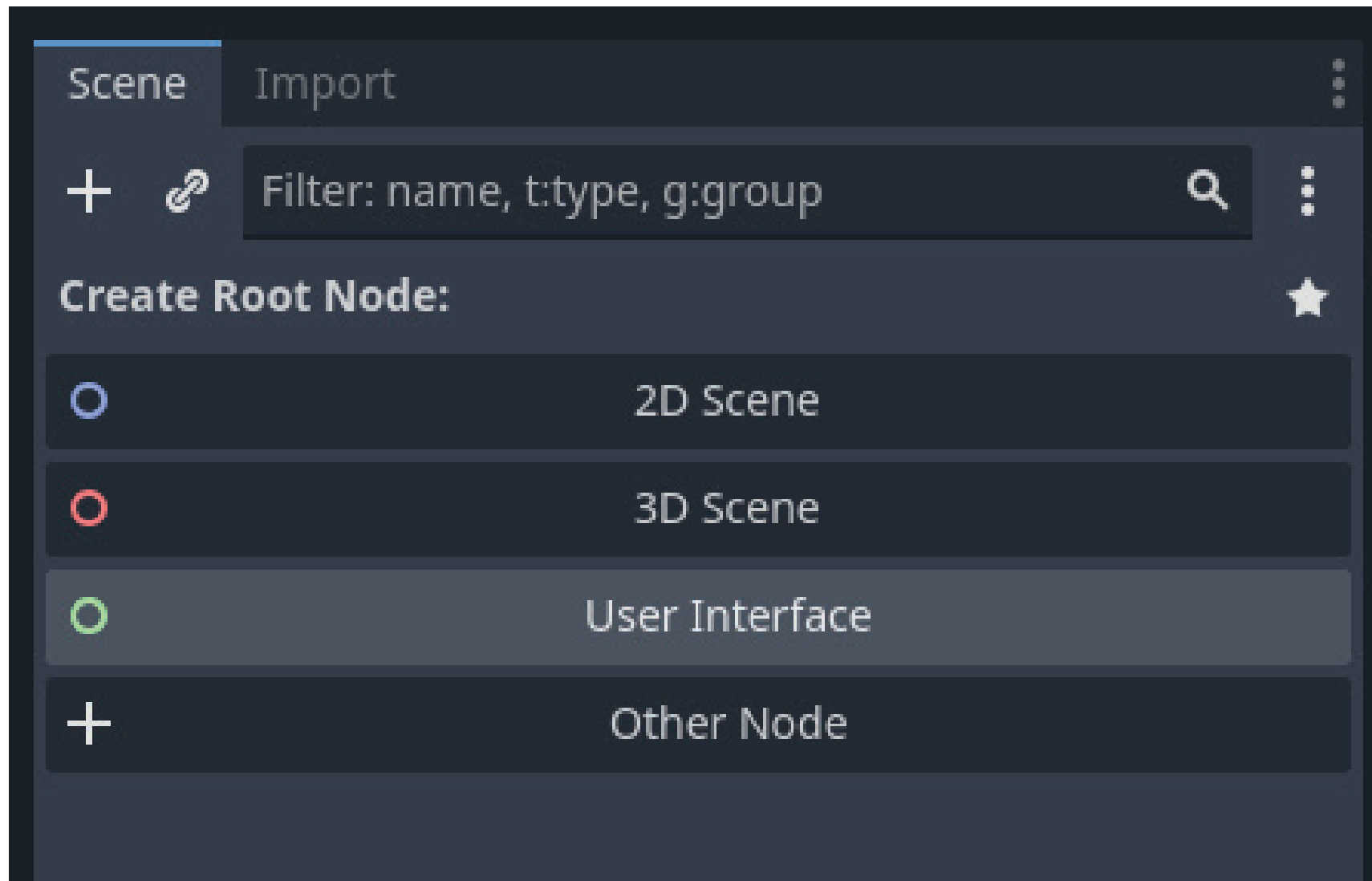
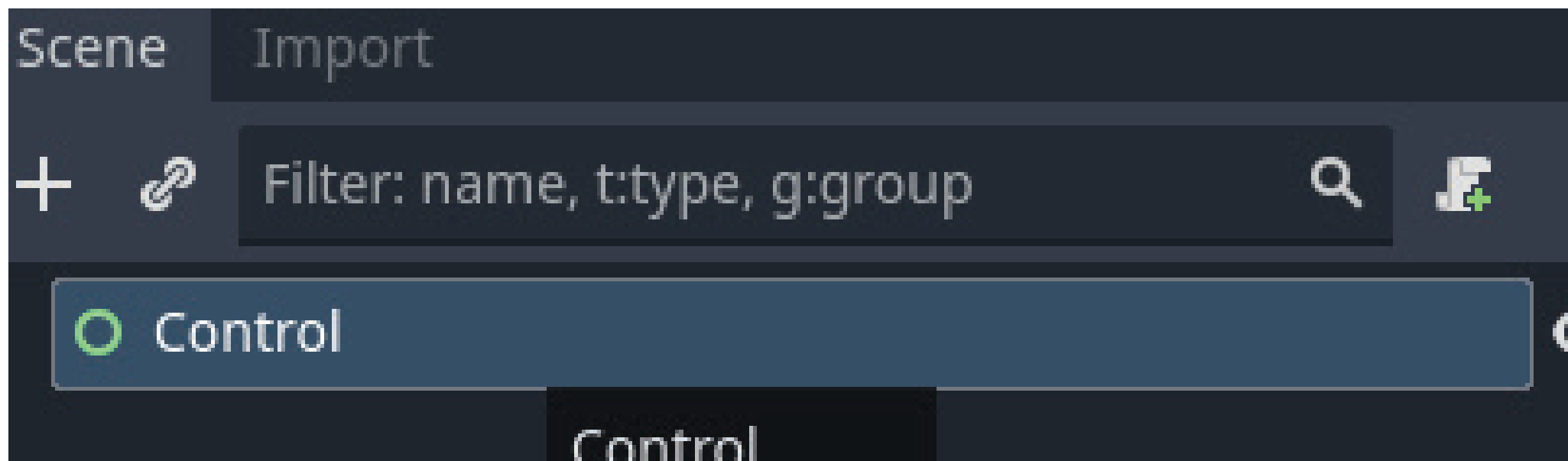
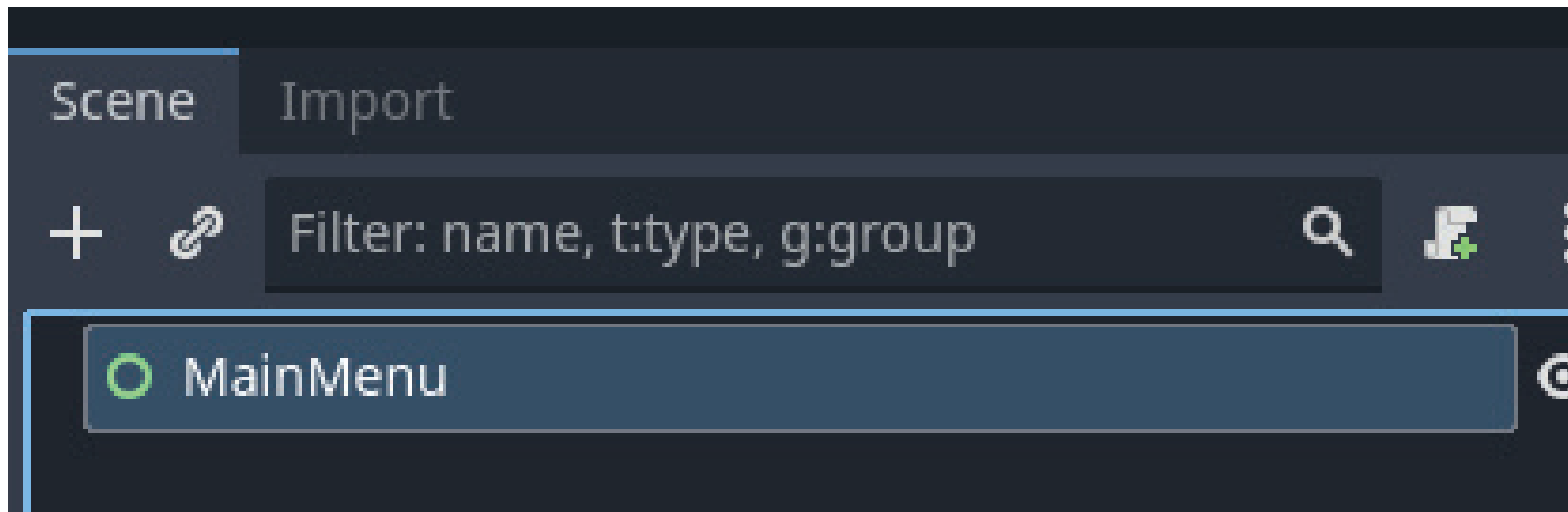


Adding a Main Menu!

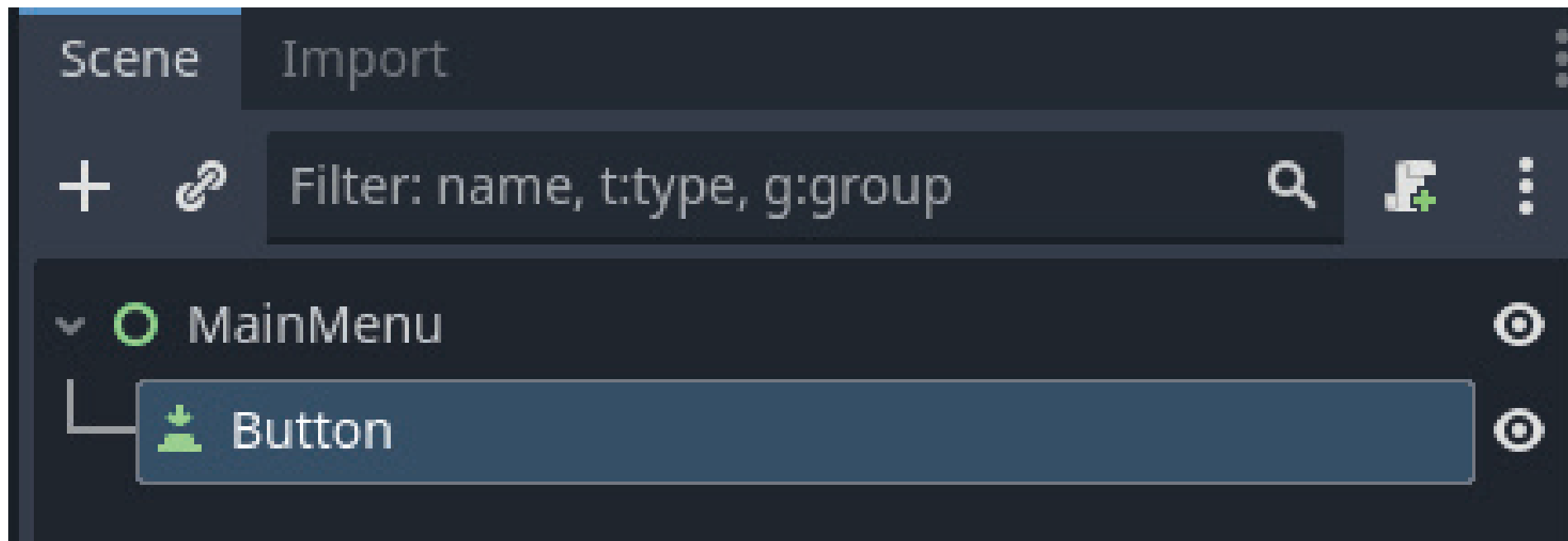


Create a new Scene
Choose > User Interface
this will create a Control
node.

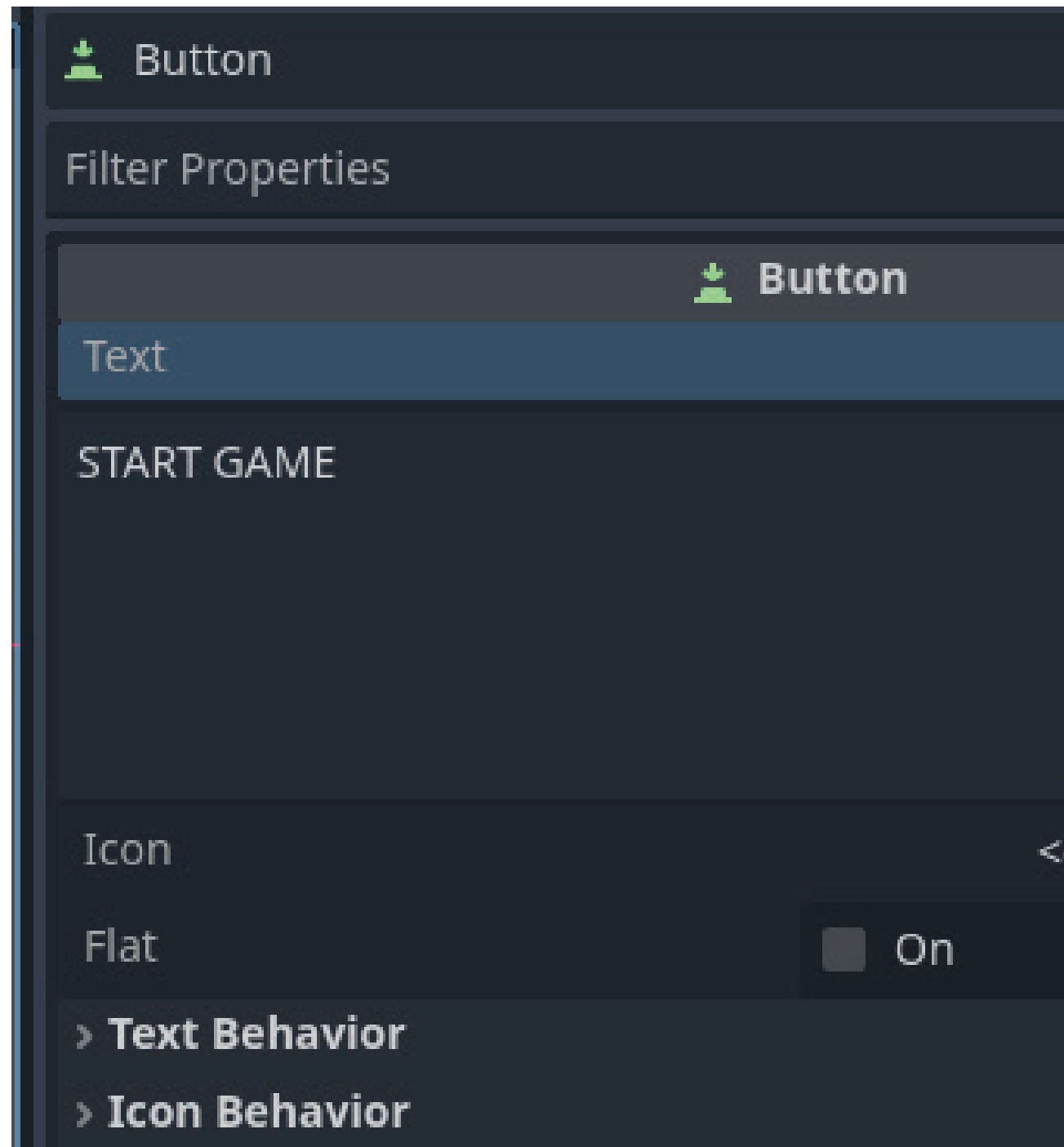




Name this node
MainMenu

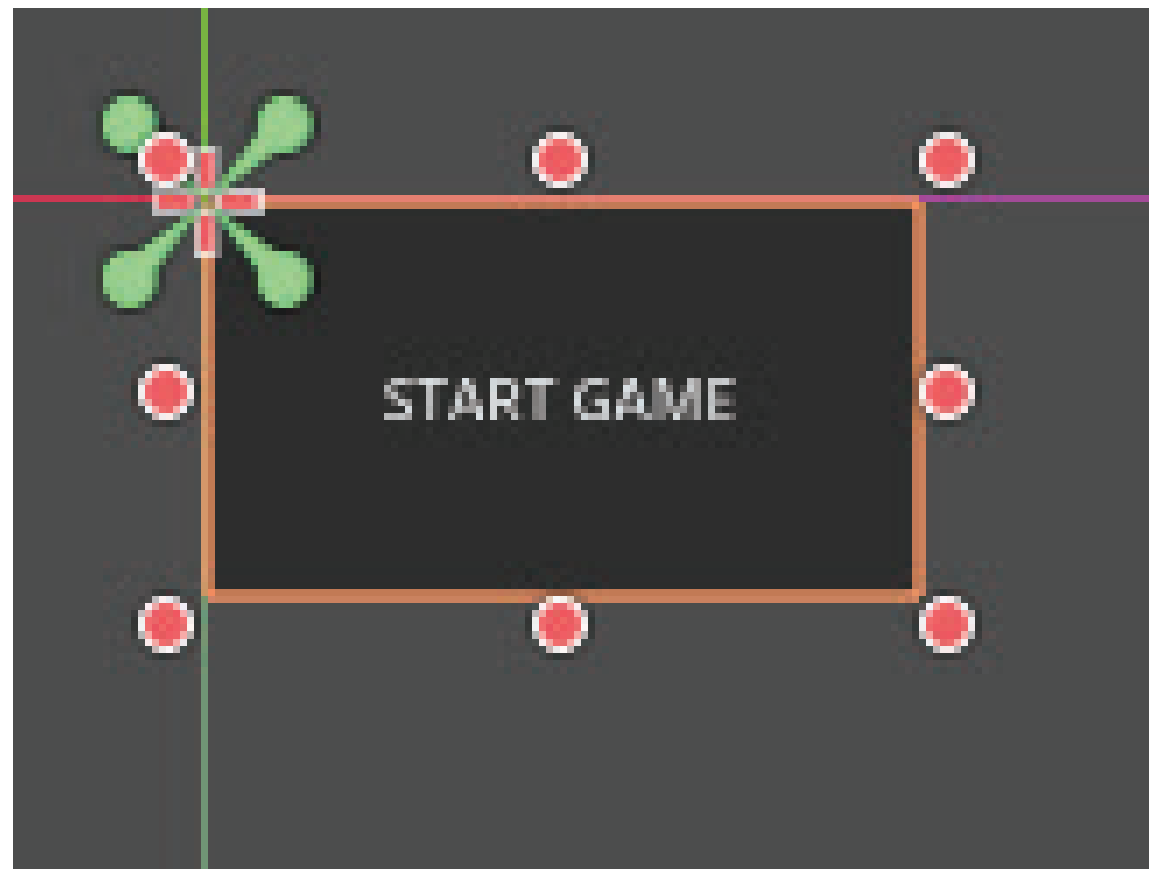


add a Child
choose Button

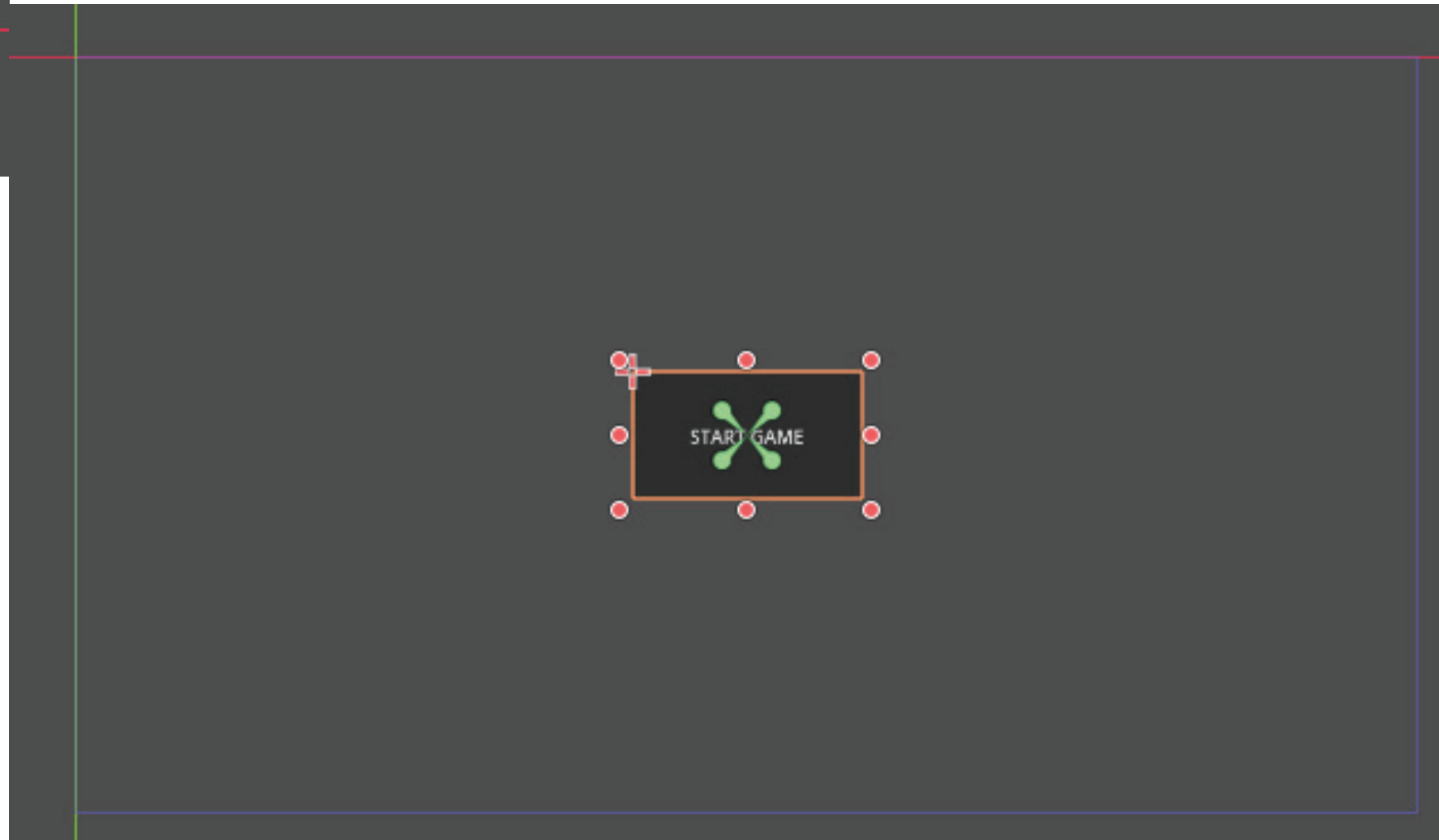
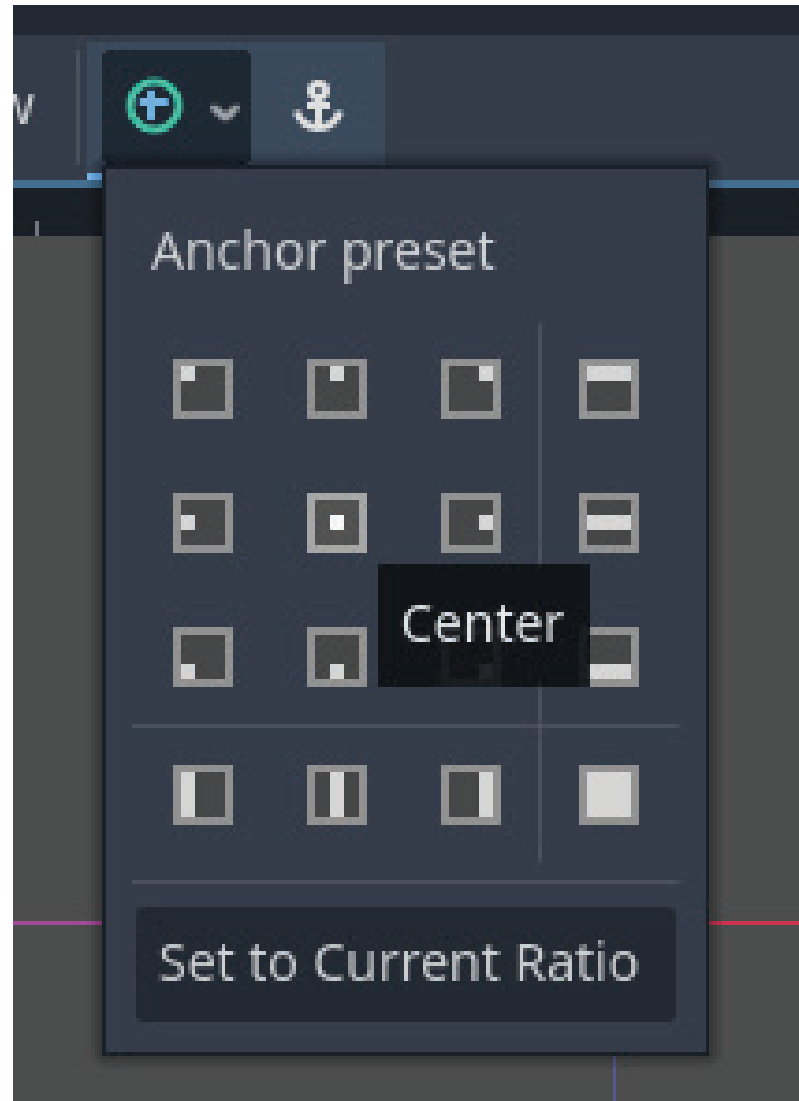


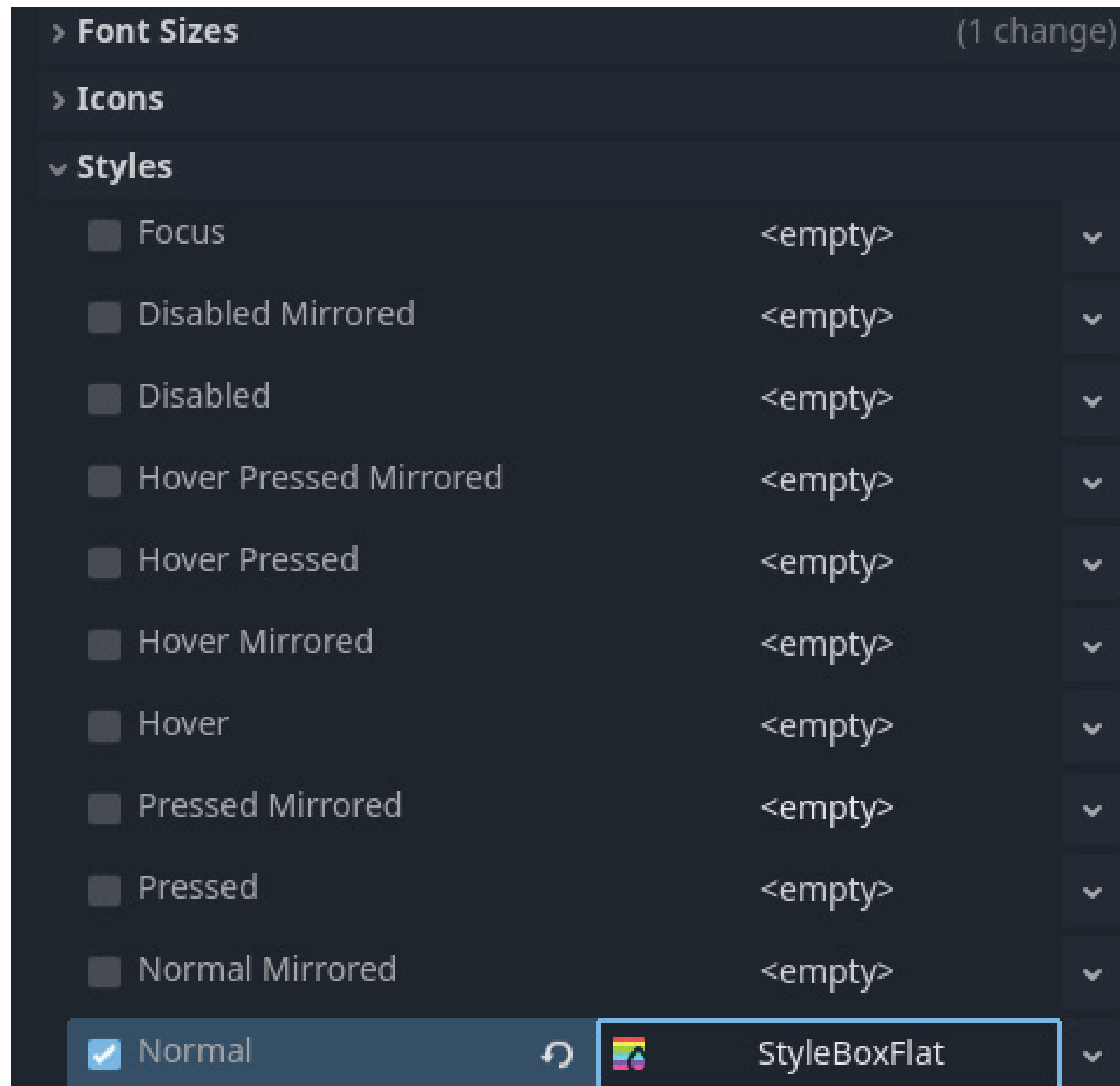
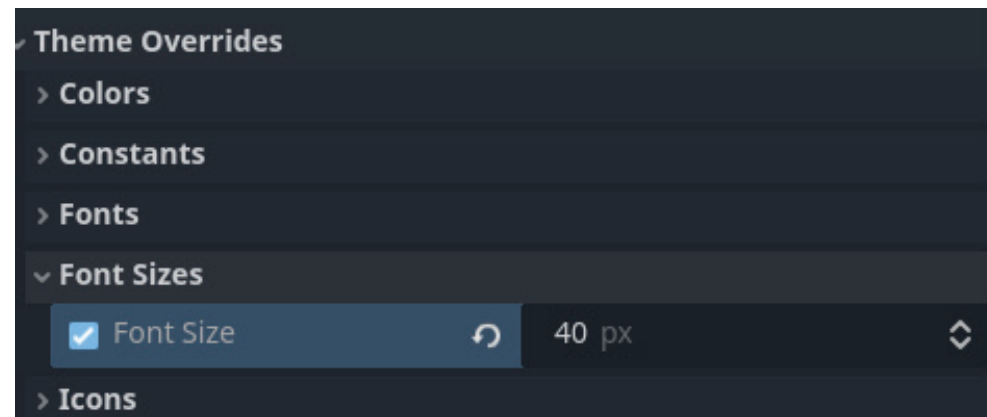
Type in the Button
Inspector > START
MENU

Make the box bigger.



Use the Anchor Preset button and use the CENTER option

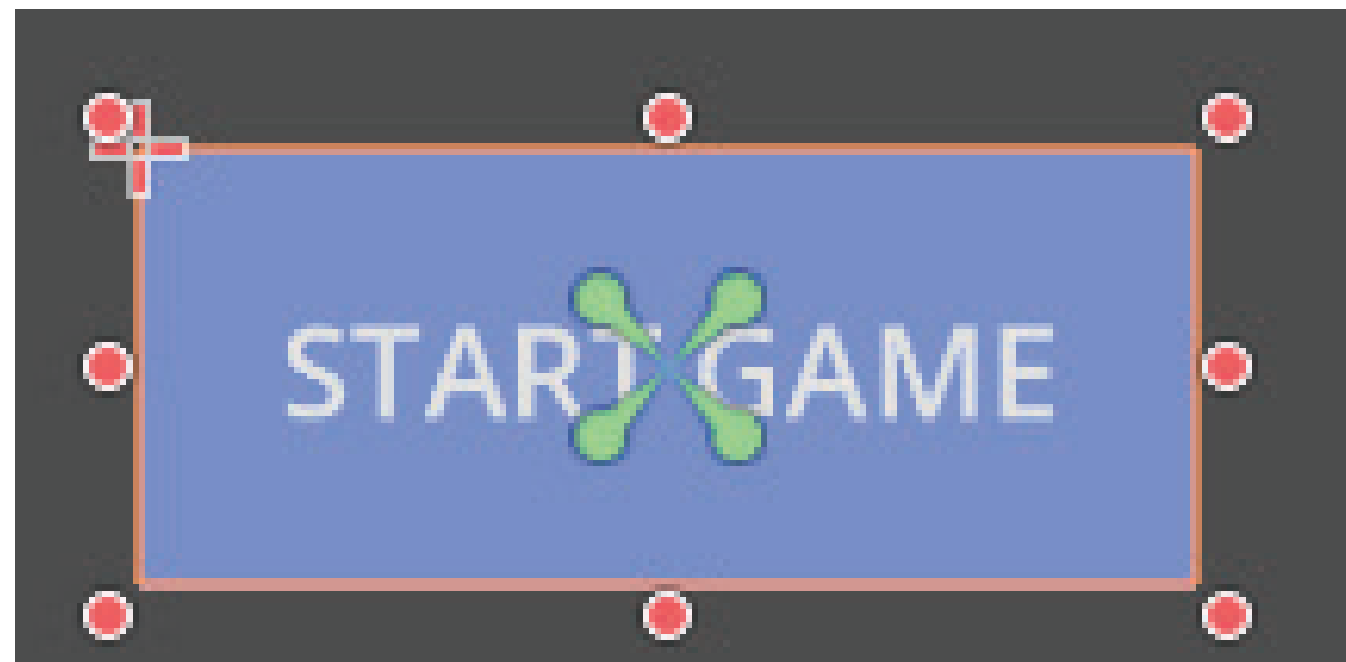
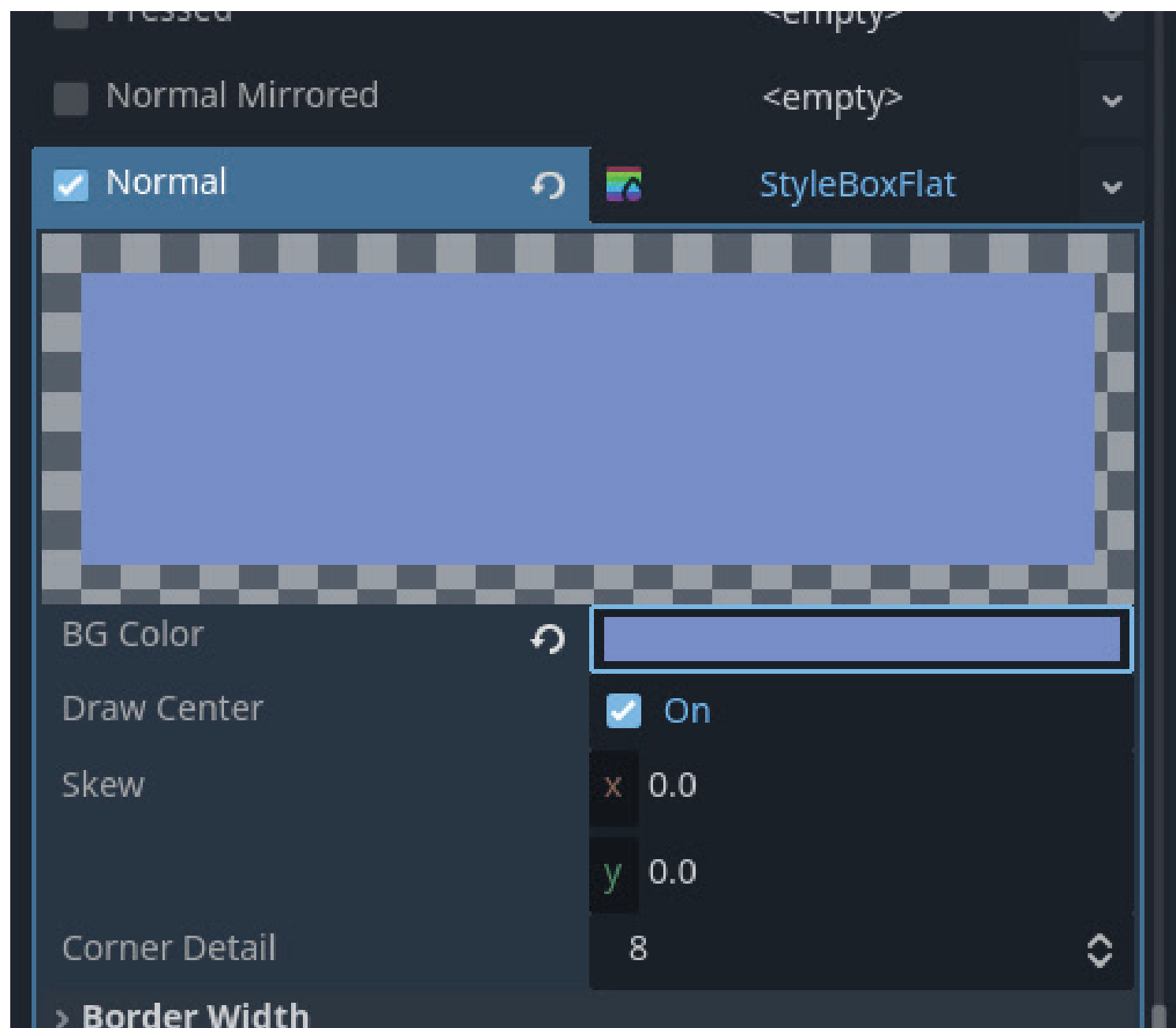




Make the font bigger >
Control > Theme Overrides
Font Sizes

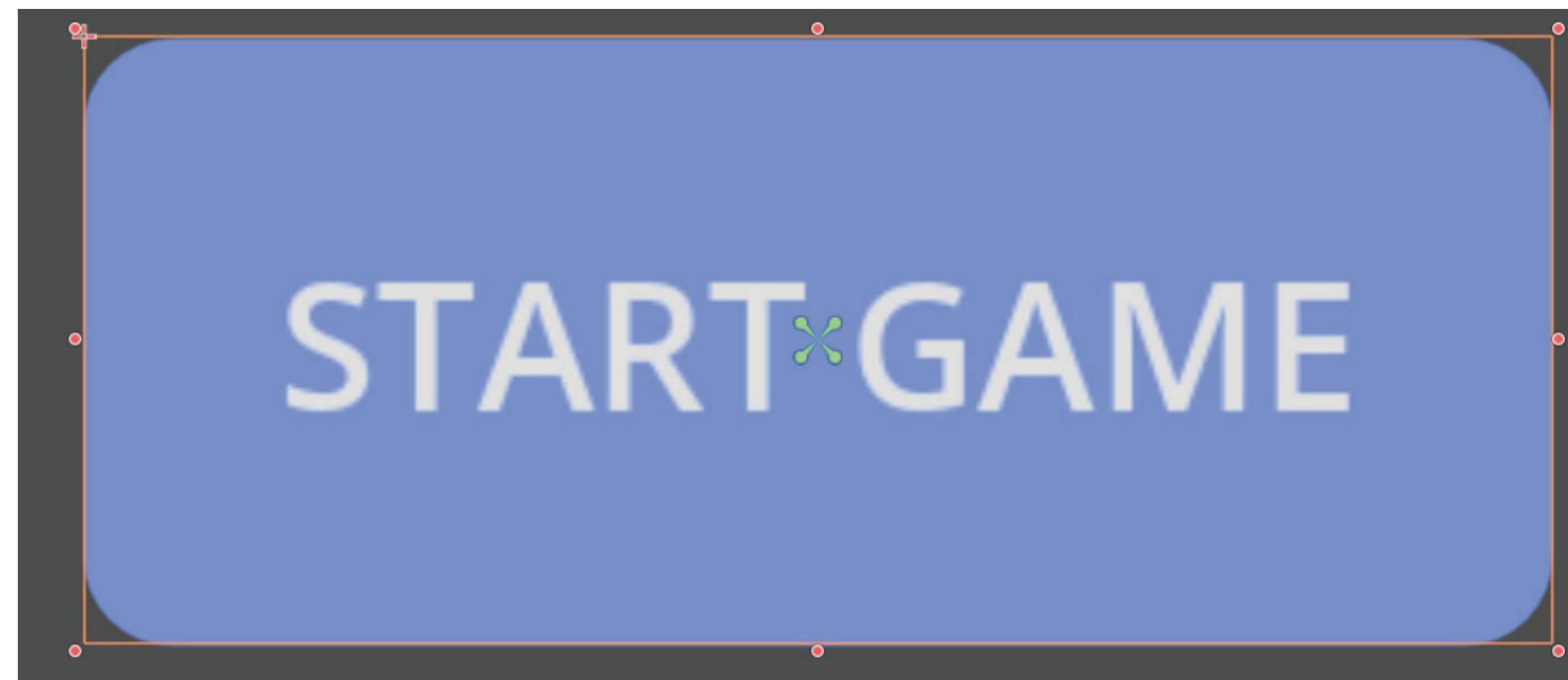
To add color in the box
Styles > Normal >
StyleBoxFlat

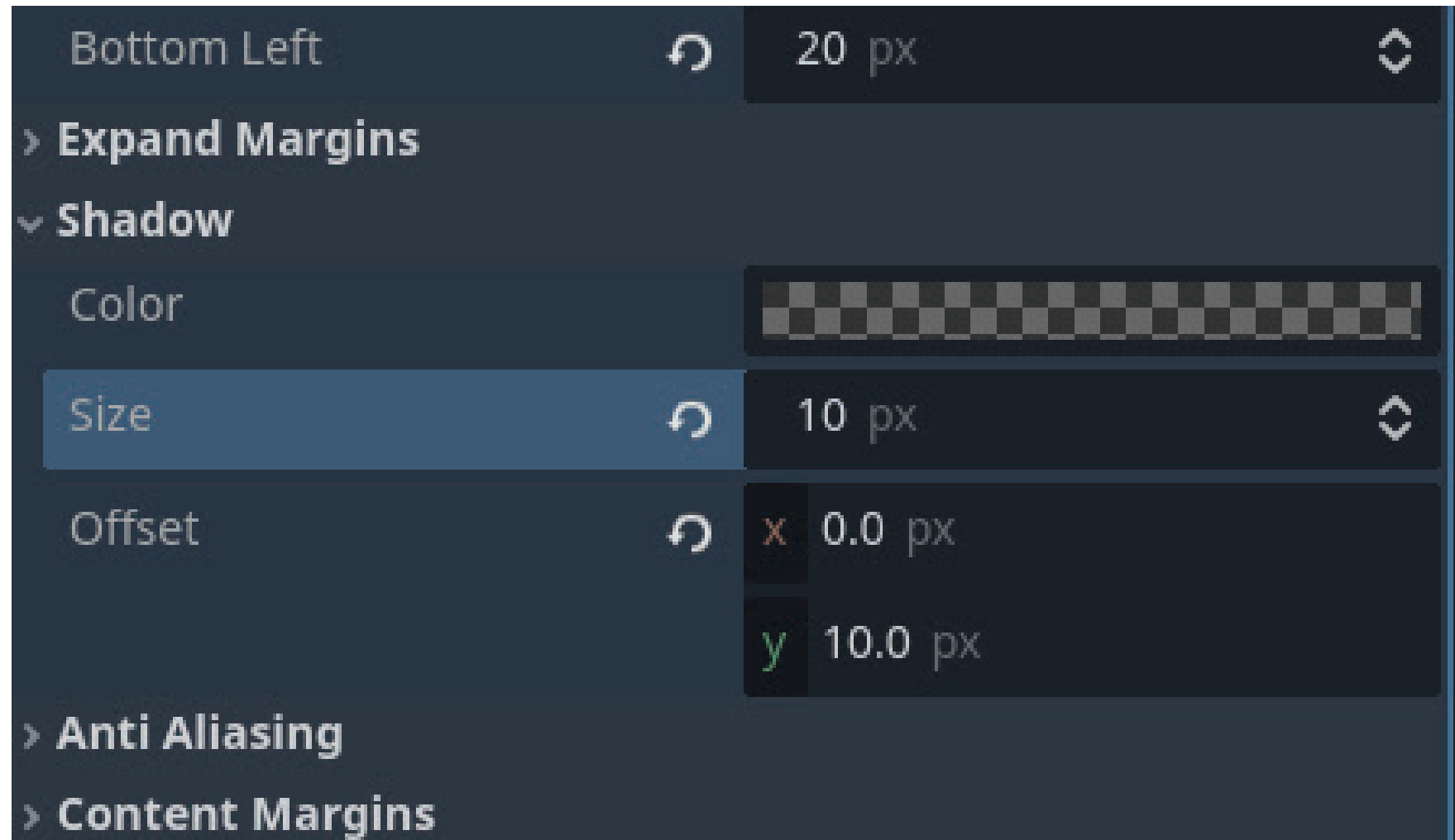
Click the StyleBoxFlat label
and change the BG Color



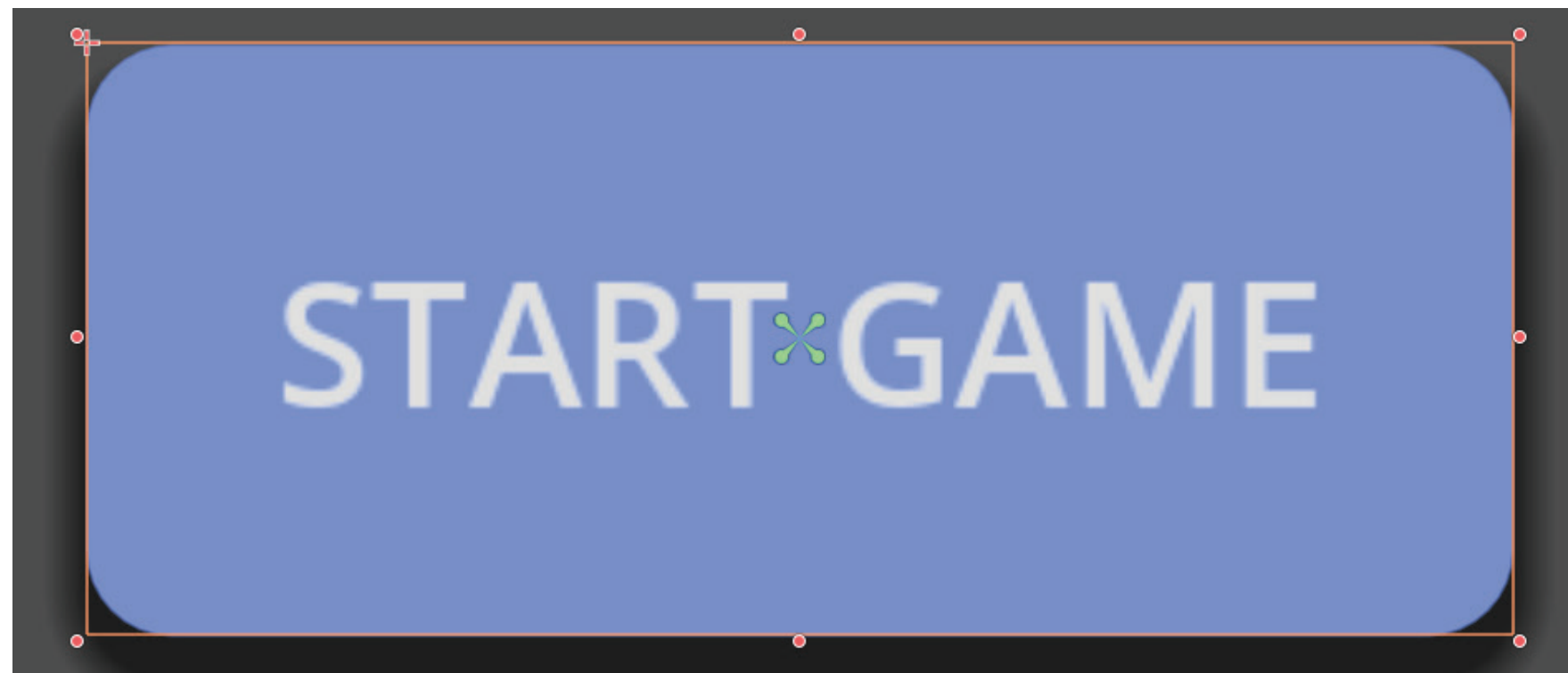


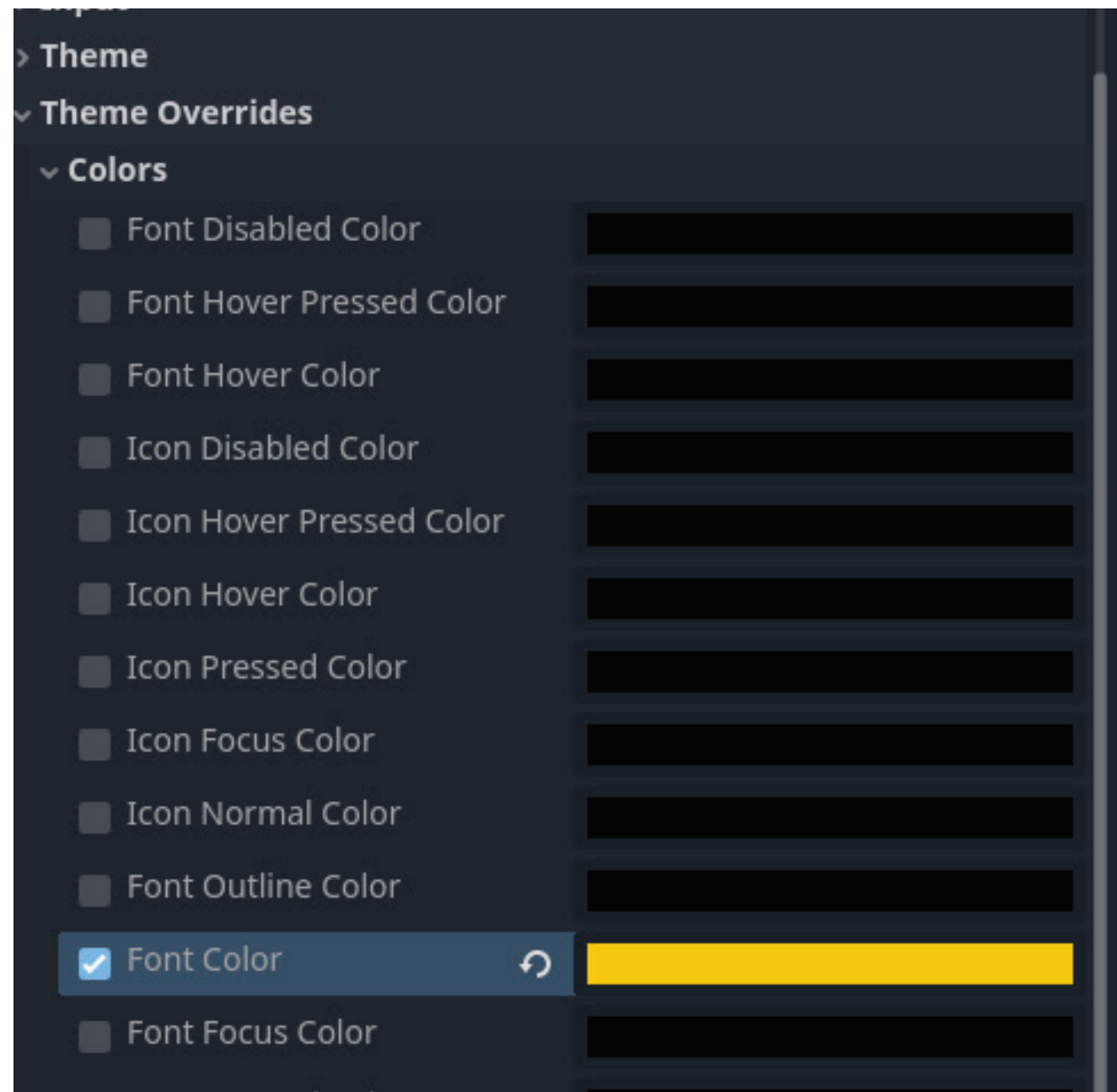
Make the box have rounded edges, by opening Corner Radius and increasing the value.





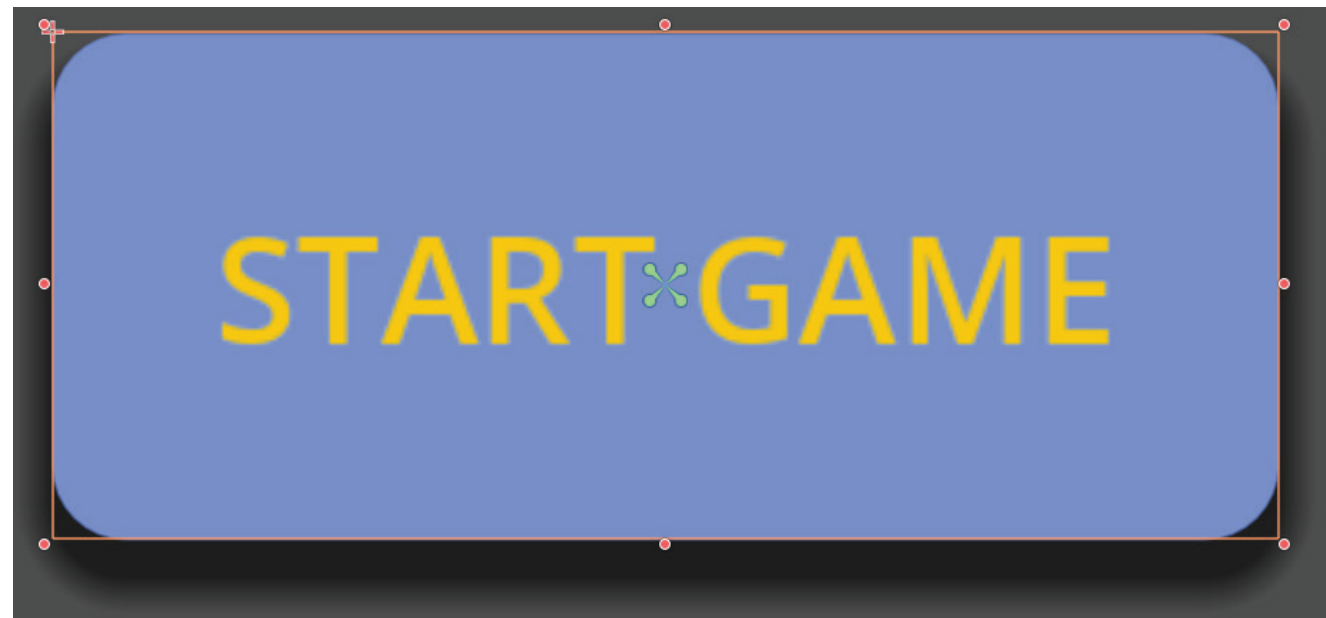
Add a Shadow by going to Shadow and increasing the Size.

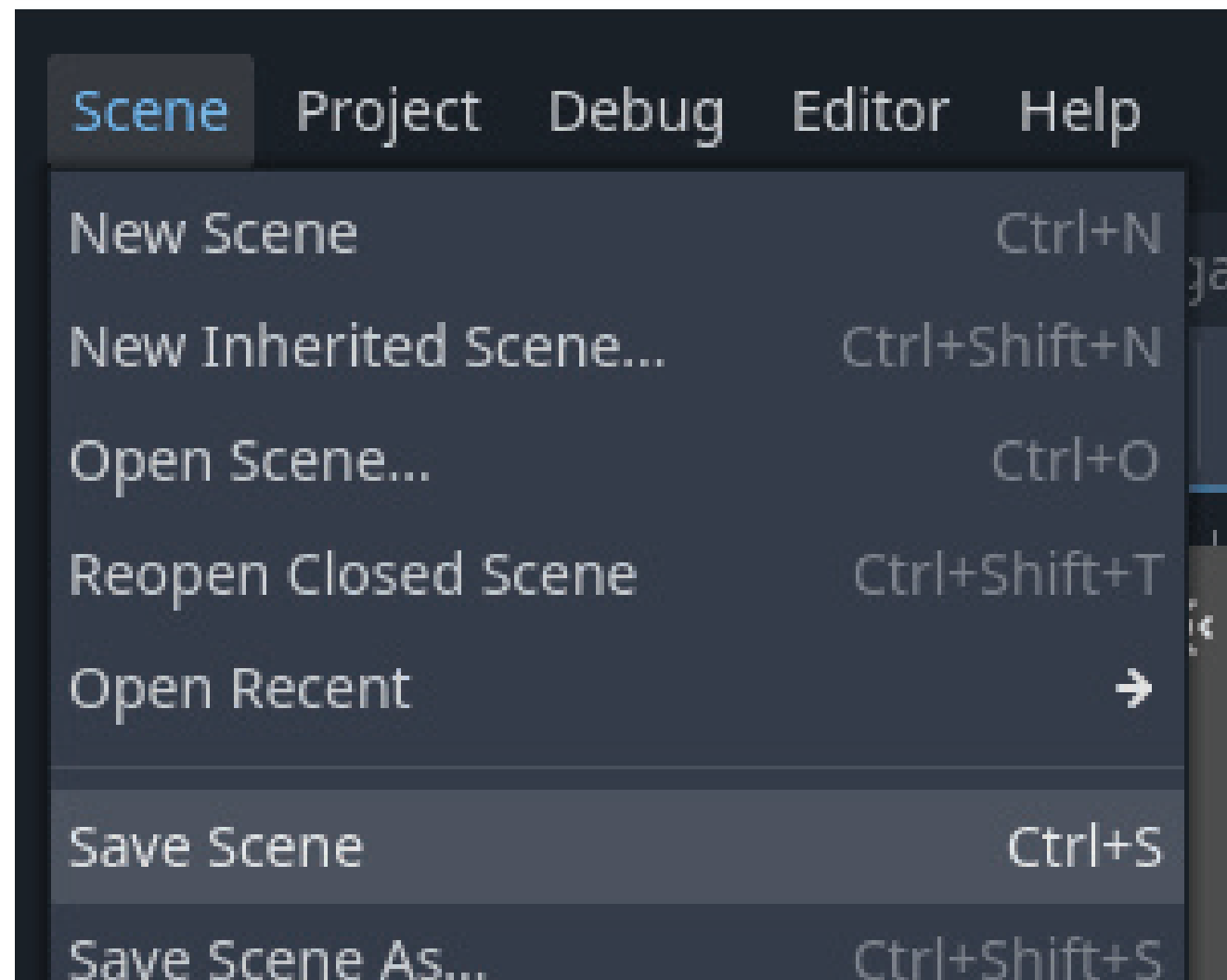




To change the Text Color
Go to
Theme Overrides> Colors

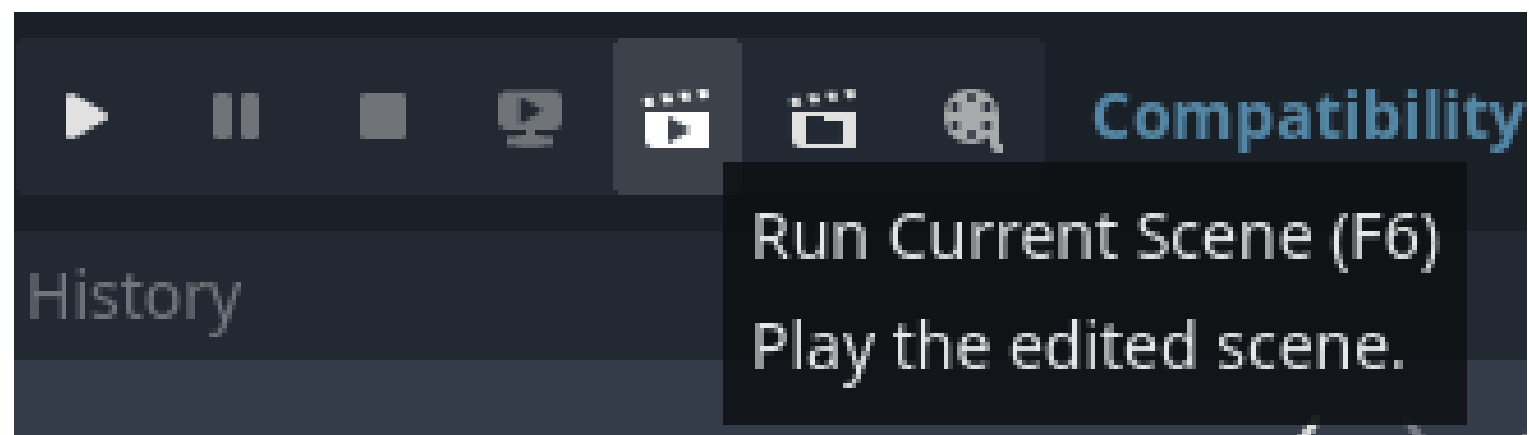
Font Color





Scene > Save Scene

To test the work press
Run Current Scene



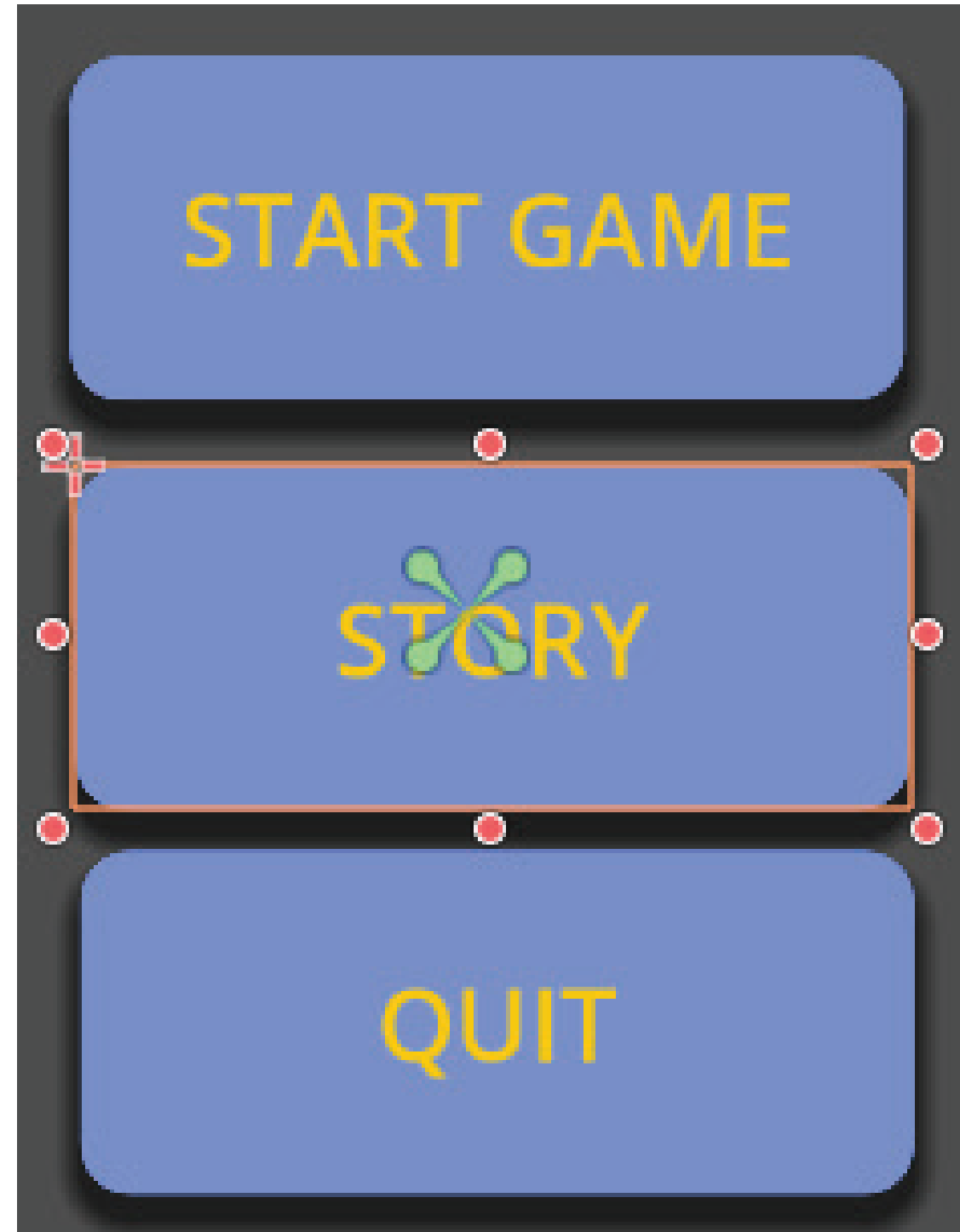
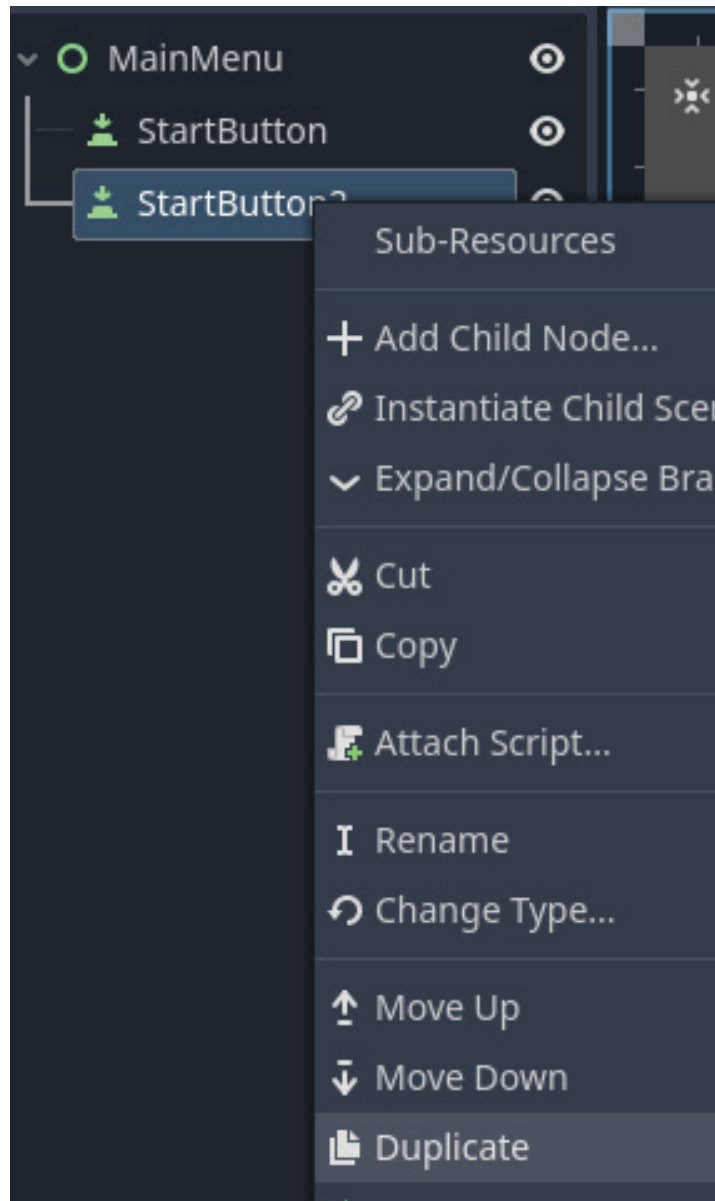


START GAME

Test the button by
Hovering
Clicking
and noticing the different
states.



START GAME

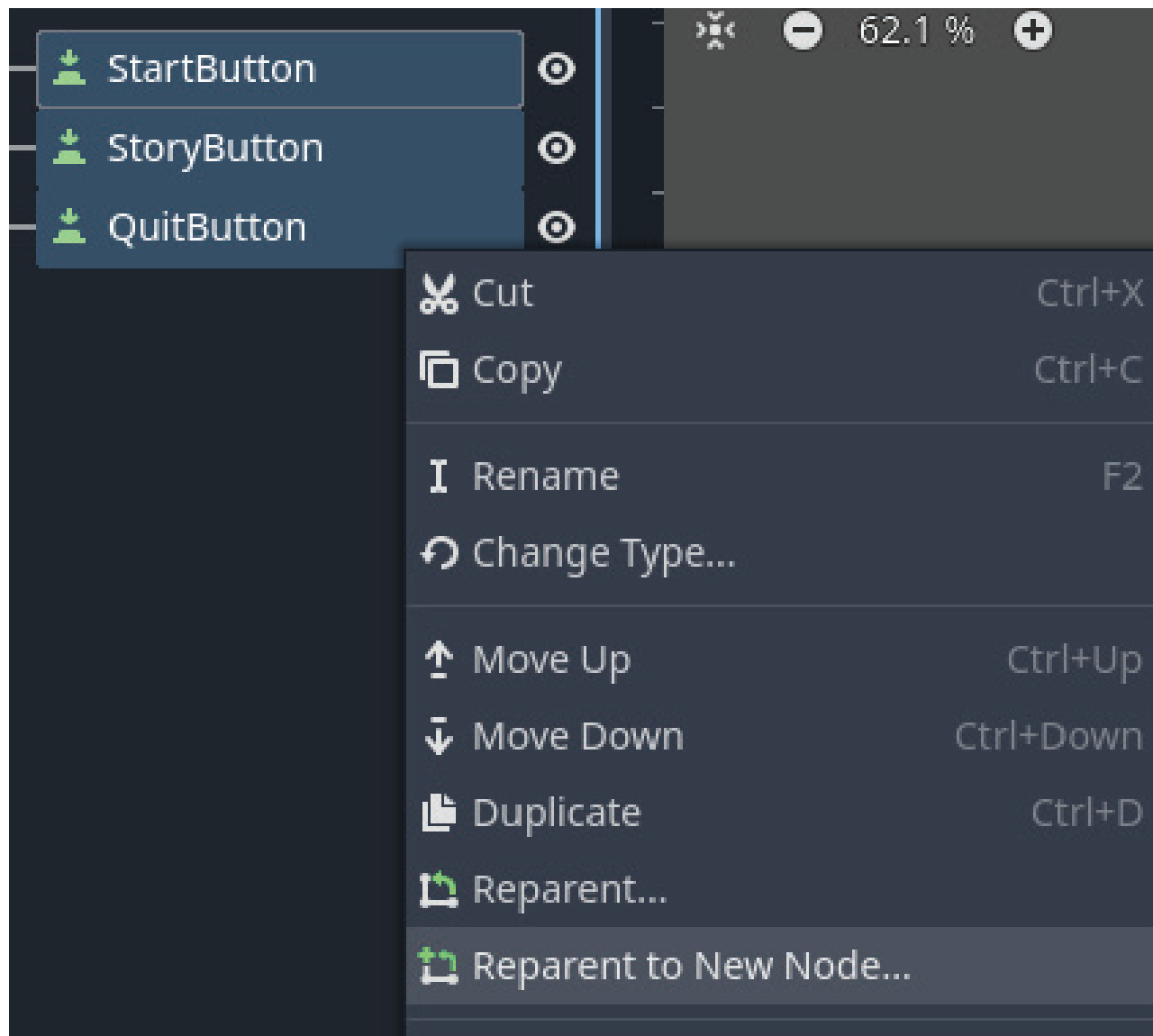


Duplicate the Start Button.

Name the next buttons, Story and Quit.

Drag them below each other.

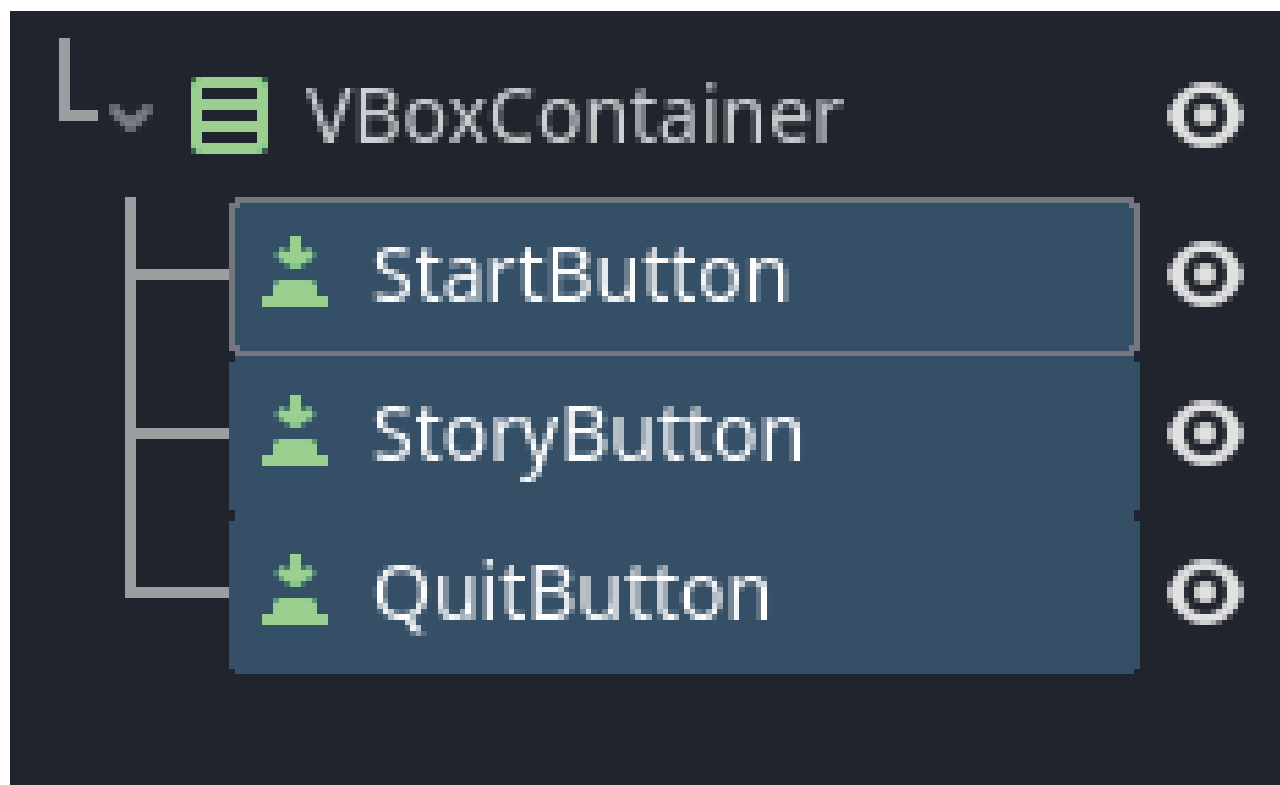
Update the Text field in each.



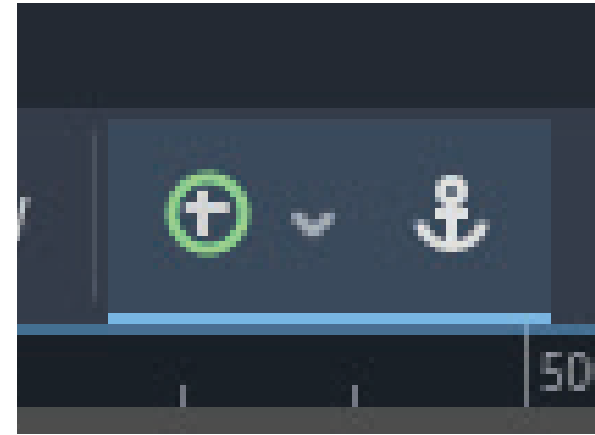
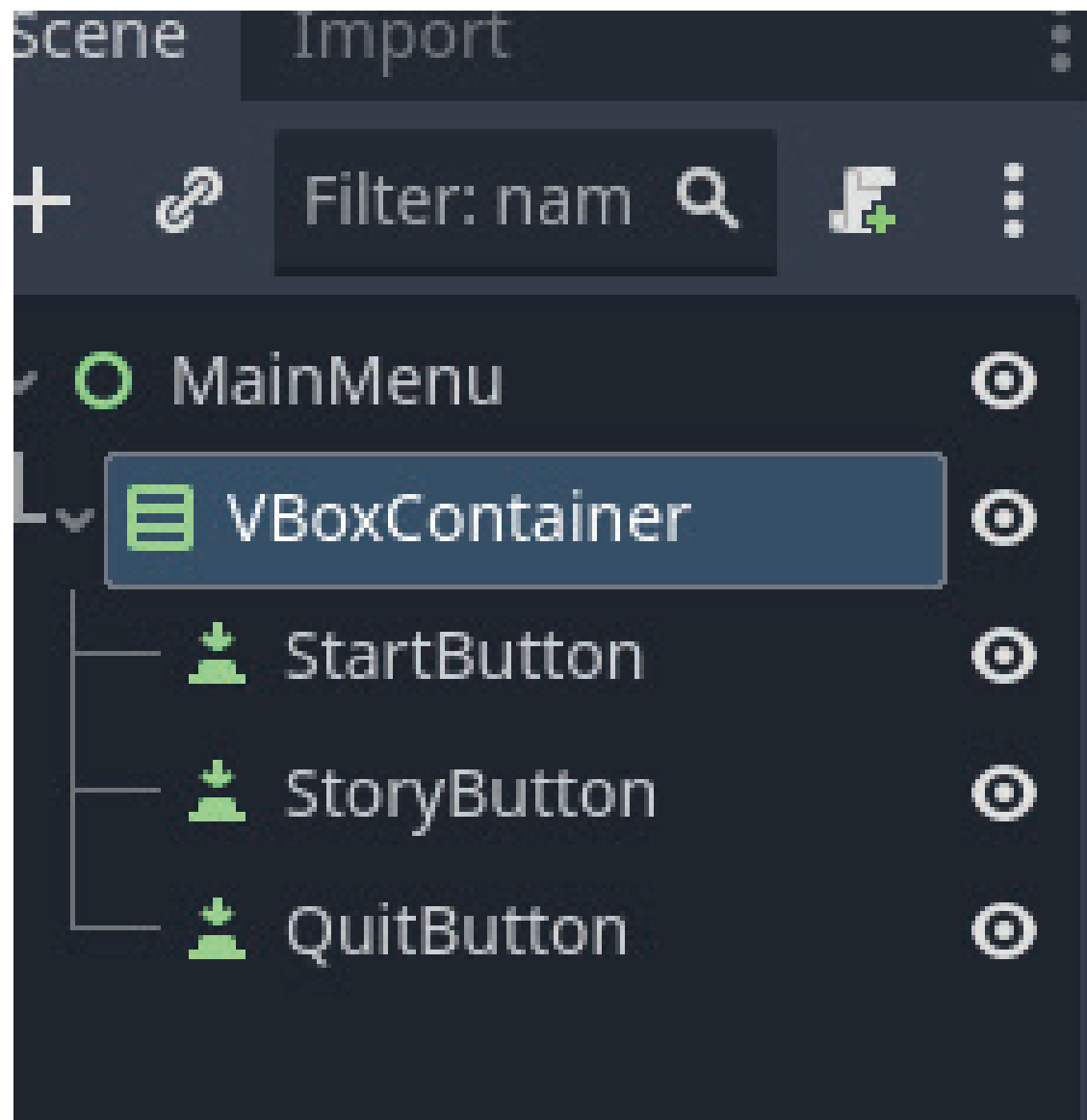
Select all the
Button Nodes.

Right Click and
choose

Reparent to New
Node



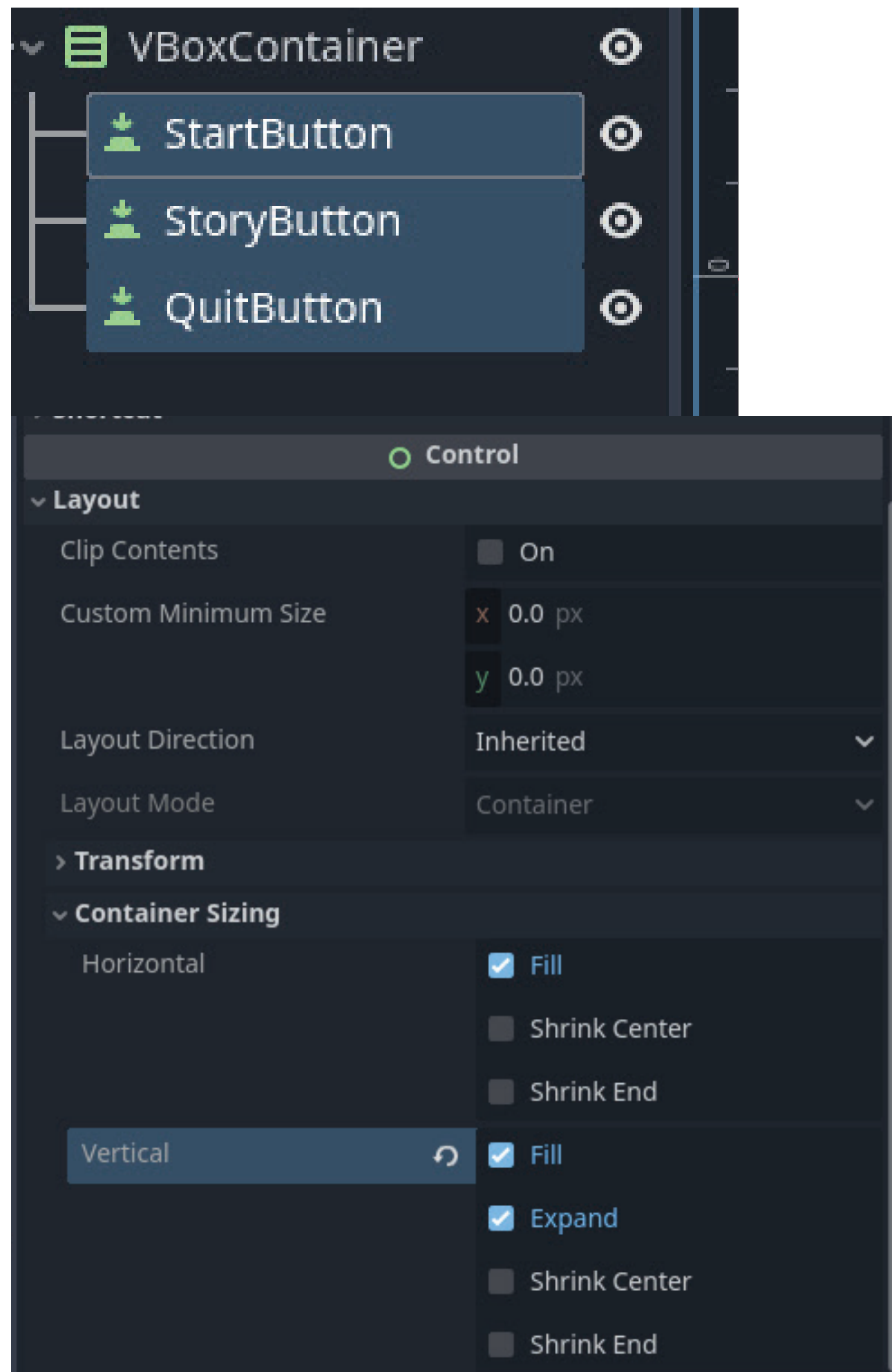
Choose
VBoxContainter



Use the Move Tool to drag the VBoxContainer to the center and make it bigger.

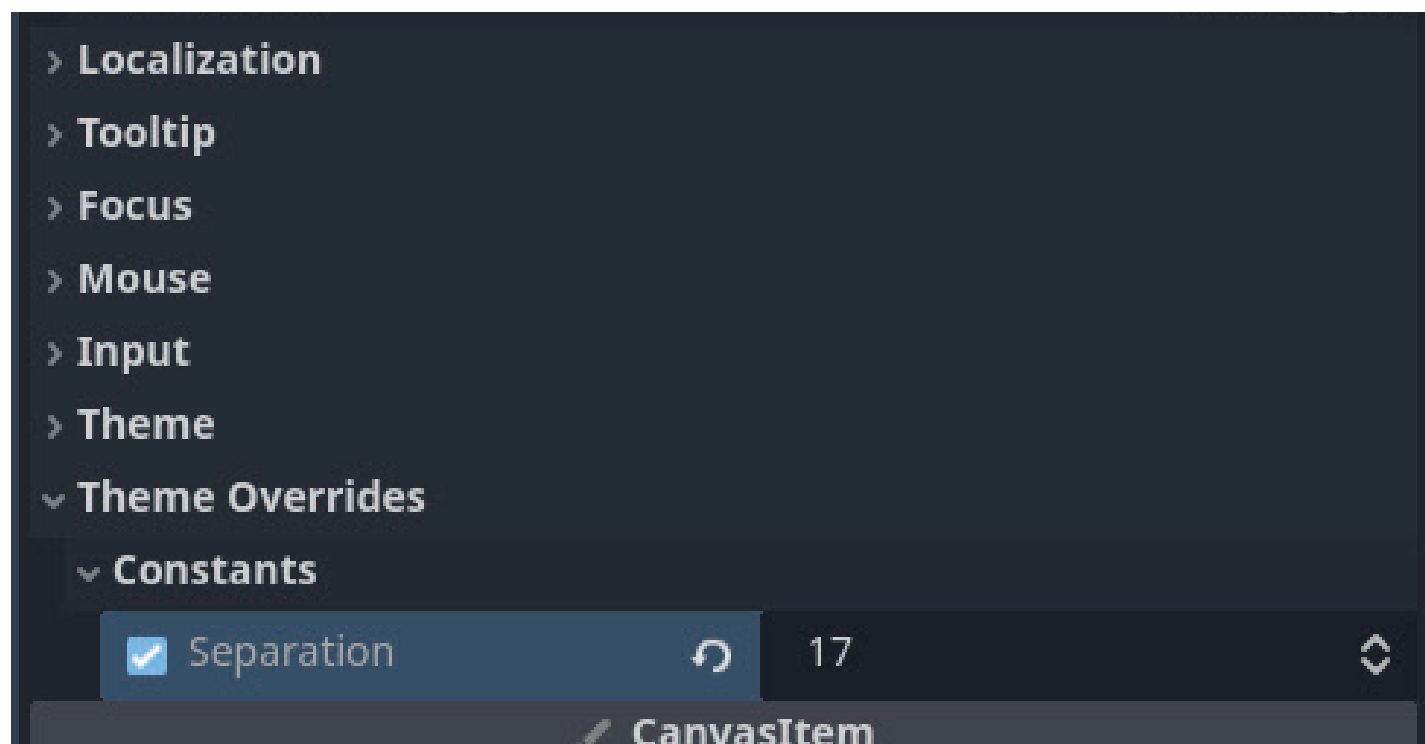
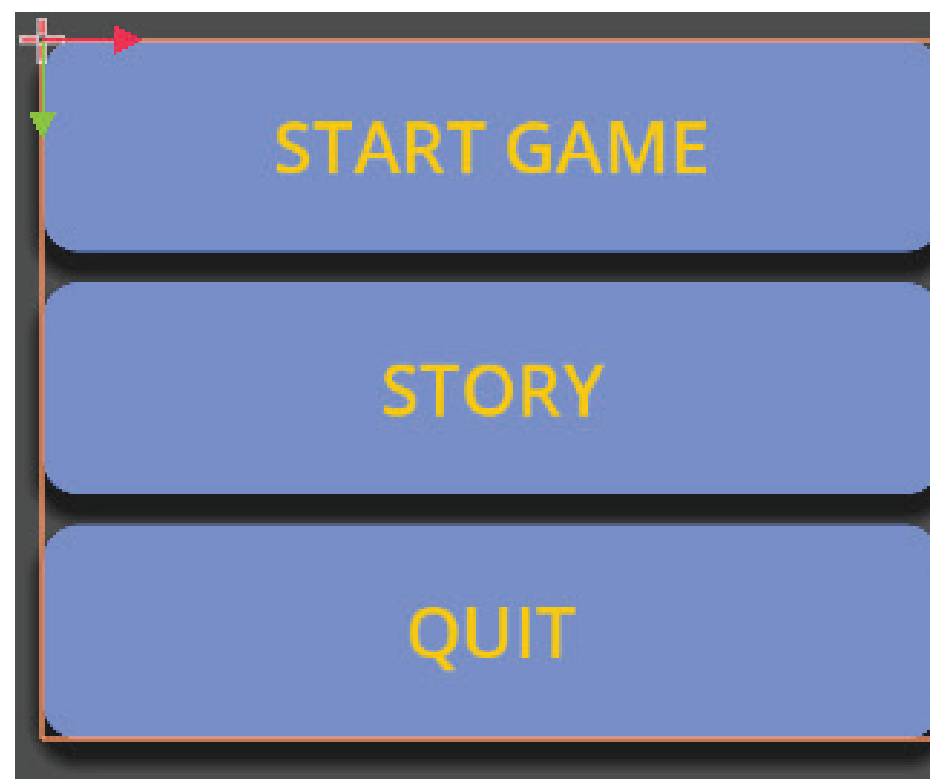
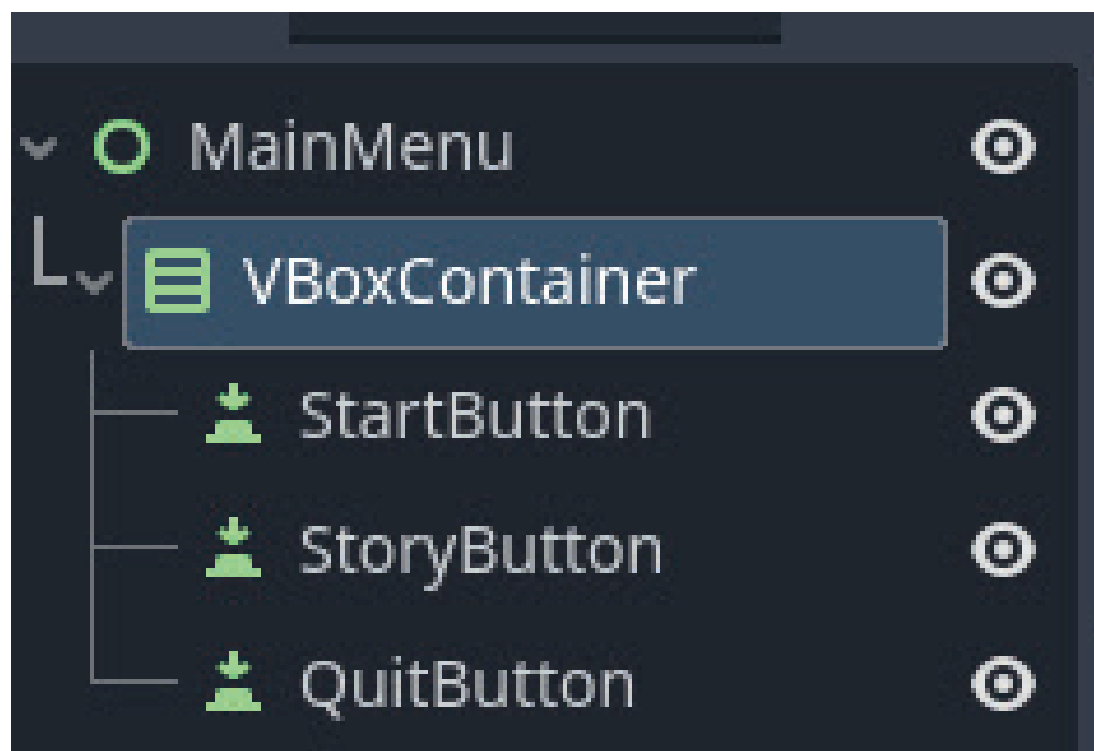
Use the Anchor Preset to center the VBoxContainer



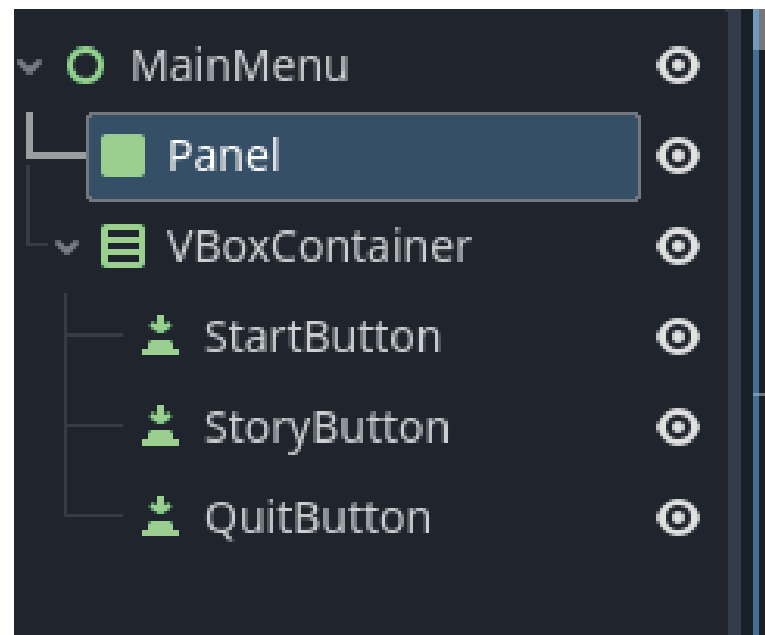


Select all the Buttons while holding shift.
Go to Inspector > Control> Layout>
Container Sizing
and press EXPAND

this will make the buttons fill the VBox

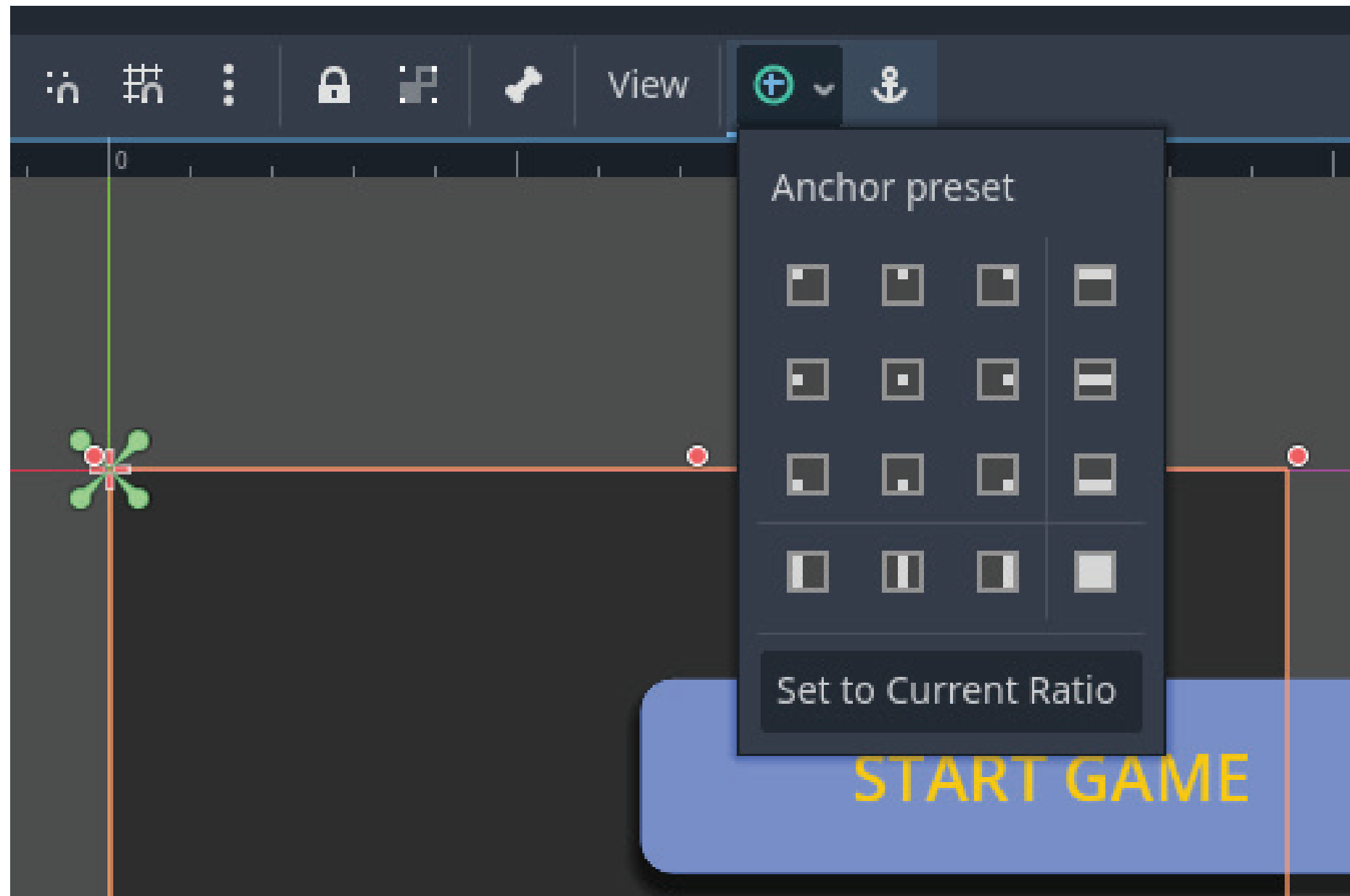


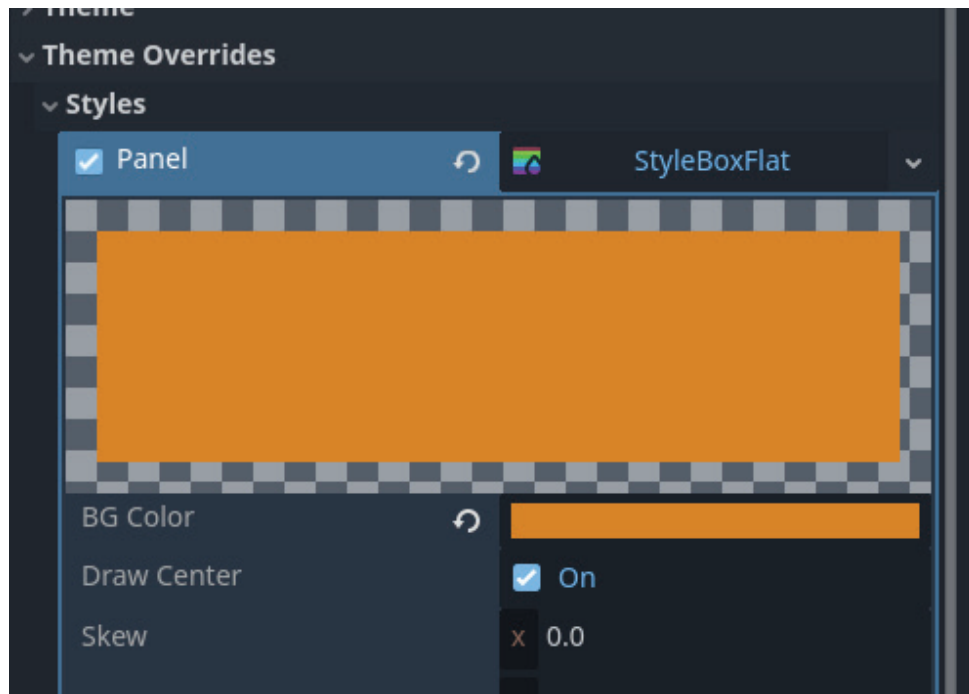
Go to VBoxContainer
Theme Overrides
Constants and increase the separation



Create a new child of the Main Menu > choose Panel

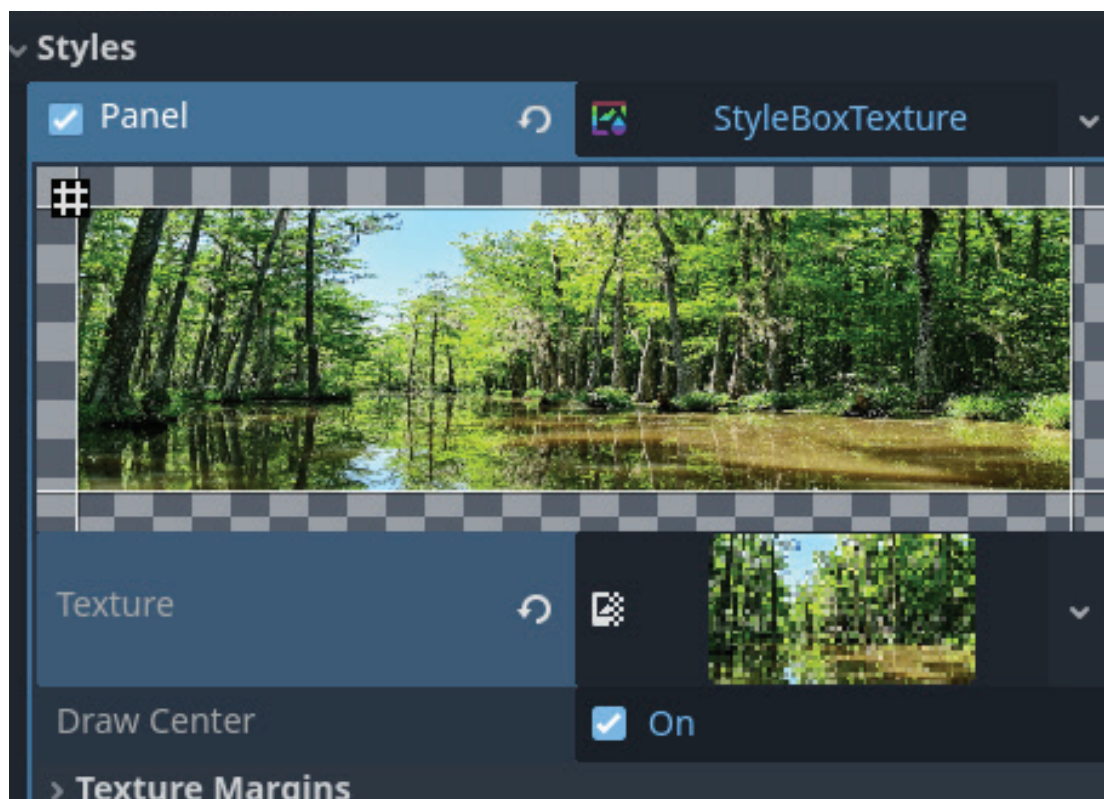
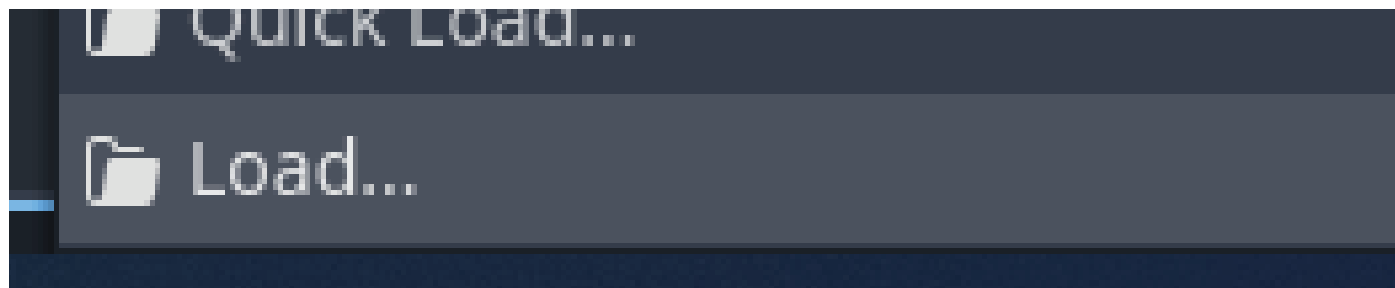
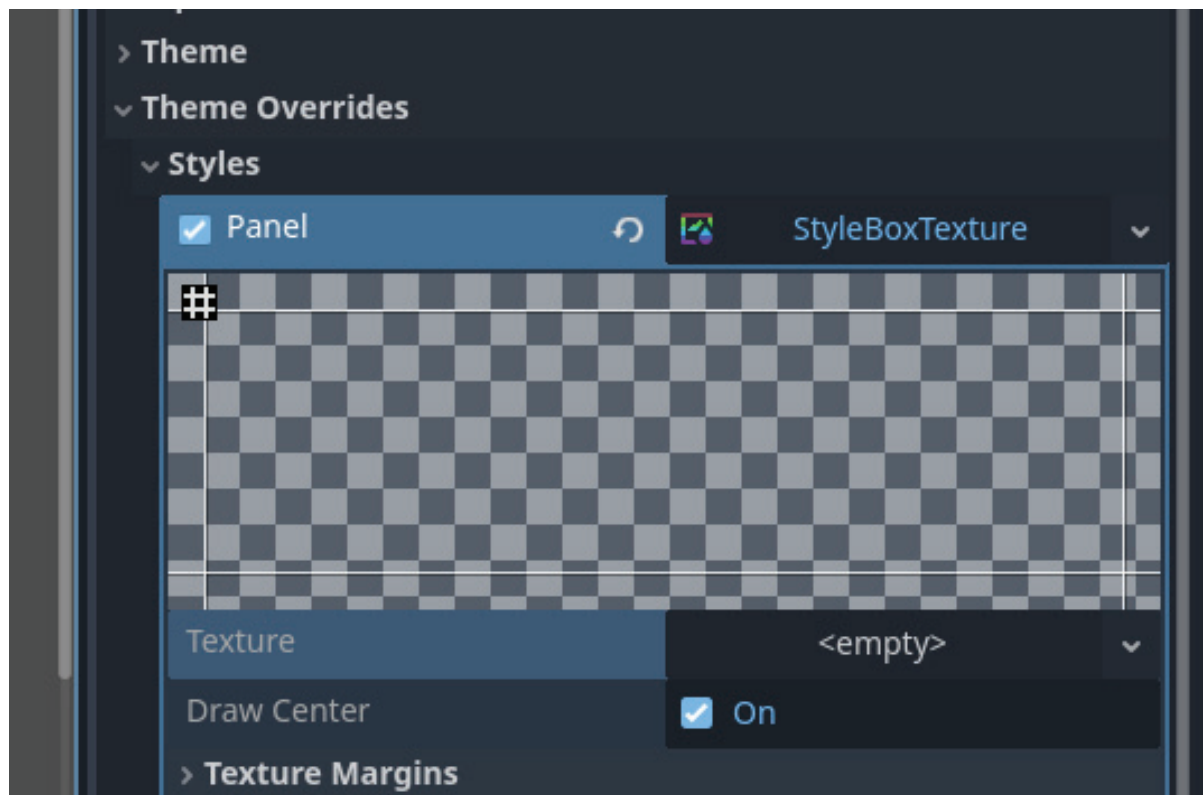
Make it larger and use the Anchor Preset to choose Full Rect, this will fill the full screen Rectangle.



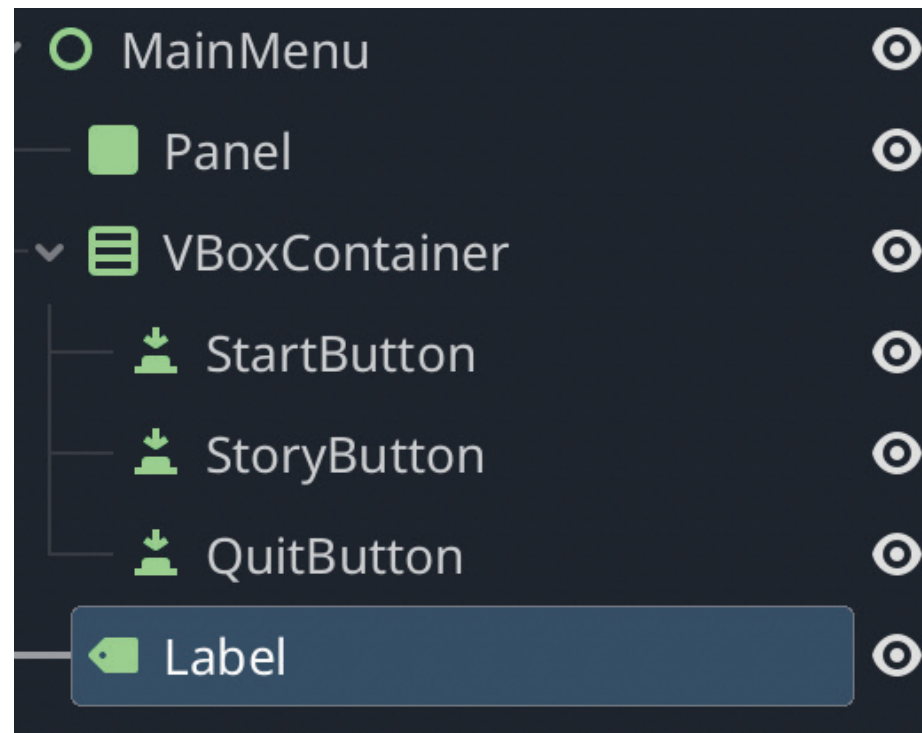


Go to Theme Overrides>
Styles> Panel and choose
empty >StyleBoxFlat and
choose a color.





If you would prefer a IMAGE
use StyleBoxTexture and load
an image in the Texture box



Create another child of the Main Menu

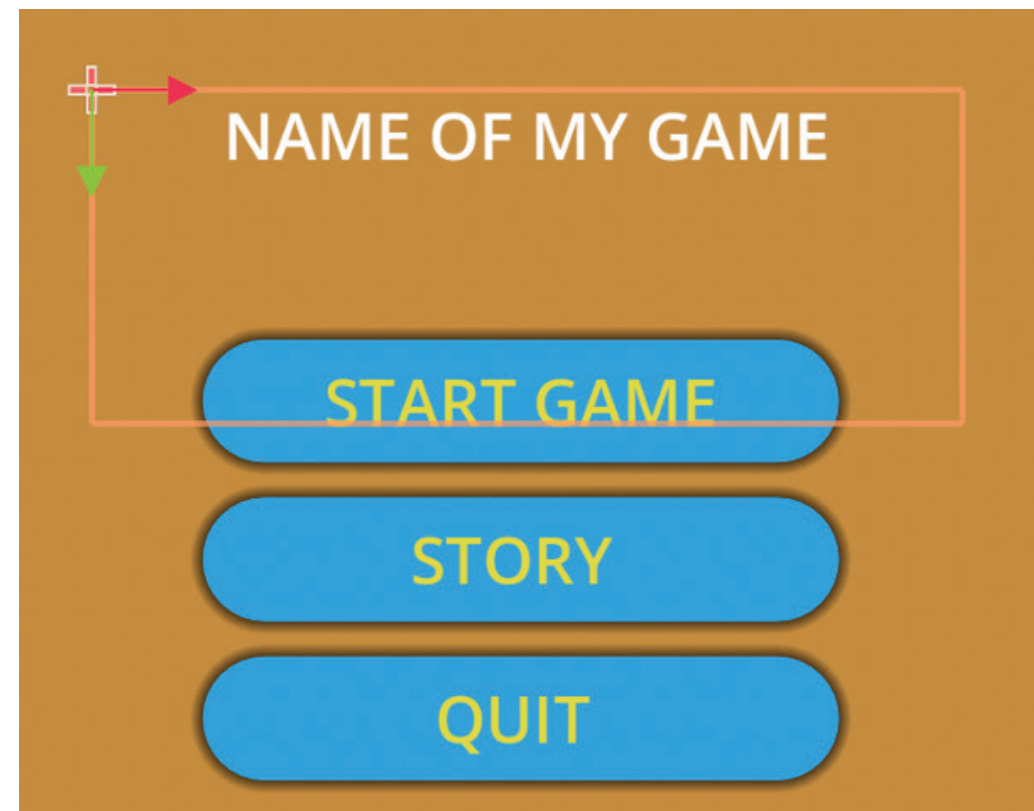
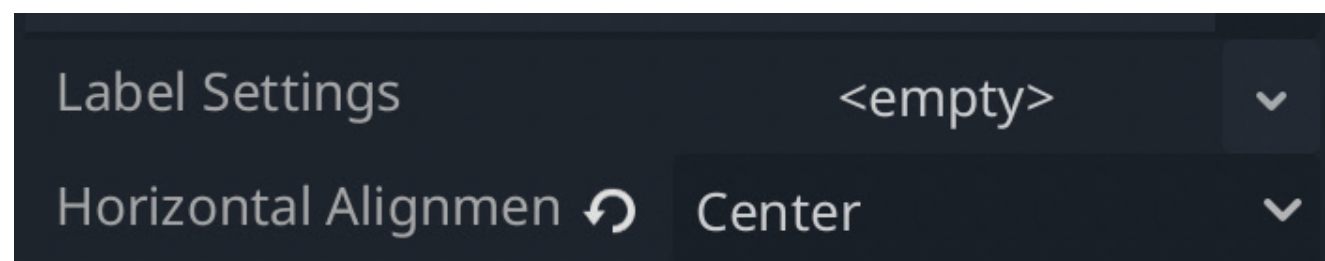
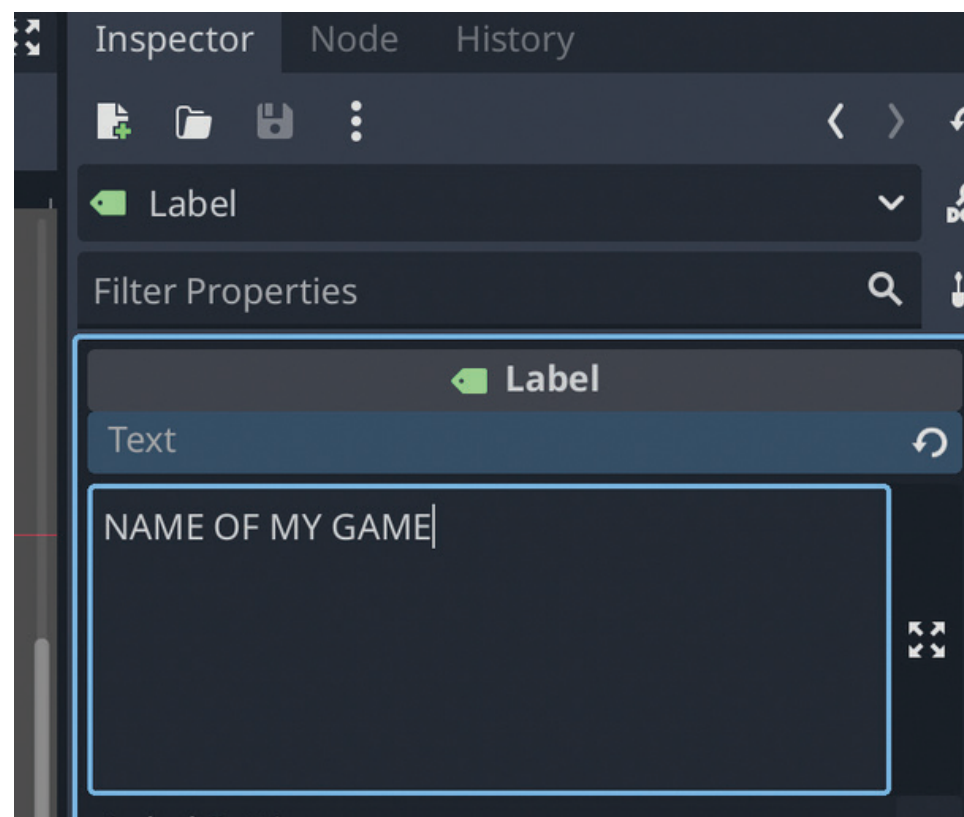
Choose Label

Write the name of your game

Go to Horizontal Alignment and choose

Center

Go to Anchor Preset and choose Center





Or

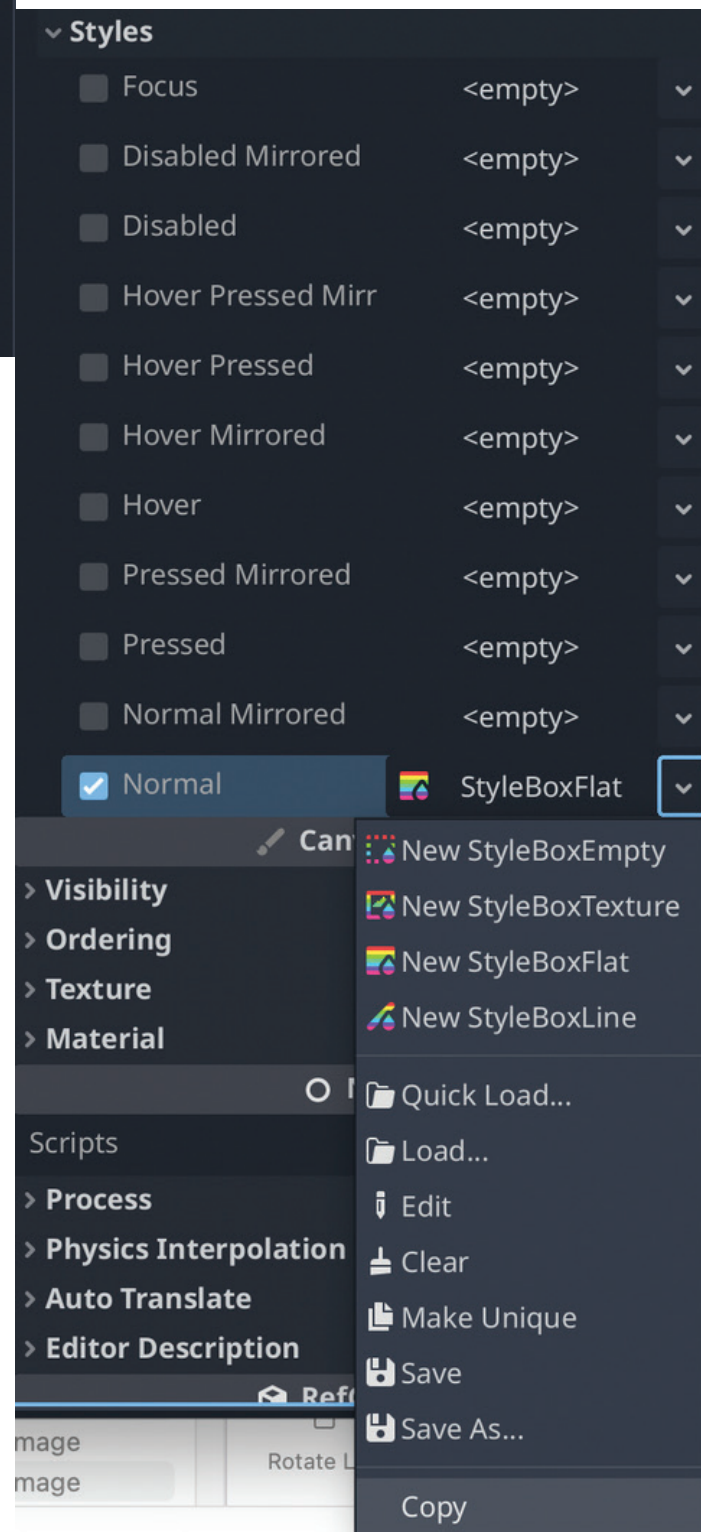
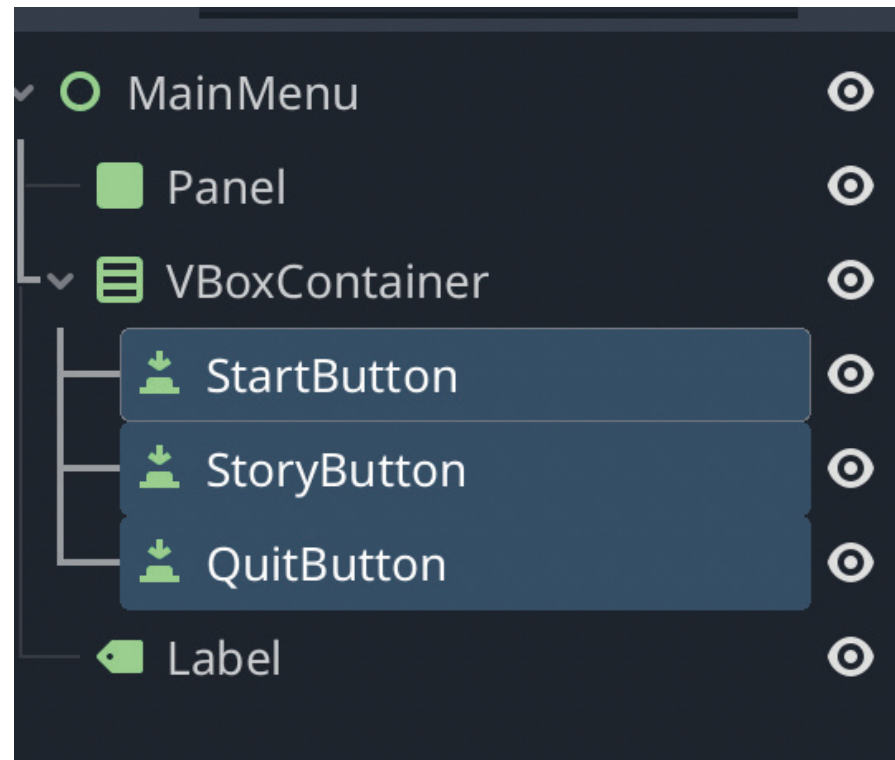
you can use your GAME LOGO as a title.

Go to Styles > StyleBoxTexture

Load your image

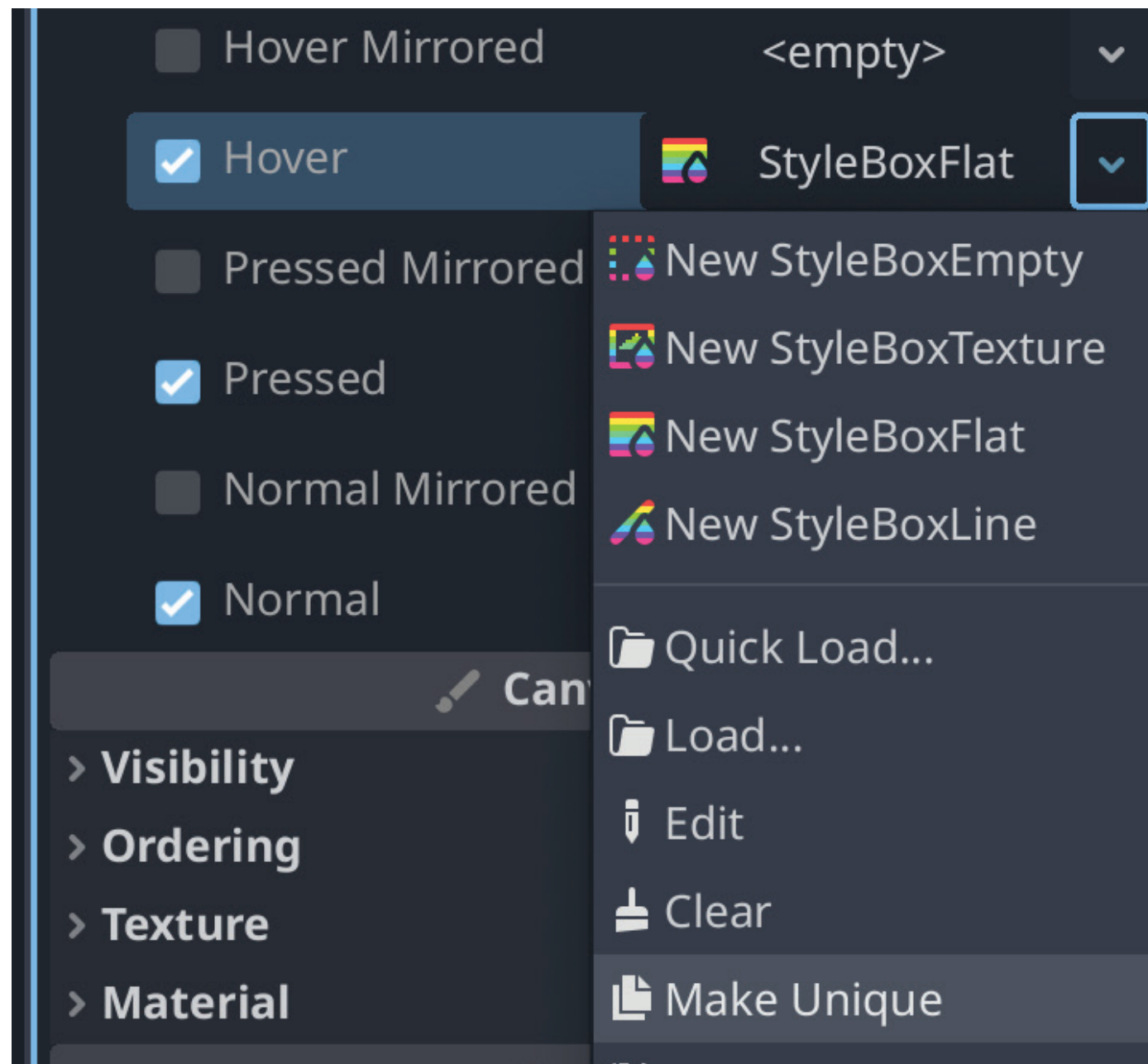
Choose Axis Stretch > Horizontal > Stretch
and Vertical > Stretch to make the image fit





To modify the styles of the buttons states, select them all.

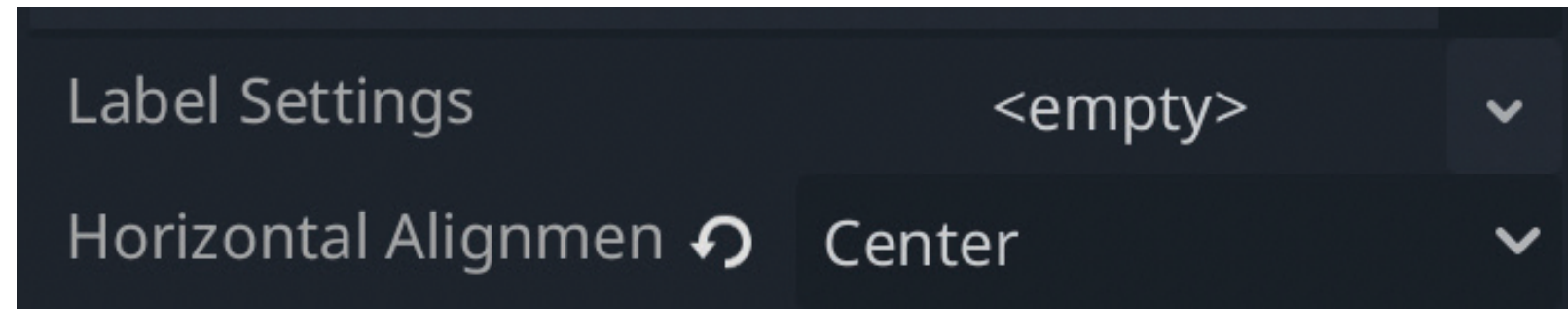
To quickly change the states for HOVER and PRESS we can copy the current style from normal and PASTE it to the others.



While all the buttons are still selected

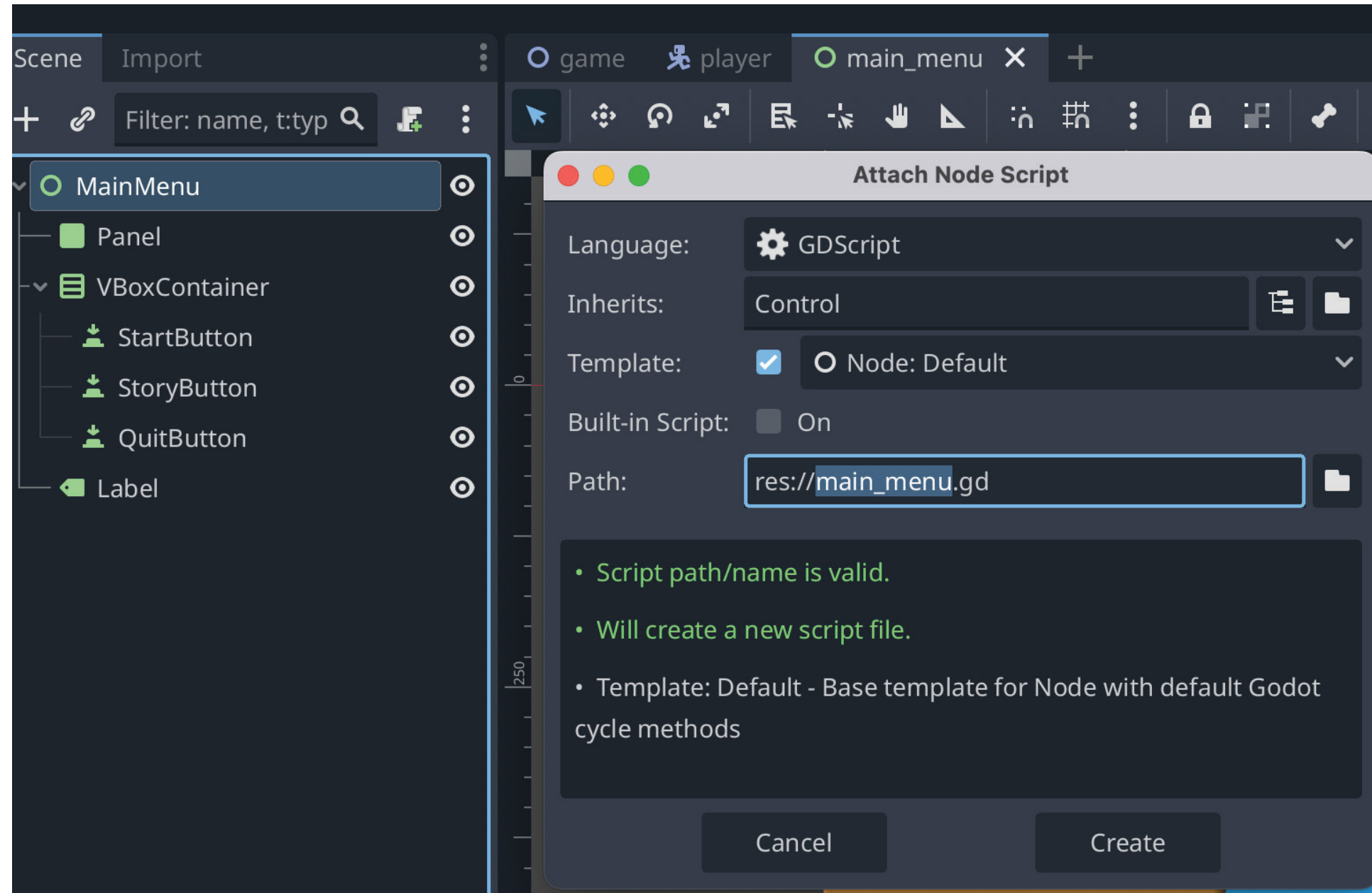
If you want to change the HOVER or PRESSED state, you can choose MAKE UNIQUE and make changes as needed.

It is usually a good idea to make a change for HOVER to let users know the buttons work.

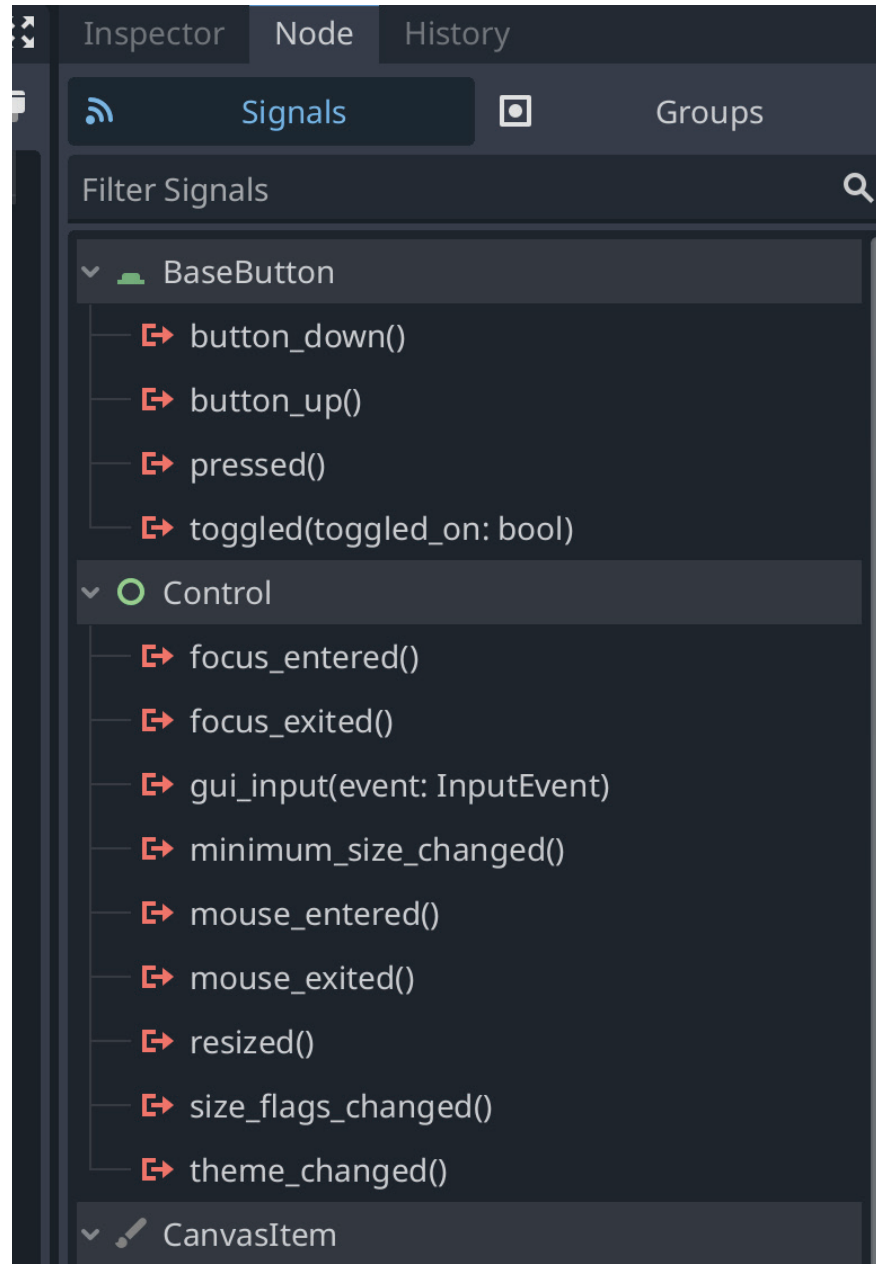
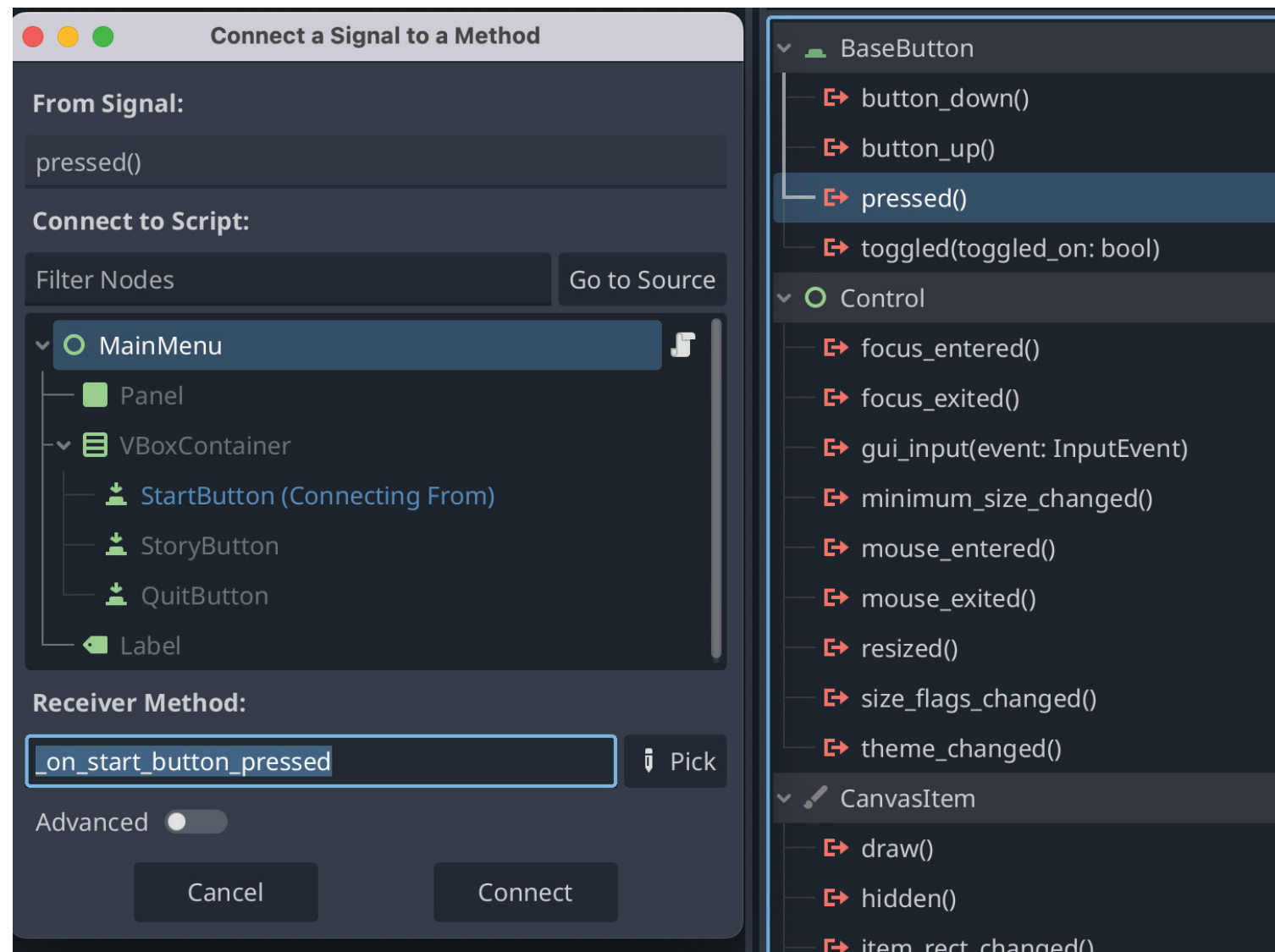
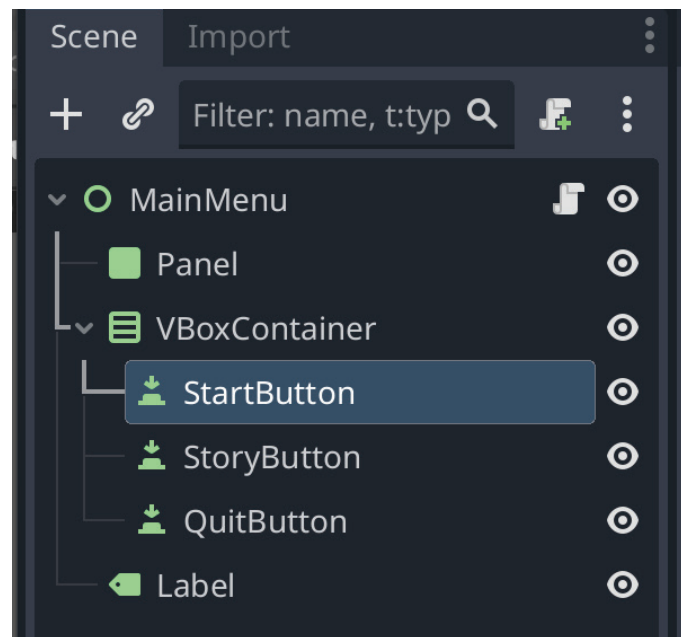


While all the buttons are still selected

To remove the small white border
go to Styles > Focus >
StyleBoxEmpty



Go to Main Menu and
attach script



Click on the Start Button
Click on the Node window to create a signal

Double click
BaseButton pressed()

```
13
→ 14  ▾ func _on_start_button_pressed() -> void:
15      »    pass # Replace with function body.
16
17
→ 18  ▾ func _on_story_button_pressed() -> void:
19      »    pass # Replace with function body.
20
21
→ 22  ▾ func _on_quit_button_pressed() -> void:
23      »    pass # Replace with function body.
24
```

The code should look like this.

*Keep the top level code above in case we need it later.

```
13
→ 14  ▼ func _on_start_button_pressed() -> void:
15      »    print("start button was pressed")
16
17
→ 18  ▼ func _on_story_button_pressed() -> void:
19      »    print("story button was pressed")
20
21
→ 22  ▼ func _on_quit_button_pressed() -> void:
23      »    print("quit button was pressed")
24
```

```
start button was pressed
story button was pressed
quit button was pressed
quit button was pressed
```

Add print code to each function to test it!

```
13
→ 14  ▼ func _on_start_button_pressed() -> void:
15      »    print("start button was pressed")
16
17
→ 18  ▼ func _on_story_button_pressed() -> void:
19      »    print("story button was pressed")
20
21
→ 22  ▼ func _on_quit_button_pressed() -> void:
23      »    print("quit button was pressed")
24
```

```
start button was pressed
story button was pressed
quit button was pressed
quit button was pressed
```

Add print code to each function to test it!

```
21
→ 22  ▾ func _on_quit_button_pressed() -> void:
23      >| print("quit button was pressed")
24      >| get_tree().quit()
```

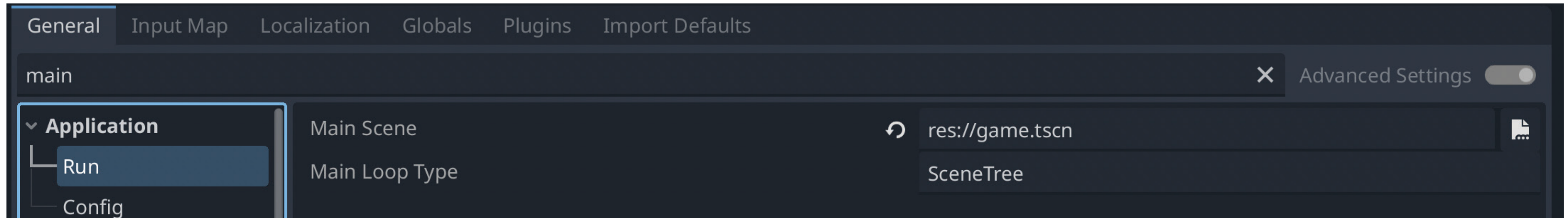
Add the following to the quit button, this should close your game on button click

`get_tree().quit()`

```
13  
→ 14  ▾ func _on_start_button_pressed() -> void:  
15      > print("start button was pressed")  
16      > get_tree().change_scene_to_file("res://game.tscn")  
17
```

Add the following to the start button, make sure to load your main game scene.

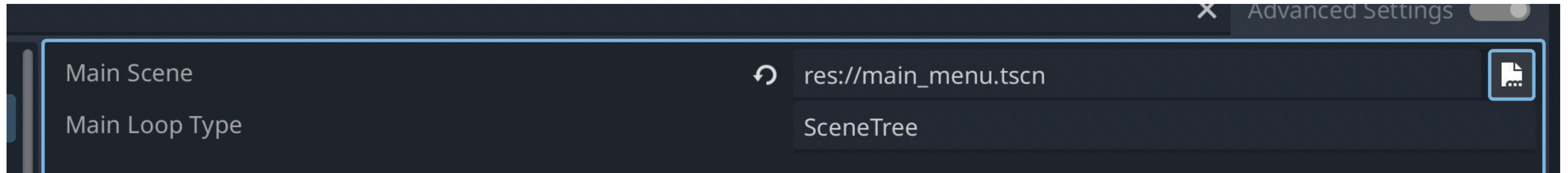
```
get_tree().change_scene_to_file("res://game.tscn")
```



To make the Menu the Main (first) scene when you start your game>

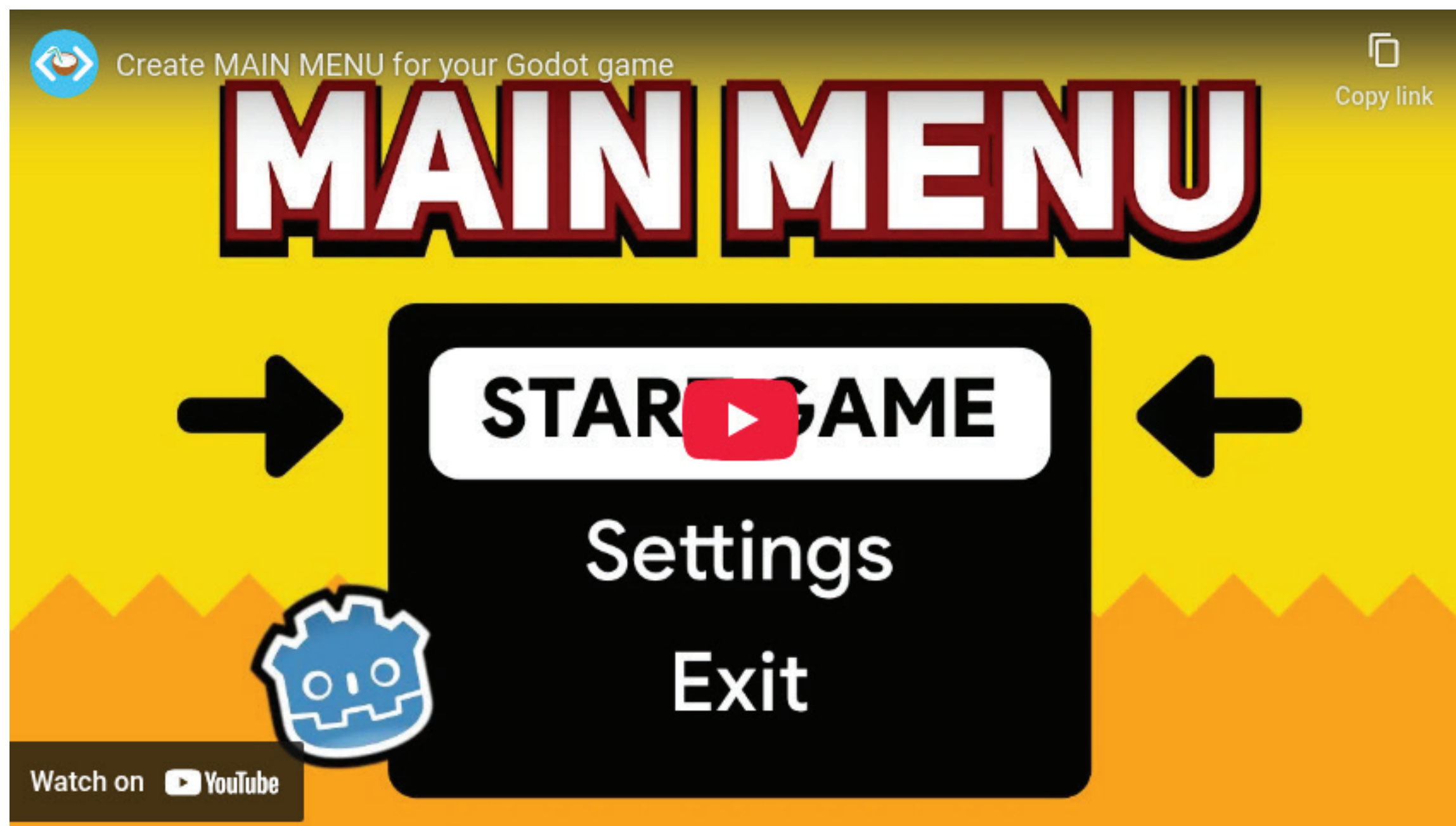
Project > Project Settings > search main

Run > Main Scene



Press the file load button and change the Main Scene to your Main Menu

Test it!



[link](#)