GettingStartedControls Part 4

In the first 3 parts of this sample we have created 3 individual scenes. In the Player it was necessary to enable and disable the displayed scenes manually. In this part we will add menu tabs to the bottom of the screen so the user can switch between scenes as he likes.

As basis for step 4 of our GettingStartedControls tutorial, please use your own created solution from the previous "GettingStartedControls" tutorial.

Our menu tabs scene can be seen on the bottom of the climate scene in below image.



Our goal is to create a tab bar to enable the user to switch between different scenes. As this bar shouldn't be displayed permanently, it will be put into a drawer control. This way it can be made visible by hovering an image and it can be hidden again by clicking on a close button. The buttons will be realized using customized radio buttons.

This tab bar will be placed on the bottom of every screen. To avoid having to create this bar inside every scene where it should be displayed, we will define a separate scene, which we can reuse on all screens.

Importing Images

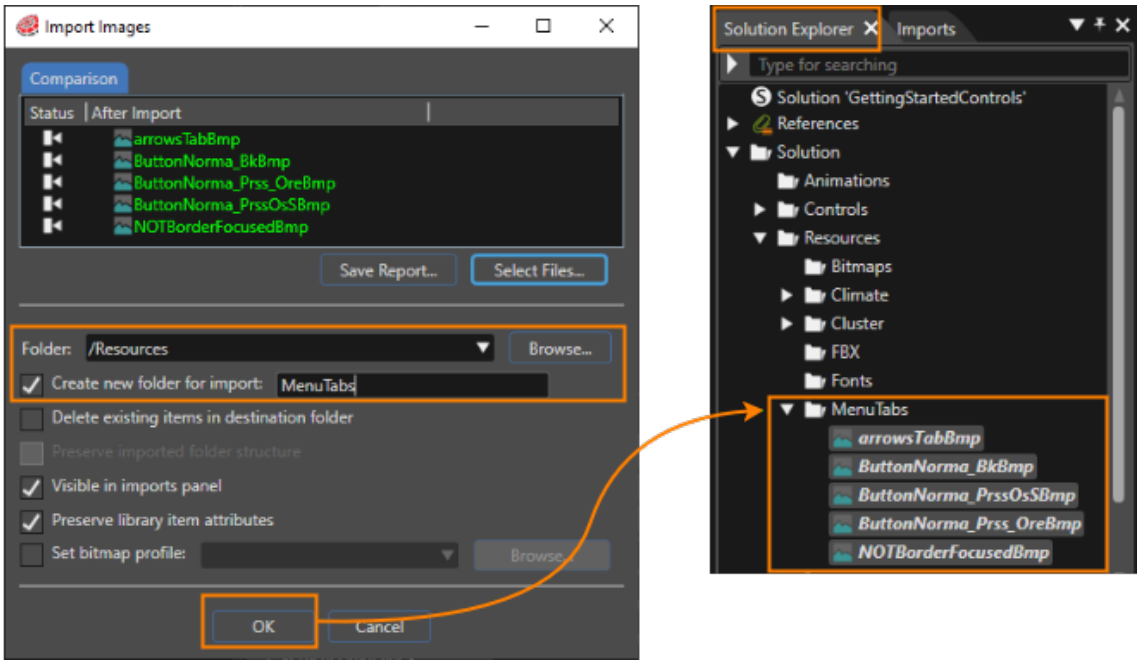
We will use the built-in Scene Composer controls but they will be customized with new images. Therefore these images have to be imported into the Scene Composer solution.

1. In the menu bar select **Import > Import Resources...**
   - The Windows **"Import Resources"** dialog will be opened.

2. In the **"Import Resources"** dialog navigate to the path given in the table below. Select all 5 files and press **[Open]**
   - The Windows **"Import Resources"** dialog will be closed.
   - Scene Composer's **"Import Images"** dialog will be opened.

Import Resources	Path
	/cgi_studio_content/Resources/GettingStartedControls/MenuTabs

3. In the **"Import Images"** dialog select folder "/Resources", make sure to create a new folder "MenuTabs" for the imported images and click **[OK]**.
   - In the Solution Explorer you can see that there is a new folder "MenuTabs" in folder "Resources" that contains the 5 imported images.
   - The **images have been imported** and can be used in the Scene Composer solution.



Creating the TabButton

We will use Scene Composer's built-in RadioButtons as controls for our tab bar. Since the usual RadioButtons' appearance doesn't fit into a tab bar optically, we are going to customize the RadioButton to create a TabButton. We will use the radio button control as our base template.

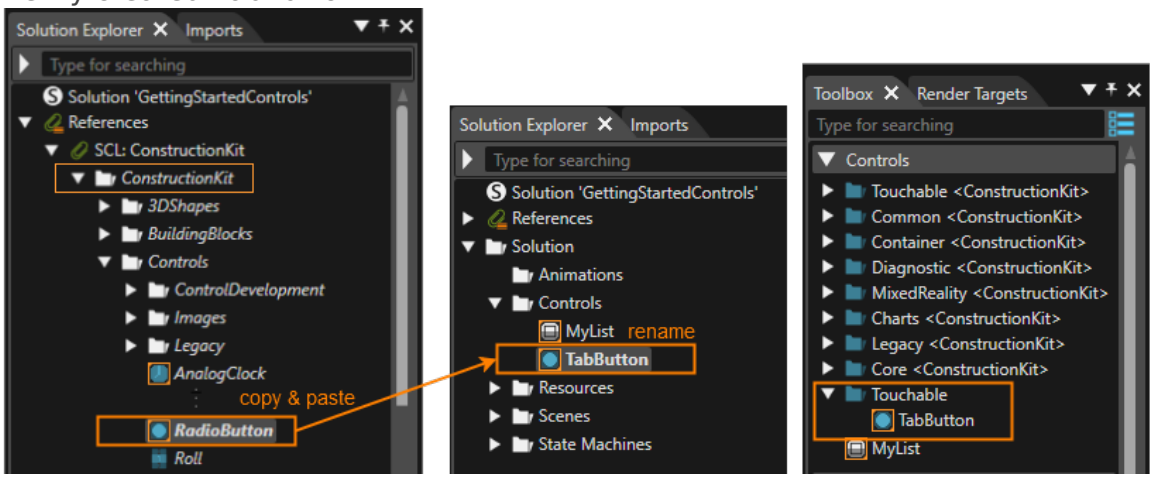
Remove the RadioButton's Knob

The RadioButton control will be customized to look like buttons: the typical RadioButton circle icon will be replaced by background images for the button's various states (normal, selected, pressed...).

1. In the Solution Explorer navigate to "References > SCL: ConstructionKit > ConstructionKit > Controls > RadioButton" and **copy the RadioButton control** either via the context menu "Copy" or the keyboard shortcut <Ctrl + C>.
   - The RadioButton control is in the clipboard and ready to be pasted.

2. In the Solution Explorer navigate to "Solution > Controls" and **paste the RadioButton control** either via the context menu "Paste" or the keyboard shortcut <Ctrl + V> into the Controls folder.
   - In Solution Explorer you can see that the RadioButton has been pasted into the Solution/Controls folder.

3. In the Solution Explorer **rename the RadioButton control** in the Solutions/Controls folder **to "TabButton"** either via the context menu "Rename" or the keyboard shortcut <F2>.
   - In the Solution Explorer you can see that the RadioButton in the Solutions/Controls folder is renamed to "TabButton".
   - In the Toolbox panel's [Controls] section you can see a new folder "Touchable" with the newly created TabButton.



4. In the Solution Explorer or in the Toolbox **double-click** the newly created **TabButton**.
   - In the Scene Editor you can see the TabButton, which currently looks like Scene Composer's RadioButton.

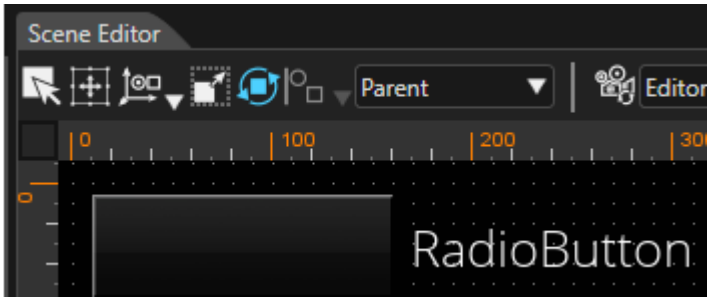
- In the Scene Tree panel you can see the complete design of the TabButton, which is a copy of the RadioButton. On the Scene Tree panel's left side you can see the node structure, on the right side you can see the applied behaviors that form the control's logic.

  > The important thing to know is that all changes you perform here will be applied to ALL instances of this control class immediately.
  > This is also valid when changing properties as long as your control class is opened. You can close it by clicking the left arrow at the top of the Scene Tree or by using the context menu of the Control in the SceneTree and choosing "Go To Previous Element".
  > If you are outside of the control's "Edit mode", all changes in the properties or behavior panel will influence a single instance of the control.

5. In the Scene Tree panel navigate to and **select** the **RadioButtonNode** (TabButton > StackLayout > OverlayLayout > RadioButtonNode). In the Scene Tree panel's right side select the RadioButtonNode **behavior actions** for setting images one by one as (see the table in the next step below). For each of the behaviors **click on the magnifying glass icon** next to the Image property in the Properties panel.
   - The Choose Item dialog will be opened

6. In the "Choose Item" dialog **select the image** as in the table below and click [OK].
   - The "Choose Item" dialog will be closed.
   - The respective **image** has been **changed**.
   - In the Scene Editor the TabButton (former RadioButton) has changed as you can see in the image below.

RadioButtonNode - Behavior	Image
SetNormalImage	/Resources/MenuTabs/ButtonNorma_BkBmp
SetSelectedImage	/Resources/MenuTabs/ButtonNorma_Prss_OreBmp
SetSelectedPressedImage	/Resources/MenuTabs/ButtonNorma_Prss_OreBmp
SetNormalPressedImage	/Resources/MenuTabs/ButtonNorma_Prss_OreBmp



Configure the Text's Position

The positioning of text and background is not quite right. The following steps will correct that.

1. In the Scene Tree panel **select** the **TextNode** (TabButton > StackLayout > TextNode). In the **Properties panel's [Layout]** section change the HorizontalAlignment and the Margin according to the table below.
   - In the Scene Editor panel minor changes in text positioning are visible.

Target Node	HorizontalAlignment	Margin	
		Left	Right
RadioButtonNode	HCenter	6	6

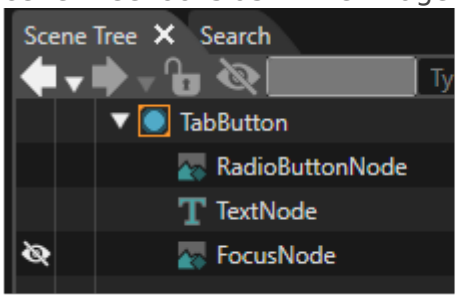
2. In the Scene Tree panel **select** the **TextNode** (TabButton > StackLayout > TextNode). In the Scene Tree panel's right side select the ProvideTextProperty. In the **Property panel** change the Text property to "TabButton".
   - In the Scene Editor the TabButton's text is changed to "TabButton".

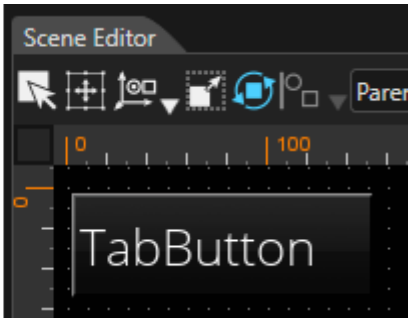
3. In the Scene Tree panel **select** the **TextNode** and **drag it onto** the root node **TabButton**, which is called "TabButton".
   - In the Scene Tree panel you can see that the TextNode is now directly under the TabButton.
   - In the Scene Editor the TextNode with the text "TabButton" is visible. It is hidden behind the Image of the RadioButtonNode.

4. In the Scene Tree panel **select** the **RadioButtonNode** and **drag it onto** the root node **TabButton** as well.
   - In the Scene Tree panel you can see that the RadioButtonNode is now between the TabButton and the TextNode.
   - In the Scene Editor the TextNode with the text "TabButton" is visible again as in the picture below.

5. In the Scene Tree panel **select** the **StackLayout** (with the OverlayLayout) and **delete** it either via the context menu "Delete" or by pressing the delete key on your keyboard.
   - The Delete Item dialog will be opened displaying which items are going to be deleted.

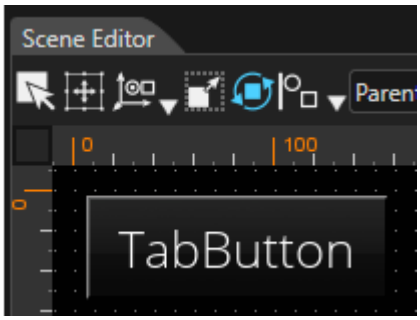
6. In the Delete Item dialog **press** the **[OK]** button.
   - In the Scene Tree panel you can see that StackLayout and OverlayLayout have been deleted.
   - The Scene Tree looks as in the image below.





7. In the Scene Tree panel **select** the **RadioButtonNode**. In the **Properties panel's [Layout]** section change Alignment and Margin properties as well as the StretchBehavior according to the table below.
   - In the Scene Editor the TextNode's text "TabButton" is neatly positioned in the center of the button (see image below).
   - We have successfully created and customized our own TabButton control, which can be used like any other control.

Target Node	Vertical	Horizontal	Margin		StretchBehavior
	Alignment		Left	Right	
RadioButtonNode	VStretch	HStretch	0	0	Fill



Creating MenuTabs Scene

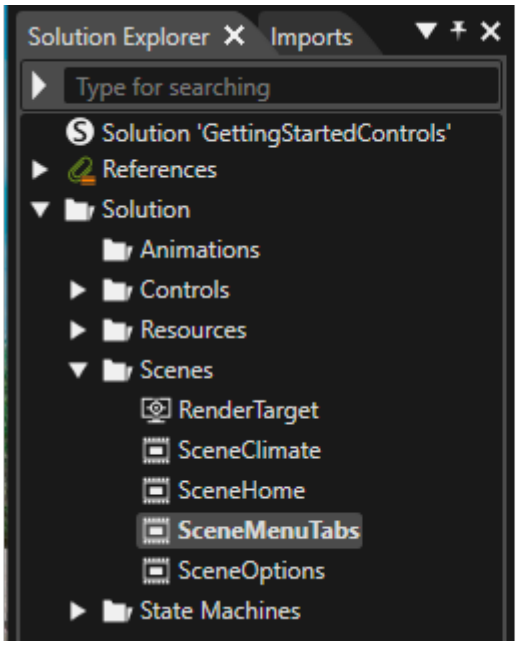
A MenuTabs scene will be created and configured with a GridLayout holding various Buttons.

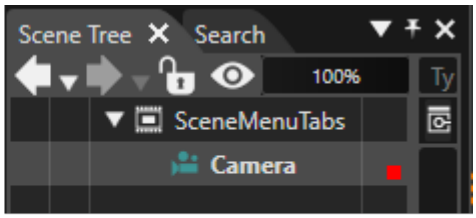
Setting up the Scene

In the following steps a scene is created and the camera as well as the viewport are configured.

1. **Create a new scene** via the menu bar "Scene" - "New 2D Scene" or press < Ctrl+2 > on your keyboard
   - The "Add New Scene 2D" dialog will be opened.

2. In the first row select "/Scenes" from the drop-down menu. Enter "**SceneMenuTabs**" as name and press **[OK]**. We use the prefix "Scene" to have all scenes listed next to each other in the Solution Explorer panel.
   - A new scene "SceneMenuTabs" is created and visible in the Solution Explorer panel in folder "Scenes".
   - The new scene is opened in the Scene Editor and in the Scene Tree panel.
   - In the Scene Tree panel you can see that the scene has a camera.
   - In the Render Targets panel you can see that the camera is already added to the RenderTarget.





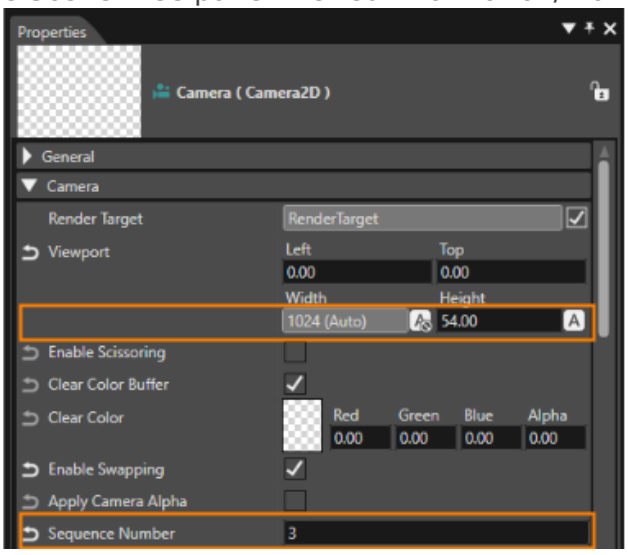
3. In the **Render Targets** panel check the checkbox of "SceneMenuTabs\Camera" in order to **enable rendering**.
   - Rendering of the scene is enabled.

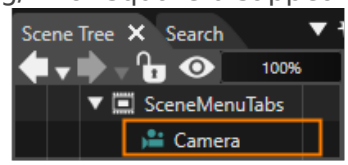
4. In the Scene Tree panel select the Camera. In the Properties panel's **[Camera]** section configure the Viewport height to 54.

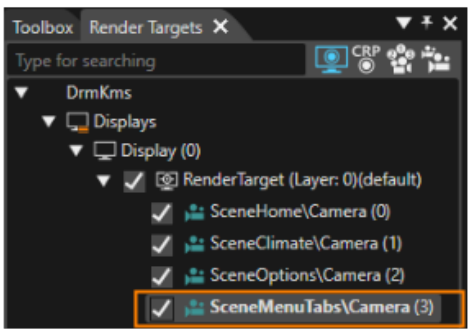
   > The value of this property is calculated automatically. In order to be able to edit it, you have to click on the **[A]** button to the property's right, which will switch auto off.

   - In the Scene Editor you can see that the height of the viewport is reduced to 54, which is about the height of the TabButton control.

5. In the Scene Tree panel select the Camera. In the **Properties panel's [Camera]** section change the **sequence number** to **3**.
   - The camera's sequence number is set to 3.
   - In the Scene Tree panel the red Information/Warning/Error square disappears.







Adding and Arranging Buttons

A GridLayout is configured to arrange 3 different kinds of buttons forming our menu of our configured car HMI.

1. From the **Toolbox panel's [Controls]** section under "Container" drag a **GroupBox** into the Scene Tree panel directly onto SceneMenuTabs.
   - In the Scene Tree panel you can see that there is a GroupBox node.
   - In the Scene Editor you can see a square with a white dashed outline.

2. In the Scene Tree panel **select the GroupBox** node. In the **Properties panel's [Layout]** section configure the Size according to the table below.

   > The value of this property is calculated automatically. In order to be able to edit it, you have to click on the **[A]** button to the property's right, which will switch auto off.

   - In the Scene Editor the white dashed outline will match the yellow line that marks the viewport.

Target Node	Size	
	X	Y
GroupBox	1024	54

3. In the Scene Tree panel click on the little triangle to the left of the GroupBox to expand its content.
   - In the Scene Tree panel the GroupBox node will open and you can see the Content anchor below.

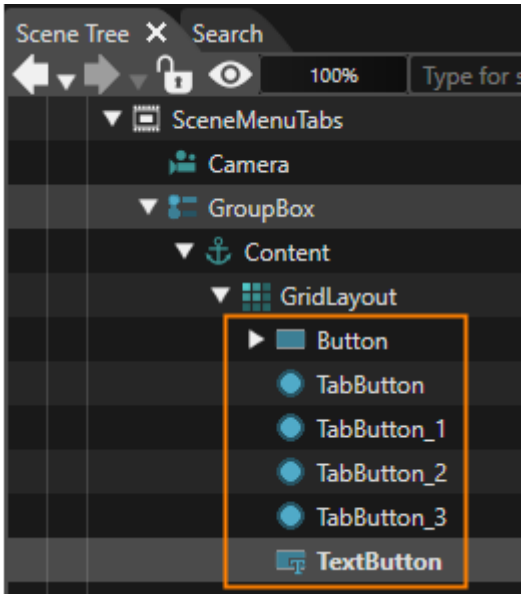
4. From the **Toolbox panel's [Node 2D]** section drag a **GridLayout** into the Scene Tree panel directly onto the **GroupBox's Content anchor**.
   - In the Scene Tree panel you can see the GridLayout node.

5. In the Scene Tree panel **select** the **GridLayout**. In the Proper**ties panel's [Layouter]** section change the Column Count to 6.
   - The Column Count has been changed to 6.

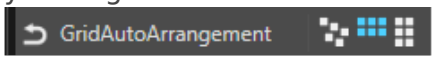
6. From the **Toolbox panel's [Controls]** section **drag Controls into the Scene Tree panel** directly onto the GridLayout according to the table below.
   - In the Scene Tree panel you can see that a Button, 4 TabButtons and a TextButton have been added to the GridLayout.
   - In the Scene Editor the blue Button and 4 TabButtons (with the dark background) are rendered directly over each other, so that not all of them are visible.

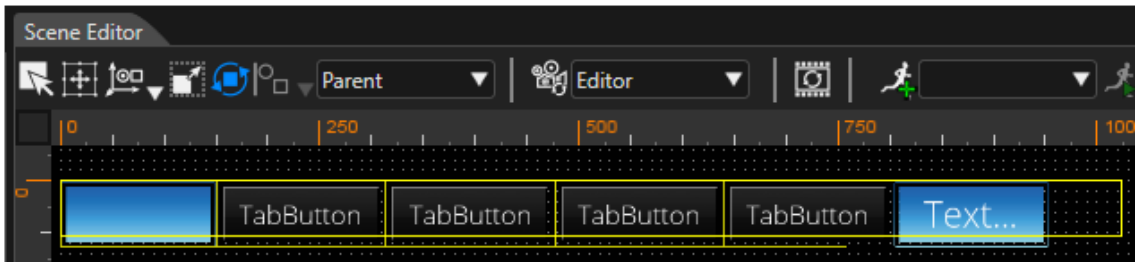
Controls Section	Controls

Touchable<ConstructionKit>	1Button
Touchable	4 TabButtons
Touchable <ConstructionKit>	1 TextButton



7. In the Scene Tree panel **select** the **GridLayout**. In the **Properties panel's [Layout]** section configure **GridAutoArrangement** for **"horizontal"**.
   - In the Scene Editor you can see that the Button, TabButtons and TextButton are neatly arranged.



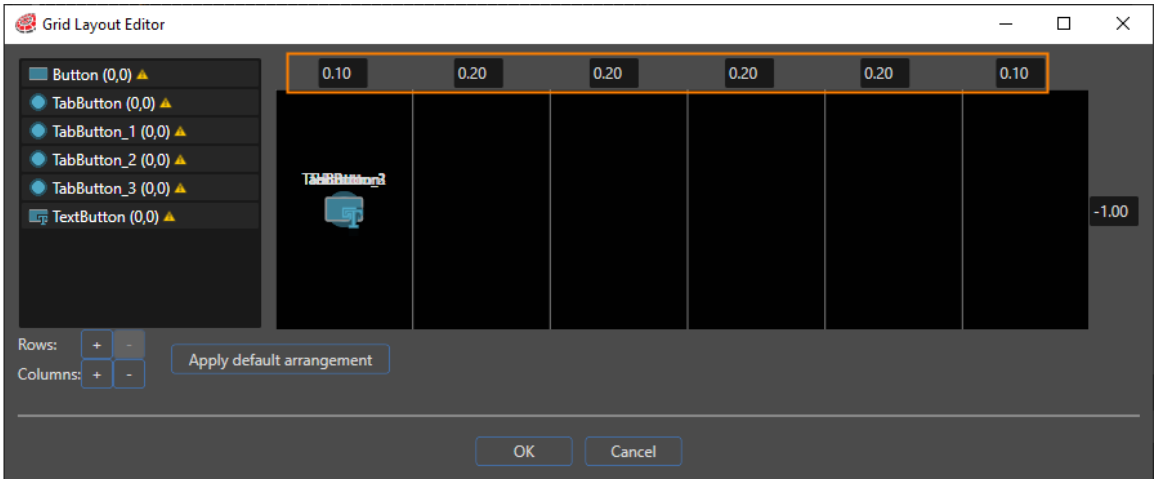


8. In the Scene Tree panel right-click on the **GridLayout** and choose **[Configure Layouter...]** from the context menu.
   - The Grid Layout Editor will be opened.
   - In the Grid Layout Editor you can see that the Controls are bunched together inside of the first cell.

> The Controls don't have to be configured for a certain column. The Controls will be arranged at runtime using the GridAutoArrangement property that we configured in the previous step.

9. In the Grid Layout Editor **configure** the **columns' widths** according to the table below and press **[OK]**.
   - In the Scene Editor you can see that the columns widths have changed.

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
0.10	0.20	0.20	0.20	0.20	0.10



Configuring the Menu Bar's Buttons

The following steps will adapt the Buttons' visual appearance to the intended design and configure the Buttons' texts.

1. In the Scene Tree panel **expand** the **Button** by clicking on the little triangle on its left.
   - In Scene Tree panel a Content anchor will get visible.

2. From the **Toolbox panel's [Controls]** section under "Common <ConstructionKit>" **drag** a **Text control** and drop it directly **onto** the **Button's Content anchor**.
   - In the Scene Tree panel the Text control appears beneath the Content anchor.
   - In the Scene Editor the Button is displayed with the text "Text" on it.

3. In the Scene Tree panel **select** the **Text control**. In the **Properties panel** change the properties according to the table below.
   - In the Scene Editor the Button is displayed with the centered sign ">".

Properties Section	Property	Value
Control	Text	>
Layout	Vertical Alignment	VCenter
	Horizontal Alignment	HCenter

4. In the Scene Tree panel **select** the **Button**. In the **Properties panel's [Control]** section change the Button's Image properties according to the table below. To change the images click on the magnifying glass icon to open the Choose Item dialog, select the appropriate image and click **[OK]**.
   - In the Scene Editor the Button's background will be displayed in almost black color.

Properties Section	Property	Value
General	Name	Next
Control	Normal Image	/Resources/MenuTabs/ButtonNorma_BkBmp
	Pressed Image	/Resources/MenuTabs/ButtonNorma_Prss_OreBmp
	Focused Image	/Resources/MenuTabs/ButtonNorma_Prss_OreBmp

5. In the Scene Tree panel **select** the **TabButtons** one by one (from top to bottom) and apply the properties in the **Properties** panel according to the table below.
   - In the Scene Tree panel all TabButtons are renamed.
   - In the Scene Editor you can see the TabButtons' texts (from left to right): HOME, CLIMATE, PHONE, OPTIONS.

General > Name	Control > Text
TabHome	HOME
TabClimate	CLIMATE
TabPhone	PHONE
TabOptions	OPTIONS

6. In the Scene Tree panel **select** the **TextButton**. In the **Properties** panel apply the properties as given in the table below.
   - In the Scene Editor the TextButton's Text and Background have changed according to the configuration
   - In the Scene Tree panel the TextButton is renamed.

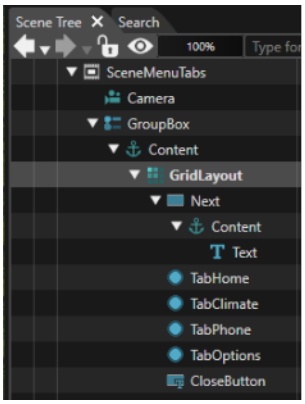
Properties Section	Property	Control > Text
General	Name	CloseButton
Control	Text	X
	Normal Image	/Resources/MenuTabs/ButtonNorma_BkBmp
	Pressed Image	/Resources/MenuTabs/ButtonNorma_Prss_OreBmp
	Focused Image	/Resources/MenuTabs/ButtonNorma_Prss_OreBmp

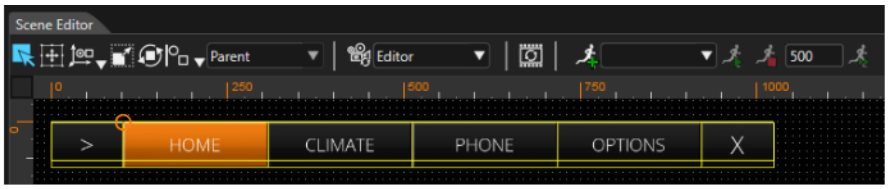
7. In the Scene Tree panel **select** the **GroupBox**. In the **Properties panel's [Control]** section click on the magnifying glass icon next to property "**Default Selected Object**".
   - The "Choose Item" dialog will be opened.

8. In the Choose Item dialog navigate to and **select** the **"TabHome"** TabButton which can be found via the path below, and click **[OK]**.

- In the Scene Editor the "TabHome" TabButton will be displayed selected as in the image below.

Default Selected Object	Path
	/Scenes/SceneMenuTabs/GroupBox/Content/GridLayout/TabHome





Integration of SceneMenuTabs

To integrate the SceneMenuTabs into the Display we have to reduce the height of our Home and Climate scenes.

1. In the **Render Targets panel select** the **cameras** one by one and change the properties in the **Properties panel's [Camera]** section according to the table below.
   - In the Scene Editor for SceneHome, SceneClimate and SceneOptions you can see that the height of the yellow camera viewport rectangle is reduced to 714.

Camera	Viewport			Enable Swapping
	Left	Top	Height	
SceneHome\Camera	0	0	714	TRUE
SceneClimate\Camera	0	0	714	TRUE
SceneOptions\Camera	0	0	714	TRUE
SceneMenuTabs\Camera	0	714	54	FALSE

2. In the menu bar go to **View > Display > Display-0**.
   - The **Display(0)** window will be opened.

3. In the **Render Targets panel** check the SceneMenuTabs\Camera and one of the other cameras.
   - In the Display(0) window you should be able to see the SceneMenuTabs on the bottom of the window and the other scene on the top just like in the image below.



Adding Functionality to the MenuTabs

When pressing a TabButton the respective scene should be displayed. This will be solved by configuring conditions and behaviors using Fusion to trigger scene activation and deactivation.

> A detailed explanation about Fusion, how to operate Fusion (how to add Fusion Nodes) and a configuration example can be found in the documentation about Fusion.

1. In the Solution Explorer **double-click** on **SceneMenuTabs**.
   - In the Scene Editor and in the Scene Tree panel SceneMenuTabs will be opened.

2. In the Scene Tree panel navigate to SceneMenuTabs > GroupBox > Content > GridLayout > TabHome. Click on the **Fusion button** on the Scene Tree panel's right side.
   - Fusion will be opened with the TabHome Fusion Node in the center.
   - The TabHome Fusion Node has a Behaviors connector.

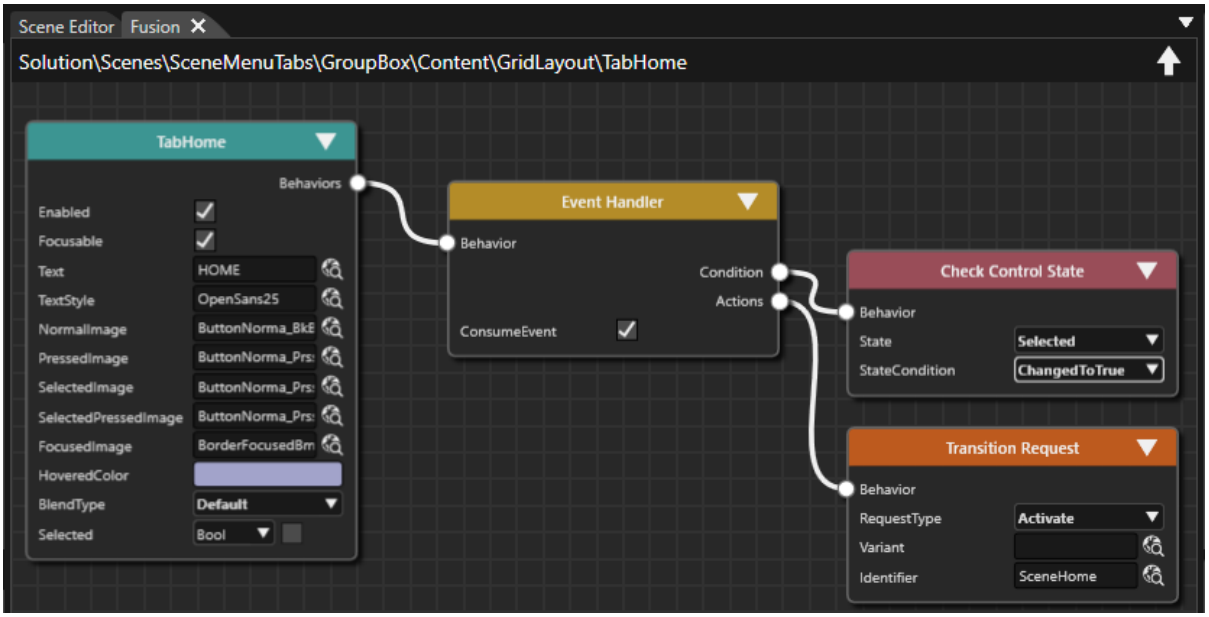
3. **Add Fusion Nodes** to Fusion diagram the according to the table below.

- 3 new Fusion node will be added to the Fusion diagram: Event Handler Fusion Node, Check Control State Fusion Node and Transition Request Fusion Node.

Connect to Fusion Node	Connector	Add Fusion Nodes
TabHome	Behaviors	Event Handler
Event Handler	Condition	Check Control State
	Action	Transition Request

4. **Configure** the newly added **Fusion nodes** according to the table below.
   - If TabHome's selected state changes to true, a transition to activate SceneHome is requested.

Fusion Node	Property	Value
Check Control State	State	Selected
	StateCondition	ChangedToTrue
Transition Request	RequestType	Activate
	Identifier	SceneHome



5. In Fusion **select** the 3 newly added EventHandler Fusion Node, the Check Control State Fusion Node and Transition Request Fusion Node. **Now copy and paste** these nodes into the Fusion diagram (either via the context menu or via the keyboard shortcuts <Ctrl>+c and <Ctrl>+v).
   - The Fusion nodes are copied, resulting in: Event Handler_1 Fusion Node, Check Control State_1 Fusion Node and Transition Request_1 Fusion Node.

6. Change the **configuration of the Fusion Nodes** created via Copy&Paste according to the table below.

- If TabHome's selected state changes to false, a transition to deactivate SceneHome is requested.

Fusion Node	Property	Value
Check Control State_1	State	Selected
	StateCondition	ChangedToFalse
Transition Request_1	RequestType	Deactivate
	Identifier	SceneHome

7. **Copy** all **Fusion Nodes** of this Fusion diagram except for the TabHome Fusion Node and paste them into the Fusion diagrams of TabClimate and TabOptions. Adapt the TransitionRequest Fusion Nodes' Identifier to SceneClimate or SceneOptions respectively according to the table below.
   - If TabClimate's selected state changes to true, a transition to activate SceneClimate is requested.
   - If TabClimate's selected state changes to false, a transition to deactivate SceneClimate is requested.
   - The same is true for TabOptions and SceneOptions.
   - With these configurations done, the SceneMenuTabs works as a navigation bar, enabling the user to switch between different scenes.

   > For this tutorial, ScenePhone has been skipped, since most controls have already been covered by other scenes.

Fusion Diagram of	Fusion Node	Identifier Value
TabClimate	Transition Request	SceneClimate
	Transition Request_1	SceneClimate
TabOptions	Transition Request	SceneOptions
	Transition Request_1	SceneOptions

8. **Save** your work and **start** the **Player** to test the functionality of your configuration.
   - Pressing TabClimate will activate SceneClimate, Pressing TabOptions will activate SceneOptions and Pressing TabHome will activate SceneHome.

Configuring the Menu Bar to Hide/Reappear

We want to make our menu bar with our TabButtons appear on hovering an indicator and disappear when clicking a button. To achieve this, the GroupBox will be moved into a Drawer Control. The indicator RenderNode2D will be enhanced with Control functionality.

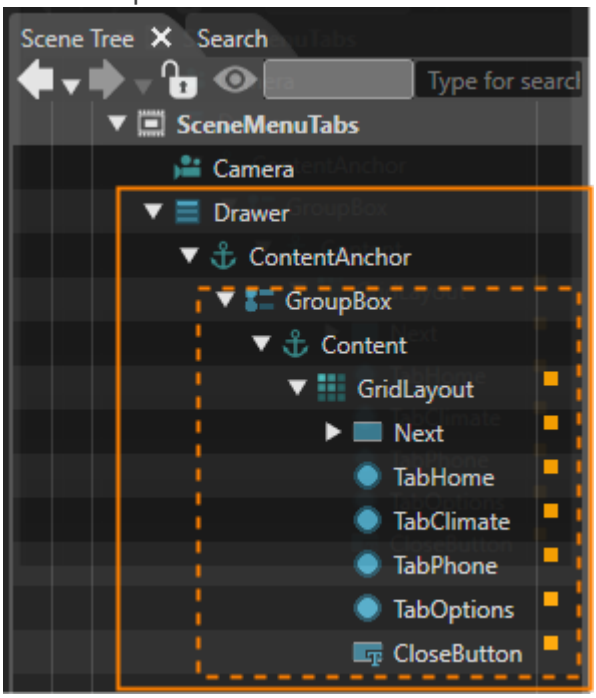
Configuring a Drawer Control

Our menu bar will be encapsulated into a Drawer Control that can then be opened and closed like a drawer at a desk, hiding or showing its contents according to its opened/closed state.

1. Open **SceneMenuTabs**. From the **Toolbox panel's [Controls]** section under "Touchable <ConstructionKit>" **drag** a **Drawer** control and drop it into the Scene Tree panel directly onto the SceneMenuTabs.
   - In the Scene Tree panel you can see that a Drawer control has been added to your scene.

2. In the Scene Tree panel **expand** the newly added **Drawer control**.
   - A ContentAnchor will be displayed.

3. In the Scene Tree panel drag the **SceneMenuTabs > GroupBox** directly onto the **Drawer's ContentAnchor.**
   - The GroupBox and all its content will be moved into the Drawer control.



4. **Configure** the **Drawer Control**'s properties in the Properties panel according to the table below.
   - The Drawer is configured to only appear if needed and to appear from the Bottom.
   - The Drawer is configured closed and its content won't be visible.

   > By unchecking the "Open" property, the Drawer Control will be closed, so it will hide its content. The content is still available, just hidden in a drawer (like

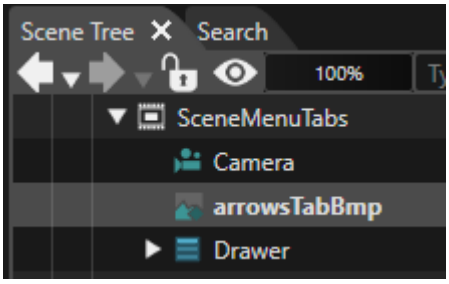
> in a desk drawer, where you can put away stuff).
> You can check and uncheck this property during editing as you like and watch it appear and disappear, but please make sure to uncheck it in the end.
> If there is no visual change when toggling the "Open" property, try forcing a refresh by pressing <F5> on your keyboard.

Drawer Properties Section	Property	Value
Control	Open	False
	Appear Direction	Bottom

5. In the Solution Explorer panel **navigate** to the image **arrowTabsBmp** which can be found following the path below and drag this image into the Scene Editor or into the Scene Tree panel.
   - In Scene Editor the arrowTabsBmp image will be displayed.
   - In Scene Tree panel a new RenderNode2D with the arrowTabBmp image will be available.

arrowTabsBmp	Solution Explorer Path
	Solution > Resources > MenuTabs > arrowsTabBmp

6. In the **Scene Tree panel** make sure that **arrowsTabBmp** is **positioned** above the Drawer control as seen in the image below.
   - The indicator is now hidden while the drawer is open and our menu bar is displayed.



7. In the Scene Tree panel select the newly created **RenderNode2D arrowsTabBmp**. In the **Properties panel** change the properties as in the table below.
   - The RenderNode2D is positioned correctly.

arrowsTabBmp Properties Section	Property		Value
Transformations	Position	X	469
	Position	Y	21
	Scale	X	0.5
	Scale	Y	0.5

Opening and closing the Drawer Control

The Drawer Control shall be opened when hovering or clicking the indicator and it shall be closed when the Close Button is pressed. This functionality is configured using Fusion.

> A detailed explanation about Fusion, how to operate Fusion (how to add Fusion Nodes) and a configuration example can be found in the documentation about Fusion.

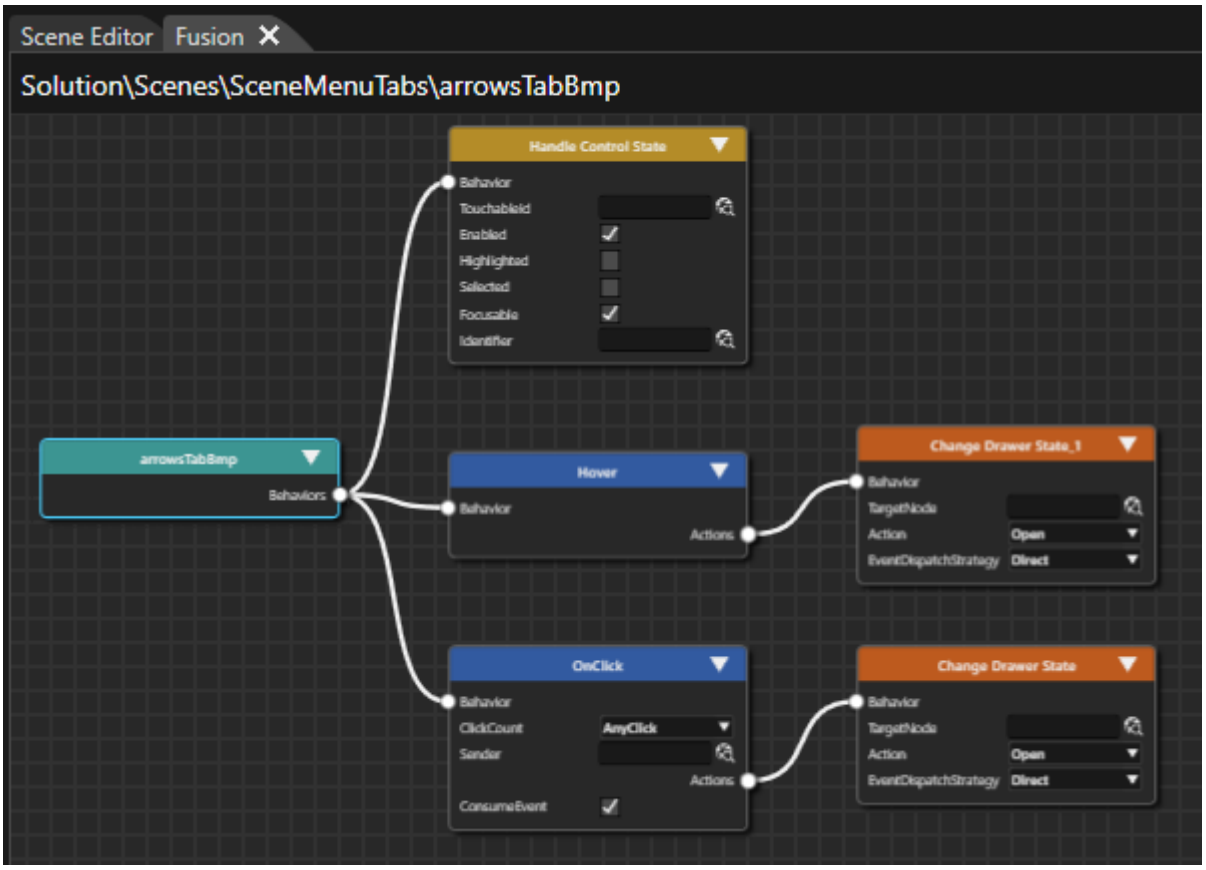
1. While arrowsTabBmp is selected in the Scene Tree panel click on the **[Fusion...]** button on the right side of the Scene Tree.
   - Fusion will be opened, displaying the **arrowsTabBmp** Fusion Node.

2. Add a **HandleControlState** Fusion Node to the arrowTabBmp Fusion Node using its Behaviors connector according to the table below.
   - A new HandleControlState Fusion Node will be added to the Fusion diagram.
   - RenderNode2D is now enhanced with control functionality.

Connect to Fusion Node	Connector	Add Fusion Nodes
arrowsTabBmp	Behaviors	HandleControlState

3. **Add Fusion Nodes** to the Fusion diagram according to the table below to build up the Fusion diagram shown in the image below.
   - 4 new Fusion nodes have been created: OnClick Fusion Node, Hover Fusion Node, Change Drawer State Fusion Node and Change Drawer State_1 Fusion Node.

Connect to Fusion Node	Connector	Add Fusion Nodes
arrowsTabBmp	Behaviors	Hover
	Behaviors	OnClick
OnClick	Actions	Change Drawer State
Hover	Actions	Change Drawer State_1



4. In Fusion **configure** the **TargetNode** of Change Drawer State Fusion Node and Change Drawer State_1 Fusion Node according to the table below. To do this, click on the magnifying glass icon on the right side of the **TargetNode** property and follow the path to the Drawer control given in the table below.
   - The Drawer will now be opened when the indicator is hovered or clicked.

TargetNode	Choose Item Path
Change Drawer State	Scenes > SceneMenuTabs > Drawer
Change Drawer State_1	Scenes > SceneMenuTabs > Drawer

5. In the Scene Tree panel **select** the **CloseButton**, which can be found in the path below. Then click on the **[Fusion...]** button on the Scene Tree panel's right side.
   - Fusion will be opened with the CloseButton's Fusion diagram.
   - The CloseButton is already a Control and has an EventHandler and a Check Click condition.

CloseButton	Scene Tree Path
	SceneMenuTabs > Drawer > ContentAnchor > GroupBox > Content > GridLayout > **CloseButton**

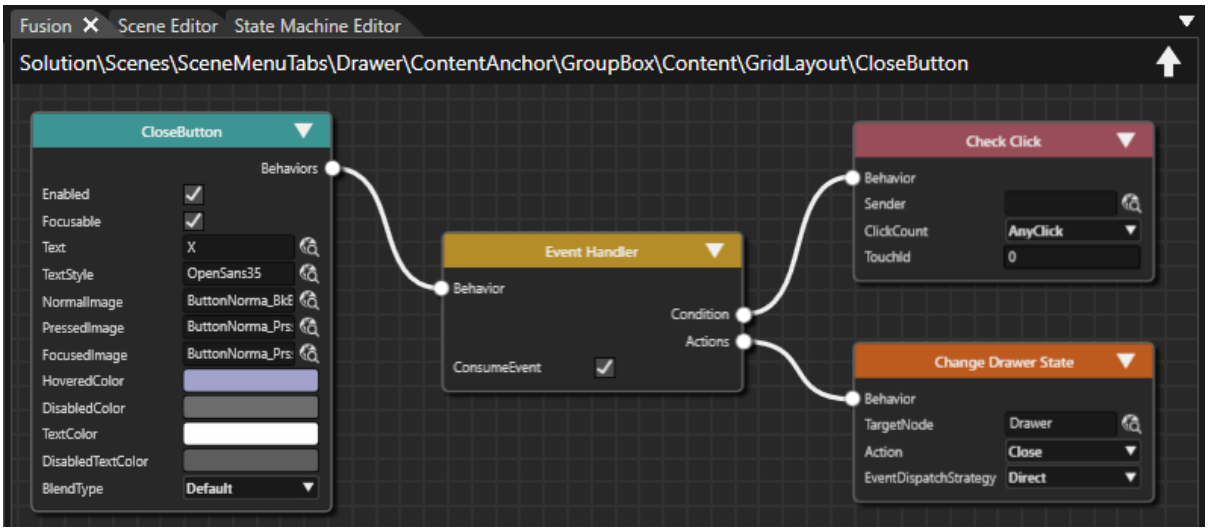
6. **Add a Fusion Node** to the Event Handler Fusion Node using its Actions connector according to the table below.Change Drawer State action.
   - In Fusion a new Fusion Node has been added to the Fusion diagram: Change Drawer State Fusion Node.

Connect to Fusion Node	Connector	Add Fusion Nodes

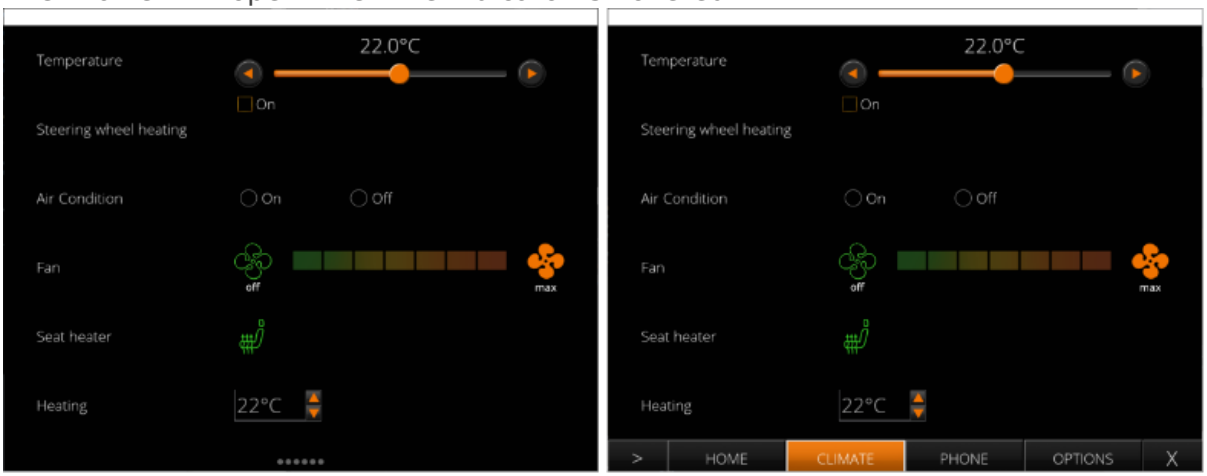
Event Handler	Actions	Change Drawer State

7. In Fusion **configure** the **Change Drawer State** Fusion Node as in the table below.
   - The Close Button is now configured to close the drawer.
   - The Fusion diagram looks like in the image below.

Change Drawer State Property	Value
TargetNode	Scenes > SceneMenuTabs > Drawer
Action	Close



8. **Save** your work and **start the Player** to test the functionality of your configuration.
   - The Drawer will close when the [Close] button is pressed.
   - The Drawer will open when the indicator is hovered.



Revision #22
Created 2023-02-16 08:55:09 UTC by Kanai Tomoaki
Updated 2025-09-12 05:30:52 UTC by Elisabeth Zauner