Control Behaviors

Construction Kit Control Behaviors

Map Control Behaviors

Map

Provides the map rendering and navigation functionality.

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id of the behavior.
Bearing	Float	Clockwise angle relative to North that the users currently faces.
Longitude	Candera::Variant	Longitude of current users location.
Latitude	Candera::Variant	Latitude of current users location.
SimulateDriving	bool	Simulate driving along computed route when in navigation mode.
NaviMessageNode	AbstractNodePointer	Node to which navigation instructions will be sent.
LocationMesh	AbstractNodePointer	Node containing mesh(es) used to render the location indicator.
RouteOverlayColor	Candera::Color	Color used by the route overlay.
AvoidTolls	bool	Avoid tolls during route computation.
AvoidMotorways	bool	Avoid motorways during route computation.
AvoidFerry	bool	Avoid ferrys during route computation.

Map Bridge

Builds a bridge to MapBehavior forwarding touch events.

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id of the behavior.
TargetNode	AbstractNodePointer	Node containing the target MapBehavior to which events will be forwarded.

Scroll Control Behaviors

List Behavior

ListBehavior is a list supporting scrolling itemwise and pagewise combined with either relative or absolute scrolling.

About implicit roles of ChildNodes

The first child node is used as content (Group2D) and the second as scrollable (Scrollbar).

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id of the behavior.
ScrollDirection	ScrollDirection::Enum	The direction in which the ListBehavior scrolls. *Possible values: Vertical, Horizontal*
Position	FeatStd::Float	The pixel position of the visible content within the list content.
AnimationDuration	FeatStd::UInt32	Duration of the scrolling animation.
DecelerationFactor	FeatStd::Float	Loss of flick animation speed over time.
MaxSpeedPerFlick	FeatStd::Float	Maximum amount of speed each flick motion adds to the flick animation.
MinSpeedPerFlick	FeatStd::Float	Minimum speed required to count a touch gesture as a flick and start or continue the flick animation.
MinMovementToScrolling	FeatStd::UInt32	Minimum distance a touch gesture needs to move to start a scrolling animation.
MaxTouchSampleAge	FeatStd::UInt32	Maximum age in milliseconds of touch movement speed counted towards generating flick animation speed.

Items	ListPropertyType<DataContextItem::SharedPointer>	The items property is used for binding of dynamic items. It can only be set via databinding. Only one visible list control should be bound to a fragment list in the model. Otherwise the list fragments in the model are not able to provide data for both list controls. Use a separate list model that is filled with the same data if such a use case is required.
ListFragmentTriggerOffset	FeatStd::SizeType	A dynamic list is requested in fragments to reduce memory consumption and overall performance overhead. This trigger offset value defines an area at the begin and end of the list fragment. Whenever the list shows a part of these areas a new fragment is requested from the model.
ListFragmentSize	FeatStd::SizeType	A dynamic list is requested in fragments to reduce memory consumption and overall performance overhead. This buffer size defines how large this fragment should be. By default it is set to 0 which results in an automatic calculation. The automatic calculation assumes the number of item nodes is configured to match the maximum number of visible nodes for one page. By multiplying this number by three and adding the trigger offset one time for the begin and one additional time for the end to calculate for the automatic buffer size the list has a spare page on each side before a new fragment has to be requested
HandleKeyboardEvents	bool	Makes the behavior handle KeyboardFocusNavigation events internally.
WrapAround	bool	Set this flag to true for static list to wrap around the list items on the top and bottom of the list

Scrollbar

Reflects the position of the scrollable behavior. E.g. ListBehavior2D.

About implicit roles of ChildNodes

A variable number of child nodes is used to control the following features.

- background: a RenderNode2D used as background
- prevItem: a control to navigate to the previous item
- nextItem: a control to navigate to the next item
- prevPage: a control to navigate to the previous page
- nextPage: a control to navigate to the next page
- marker: a control used as a thumb

Depending on the number of provided child nodes, the nodes are interpreted as follows.

- 1 child: marker
- 2 children: background, marker
- 3 children: prevPage, nextPage, marker
- 4 children: background, prevPage, nextPage, marker
- 5 children: prevItem, nextItem, prevPage, nextPage, marker

- 6 children: background, prevItem, nextItem, prevPage, nextPage, marker

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id of the behavior.
TouchListenerNode	Candera::AbstractNodePointer	This node defines the area on which the ScrollbarBehavior will listen for the touch events.
ScrollDirection	FeatStd::Optional< ScrollDirection::Enum >	The direction in which the ListBehavior scrolls. By default it is configured as a vertical scrollbar.
ScrollbarVisibility	ScrollbarVisibility::Enum	Configure the visibility of the scrollbar. By default it is set to Automatic which will determine the visibility based on the actual content. *Possible values: Automatic, Collapsed, Visible*
HandleMinimumSize	FeatStd::UInt32	The minimum height (for vertical scrollbar) or minimum width (for horizontal scrollbar) of the handle in pixels.
MinMovementToScrolling	FeatStd::UInt32	Minimum distance a touch gesture needs to move to start scrolling.
ScrollableNode	Candera::AbstractNodePointer	Node of the scrollable behavior.

Scroll View

Scroll View is a Touchable Control that has a ContentAnchor which can be filled with various content. Depending on the size of the content and the configured Scroll View's Layout Size, a horizontal and/or vertical scrollbar appears.

Because of the fact that Scroll View supports horizontal and vertical scrolling, it receives the ScrollableEvent from Send Scrollable Event Behavior (can also be forwarded from a Scrollbar Behavior Scrollbar via Event Bridge Behavior).

Example Usage

- Scroll View Example

Common Properties

This behavior does not have any configurable properties.

About implicit roles of ChildNodes

The first child node is used as content (Group2D), the second as vertical scrollbar (Scrollbar) and the third as horizontal scrollbar (Scrollbar).

Snap List Item Configuration

Allows the user to define the snap position of a list or list item by setting its item alignment (for example, Center) and offset. This is then applied when navigating or scrolling to align the list item.

Can be used with the List Control.

Common Properties

Property Name	Type	Description
ItemAlignment	SnapAlignmentOrientation::Enum	The location of the alignment base point within the list item. *Possible values: Begin, Center, End*
ItemAlignmentOffset	FeatStd::Float	The offset of the alignment point from the configured alignment base point. For End alignment a positive offset will move the alignment point from the list item end towards the list item begin.
ItemAlignmentOffsetInPercent	bool	The offset is either in percent (when set to true) or in pixel (when set to false). In case of percent the item size will be taken to calculate the pixel offset.

Snap List Item Extension

SnapListItemExtensionBehavior registers a ListBehaviorExtension on the given ListBehavior for snapping the focused item to the configured list according to the snapping configuration of the list and the list item node.

Information on how to use this behavior can be found in the chapter of the List Control.

Common Properties

Property Name	Type	Description
ListBehavior	ListBehavior*	The ListBehavior that shall be extended. *This property can be set in the right side of the Scene Tree Panel.*

ListAlignment	SnapAlignmentOrientation::Enum	The location of the alignment base point within the list item. *Possible values: Begin, Center, End*
ListAlignmentOffset	FeatStd::Float	The offset of the alignment point from the configured alignment base point. For End alignment a positive offset will move the alignment point from the list item end towards the list item begin.
ListAlignmentOffsetInPercent	bool	The offset is either in percent (when set to true) or in pixel (when set to false). In case of percent the item size will be taken to calculate the pixel offset.
ItemAlignment	SnapAlignmentOrientation::Enum	The location of the alignment base point within the list item. *Possible values: Begin, Center, End*
ItemAlignmentOffset	FeatStd::Float	The offset of the alignment point from the configured alignment base point. For End alignment a positive offset will move the alignment point from the list item end towards the list item begin.
ItemAlignmentOffsetInPercent	bool	The offset is either in percent (when set to true) or in pixel (when set to false). In case of percent the item size will be taken to calculate the pixel offset.

SubList Behavior

SubListBehavior extends the ListBehavior by special handling when being used in a child node of a ListBehavior.

SubListBehavior is a list supporting scrolling itemwise and pagewise combined with either relative or absolute scrolling.

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id of the behavior.
ScrollDirection	ScrollDirection::Enum	The direction in which the SubListBehavior scrolls. *Possible values: Vertical, Horizontal*
Position	FeatStd::Float	The pixel position of the visible content within the list content.

AnimationDuration	FeatStd::UInt32	Duration of the scrolling animation.
DecelerationFactor	FeatStd::Float	Loss of flick animation speed over time.
MaxSpeedPerFlick	FeatStd::Float	Maximum amount of speed each flick motion adds to the flick animation.
MinSpeedPerFlick	FeatStd::Float	Minimum speed required to count a touch gesture as a flick and start or continue the flick animation.
MinMovementToScrolling	FeatStd::UInt32	Minimum distance a touch gesture needs to move to start a scrolling animation.
MaxTouchSampleAge	FeatStd::UInt32	Maximum age in milliseconds of touch movement speed counted towards generating flick animation speed.
Items	ListPropertyType< ::Courier::DataContextItem::SharedPointer >	The items property is used for binding of dynamic items. It can only be set via databinding. Only one visible list control should be bound to a fragment list in the model. Otherwise the list fragments in the model are not able to provide data for both list controls. Use a separate list model that is filled with the same data if such a use case is required.
ListFragmentTriggerOffset	FeatStd::SizeType	A dynamic list is requested in fragments to reduce memory consumption and overall performance overhead. This trigger offset value defines an area at the begin and end of the list fragment. Whenever the list shows a part of these areas a new fragment is requested from the model.
ListFragmentSize	FeatStd::SizeType	A dynamic list is requested in fragments to reduce memory consumption and overall performance overhead. This buffer size defines how large this fragment should be. By default it is set to 0 which results in an automatic calculation. The automatic calculation assumes the number of item nodes is configured to match the maximum number of visible nodes for one page. By multiplying this number by three and adding the trigger offset one time for the begin and one additional time for the end to calculate for the automatic buffer size the list has a spare page on each side before a new fragment has to be requested
HandleKeyboardEvents	bool	Makes the behavior handle KeyboardFocusNavigation events internally.
ParentListNode	Candera::AbstractNodePointer	The node with a ListBehavior that acts as a parent to this SubListBehavior.

Circular Slider Behavior

This behavior provides the functionality of CircularSlider control by creating a radial mask and setting a rotation of knob node. This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
Receivers	Candera::ProcessValueBehaviorArrayProperty	The process value behaviors that will receive the value as next in the value processing chain. *This property can be set in the right side of the Scene Tree Panel.*
Value	FeatStd::Variant	Value of the slider
MinimumValue	FeatStd::Variant	Minimum value of the slider
MaximumValue	FeatStd::Variant	Maximum value of the slider
StepSize	FeatStd::Variant	Step size for Value
MinimumAngle	FeatStd::Float	Angle of the slider, corresponding to minimum value.
MaximumAngle	FeatStd::Float	Angle of the slider, corresponding to maximum value.
ThumbNode	Candera::AbstractNodePointer	Thumb node which will be rotated.
ImageNode	Candera::AbstractNodePointer	Image node which will be masked.

Drawer Behavior

The Drawer is a Layouting Behavior that can hide or show its content via Events or Touch Inputs. This will only work if the Node has a Group called 'Content' as Child. To support Touch Inputs there needs to be a Node called 'Knob' as Child.

Drawer Behavior to switch node on/off.

This behavior is available for 2D scenes only.

About implicit roles of ChildNodes

The first child node is used as content (Group2D) and the second as knob (RenderNode2D).
The knob node must be named "Knob" for proper dragging support (Touch).

Common Properties

Property Name	Type	Description

Receivers	Candera::ProcessValueBehaviorArrayProperty	The process value behaviors that will receive the value as next in the value processing chain. *This property can be set in the right side of the Scene Tree Panel.*
Open	bool	Shows the Drawer in its opened appearance.
AppearDirection	AppearDirection::Enum	The direction from which the Drawer will open. *Possible values: Top, Left, Right, Bottom*
DragEnabled	bool	Set to true to allow the Drawer to be controlled by Touch Inputs.

Linear Slider Behavior

This behavior provides the functionality of a linear Slider control. It must be used as a receiver in a process value chain and expects a normalized value in the range of 0.0 to 1.0.

This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
Receivers	Candera::ProcessValueBehaviorArrayProperty	The process value behaviors that will receive the value as next in the value processing chain. *This property can be set in the right side of the Scene Tree Panel.*
KnobNode	Candera::AbstractNodePointer	Knob node which will be shifted.
SliderBarNode	Candera::AbstractNodePointer	Image node which will be scaled according to value.
BackgroundNode	Candera::AbstractNodePointer	Image node which will provide background image and also control slider total size.
UseBitmapClipping	bool	If true, the slider bar image will be resized using bitmap clipping instead of the layouter.

Internal Control Behaviors

Caret

Behavior which receives the CaretPositionChanged Event from the EditTextBehavior. It forwards the value to other behaviors (e.g. SetSize) and switches the blinking state.

Common Properties

Property Name	Type	Description
Receivers	Candera::ProcessValueBehaviorArrayProperty	The process value behaviors that will receive the value as next in the value processing chain. *This property can be set in the right side of the Scene Tree Panel.*
CaretNode	Candera::AbstractNodePointer	Node which will be set as caret..
IntervalSwitchBehavior	Candera::Behavior*	The Caret Behavior will change the state of the given IntervalSwitch Behavior. *This property can be set in the right side of the Scene Tree Panel.*

Handle Control State

Manages how a control responds to user actions like clicks, touches, or hover. Sends state events when the control is pressed, selected, or highlighted. State conditions behaviors react to these events to update the control accordingly.

Example Usage

- Hover - Hover Leave Example
- Move Node Example
- OnClick Example
- Set Alpha Value Example
- Set Position Example
- Set Rotation Example
- Start Timer Example
- Switch Image Example
- Perform Transition Requests Example

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id of the behavior.

Enabled	bool	If set to false it will not react on focus events and does not have to support different visualizations for the different states.
Highlighted	bool	If true, the controls state Highlighted is set to true.
Selected	bool	If true, the controls state Selected is set to true.
Focusable	bool	If true, the control is focusable. Otherwise the control can not be focused.
Identifier	Candera::AbstractNodePointer	Identifier which is used when emitting an Event (e.g. Click Event).

DataContext

This behavior provides a local data context on the associated node that can act as a set of binding sources (if a binding source is set at that data context). It is internally used by other behaviors like the list behavior to set individual binding sources for items.

Common Properties

This behavior does not have any configurable properties.

Edit Text

Control Behavior to make a TextNode editable. Receives KeyEvents and emits CaretPositionChanged Events. These can be received by the CaretBehavior to set the position of the caret node.

Common Properties

Property Name	Type	Description
Receivers	Candera::ProcessValueBehaviorArrayProperty	The process value behaviors that will receive the value as next in the value processing chain. *This property can be set in the right side of the Scene Tree Panel.*
InputText	FeatStd::String	Text which will be set as default text.
TextNode	Candera::AbstractNodePointer	Text Node which will be edited.
Placeholder	Candera::AbstractNodePointer	Text which will be shown as placeholder.
SetValueOnEachChange	bool	If true, the value will be set on each input.
MaxTextLength	FeatStd::UInt16	Maximum count of characters.

JustDigits	bool	If true, the Inputfield will receive just digits.

Selection Group

Provides an ArrayProperty for Selection Objects. If one of them gets selected the others get deselected (Objects need a ChangeSelectionActionBehavior). Needs to be attached to a value Behavior.

Common Properties

Property Name	Type	Description
DefaultSelectedObject	Candera::AbstractNodePointer	Selection Object (e.g. RadioButton) which shall be selected by default.

Public Property Control Behaviors

Provide Alignment

Provides properties to set the horizontal and vertical alignment. Especially for Controls.

Common Properties

Property Name	Type	Description
HorizontalAlignment	Candera::HorizontalAlignment	Provides property to set the Horizontal Alignment (especially for Controls). *Possible values: HLeft, HCenter, HRight, Hstretch, HstrechRight, HstrechCenter*
VerticalAlignment	Candera::VerticalAlignment	Provides property to set the Vertical Alignment (especially for Controls). *Possible values: VTop, VCenter, VBottom, Vstretch, VstrechCenter, VstrechBottom*

Copy Size

Copies the layout size to a target node. Especially for Controls to transfer the size from the control node to an inner node.

Common Properties

Property Name	Type	Description
TargetNode	Candera::AbstractNodePointer	The target node to which the events are routed from the source node. If the property is not set, the event will be routed to the node of the behavior

Provide Color

Sets the color of a 2D node with a Color effect (e.g. BitmapBrushColorBlend, SolidColorBrush, MVGRender) or the diffuse color of a material in 3D.

Common Properties

Property Name	Type	Description
PropertyReference	CgiStudioControl::PropertyReference::Enum	Choose how the property gets set. Via own property or from another Behavior. *Possible values: Self, Other*
Color	Candera::Color	The color to set. *This property is only available if the PropertyReference is set to "Self".*

Advanced Properties

Property Name	Type	Description
PropertyProvider	Behavior*	Property Behavior which contains the color which has to be set on the effect. *This property can be set in the right side of the Scene Tree Panel.* *This property is only available if PropertyReference is set to "Other".*

Provide Margin

Provides property to set the margin. Especially for Controls.

Common Properties

Property Name	Type	Description
Margin	Candera::Margin	Provides property to set the Margin (especially for Controls).

Property Provider

Offers to set a property of a selectable type. This property can then be used for other Behaviors (e.g. an image for Set Image or Provide Image).

Common Properties

Property Name	Type	Description
PropertyDataType	CgiStudioControl::PropertyDataTypes::Enum	Choose the type which should be offered by this behavior. *Possible values: Color, Text, Style, Image, Appearance, Value, BlendType, Layout, VertexBuffer*

Advanced Properties

Property Name	Type	Description
Color	Candera::Color	The color which can be used from other Behaviors. *This property is only available if PropertyDataType is set to "Color".*
Text	FeatStd::String	The Text that will be set to the Text Node. *This property is only available if PropertyDataType is set to "Text".*
Style	SharedPointer<Candera::TextRendering::SharedStyle>	The Style that will be set to the Text Node. *This property is only available if PropertyDataType is set to "Style".*
Image	SharedPointer<Candera::Image2D>	The image to set on the effect. *This property is only available for 2D scenes if PropertyDataType is set to "Image".*
Texture	SharedPointer<Candera::TextureImage>	The texture to set on the effect. *This property is only available for 3D scenes if PropertyDataType is set to "Image".*
Appearance	SharedPointer<Candera::Appearance>	The appearance to set on the node. *This property is only available for 3D scenes if PropertyDataType is set to "Appearance".*
Value	FeatStd::Variant	The value that will be set to the Send Value as Event action. *This property is only available if PropertyDataType is set to "Value".*
BlendType	BlendType::Enum	The blend type that will be set. *Possible values: Default, DefaultPremultiplied, DefaultTransparentLayer, Add, Multiply, Screen, Darken, Lighten, SourceAtop, SourceIn, SourceOut, SourceOver, DestinationIn, DestinationOut, DestinationOver, Off.* *This property is only available if PropertyDataType is set to "BlendType".*

Layout	CgiStudioControl::LayoutType::Enum	The layouter type for node. See Set Layout for more information. *Possible values: Horizontal, Vertical* *This property is only available for 2D scenes if PropertyDataType is set to "Layout".*
StackArrangement	StackLayouter::Arrangement	Selects arrangement for stack layouter. *This property is only available for 2D scenes if PropertyDataType is set to "Layout" and the property Layout is set to "Stack".*
VertexBuffer	SharedPointer<Candera::VertexBuffer>	*You can set the VertexBuffer held by the mesh created when importing FBX. This property is only available for 3D scenes if PropertyDataType is set to "VertexBuffer".*

Render Node

Sets rendering of a node to enabled or disabled and provides the setting as a property for Controls.

Common Properties

Property Name	Type	Description
EnableRendering	bool	If true, rendering is enabled.
PostponeChange	bool	If set to true the render change will be handled during the next Update cyle, if false it will be handled immediately.

Provide Appearance

Sets the Appearance of a Render Node and provides the Appearance as a property for Controls.

This behavior is available for 3D scenes only.

Common Properties

Property Name	Type	Description
PropertyReference	CgiStudioControl::PropertyReference::Enum	Choose how the property gets set. Via own property or from another Behavior. *Possible values: Self, Other*
Appearance		*This property is only available if "PropertyReference" is set to "Self".*

PropertyProvider	Behavior*	Property Behavior which contains the color which has to be set on the effect. *This property is only available if "PropertyReference" is set to "Other".* *This property can be set in the right side of the Scene Tree Panel.*

Provide Image

Sets the image of a node and provides the Image as a property for Controls.

Example Usage

- Set Image Example

Common Properties

Property Name	Type	Description
PropertyReference	CgiStudioControl::PropertyReference::Enum	Choose how the property gets set. Via own property or from another Behavior. *Possible values: Self, Other*

Advanced Properties

Property Name	Type	Description
PropertyProvider	Behavior*	Property Behavior which contains the color which has to be set on the effect. *This property is only available if "PropertyReference" is set to "Other".* *This property can be set in the right side of the Scene Tree Panel.*
Image	SharedPointer<Candera::Image2D>	The image to set on the effect. *This property is only available for 2D scenes if "PropertyReference" is set to "Self".*
Texture	SharedPointer<Candera::TextureImage>	The image to set on the appearance. *This property is only available for 3D scenes if "PropertyReference" is set to "Self".*

Set Layout

Class Reference

Provides a published property to set the layout for Controls.

Limitations

- Only certain built-in layout mechanisms are supported
- Scene Composer does not support displaying layout-related properties when layouter is assigned using SetLayout behavior.

Common Properties

Property Name	Type	Description
Layout	CgiStudioControl::LayoutType::Enum	Selects layout type for node. *Possible values: None, Overlay, Stack, Baseline.* *This property is only available if "PropertyReference" is set to "Self".*
PropertyReference	CgiStudioControl::PropertyReference::Enum	Choose how the property gets set. Via own property or from another Behavior. *Possible values: Self, Other*

Advanced Properties

Property Name	Type	Description
StackArrangement	Candera::StackLayouter::Arrangement	Selects arrangement for stack layouter. *This property is only available if Layout is set to "Stack".*
PropertyProvider	Behavior*	Property Behavior which contains the layout which has to be set on the effect. See Property Provider for more information. *This property is only available if "PropertyReference"" is set to "Other".*

Provide Text

Modifies the text of a TextNode or CanvasText and provides the text as property for Controls.

Example Usage

- Set Text Example

Common Properties

Property Name	Type	Description
PropertyReference	CgiStudioControl::PropertyReference::Enum	Choose how the property gets set. Via own property or from another Behavior.
Text	FeatStd::String	The Text that will be set to the Text Node.

Style	SharedPointer< Candera::TextRendering::SharedStyle>	The Style that will be set to the Text Node.
Alignment	Candera::TextNode2DLayouter::TextAlignment	The horizontal text alignment, using the culture text direction, is used to define how text will be horizontally aligned within its layout area. *Possible values: Auto, Left, Center, Right.*
Multiline	bool	If multi line is disabled then the new line control characters will be ignored and text will always be rendered in one line.
LineSpacing	Candera::TextRendering::PixelSize	Defines the distance between the baselines of successive lines of text.
WordWrap	bool	If word wrapping is enabled then the lines that exceed the client area will be split during arrangement into multiple lines.
Trimming	Candera::TextNode2DLayouter::Trimming	Trimming defines the behavior to employ when content overflows the content area. *Possible values: None, Ellipsis, CustomText.*

Advanced Properties

Property Name	Type	Description
PropertyProvider	Behavior*	Property Behavior which contains the color which has to be set on the effect. *This property is only available if "PropertyReference"" is set to "Other".*
TrimmingText	FeatStd::String	Trimming text defines the text which replaces content overflowing the content area. *This property is only available if "Trimming" is set to "CustomText".*

Align Pivot

Aligns all child nodes according to their pivot point.

Example Usage

- Align Pivot Example

Common Properties

This behavior does not have any configurable properties.

Android Event Bridge

Builds a bridge for events between CGIStudio and an Android application. Receives JsonEvents from an Android Control and transforms them to an equivalent CGI Studio event.

This behavior is only available for Android platform.

Common Properties

Property Name	Type	Description
TargetNode	Candera::AbstractNodePointer	The target node to which the events are sent. If the property is not set, the event will be routed to the node of the behavior.

Arc Align

Aligns child nodes in a circular layout. Provides properties for the radius, orientation offset and the min and max angle.

Example Usage

- Arc Align Example

Common Properties

Property Name	Type	Description
Radius	Float	Radius of the digital needle indicators.
Orientation Offset	Float	Offset in the orientation of the needle indicators.
Angle Min Max	Vector2	Min/Max angle values (e.g. indicators displayed from -25 to 210 degrees).

Blend Mode

Class Reference

Sets a blend mode from a given list of options. Depending on that option the behavior chooses the appropriate blend factors and operations. In 2D it sets the values on the effect and in 3D on the render-mode if one is attached on the same node.

The following blend modes are available:



	• Default • DefaultPremultiplied • DefaultTransparentLayer • Add • Multiply • Screen • Darken (3D only) • Lighten (3D only) • Source Atop • Source In • SourceOut • Source Over • Destination Atop • Destination In • Destination Out • Destination Over • Off

The default blend mode comes in different variations, e.g. pre-multiplied, which should be applied if the alpha channel is pre-multiplied. The other blend modes are implemented according to the Porter-Duff composition modes.
In 3D there are two additional blend modes available:

- Lighten
- Darken

Common Properties

Property Name	Type	Description
PropertyReference	CgiStudioControl::PropertyReference::Enum	Choose how the property gets set. Via own property or from another Behavior. *Possible values: Self, Other*

PropertyProvider	PropertyProviderBehavior*	Property Behavior which contains the color which has to be set on the effect. *This property is only available if PropertyReference is set to "Other".* *This property can be set in the right side of the Scene Tree Panel.*
Mode	BlendType::Enum	Sets a blend mode from a given list of options. *Possible values: Default, DefaultPremultiplied, DefaultTransparentLayer, Add, Multiply, Screen, Darken (3D only), Lighten (3D only), Source Atop, Source In, SourceOut, Source Over, Destination Atop, Destination In, Destination Out, Destination Over, Off*

Breadcrumb Behavior

A breadcrumb used as a navigational aid in user interfaces. It allows users to keep track and maintain awareness of their location.

Common Properties

Property Name	Type	Description
StateMachineNode	Candera::AbstractNodePointer	The node of the State Machine
EllipsisText	FeatStd::String	When more elements are added than the breadcrumb can show, one element will be replaced with the text from this field (collapse effect).
PathDelimiter	FeatStd::String	Tag that separates the elements that a path contains.

Disable Control

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node **gets disabled**.

Enable Control

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node **gets enabled**.

Event Bridge

Builds a bridge for events between two nodes for direct event dispatching. The events are routed from the source node to the target node (only in one direction).

Common Properties

Property Name	Type	Description
TargetNode	Candera::AbstractNodePointer	The target node to which the events are routed from the source node. If the property is not set, the event will be routed to the node of the behavior.

Focus

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node **gets focused**.

Focus Leave

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node's **focus is left**.

Hover

See also: HoverLeave

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node **gets hovered**.

Example Usage

- Hover - Hover Leave Example

HoverLeave

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node's **hovering is left**.

Example Usage

- Hover - Hover Leave Example

IntervalSwitch

Sends Off or On value to Trigger Behaviors that belong to same node.

Common Properties

Property Name	Type	Description
State	IntervalSwitchState::Enum	If set to On, sends only On value to Trigger Behaviors in the same node. If set to Off, sends only Off value to Trigger Behaviors in the same node. If set to Blinking, alternates between sending On or Off values. Use On Time Ms and Off Time Ms to specify the speed of blinking. *Possible values: On, Off, Blinking*
OnTimeMs	FeatStd::UInt32	If State is set to Blinking, IntervalSwitchBehavior will be sending On value for On Time Ms milliseconds before switching to sending Off value.
OffTimeMs	FeatStd::UInt32	If State is set to Blinking, IntervalSwitchBehavior will be sending Off value for Off Time Ms milliseconds before switching to sending On value.
AcceptValueEvent	bool	Set this flag to true to make the behavior changeable by events/actions.

Keyboard

This behavior provides the functionality of Keyboard control by managing the Keyboard Keys and setting their properties according to the Configuration Script.

This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
ConfigurationScript	FeatStd::String	String that specifies symbols and layout of the Control
KeyContent	AbstractNodePointer	Node containing Keyboard Keys.
StateMachineNode	AbstractNodePointer	State machine that reacts on events sent by a special key to change keyboard layout
KeyBackContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of KEY_BACK special key. *Possible values: Text, Icon*

KeyBackText	FeatStd::String	KEY_BACK is labeled with the text. *This property is only visible if KeyBackContentType is set to "Text".*
KeyBackImage	SharedPointer<Candera::Image2D>	KEY_BACK is labeled with the image. *This property is only visible if KeyBackContentType is set to "Icon".*
KeyEnterContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of KEY_ENTER special key. *Possible values: Text, Icon*
KeyEnterText	FeatStd::String	KEY_ENTER is labeled with the text. *This property is only visible if KeyBackContentType is set to "Text".*
KeyEnterImage	SharedPointer<Candera::Image2D>	KEY_ENTER is labeled with the image. *This property is only visible if KeyBackContentType is set to "Icon".*
KeySpaceContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of KEY_SPACE special key. *Possible values: Text, Icon*
KeySpaceText	FeatStd::String	KEY_SPACE is labeled with the text. *This property is only visible if KeyBackContentType is set to "Text".*
KeySpaceImage	SharedPointer<Candera::Image2D>	KEY_SPACE is labeled with the image. *This property is only visible if KeyBackContentType is set to "Icon".*
KeySwitchContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of SWITCH special key
KeySwitchText	FeatStd::String	SWITCH is labeled with the text. *This property is only visible if KeySwitchContentType is set to "Text".*
KeySwitchImage	SharedPointer<Candera::Image2D>	SWITCH is labeled with the image. *This property is only visible if KeySwitchContentType is set to "Icon".*
KeyShiftContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of SHIFT special key
KeyShiftText	FeatStd::String	SHIFT is labeled with the text. *This property is only visible if KeyShiftContentType is set to "Text".*
KeyShiftImage	SharedPointer<Candera::Image2D>	SHIFT is labeled with the image. *This property is only visible if KeyShiftContentType is set to "Icon".*
KeyShiftOnceContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of SHIFT ONCE special key
KeyShiftOnceText	FeatStd::String	SHIFT ONCE is labeled with the text*:* *This property is only visible if KeyShiftOnceContentType is set to "Text".*

KeyShiftOnceImage	SharedPointer<Candera::Image2D>	SHIFT ONCE is labeled with the image. *This property is only visible if KeyShiftOnceContentType is set to "icon".*
KeyShiftLockContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of SHIFT LOCK special key
KeyShiftLockText	FeatStd::String	SHIFT LOCK is labeled with the text. *This property is only visible if KeyShiftLockContentType is set to "Text".*
KeyShiftLockImage	SharedPointer<Candera::Image2D>	LEFT is labeled with the image. *This property is only visible if KeyShiftLockContentType is set to "Icon".*
KeyCloseContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of SHIFT LOCK special key
KeyCloseText	FeatStd::String	CLOSE is labeled with the text. *This property is only visible if KeyCloseContentType is set to "Text".*
KeyCloseImage	SharedPointer<Candera::Image2D>	CLOSE is labeled with the image. *This property is only visible if KeyCloseContentType is set to "Icon".*
KeyLeftContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of LEFT special key
KeyLeftText	FeatStd::String	LEFT is labeled with the text. *This property is only visible if KeyLeftContentType is set to "Text".*
KeyLeftImage	SharedPointer<Candera::Image2D>	LEFT is labeled with the image. *This property is only visible if KeyLeftContentType is set to "Icon".*
KeyRightContentType	CgiStudioControl::KeyContent::Enum	Switch for Content of RIGHT special key
KeyRightText	FeatStd::String	RIGHT is labeled with the text. *This property is only visible if KeyRightContentType is set to "Text".*
KeyRightImage	SharedPointer<Candera::Image2D>	RIGHT is labeled with the image. *This property is only visible if KeyRightContentType is set to "Icon".*
NavigateUpKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that moves focus to a key above the current focused
NavigateDownKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that moves focus to a key below the current focused
NavigateNextKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that moves focus to a key on the right from the current focused

NavigatePreviousKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that moves focus to a key on the left from the current focused
NavigateStartOnDown	bool	Enables starting navigation with the same key as NavigateDownKeyCode
NavigateEndOnUp	bool	Enables finishing navigation with the same key as NavigateUpKeyCode
NavigateStartKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that starts navigation within the Control
NavigateEndKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that finishes navigation within the Control
NavigateCloseKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that hides the Control
NavigateInvokeKeyCode	Candera::KeyboardKeyCode::Enum	Key from a physical keyboard that emulates clicking the current focused Key from the Control

Configuration String

The most important feature of the Keyboard Control is Configuration Script. It is parsed during layouting a scene which a keyboard is embedded on and keys (their appearance and their special functionalities) are arranged according to it.

The configuration script is simply a string. Each line of the string specifies a row of keys in the instance of keyboard and consists of space-separated list of key strings. A key string specifies a value that will be sent to a connected InputField. The most typical key string is simply a character or figure but it is not limited to it.

Following key strings have a special meaning:

Key String	Meaning
enum from Candera::KeyboardKeyCode::Enum: KEY_BACK, KEY_SPACE, KEY_ENTER	For most keys KEY_* has the same meaning as * (e.g. „KEY_A“ is the same as simply „A“); it enables keys with non-printable characters like backspace etc.
CLOSE()	Represents a key that sends Close KeyboardAction to a state machine; it is designed to enable hiding (setting EnableRendering to false) an instance of the Control
SWITCH(layoutName)	Represents a key that sends ¬¬to a state machine an event with its argument; it is designed to enable permanent changing layout e.g. to uppercase letters, symbols or digits

SHIFT(layoutName)	Mostly the same as SWITCH() but is separated to enable changing layout only for a single input, it means after entering a single letter (e.g. first capital letter in a sentence) the layout will be reset to the previous one
NULL()	Represents a key that will NOT be rendered; it is designed for a layouting/padding purpose

Keyboard Control has public properties that customize labels of keys with the special meaning described above. One can set them custom text or image label.

Each key string (both regular and special) can be extended by a scaling factor in square brackets, e.g. SHIFT(Uppercase/en/US)[2] defines a key that sends to a state machine event with “Uppercase/en/US” string and the key itself is counted “2” times (like two different keys) when counting default width of a key in its row.

It is not obligatory to connect the Control to a state machine. There is a default set of layouts with lower- and uppercase english letters as well as digits and symbols with full switching support without a state machine.

Menu Item

Class Reference

Menu Item Behavior defines the attached Node as an Menu Item. A Menu Item can interact with the Menu Screen Behavior. Sends Off or On value to Trigger Behaviors that belong to same node.

This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
DeleteButtonNode	Candera::AbstractNodePointer	Delete button node needed to control visibility (render on/off).
Focusable	bool	If true then focus node will be visible.
FocusNode	Candera::AbstractNodePointer	Focus node needed to control visibility (render on/off) and change focus image.
Enabled	bool	If true then item is visible.
FocusImageOnSelection	SharedPointer<Candera::Image2D>	The image to set on the effect.
FocusImageOnDragging	SharedPointer<Candera::Image2D>	The image to set on the effect.

MenuScreen

Class Reference

The Control represents a menu in the form of a grid of icons (instances of the Menu Item Control). It has configurable number of rows and columns as well as default and maximum number of screens. A user can swipe between neighbouring screens. The Control handles user input and forwards proper information to children MenuItem nodes. The children nodes may be removed and reordered in runtime.

This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
ActiveScreenIndex	UInt32	Currently active index.
NumberOfScreens	UInt32	Number of active screens.
LimitOfScreens	UInt32	Limit of active screens.
Rows	UInt32	Number of rows.
Columns	UInt32	Number of columns.
ScreenChangeTime	Float	Time of dragging a Menu Item outside boundaries of the Menu Screen to change its screen to a neighbour.
SwapItemAnimationEasingFunction	EasingFunctions::Enum	Swap item animation easing function. *Possible values: CircularInOut, Linear, SineIn, SineOut, SineInOut*
SwapItemAnimationLength	Float	Swap item animation length
ChangeScreenAnimationEasingFunction	EasingFunctions::Enum	Change screen animation easing function. *Possible values: CircularInOut, Linear, SineIn, SineOut, SineInOut*

ChangeScreenAnimationLength	Float	Change screen animation length.
KeyCodeToMoveSelectionLeft	KeyboardKeyCode::Enum	Key code to move selection left. *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*
KeyCodeToMoveSelectionRight	KeyboardKeyCode::Enum	Key code to move selection right. *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*
KeyCodeToMoveSelectionUp	KeyboardKeyCode::Enum	Key code to move selection up. *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*
KeyCodeToMoveSelectionDown	KeyboardKeyCode::Enum	Key code to move selection down. *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*
PageIndicatorNode	Candera::AbstractNodePointer	Page indicator node.
Items	ListPropertyType<::Courier::DataContextItem::SharedPointer>	The items property is used for binding of dynamic items. It can only be set via databinding. Only one visible list control should be bound to a fragment list in the model. Otherwise the list fragments in the model are not able to provide data for both list controls. Use a separate list model that is filled with the same data if such a use case is required.
ContentNode	Candera::AbstractNodePointer	Content node.

Navigation Instruction View

Behavior for displaying the instructions during navigation.

Common Properties

Property Name	Type	Description
SignDisplayNode	Candera::AbstractNodePointer	Node used to display the graphical sign.
PrimaryTextNode	Candera::AbstractNodePointer	Node used to display the primary message.
SecondaryTextNode	Candera::AbstractNodePointer	Node used to display the secondary message.

ManeuverTypeTextNode	Candera::AbstractNodePointer	Node used to display the maneuver type (as text).

OnClick

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node **is clicked**.

OnClick behavior allows to check for single, double or tripple (or any) Click events.

Note:
Click is a synthetic event composed by combining two basic events: Press and Release.

Example Usage

- Enable Rendering Example
- Move Node Example
- OnClick Example
- Set Position Example
- Set Rotation Example
- Start Timer Example
- Perform Transition Requests Example
- Switch Image Example

OnPressed

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A set of actions can be configured, which will be executed when the node **is pressed**.

OnPressed behavior allows to check for Pressed and Released (and LongPressed) events.

Example Usage

- OnPressed Example

OnTimer

This behavior is a Behavior Block. It can be dragged&dropped on nodes just like any Behavior.

A timer node and a set of actions can be configured. These actions will be executed when the **timer expires**.

See also: Check Timer, Start Timer

This behavior is a Behavior Block.

Example Usage

- Start Timer Example

Orbit Camera Transform

When attached to a Node, the specified camera will orbit around that Node. PLAYER USE ONLY.

And the options to limit the movement of the camera to specific parts of the spherical orbit. Setting the flag IsOrbitLimited allows the user to set minimum and maximum angles between the camera and the axes of the coordinate system, preventing the camera from moving too close to an axis. In this way, the user can define unwanted areas that the camera cannot move to, e.g., below the ground.

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id
Camera	CameraType	The camera that looks at the associated node.

Advanced Properties

Property Name	Type	Description
IsOrbitLimited	bool	Enables setting of boundaries for the camera within the orbit.
AngleMinX	FeatStd::Float	Min angle between camera and X-axis (from 0 to 180 degrees). Must be less than AngleMaxX.
AngleMaxX	FeatStd::Float	Max angle between camera and X-axis (from 0 to 180 degrees). Must be greater than AngleMinX.
AngleMinY	FeatStd::Float	Min angle between camera and Y-axis (from 0 to 180 degrees). Must be less than AngleMaxY.

AngleMaxY	FeatStd::Float	Max angle between camera and Y-axis (from 0 to 180 degrees). Must be greater than AngleMinY.
AngleMinZ	FeatStd::Float	Min angle between camera and Z-axis (from 0 to 180 degrees). Must be less than AngleMaxZ.
AngleMaxZ	FeatStd::Float	Max angle between camera and Z-axis (from 0 to 180 degrees). Must be greater than AngleMinZ.

Page Indicator

Class Reference

Send events to a PageIndicator behavior to update its state. Can change the number of pages, set the active page, or activate a specific dot via external controls or PageIndicatorElement. This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
TargetNode	Candera::AbstractNodePointer	Target node that will receive a ChangeValueEvent when ActivePageIndex changes
ActivePageIndex	FeatStd::Int32	Index of active page (range from 0 to NumberOfPages - 1)
NumberOfPages	FeatStd::Int32	Total number of pages (you have to ensure that PageIndicator has at least this number of child elements)
DecrementKeyCode	Candera::KeyboardKeyCode::Enum	Key used for decrementing value (if value is actually decremented, focus handling is suspended) *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*
IncrementKeyCode	Candera::KeyboardKeyCode::Enum	Key used for incrementing value (if value is actually incremented, focus handling is suspended) *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*

Rectangle Clipping Behavior

Clips the control to given rectangle.

Common Properties

Property Name	Type	Description
Width	Float	Width of the clipping rectangle.

Height	Float	Height of the clipping rectangle.
ClipPositionEnabled	bool	Enable Clipping Position.
ClipPosition	Candera::Vector2	Position of the clipping rectangle. *This property is only available if ClipPositionEnabled is set to true.*

Roll

Class Reference

Displays a list of float values within a specified range, and lets the user select a value by rolling with touch or scrolling via configured arrow keys. This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
ValueGroup	Candera::AbstractNodePointer	Node containing a range of displayed values
LensGroup	Candera::AbstractNodePointer	Node containing current value. *This property is only available if 'LensVisible' is set to 'true'.*
DecorationGroup	Candera::AbstractNodePointer	Node containing lens decoration. *This property is only available if 'LensVisible' is set to 'true'.*
Value	FeatStd::Float	Current value
Min	FeatStd::Float	Minimum value
Max	FeatStd::Float	Maximum value
LensVisible	bool	Is lens group visible?
UseCyclicRoll	bool	Is roll cyclic?
AllowDraggingPastLimit	bool	Allow temporary over-dragging (dragging to blank values before first and after last entry) *This property is only available if 'UseCyclicRoll' is set to 'false'.*
DecrementKeyCode	Candera::KeyboardKeyCode::Enum	Key used for decrementing value (if value is actually decremented, focus handling is suspended) *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*

IncrementKeyCode	Candera::KeyboardKeyCode::Enum	Key used for incrementing value (if value is actually incremented, focus handling is suspended) *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8*
TextEditNode	Candera::AbstractNodePointer	Node containing text edit field
Step	FeatStd::Float	Value step (positive float value).

Advanced Properties

Property Name	Type	Description
Format	FeatStd::String	Value display format (use '%f' for value) *This property is only available if 'LensVisible' is set to 'false'.*
EditKeyCode	Candera::KeyboardKeyCode::Key_Enter	Key used for entering text edit mode *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8* *This property is only available if 'LensVisible' is set to 'true' and a 'TextEditNode' is configured.*

Touch Indication

Used to display a touch indicator at the position where the screen is touched. Control must be positioned at position (0;0) if no offset to the touch location is desired.

Common Properties

Property Name	Type	Description
TouchableId	String	Touchable Id
IndicationNode	Candera::AbstractNodePointer	The node moved by the Touch Indication Behavior. It is moved to the touch location to point it out.
TouchAreaNode	Candera::AbstractNodePointer	The node whose area the Touch Indication Behavior should indicate touch events on.
Camera	Candera::AbstractNodePointer	The camera used to render the touch indicator. Will be used to correctly match touch position with object position in case the camera has a position offset.
DistanceFromCamera	FeatStd::Float	How far from the camera should the touch indication node be rendered. *This property is only available for 3D scenes.*

OnAnimation	Candera::MemoryManagement::SharedPointer< Candera::Animation::AnimationPlayer>	Optional animation played when the indicator is shown.
OffAnimation	Candera::MemoryManagement::SharedPointer< Candera::Animation::AnimationPlayer>	Optional animation played when the indicator is hidden.
HideIndicatorOnMove	bool	Hide the touch indicator when a move is detected, otherwise the indicator will stay visible during movement until the finger is lifted from the screen.

Transform Mesh2D

Class Reference

Performs a perspective projection of a 3D rotated effect (e.g. Bitmap or SolidColor effects) based on the configured origin, rotation and field of view.

This behavior is available for 2D scenes only.

Common Properties

Property Name	Type	Description
Rotation	FeatStd::Float	Defines the rotation angle in degree around configured rotation axis at the configured origin
CameraDistance	FeatStd::Float	Defines a normalized camera distance (the projection is based on a normalized geometry of widht and height 1) with a default value of 1.
FitToMaximumSizeRotation	FeatStd::Float	Defines the rotation angle in degree that will be uses as maximum for the FitToMaximumSize fit strategy. Default is an angle of 90 degree.
RotationAxis	Candera::Vector3	Defines the axis (default is the y axis) for the rotation as a 3D direction vector (the vector will be internally normalized).
FieldOfView	FeatStd::Float	The area on the Y axis view field in degrees. It influences the perspective distortion of the projection. Higher angles will result in more distorted projection. Lower angles will result in less perspective distorted projections. The angle has to be higher than 0 and less than 180.
FitStrategy	TransformMesh2DFitStrategy::Enum	Defines how the perspective projected mesh should fit into the original boundaries.

HorizontalOrigin	CgiStudioControl::HorizontalOrigin::Enum	The origin influences where the rotation is performed and how the perspective projection looks. It is defined as a normalized (bitmap resolution independent) origin on the y axis as Left (0), Center (0.5), Right (1.0) or Custom (see CustomHorizontalOrigin).
VerticalOrigin	CgiStudioControl::VerticalOrigin::Enum	The origin influences where the rotation is performed and how the perspective projection looks. It is defined as a normalized (bitmap resolution independent) origin on the y axis as Top (0), Center (0.5), Bottom (1.0) or Custom (see CustomVerticalOrigin).
CustomHorizontalOrigin	FeatStd::Float	Defines a normalized (bitmap resolution independent) custom origin that can not be defined with Left, Center or Right.
CustomVerticalOrigin	FeatStd::Float	Defines a normalized (bitmap resolution independent) custom origin that can not be defined with Top, Center or Bottom.
OriginOffset	Candera::Vector2	In addition to the configured normalized origin this pixel based origin offset is added.

Event Handler

Passes a value to a Condition Behavior and if the response is positive, invokes an Action behavior.

Example Usage

- Render Child Nodes Example

Common Properties

Property Name	Type	Description
Condition	Candera::ConditionBehavior *	The ConditionBehavior that TriggerBehavior passes a value to and expect a response from to see if Action needs to be invoked.
Actions	ActionBehaviorArrayProperty	The set of ActionBehaviors that TriggerBehavior invokes in the even that it receives a positive response from Condition.
ConsumeEvent	bool	Set this flag to false if the event should not be consumed by the trigger. By default it is set to true and if the trigger evaluates to true the event will not be routed to any other behavior.

Virtual Key

Receives a Change Value Event with a boolean. True will start emitting Key Events, false will stop emitting Key Events.

Common Properties

Property Name	Type	Description
KeyType	CgiStudioControl::KeyType::Enum	Choose the type of key. With KeyCode you can select any key. *Possible values: Modifier, KeyCode*
KeyCode	Candera::KeyboardKeyCode::Enum	With KeyCode you can select any key. *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8* *This property is only available if "KeyType" is set to "KeyCode".*
Modifier	Candera::KeyModifierEnum::Enum	Modifier Key (e.g. Shift) *Possible values: None, Key_Shift, Key_Alt, Key_Ctrl, Key_CapsLock, Key_Num* *This property is only available if "KeyType" is set to "Modifier".*

Revision #37
Created 2023-02-27 10:21:54 UTC by Kanai Tomoaki
Updated 2025-12-26 06:52:11 UTC by Tsuyoshi.Kato