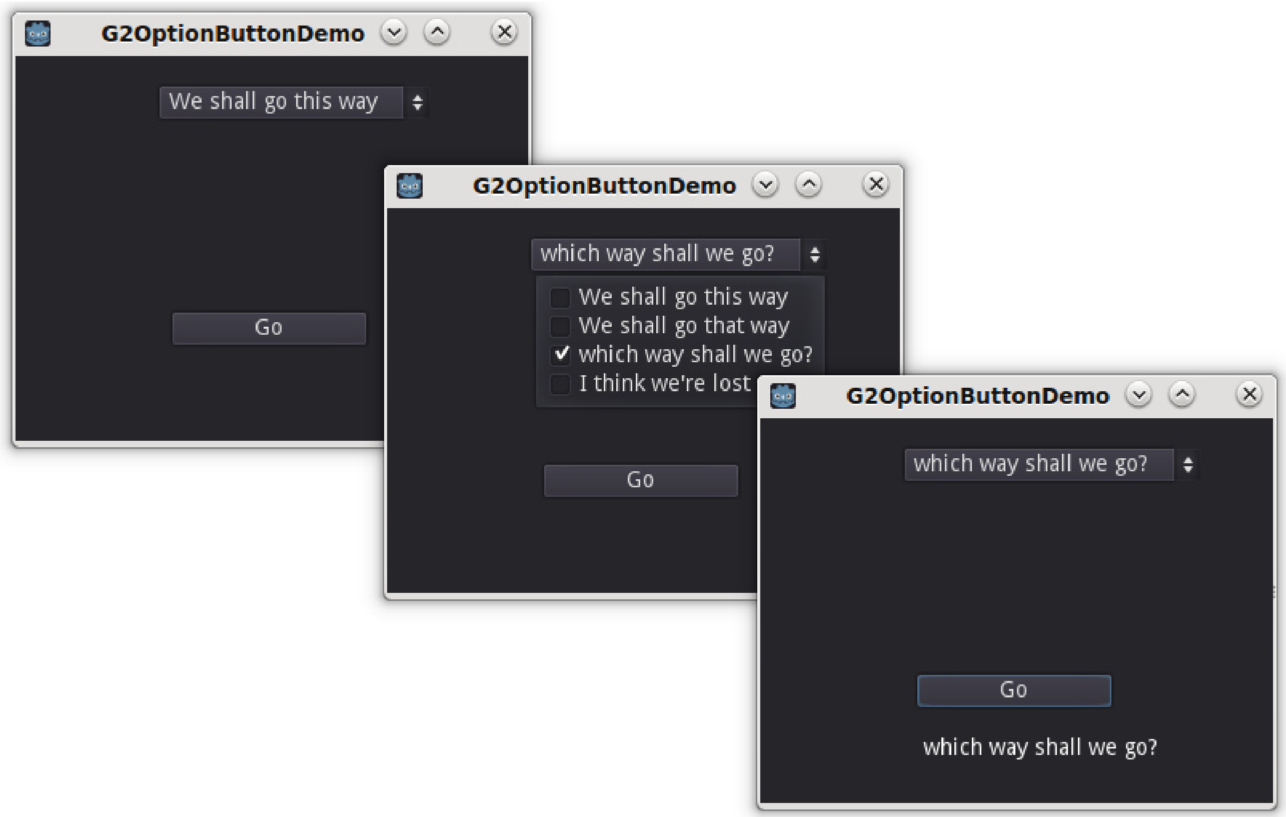
Godot 2 OptionButton Node

This article demonstrates using the ***OptionButton Node*** in Godot 2.

Using the OptionButton Node

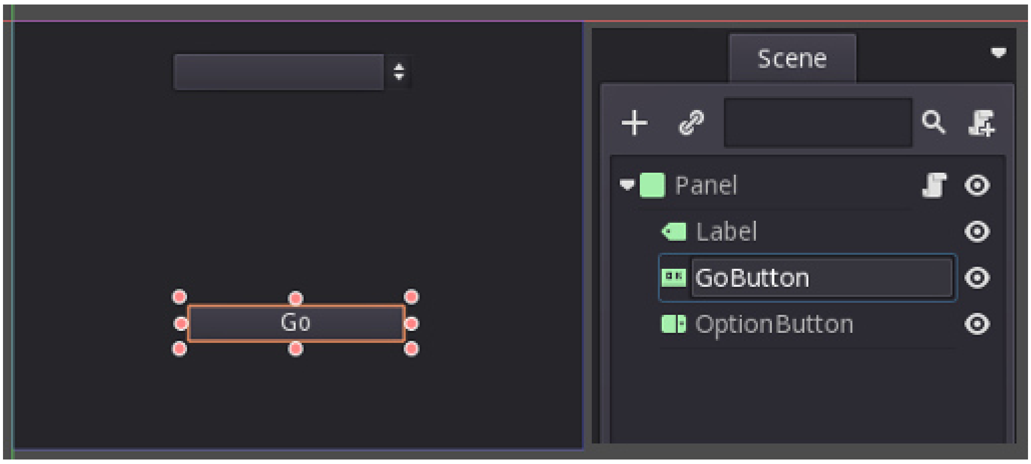
In this example the OptionButton *node* is loaded with the list components using code, and a button is used to take the selected item and display it in a label.



This demonstrates the interface in action.

Download the demonstration project from here:***G2OptionButtonDemo.zip***

Scene Setup



The scene is comprised of;

- a *Panel* for the root node,
- a *Button* node (Renamed “GoButton”),
- a *Label* node in which to display the selected item’s text.,
- and the *OptionButton* node.

The *Panel*, and the *Labels* are all using the default settings with the only changes being the Panel's *Size* which is set to match the project's window size (320x240).

Similarly the *Button* is just a default button with the only change being the addition of “Go” to the *Text* field.

The Script

In order to make it all work, a script is required. This loads the items into the *OptionButton* node and controls the query to the *OptionButton* node when the “Go” button is pressed.

Attached to the *Panel* node is the following script:

```
extends Panel

var ItemListContent = ["We shall go this way","We shall go that way","which way
shall we go?","I think we're lost"]

func _ready():
	#Load the OptionButton content
	for ItemID in range(4):
		get_node("OptionButton").add_item(ItemListContent[ItemID],ItemID)

	get_node("OptionButton").select(0)  #This sets a default so we don't
	# have to do error catching if an empty selection is captured.

	get_node("GoButton").connect("pressed",self,"ReportListItem")

func ReportListItem():
	var ItemNo = get_node("OptionButton").get_selected()
		#The output ItemNo is the number of the selected item
	var SelectedItemtext = ItemListContent[ItemNo]
	get_node("Label").set_text(str(SelectedItemtext))
	print(ItemNo)
```

In the `ready()` part of the script the first loop loads the *OptionButton* node with the members of the `ItemListContent` array. In this example we use `.select()` to set a default value just in case the user gets excited and hits the “Go” button without making a selection.

When the “`GoButton`” is pressed the script runs the “`ReportListItem`” function. This function gets the selected items from the *OptionButton node* and returns the list of position value of the selected item in the list. We can use this to find the matching line of text in the `ItemListContent` array and use this to display in the *Label*. We also print the selected item array to the console.

It can be made a bit prettier with the application of *Themes*.