UI events

To interact with Unity's built-in UI elements, you need to perform extra steps, particularly if you're dealing with 3D-tracked devices. The XR Interaction Toolkit package provides a number of new components that you can use to convert an XR controller to work seamlessly with the UI, as well as helper menu options that handle basic configuration settings.

Before adding a button, a slider, or any other UI element, click on GameObject > UI > Canvas to generate a Canvas which will contain all UI elements, as well as an EventSystem component. In the Canvas properties, change the Render Mode to **World Space**, and set the Event Camera to the Main Camera. Then, use the Scale property to make the Canvas fit your scene.

> **Important:** If you directly modify the Width and Height of your Canvas instead of scaling it, the Canvas might not work as intended. This also applies to Buttons and other UI elements.

The Event System component acts as a central dispatch for UI events to process input, and update individual active canvases. Additionally, each Event System needs an Input Module to process input. The EventSystem component might have a Standalone Input Module or an Input System UI Module, which will prevent proper input processing. Remove the component, and add an **XR UI Input Module** to the event system.

Finally, add a Tracked Device Graphic Raycaster component to the Canvas, and set the Raycast Trigger Interaction to "Collide". This will allow your Raycast to actually hit elements on the Canvas instead of passing through them.

To test your setup, add a Button object as a child of the Canvas. If you select the button by pressing the trigger on your Right controller, the button should change color. Now, you can add your own script to the Button object, and change the `On Click ()` property to run that script when the button is pressed.

Revision #2
Created 23 March 2023 14:59:39 by Matilda Fogato
Updated 24 March 2023 10:06:11 by Matilda Fogato