     this.style.minWidth = 100;
62     this.style.minHeight = 100;
63
64     /// Optional: this shows the syntax for setting up a basic USS transition using the C# syntax
65     /// More information: https://docs.unity3d.com/Manual/UIE-Transitions.html
66
67     // Omitted for simplicity. Uncomment this block to add some "built-in" USS Transitions:
68
69     // TransitionDuration needs a List of TimeValues
70
71     List<TimeValue> durations = new List<TimeValue>();
72     durations.Add(new TimeValue(0.5f, TimeUnit.Second));
73     this.style.transitionDuration = new StyleList<TimeValue>(durations);
74
75     // easingFunctions also needs a List
76
77     List<EasingFunction> easingFunctions = new List<EasingFunction>();
78     easingFunctions.Add(new EasingFunction(EasingMode.EaseInOut));
79
80     this.style.transitionTimingFunction = new StyleList<EasingFunction>(easingFunctions);
81
82
83     // Use the Clickable Manipulator to toggle between active and inactive color.
84     // The clicked property is a shorthand way to subscribe to click events (versus RegisterCallback).
85     this.clicked += OnClick;
86
87 }
88
89 // Note: CustomColorButton instance is tightly bound to its OnClick handler; when the button is destroyed, the OnClick handler will be
90 // be garbage collected alongside the button.
91 1 reference
92 private void OnClick()
93 {
94     m_IsActive = !m_IsActive;
95     this.style.backgroundColor = m_IsActive ? activeColor : inactiveColor;
96 }
97 }
```


Manipulator → Dragger

```
21 public SimpleDragManipulator()
22 {
23     // This filter determines what condition starts the Manipulator (Left Mouse Button down)
24     activators.Add(new ManipulatorActivationFilter { button = MouseButton.LeftMouse });
25 }
26
27 // Event subscriptions
28 0 references
29 protected override void RegisterCallbacksOnTarget()
30 {
31     target.RegisterCallback<MouseDownEvent>(OnMouseDown);
32     target.RegisterCallback<MouseMoveEvent>(OnMouseMove);
33     target.RegisterCallback<MouseUpEvent>(OnMouseUp);
34 }
35
36 // Event unsubscriptions
37 0 references
38 protected override void UnregisterCallbacksFromTarget()
39 {
40     target.UnregisterCallback<MouseDownEvent>(OnMouseDown);
41     target.UnregisterCallback<MouseMoveEvent>(OnMouseMove);
42     target.UnregisterCallback<MouseUpEvent>(OnMouseUp);
43 }
44
45 // Check if we can click and drag an element and then initialize some values
46 2 references
47 private void OnMouseDown(MouseDownEvent evt)
48 {
49     // Checks whether MouseEvent satisfies all of the ManipulatorActivationFilter requirements.
50     if (CanStartManipulation(evt))
51     {
52         // Save the mouse and element start positions
53         m_StartMousePosition = evt.mousePosition;
54         m_StartElementPosition = new Vector2(target.layout.x, target.layout.y);
55
56         // Flag that this is active and receive all mouse events, even if the mouse pointer leaves
57         m_IsDragging = true;
58         target.CaptureMouse();
59
60         // Limit the event to this element, do not send up or down the hierarchy
61         evt.StopPropagation();
62     }
63 }
```

Manipulator → Dragger

```
61
62 // Offset the dragged element based on the difference from the start position
2 references
63 private void OnMouseMove(MouseMoveEvent evt)
64 {
65     // Checks whether the MouseEvent is related to this Manipulator and dragging is active
66     if (CanStopManipulation(evt) && m_IsDragging)
67     {
68         // Use the difference in mouse position to offset the element as well
69         Vector3 mouseDelta = evt.mousePosition - m_StartMousePosition;
70
71         // Convert the pixel offset into a new left/top StyleLength
72         target.style.left = CreatePixelLength(m_StartElementPosition.x + mouseDelta.x);
73         target.style.top = CreatePixelLength(m_StartElementPosition.y + mouseDelta.y);
74
75         // Limit the event to this element, do not send up or down the hierarchy
76         evt.StopPropagation();
77     }
78 }
79
80 // Creates a new StyleLength representing a length in pixels.
2 references
81 private StyleLength CreatePixelLength(float value)
82 {
83     return new StyleLength(new Length(value, LengthUnit.Pixel));
84 }
```

Campus Virtual → Documentation

QuizU documentation

Documentation for the QuizU - an official Unity sample demonstrating MVP, state pattern, managing menu screens and much more using UI Toolkit available on the [Unity Asset Store](#).

Version 1.01 Nov 1st 2023



Table of contents

Welcome to the new sample project QuizU.....	4
Key features.....	4
UI Toolkit demos.....	6
The mini-game.....	7
QuizU: Exploring the demo scenes.....	9
Demo selection.....	9
UXML and visual trees.....	10
GroupBox versus VisualElement.....	11
Flexbox rules and layout.....	11
USS styles.....	12
UQuery.....	13
Pseudo-classes.....	14
USS Transitions.....	15
Events.....	16
Event dispatch and propagation.....	17
Manipulators.....	18
Custom controls.....	19
Continuing with UI Toolkit.....	20

Índice

- Project Samples
 - Dragon Crashers
 - Custom Controls
 - Tabbed menú
 - Radial progress bar
 - Quizu
 - Demos
 - CustomControls
 - Manipulators
 - **Quiz**
 - **Data**
 - **Scripts**
 - Coroutines
 - ScriptablesObjects
 - Actions
- Proyecto

Quiz → UI Toolkit



4

Question 1 of 20

Which of the following is NOT a valid way to set style properties in UI Toolkit?

SELECT ONE:

Using a UXML file



Using a GUI Style

Using a USS file

Using a C# script

SUBMIT

Quiz → UI Toolkit



4

Question 1 of 20

Which of the following is NOT a valid way to set style properties in UI Toolkit?

SELECTED: **D**

Using a UXML file

Using a GUI Style

Using a USS file



Using a C# script

Correct


In UI Toolkit, you can set style properties using a UXML file, a USS file, or a C# script. However, UI Toolkit does not support setting style properties using a GUI Style (IMGUI), as it uses a separate system for styling UI elements.

CONTINUE

Quiz → UI Toolkit



4



Question 2 of 20

In UI Toolkit, which feature allows you to define how elements should be aligned and spaced within a container?

SELECT ONE:

Flexbox



Stack

Grid


Box

SUBMIT

Quiz → UI Toolkit



2 in a row!

4 

Question 2 of 20

In UI Toolkit, which feature allows you to define how elements should be aligned and spaced within a container?

SELECTED: **A**

Flexbox

Stack

Grid



Box

Correct


Flexbox allows you to define how elements should be aligned and spaced within a container. It can help you build complex UI layouts and automatically handles layout adjustments for different screen sizes.

CONTINUE

Quiz → UI Toolkit



2 in a row!

4 

Question 3 of 20

How can you specify a style that should be used when a specific condition is met in UI Toolkit?

SELECT ONE:

Using Conditional Styles



Using UI conditions

Using UI rules


Using Pseudo-classes

SUBMIT

Quiz → UI Toolkit



3 in a row!

4 

Question 3 of 20

How can you specify a style that should be used when a specific condition is met in UI Toolkit?

SELECTED: **C**

Using Conditional Styles

Using UI conditions

Using UI rules



Using Pseudo-classes

Correct


In UI Toolkit, **Pseudo-classes** are used to specify a style that should be used when a specific condition is met. This allows you to create dynamic, responsive designs.

CONTINUE

Quiz → UI Toolkit



3 in a row!

4 

Question 4 of 20

In UI Toolkit, what is the purpose of a **ScrollView**?

SELECT ONE:

To create a view that can be scrolled horizontally or vertically



To create a scrolling background

To create a view that can be rotated


To create a view that can be zoomed in or out

SUBMIT

Quiz → UI Toolkit



4 in a row!

4 

Question 4 of 20

In UI Toolkit, what is the purpose of a **ScrollView**?

SELECTED: **A**

To create a view that can be scrolled horizontally or vertically

To create a scrolling background

To create a view that can be rotated



To create a view that can be zoomed in or out

Correct


In UI Toolkit, a **ScrollView** is used to create a view that can be scrolled horizontally or vertically. This is especially useful when the content is too large to fit within the visible area.

CONTINUE

Quiz → UI Toolkit



4 in a row!

4 

Question 5 of 20

Which of the following is NOT a type of UI element in UI Toolkit?

SELECT ONE:

TextField



Scroll

Button


Label

SUBMIT

Quiz → UI Toolkit



5 in a row!

4 

Question 5 of 20

Which of the following is NOT a type of UI element in UI Toolkit?

SELECTED: **D**

TextField

Button

Label



Scroll

Correct


While Button, Label, and TextField are all types of UI elements in UI Toolkit, there is no 'Scroll' element. For scrolling functionality, you would use a ScrollView.

CONTINUE

Quiz → UI Toolkit



5 in a row!

4 

Question 6 of 20

What is the purpose of the **VisualTreeAsset** in UI Toolkit?

SELECT ONE:

To apply visual effects to UI elements

To represent a UI hierarchy

To manage UI animations

To hold visual styles for UI elements

SUBMIT

Quiz → UI Toolkit

6 in a row!

4

Question 6 of 20

What is the purpose of the **VisualTreeAsset** in UI Toolkit?



SELECTED: D

- To apply visual effects to UI elements
- To represent a UI hierarchy
- To manage UI animations
- To hold visual styles for UI elements


Correct
A **VisualTreeAsset** in UI Toolkit represents a UI hierarchy, which you can design in the UI Builder.

CONTINUE

Quiz → UI Toolkit



6 in a row!

4 

Question 7 of 20

Which of the following would you use to control the layout of multiple UI elements in UI Toolkit?

SELECT ONE:

VisualElement

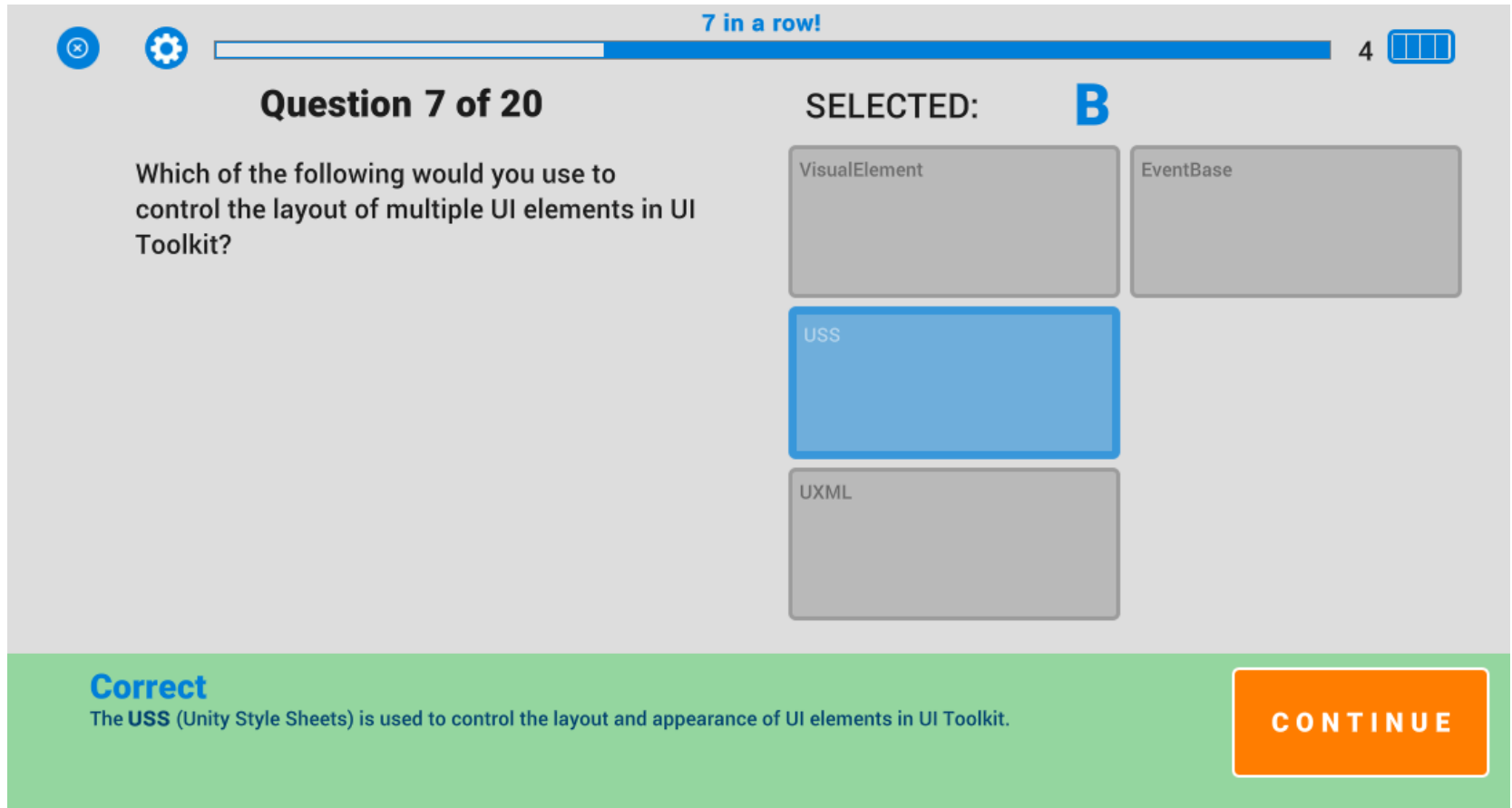
EventBase

USS

UXML

SUBMIT

Quiz → UI Toolkit



The image shows a quiz interface for the UI Toolkit. At the top, there's a progress bar with the text "7 in a row!" and a score of 4. The question is "Question 7 of 20" and asks "Which of the following would you use to control the layout of multiple UI elements in UI Toolkit?". The options are "VisualElement", "EventBase", "USS", and "UXML". "USS" is selected, indicated by a blue border and a blue "B" in the "SELECTED:" label. The bottom of the screen is green and displays "Correct" and the explanation: "The **USS** (Unity Style Sheets) is used to control the layout and appearance of UI elements in UI Toolkit." An orange "CONTINUE" button is in the bottom right corner.

7 in a row!

4

Question 7 of 20

Which of the following would you use to control the layout of multiple UI elements in UI Toolkit?

SELECTED: **B**

VisualElement

EventBase

USS



UXML

Correct


The **USS** (Unity Style Sheets) is used to control the layout and appearance of UI elements in UI Toolkit.

CONTINUE

Quiz → UI Toolkit



7 in a row!

4 

Question 8 of 20

Which of the following are correctly paired when describing parts of UI Toolkit?

SELECT ONE:

Hierarchy/Structure - USS,
Style/Layout - UXML,
Functionality/Logic - C#



Hierarchy/Structure - C#,
Style/Layout - UXML,
Functionality/Logic - USS

Hierarchy/Structure - UXML,
Style/Layout - C#,
Functionality/Logic - USS


Hierarchy/Structure - UXML,
Style/Layout - USS,
Functionality/Logic - C#

SUBMIT

Quiz → UI Toolkit



8 in a row!

4 

Question 8 of 20

Which of the following are correctly paired when describing parts of UI Toolkit?

SELECTED: C

Hierarchy/Structure - USS,
Style/Layout - UXML,
Functionality/Logic - C#

Hierarchy/Structure - C#,
Style/Layout - UXML,
Functionality/Logic - USS

Hierarchy/Structure - UXML,
Style/Layout - C#,
Functionality/Logic - USS



Hierarchy/Structure - UXML,
Style/Layout - USS,
Functionality/Logic - C#

Correct


In UI Toolkit, UXML is used for declaring the hierarchy/structure of the UI, USS is used for styling and layout, and C# is used for implementing functionality and logic.

CONTINUE

Quiz → UI Toolkit



8 in a row!

4 

Question 9 of 20

In UI Toolkit, which of the following is used to handle events?

SELECT ONE:

EventTrigger



EventBase

EventSystem


EventTarget

SUBMIT

Quiz → UI Toolkit



9 in a row!

4 

Question 9 of 20

In UI Toolkit, which of the following is used to handle events?

SELECTED: **D**

EventTrigger

EventSystem



EventTarget

EventBase


Correct
EventBase is the base class for all events in UI Toolkit.

CONTINUE

Quiz → UI Toolkit



9 in a row!

4 

Question 10 of 20

Which class in UI Toolkit allows you to create a dropdown menu?

SELECT ONE:

DropdownMenu



DropdownElement

MenuItem


PopupMenu

SUBMIT

Quiz → UI Toolkit



10 in a row!

4 

Question 10 of 20

Which class in UI Toolkit allows you to create a dropdown menu?

SELECTED: **A**

DropDownMenu

DropDownElement



MenuElement

PopupMenu


Correct
The DropDownMenu class in UI Toolkit is used for creating dropdown menus.

CONTINUE

Quiz → UI Toolkit



10 in a row!

4 

Question 11 of 20

In UI Toolkit, how can you specify that a style should only apply when an element is hovered over by the mouse?

SELECT ONE:

Using **:hover** pseudo-class



Using **HoverHandler** class

Using **hover** attribute


Using **onMouseHover** event

SUBMIT

Quiz → UI Toolkit



11 in a row!

4 

Question 11 of 20

In UI Toolkit, how can you specify that a style should only apply when an element is hovered over by the mouse?

SELECTED: **A**

Using `:hover` pseudo-class

Using `HoverHandler` class



Using `hover` attribute

Using `onMouseHover` event


Correct
In Unity's UI Toolkit, the `:hover` pseudo-class is used to apply styles when an element is hovered over by the mouse. This allows you to create interactive and responsive designs in your user interface.

CONTINUE

Quiz → UI Toolkit



11 in a row!

4 

Question 12 of 20

In Unity's UI Toolkit, which class is used to read input from users?

SELECT ONE:

InputReader

UserEvent

UserInput

PointerEvent

SUBMIT

Quiz → UI Toolkit

The screenshot shows a quiz interface with a grey background. At the top, there's a progress bar with the text "12 in a row!" in blue. To the left of the progress bar are two circular icons: one with a cross and one with a gear. To the right is a score indicator showing "4" and a full battery icon. The main question area on the left is titled "Question 12 of 20" and contains the text: "In Unity's UI Toolkit, which class is used to read input from users?". On the right, under the heading "SELECTED:", there's a large blue letter "C". Below this, five class names are listed in grey boxes: "InputReader", "UserEvent", "UserInput", and "PointerEvent". The "PointerEvent" box is highlighted with a blue border. At the bottom, a green banner contains the word "Correct" in blue, followed by the text: "PointerEvent is used in Unity's UI Toolkit to read user input, specifically, it handles pointer-based input like mouse and touch events." On the right side of the green banner is an orange button with the text "CONTINUE" in white.

12 in a row!

4

Question 12 of 20

In Unity's UI Toolkit, which class is used to read input from users?



SELECTED: **C**

- InputReader
- UserEvent
- UserInput
- PointerEvent**


Correct
PointerEvent is used in Unity's UI Toolkit to read user input, specifically, it handles pointer-based input like mouse and touch events.

CONTINUE

Quiz → UI Toolkit



12 in a row!

4 

Question 13 of 20

In UI Toolkit, which class allows you to create a progress bar?

SELECT ONE:

ProgressBar

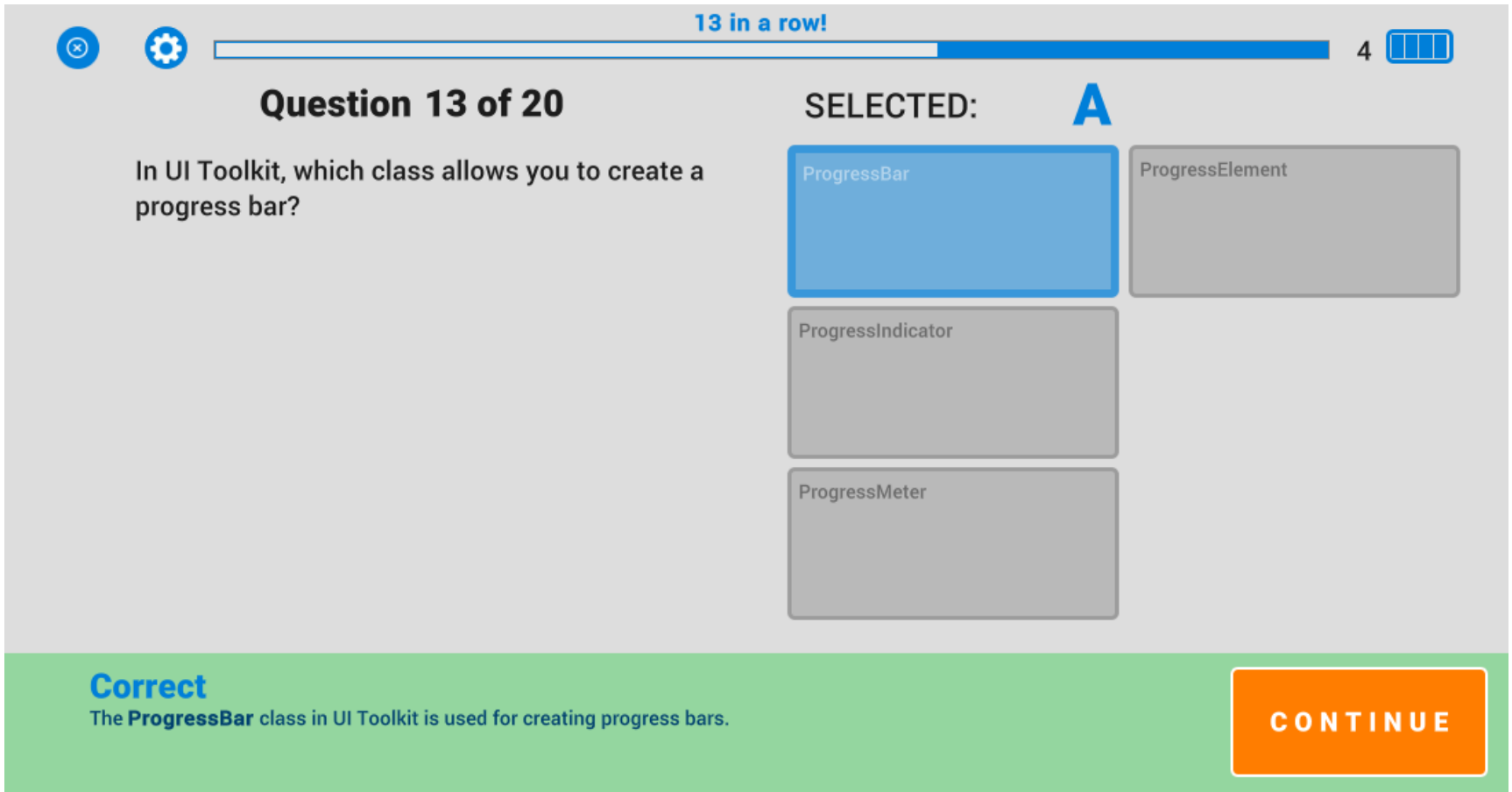
ProgressIndicator

ProgressMeter

ProgressElement

SUBMIT

Quiz → UI Toolkit



The image shows a quiz interface for the UI Toolkit. At the top, there's a progress bar indicating '13 in a row!' and a score of 4 out of 5. The question is 'Question 13 of 20' and asks 'In UI Toolkit, which class allows you to create a progress bar?'. The selected answer is 'A', which corresponds to 'ProgressBar'. Other options are 'ProgressElement', 'ProgressIndicator', and 'ProgressMeter'. The interface is green at the bottom, indicating a correct answer, with a 'CONTINUE' button.

13 in a row!

4

Question 13 of 20

In UI Toolkit, which class allows you to create a progress bar?

SELECTED: **A**

ProgressBar

ProgressElement

ProgressIndicator



ProgressMeter

Correct


The **ProgressBar** class in UI Toolkit is used for creating progress bars.

CONTINUE

Quiz → UI Toolkit



13 in a row!

4 

Question 14 of 20

In UI Toolkit, what is the purpose of **BindableElement**?

SELECT ONE:

It allows elements to be bound to data sources



It allows elements to be bound to animations

It is a base class for all interactive UI elements


It allows elements to be bound to event handlers

SUBMIT

Quiz → UI Toolkit



14 in a row!

4 

Question 14 of 20

In UI Toolkit, what is the purpose of **BindableElement**?

SELECTED: **A**

It allows elements to be bound to data sources

It allows elements to be bound to animations

It is a base class for all interactive UI elements



It allows elements to be bound to event handlers

Correct


The **BindableElement** class in UI Toolkit allows elements to be bound to data sources, such as variables or properties in your scripts, allowing them to automatically update to reflect changes in the underlying data.

CONTINUE

Quiz → UI Toolkit



14 in a row!

4 

Question 15 of 20

In UI Toolkit, what does the **flex-grow** property do in a Flexbox layout?

SELECT ONE:

It specifies the initial size of the flex item



It defines the ability for a flex item to grow if necessary

It determines the alignment along the main axis


It determines if the flex items are laid out in rows or columns

SUBMIT

Quiz → UI Toolkit



15 in a row!

4 

Question 15 of 20

In UI Toolkit, what does the **flex-grow** property do in a Flexbox layout?

SELECTED: **D**

It specifies the initial size of the flex item

It determines the alignment along the main axis

It determines if the flex items are laid out in rows or columns



It defines the ability for a flex item to grow if necessary

Correct


The **flex-grow** property in a Flexbox layout defines the ability for a flex item to grow if necessary. It accepts a unitless value that serves as a proportion, dictating the amount of space inside the flex container the item should take up.

CONTINUE

Quiz → UI Toolkit



15 in a row!

4 

Question 16 of 20

Which of the following is not a direct subclass of **VisualElement** in the UI Toolkit?

SELECT ONE:

ScrollView



DropDownMenu

Label


Button

SUBMIT

Quiz → UI Toolkit



16 in a row!

4 

Question 16 of 20

Which of the following is not a direct subclass of **VisualElement** in the UI Toolkit?

SELECTED: **D**

ScrollView

Label



Button

DropDownMenu


Correct
DropDownMenu is not a direct subclass of **VisualElement**. It is a part of the UI Toolkit, but it does not derive directly from **VisualElement**.

CONTINUE

Quiz → UI Toolkit



16 in a row!

4 

Question 17 of 20

In the UI Toolkit, what is the purpose of a **VisualElement**?

SELECT ONE:

It's a class used for audio elements in the UI



It's the base class for all elements in the UI Toolkit

It's used for applying visual effects to UI elements


It's used for creating 3D visual elements in the UI

SUBMIT

Quiz → UI Toolkit



17 in a row!

4 

Question 17 of 20

In the UI Toolkit, what is the purpose of a **VisualElement**?

SELECTED: **D**

It's a class used for audio elements in the UI

It's used for applying visual effects to UI elements



It's used for creating 3D visual elements in the UI

It's the base class for all elements in the UI Toolkit


Correct
VisualElement is the base class for all elements in the UI Toolkit. It is the core primitive from which all user interface controls and containers derive.

CONTINUE

Quiz → UI Toolkit



17 in a row!

4 

Question 18 of 20

In the UI Toolkit, what is the **UQueryBuilder** used for?

SELECT ONE:

It is used to manage multi-threading in the UI



It is used to bind UI elements to data sources

It is used to construct complex UI animations


It is used to query the UI hierarchy and find elements

SUBMIT

Quiz → UI Toolkit



18 in a row!

4 

Question 18 of 20

In the UI Toolkit, what is the **UQueryBuilder** used for?

SELECTED: **C**

It is used to manage multi-threading in the UI

It is used to bind UI elements to data sources

It is used to construct complex UI animations



It is used to query the UI hierarchy and find elements

Correct


The **UQueryBuilder** is used to query the UI hierarchy and find elements. It allows for complex queries to be constructed to find specific elements.

CONTINUE

Quiz → UI Toolkit



18 in a row!

4 

Question 19 of 20

Which method is used to add a child element to a **VisualElement** in the UI Toolkit?

SELECT ONE:

Append()



AddChild()

Add()


InsertChild()

SUBMIT

Quiz → UI Toolkit



19 in a row!

4 

Question 19 of 20

Which method is used to add a child element to a **VisualElement** in the UI Toolkit?

SELECTED: **B**

Append()

AddChild()



Add()

InsertChild()


Correct
The **Add()** method is used to add a child element to a **VisualElement** in the UI Toolkit.

CONTINUE

Quiz → UI Toolkit



19 in a row!

4 

Question 20 of 20

What is the purpose of the **USS** file in the UI Toolkit?

SELECT ONE:

It defines the visual style and layout of the UI



It is used for localization of the UI

It contains the script logic for the UI


It defines the structure of the UI

SUBMIT

Quiz → UI Toolkit



20 in a row!

4 

Question 20 of 20

What is the purpose of the **USS** file in the UI Toolkit?

SELECTED: **A**

It defines the visual style and layout of the UI

It is used for localization of the UI













It contains the script logic for the UI

It defines the structure of the UI

Correct
The **USS** (Unity Style Sheets) file is used to define the visual style and layout of the UI in the UI Toolkit.

CONTINUE

Quiz

UNITY > sample1 > Assets > Unity Technologies > QuizU - A UI Toolkit demo > Quiz >				
Nombre	Fecha de modificación	Tipo	Tamaño	
 Data	18/03/2024 17:46	Carpeta de archivos		
 Prefabs	18/03/2024 17:46	Carpeta de archivos		
 Scripts	18/03/2024 17:46	Carpeta de archivos		
 Textures	18/03/2024 17:46	Carpeta de archivos		
 TutorialInfo	18/03/2024 17:46	Carpeta de archivos		
 UI	18/03/2024 17:46	Carpeta de archivos		
 Data.meta	06/12/2023 14:48	Archivo META	1 KB	
 Prefabs.meta	06/12/2023 14:48	Archivo META	1 KB	
 Scripts.meta	06/12/2023 14:48	Archivo META	1 KB	
 Textures.meta	06/12/2023 14:48	Archivo META	1 KB	
 TutorialInfo.meta	06/12/2023 14:49	Archivo META	1 KB	
 UI.meta	06/12/2023 14:50	Archivo META	1 KB	

















Data → JSON

› UNITY › sample1 › Assets › Unity Technologies › QuizU - A UI Toolkit demo › Quiz › Data ›

Nombre	Fecha de modificación	Tipo	Tamaño
JSON	18/03/2024 17:46	Carpeta de archivos	
Questions	18/03/2024 17:46	Carpeta de archivos	
Resources	18/03/2024 17:46	Carpeta de archivos	
Unused	18/03/2024 17:46	Carpeta de archivos	
JSON.meta	06/12/2023 14:49	Archivo META	1 KB
Questions.meta	06/12/2023 14:49	Archivo META	1 KB
Resources.meta	06/12/2023 14:48	Archivo META	1 KB
Unused.meta	06/12/2023 14:50	Archivo META	1 KB

```
1  {
2    "text": "How many audio listeners should be present in a Unity scene?",
3    "skills": "Implement audio in Unity, Produce customized results by correctly configuring audio in a scene",
4    "shuffleOptions": true,
5    "options": [
6      {
7        "text": "As many as there are sound producing objects",
8        "isCorrect": false,
9        "feedback": "An audio listener plays any audio sources that are within range in a scene. There is only ever one audio listener, generally on the main camera."
10     },
11     {
12       "text": "Two",
13       "isCorrect": false,
14       "feedback": "An audio listener plays any audio sources that are within range in a scene. There is only ever one audio listener, generally on the main camera."
15     },
16     {
17       "text": "One for each directly interactable object",
18       "isCorrect": false,
19       "feedback": "An audio listener plays any audio sources that are within range in a scene. There is only ever one audio listener, generally on the main camera."
20     },
21     {
22       "text": "One",
23       "isCorrect": true,
24       "feedback": "There is only ever one audio listener, generally on the main camera."
25     }
26   ],
27   "feedbackCorrect": "There is only ever one audio listener, generally on the main camera.",
28   "feedbackIncorrect": ""
29 }
```

Scripts

» UNITY » sample1 » Assets » Unity Technologies » QuizU - A UI Toolkit demo » Quiz » Scripts »				
Nombre	Fecha de modificación	Tipo	Tamaño	
 Editor	18/03/2024 17:46	Carpeta de archivos		
 Events	18/03/2024 17:46	Carpeta de archivos		
 Gameplay	18/03/2024 17:46	Carpeta de archivos		
 Managers	18/03/2024 17:46	Carpeta de archivos		
 ScriptableObjects	18/03/2024 17:46	Carpeta de archivos		
 StateMachine	18/03/2024 17:46	Carpeta de archivos		
 UI	18/03/2024 17:46	Carpeta de archivos		
 Utilities	18/03/2024 17:46	Carpeta de archivos		
 Editor.meta	06/12/2023 14:49	Archivo META	1 KB	
 Events.meta	06/12/2023 14:48	Archivo META	1 KB	
 Gameplay.meta	06/12/2023 14:49	Archivo META	1 KB	
 Managers.meta	06/12/2023 14:48	Archivo META	1 KB	
 ScriptableObjects.meta	06/12/2023 14:49	Archivo META	1 KB	
 StateMachine.meta	06/12/2023 14:48	Archivo META	1 KB	
 UI.meta	06/12/2023 14:49	Archivo META	1 KB	
 Utilities.meta	06/12/2023 14:49	Archivo META	1 KB	

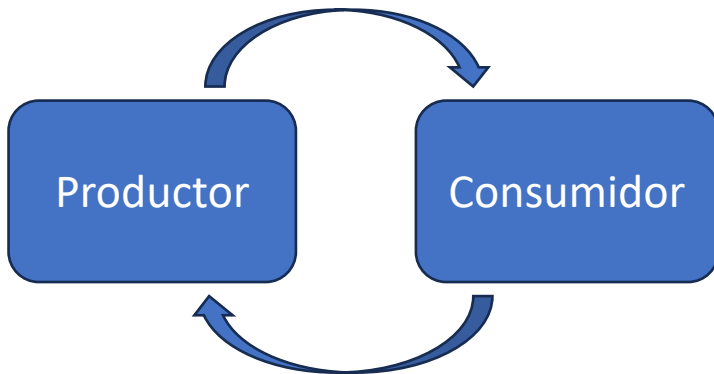
State Machine → Coroutines

```
var q := new queue

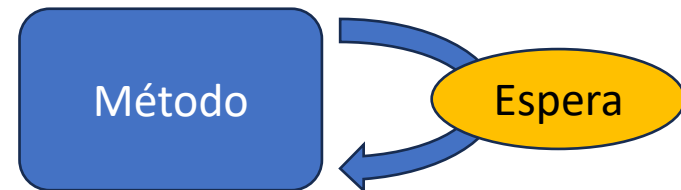
coroutine produce
  loop
    while q is not full
      create some new items
      add the items to q
    yield to consume

coroutine consume
  loop
    while q is not empty
      remove some items from q
      use the items
    yield to produce

call produce
```



```
IEnumerator Fade()
{
    Color c = renderer.material.color;
    for (float alpha = 1f; alpha >= 0; alpha -= 0.1f)
    {
        c.a = alpha;
        renderer.material.color = c;
        yield return new WaitForSeconds(.1f);
    }
}
```











State Machine → Coroutines

```
124 | 1 reference
125 | protected virtual IEnumerator Loop()
126 | {
127 |     while (true)
128 |     {
129 |         if (CurrentState != null && m_CurrentPlayCoroutine == null) //current state is done playing
130 |         {
131 |             if (CurrentState.ValidateLinks(out var nextState))
132 |             {
133 |                 if (m_PlayLock)
134 |                 {
135 |                     //finalize current state
136 |                     CurrentState.Exit();
137 |                     m_PlayLock = false;
138 |                 }
139 |                 CurrentState.DisableLinks();
140 |                 SetCurrentState(nextState);
141 |                 CurrentState.EnableLinks();
142 |             }
143 |         }
144 |         yield return null;
145 |     }
146 | }
147 |
148 |
149 | 0 references
150 | public bool IsRunning => m_LoopCoroutine != null;
151 | }
```

ScriptableObjects

- Objetos que son instancias de clases que heredan de MonoBehaviour (“MonoBehaviourObjects”) son almacenados en memoria y existen solo mientras se juega el juego.
- **ScriptableObjects** → son almacenados en el disco duro en forma serializada (existen durante y después del juego). Son usados como un contenedor/plantilla para almacenar datos.

UNITY > sample1 > Assets > Unity Technologies > QuizU - A UI Toolkit demo > Quiz > Scripts > ScriptableObjects			
Nombre	Fecha de modificación	Tipo ^	Tamaño
 AudioSettingsSO.cs	06/12/2023 14:50	Archivo de origen ...	6 KB
 DescriptionSO.cs	06/12/2023 14:48	Archivo de origen ...	1 KB
 QuestionSO.cs	06/12/2023 14:49	Archivo de origen ...	5 KB
 QuizSO.cs	06/12/2023 14:49	Archivo de origen ...	3 KB
 AudioSettingsSO.cs.meta	06/12/2023 14:50	Archivo META	1 KB
 DescriptionSO.cs.meta	06/12/2023 14:48	Archivo META	1 KB
 QuestionSO.cs.meta	06/12/2023 14:49	Archivo META	1 KB
 QuizSO.cs.meta	06/12/2023 14:49	Archivo META	1 KB

AudioSettingsSO

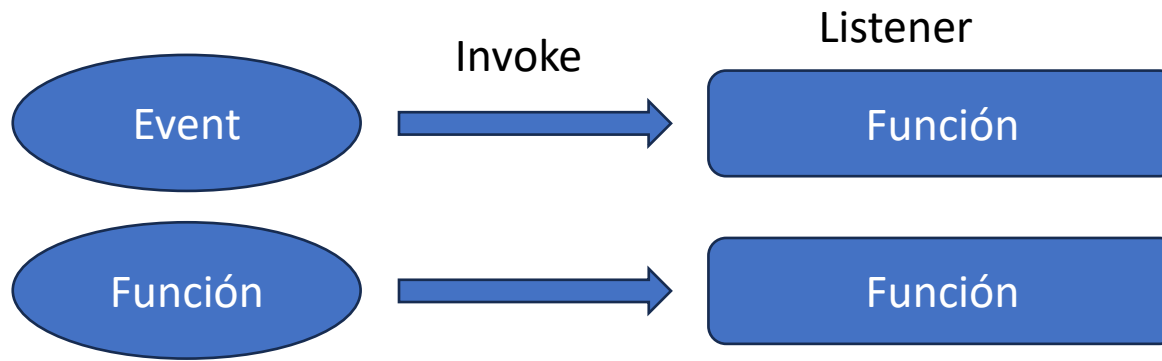
```
14 [CreateAssetMenu(fileName = "AudioSettings", menuName = "Quiz/AudioSettings", order = 1)]
15 0 references
16 public class AudioSettingsSO : DescriptionSO
17 {
18     // Default ScriptableObject data
19     1 reference
20     const float k_DefaultMasterVolume = 1f;
21     1 reference
22     const float k_DefaultSFXVolume = 1f;
23     1 reference
24     const float k_DefaultMusicVolume = 0f;
25
26     [Header("Mixer")]
27     [Tooltip("The AudioMixer that controls the audio levels for the game")]
28     1 reference
29     [SerializeField] private AudioMixer m_AudioMixer;
30
31     [Header("Volume Settings")]
32     [Tooltip("The master volume level (0 to 1)")]
33     2 references
34     [SerializeField] private float m_MasterVolume = k_DefaultMasterVolume;
35
36     [Tooltip("The sound effects volume level (0 to 1)")]
37     2 references
38     [SerializeField] private float m_SoundEffectsVolume = k_DefaultSFXVolume;
39
40     [Tooltip("The music volume level (0 to 1)")]
41     2 references
42     [SerializeField] private float m_MusicVolume = k_DefaultMusicVolume;
43
44     // Convert bool to 1 and 0 and multiply by MixerGroup (unused in this project, here for demo purposes)
45     [Header("Mute Settings")]
46     [Tooltip("Mute or unmute the master volume")]
47     2 references
48     [SerializeField] private bool m_IsMasterMuted = false;
49
50     [Tooltip("Mute or unmute the sound effects volume")]
51     2 references
52     [SerializeField] private bool m_IsSoundEffectsMuted = false;
53
54     [Tooltip("Mute or unmute the music volume")]
55     2 references
56     [SerializeField] private bool m_IsMusicMuted = false;
57
58 }
```

QuestionSO

```
33 public class QuestionSO : ScriptableObject
34 {
35     [Tooltip("List of question text, supporting different font sizes.")]
36     [SerializeField] List<QuestionText> m_QuestionText;
37     [Tooltip("Optional image associated with the question.")]
38     [SerializeField] [Optional] Sprite m_QuestionImage;
39     [TextArea(3, 5)]
40     [Tooltip("Skills tested or used in the question.")]
41     [SerializeField] string m_Skills;
42     [Tooltip("Should the answers should be shuffled?")]
43     [SerializeField] bool m_ShuffleAnswers;
44     [Tooltip("Additional information about the correct/incorrect responses")]
45     [SerializeField] List<Answer> m_Answers;
46     [TextArea(3, 5)]
47     [Tooltip("Additional information provided when the correct answer is chosen.")]
48     [SerializeField] string m_FeedbackCorrect;
49     [TextArea(3, 5)]
50     [Tooltip("Additional information provided when an incorrect answer is chosen.")]
51     [SerializeField] string m_FeedbackIncorrect;
52     [Tooltip("Optional link to the documentation for more information.")]
53     [SerializeField][Optional] string m_FeedbackLink;
54 }
```

Actions

- En clase hemos visto Delegates, events y callbacks. Resumiendo:
 - Delegates en C# son punteros a funciones. Los Delegates nos permiten usar las funciones como parámetros de entrada de otras funciones. Esto nos permite crear Callbacks.
 - En los Callbacks, un evento de sistema dispara una función.
 - En las Actions, el final de una función dispara otra función.
 - Observer Pattern:



Events

```
15 public class AudioSettingsSO : DescriptionSO
76
77     // Event subscriptions
    0 references
78     private void OnEnable()
79     {
80         SettingsEvents.MasterVolumeChanged += SettingsEvents_MasterVolumeChanged;
81         SettingsEvents.MusicVolumeChanged += SettingsEvents_MusicVolumeChanged;
82         SettingsEvents.SFXVolumeChanged += SettingsEvents_SFXVolumeChanged;
83
84         // Note: we disable validating the Sound Effects AudioClips here, instead we
85         // validate it from the GameplaySounds class
86
87         // NullRefChecker.Validate(this);
88     }
89
90     // Event unsubscriptions
    0 references
91     private void OnDisable()
92     {
93         SettingsEvents.MasterVolumeChanged -= SettingsEvents_MasterVolumeChanged;
94         SettingsEvents.MusicVolumeChanged -= SettingsEvents_MusicVolumeChanged;
95         SettingsEvents.SFXVolumeChanged -= SettingsEvents_SFXVolumeChanged;
96     }
97
```

Game events → Actions

```
0 references
12 public static class GameEvents
13 {
14     #region Gameplay events
15
16     0 references
17     public static Action<int> IndexUpdated;
18
19     // Refreshed the current Question with new ScriptableObject data
20     0 references
21     public static Action<QuestionSO> QuestionUpdated;
22
23     // Selected one or more user selected responses
24     0 references
25     public static Action<List<string>> AnswerSelected;
26
27     // Submitted the selected answer
28     0 references
29     public static Action AnswerSubmitted;
30
31     // Passes feedback for correct/incorrect responses
32     0 references
33     public static Action<string> CorrectlyAnswered;
34
35     0 references
36     public static Action<string> IncorrectlyAnswered;
37 }
```

Índice

- Project Samples
 - Dragon Crashers
 - Custom Controls
 - Tabbed menú
 - Radial progress bar
 - Quizu
 - Demos
 - CustomControls
 - Manipulators
 - Quiz
 - Data
 - Scripts
 - Coroutines
 - ScriptablesObjects
 - Actions

• Proyecto

Proyecto

- El proyecto se hace individual o en grupos de 2-3 personas.
- Tiene que recoger más o menos todo lo que hemos ido viendo en las prácticas.
- Crearos una cuenta en Github (si no la tenéis ya).
- En el campus virtual hay dos enlaces:
 - 1. PDF con el prototipo en Figma.
 - 2. Fichero de texto con el enlace a github para el proyecto.

Último día presentación → Proyecto y prácticas

Mayo 2024						
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26

Examen (si no proyecto)

Miércoles 29	9:00	2ºDSI
	13:00	
de mayo	17:00	CSV
Jueves 30	9:00	IW
	13:00	
de mayo	17:00	EIRV
Viernes 31	9:00	2ºP2
	13:00	
de mayo	17:00	3ºP3
		Gr. Videojuegos