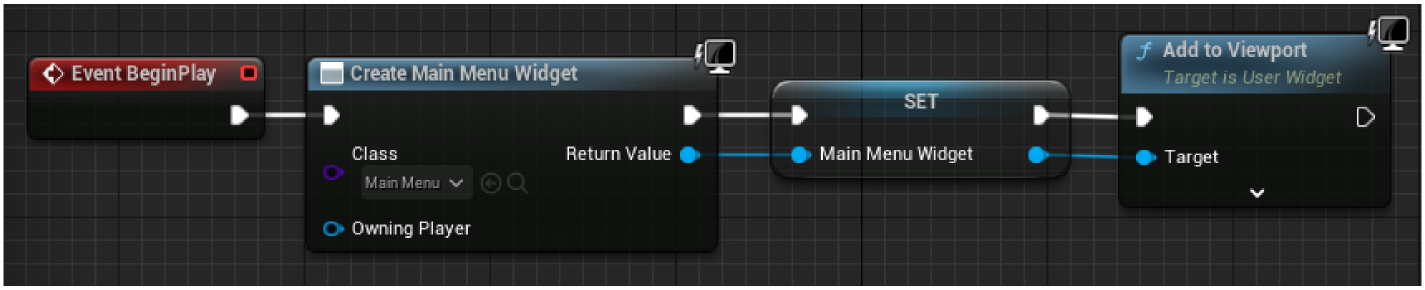
Creating Widgets

How to create and display in game a Widget Blueprint.

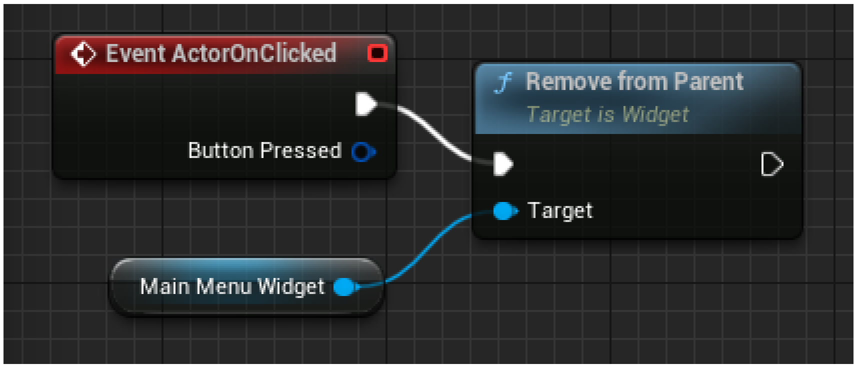
Once you have created and laid out your **Widget Blueprint**, in order for it to be displayed in-game, you will need call it by using the **Create Widget** and **Add to Viewport** nodes inside another Blueprint (**Level Blueprint** or a **Character Blueprint** for example).



In the example above, the **Create Widget** node is calling the Widget Blueprint specified under the **Class** section and the **Return Value** is the resulting product (**Owning Player** is the **Player Controller** and is applied to blank results in the default Player Controller). The **Add to Viewport** is what draws the Widget Blueprint onscreen and we are specifying that the Main Menu variable, which holds the created Widget, is the added Target.

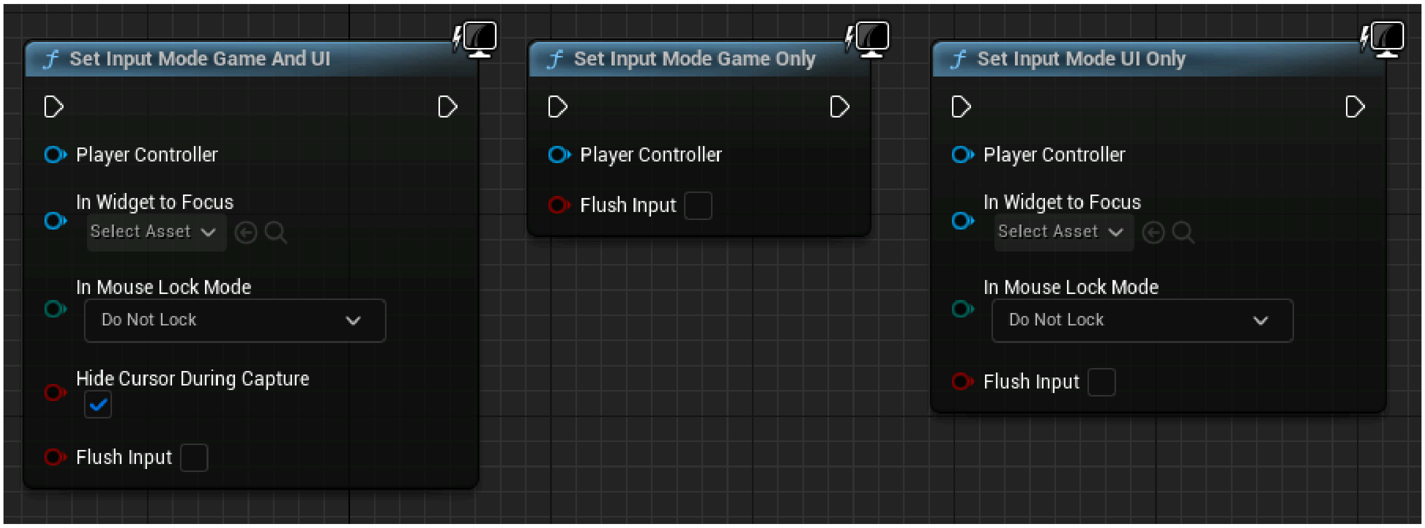
Additionally, the Return Value is being assigned to a variable titled Main Menu, which allows us to access the Widget Blueprint later instead of re-creating the Widget and, if needed, to remove the widget.

To remove the Widget from being displayed, use the **Remove from Parent** node and specify the Target Widget Blueprint.



Set Input Mode and Show Cursor

There may be instances when you want the player to interact with your UI and there may be instances when you want them to completely ignore it. There are some nodes that can be used to determine how the player interacts with the UI, which are of the **Set Input Mode** type as seen below.

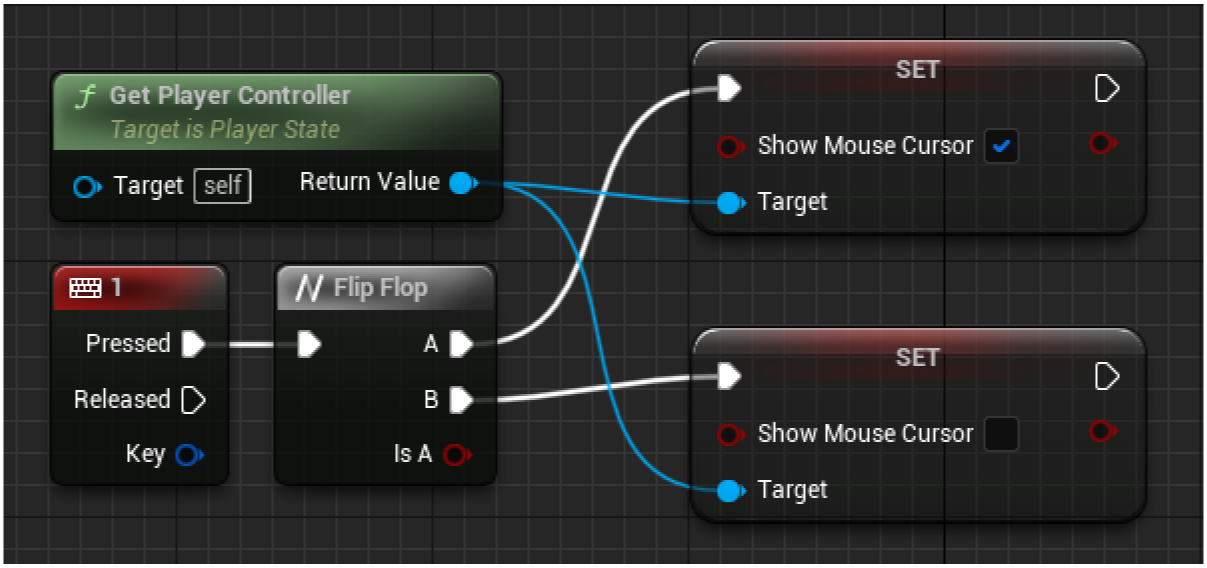


- The left node above, **Set Input Mode Game and UI**, does as the name implies and allows the player to manipulate the game via inputs as well as UI (for example, controlling a character onscreen but also having access to clicking on any buttons or UI elements you may have).
- The center node above, **Set Input Mode Game Only**, enables input only for the game and ignores UI elements (this is perfect for non-interactive UI elements like Health, Points, or

Time display).

- The right node above, **Set Input Mode UIOnly**, is an edge case node and is used when you only want to allow for UI navigation only and no game input. This completely disables all game controls and UI consumes all input, so use this node sparingly.

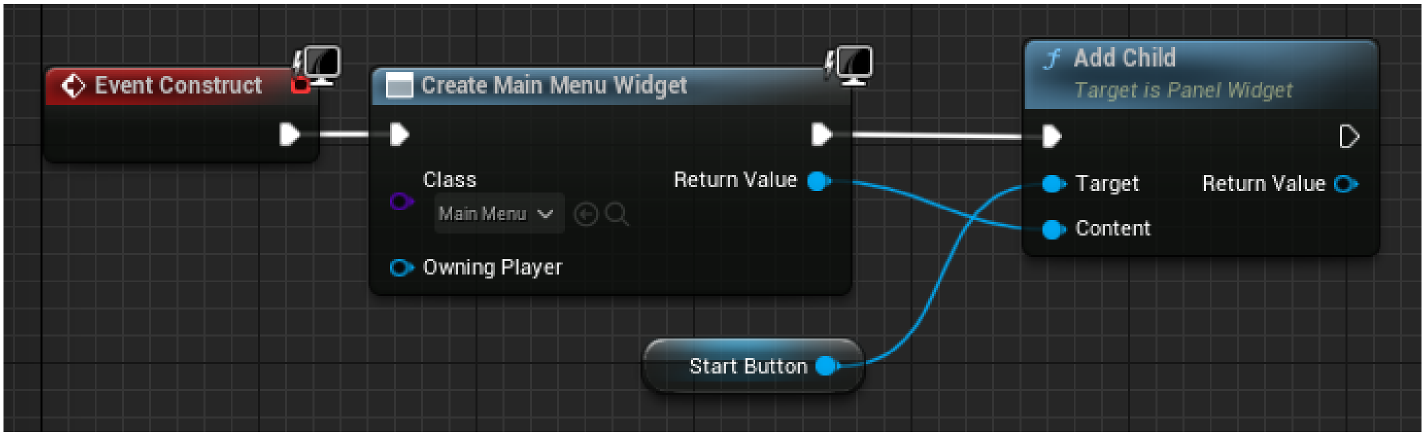
To supplement the nodes above, you may want to **Enable/Disable** the **Mouse Cursor** from being displayed. You can do so by using the **Set Show Mouse Cursor** node. Drag off a **Get Player Controller** node, then use the **Set Show Mouse Cursor** node and check it to True or False to display or hide the Mouse Cursor.



Above, **I** is used to toggle between showing/hiding the Mouse Cursor.

Adding Widgets to Widgets

You can add widgets to other widgets by creating a Parent-Child relationship where the second widget is nested under the first widget. To do this, simply append the child widget to the parent widget. You do not need to use the **Add to Viewport** function.



The example above shows how a **Scroll Box** Widget called start_Button is appended to a new Widget called Main Menu using the **Add Child** function.

> The **Add Child** node is for parenting one widget to another in a panel while the **Add to Viewport** adds the widget to the root viewport as if it is a new window. In order to remove a child widget, you need to get your Parent and call **RemoveChild**.