Condition Behaviors

Control Condition Behaviors

Check Animation State

Checks the current state of an animation (for example, play, paused, or direction changed) from the Configure and Play Animation behavior and triggers an action behavior when the condition is met.

Common Properties

Property Name	Type	Description
AnimationState	AnimationState::Enum	State of the AnimationEvent which fulfills the Condition. *Possible values: PastEnd, Start, Stop, Finish, Resume, Pause, ApplyChanges, DirectionChange*
Sender	SharedPointer<Candera::Animation::AnimationPlayer>	The sender of AnimationEvent.

> When this condition is used in a global state machine, the event dispatch strategy of 'Configure and Play Animation Behavior' must be set to bubble or tunnel.

Check Enter or Exit

Expects an EnterExit Event (generated by Edit Text Behavior) and returns either a positive or negative response.

Common Properties

Property Name	Type	Description
EnterOrExit	bool	True means Enter, false means Exit.

Check Interval Switch

Checks if the visualization state from the IntervalSwitch behavior matches the configured state (On or Off). Triggers an action when the condition is met. Can be used to check the state of a IntervalSwitch behavior.

Common Properties

Property Name	Type	Description
VisualizationState	IntervalSwitchState::Enum	If the value (received by the IntervalSwitchConditionBehavior) matches the Visualization State set, it will return a positive response. *Possible values: On, Off, Blinking*

Check Selection ID

Compares the ID from the ChangeSelectionEvent with the ID from its own Control.

Common Properties

Property Name	Type	Description
SelectionCondition	SelectionConditionEnum::Enum	The condition that the ID from the ChangeSelectionEvent has to fulfill. *Possible values: Same, NotSame*

Check Control State

Checks a specific state event generated by the ControlState behavior (pressed, enabled, etc.) in conjunction with the configured state and condition (e.g. pressed and is true). If the condition is met, an action behavior is triggered.

Common Properties

Property Name	Type	Description
State	ControlStateEnum::Enum	State which has to be checked. *Possible values: Selected, Pressed, Enabled, Highlighted, KeyboardFocused, ScopeFocused, Hover, Touched*
StateCondition	StateConditionEnum::Enum	The condition the state has to fulfill. *Possible values: ChangedToTrue, ChangedToFalse, IsTrue, IsFalse*

Check Timer

See also: Start Timer, OnTimer

Checks the time (in milliseconds) received from the StartTimer behavior. If the timer reaches the configured delay, it invokes an Action behavior. Can be used as a transition condition in a state machine.

Common Properties

Property Name	Type	Description
Sender	AbstractNodePointer	The node from which the event has to be emitted. If it is not set then all events of the proper event type are handled.

Check Changed Value

Evaluates the configured value condition. To be used within a trigger behavior to trigger a certain action if the value matches the condition.

Common Properties

Property Name	Type	Description
Sender	AbstractNodePointer	The node from which the event has to be emitted. If it is not set then all events of the proper event type are handled.
ConditionOperator	Operators::Enum	The condition operator. *Possible values: GreaterThan, LowerThan, Equal, AlmostEqual, GreaterOrEqual, LowerOrEqual*
ConditionValue	FeatStd::Variant	The condition value.

Check Processed Value

Checks whether a Value meets certain condition.

The following values can take part in the comparison, depending on the Operation Mode selected:

- Value sent in Process Value
- External value from ValueBehavior
- Value set directly in this ValueConditionBehavior

Example Usage

- Thermometer Example

Common Properties

Property Name	Type	Description
OperationMode	Operators::Enum	Operation mode of behavior, defining which values should be compared. *Possible values: CompareProcessedValueWithConditionValue, CompareExternalValueWithConditionValue, CompareProcessedValueWithExternalValue*
ConditionOperator	Operators::Enum	The condition operator. *Possible values: GreaterThan, LowerThan, Equal, AlmostEqual, GreaterOrEqual, LowerOrEqual*
ConditionValue	FeatStd::Variant	The condition value. *This property is only available if "OperationMode" is set to "CompareProcessedValueWithConditionValue" or "CompareExternalValueWithConditionValue".*
ValueBehavior	CgiStudioControl::ValueBehavior *	Value behavior to be checked for changes if OperationMode requires external value. *This property is only available if "OperationMode" is set to "CompareExternalValueWithConditionValue" or "CompareProcessedValueWithExternalValue".* *This property can be set in the right side of the Scene Tree Panel.*

Check Video State

Checks the state of an VideoStateEvent received from an 'Video Stream Behavior'.

Expects a VideoState event (generated by Video Stream Behavior) and evaluates the configured state condition. To be used within a trigger behavior.

Common Properties

Property Name	Type	Description
VideoState	VideoState::Enum	State of the VideoEvent which fulfills the Condition. *Possible values: Null, Ready, Playing, Paused, EndOfStream*

Input Condition Behaviors

Check Click

Evaluates the configured click event condition. To be used within a trigger behavior to trigger a certain action.

Example Usage

- Render Child Nodes Example

Common Properties

Property Name	Type	Description
Sender	AbstractNodePointer	The node from which the event has to be emitted. If it is not set then all events of the proper event type are handled.
ClickCount	CgiStudioControl::ClickCount::Enum	If the value ClickConditionBehavior receives matches the Click Count set, it will return a positive response. *Possible values: AnyClick, SingleClick, DoubleClick, TripleClick*
TouchId	FeatStd::UInt	The id of the finger that performed the click (the default value of 0 indicates the first finger, 1 the second ...).

Check Drag Drop

Evaluates the configured DragDrop event condition. To be used within a trigger behavior to trigger a certain action.

Common Properties

Property Name	Type	Description
Sender	AbstractNodePointer	The node from which the event has to be emitted. If it is not set then all events of the proper event type are handled.
Type	FeatStd::Optional <DragDropEventType::Enum>	The optional type of the event. If not provided then any DragDropEvent will be evaluated to true. *Possible values: Drag, DragMove, Drop, DragEnter, DragLeave*
TouchId	FeatStd::UInt	The id of the finger that performed the drag (the default value of 0 indicates the first finger, 1 the second ...).

Check Hover

Evaluates the configured Hover event condition. To be used within a trigger behavior to trigger a certain action.

Common Properties

Property Name	Type	Description
Sender	AbstractNodePointer	The node from which the event has to be emitted. If it is not set then all events of the proper event type are handled.
Type	FeatStd::Optional <DragDropEventType::Enum>	The optional type of the event. If not provided then any DragDropEvent will be evaluated to true. *Possible values: Hover, HoverEnter, HoverLeave*
TouchId	FeatStd::UInt	The id of the finger that performed the drag (the default value of 0 indicates the first finger, 1 the second ...).

Check Key Input

Checks a received KeyEvent (Control or Node with ControlStateBehavior / 'Handle Control State' must have focus).

Common Properties

Property Name	Type	Description
KeyType	CgiStudioControl::KeyType::KeyCode	Choose type of key. *Possible values: KeyCode, Modifier*
KeyCode	Candera::KeyboardKeyCode::Enum	Check the KeyCode of the pressed key (can also be modifier). *Possible values: Key_None, ...<all keyboard key codes>... Key_OEM8* *This property is only available if "KeyType" is set to "KeyCode".*

Advanced Properties

Property Name	Type	Description
Modifier	Candera::KeyModifierEnum::Enum	Check a modifier key. *Possible values: None, Key_Shift, Key_Alt, Key_Ctrl, Key_CapsLock, Key_Num* *This property is only available if "KeyType" is set to "Modifier".*

Check Pressed

Evaluates the configured Pressed event condition. To be used within a trigger behavior to trigger a certain action.

Common Properties

Property Name	Type	Description
Sender	AbstractNodePointer	The node from which the event has to be emitted. If it is not set then all events of the proper event type are handled.
Type	FeatStd::Optional <DragDropEventType::Enum>	The optional type of the event. If not provided then any DragDropEvent will be evaluated to true. *Possible values: Pressed, Released, LongPressed*
TouchId	FeatStd::UInt	The id of the finger that performed the drag (the default value of 0 indicates the first finger, 1 the second ...).

Swipe Gesture

The swipe condition checks if a swipe/flick has been performed according to the configured settings.

Configuration

The type of swipe event and the direction can be configured. By default, it is configured to the end of the swipe (removing the finger from the screen) in any direction. Other configurations are:

Begin

The finger has touched the control and moved over the threshold.

- The threshold in pixel is independent of the unit size of the control local coordinate system. You do not have to take care of that transformation.
- The default threshold is set to 10 pixel.
- The threshold can be changed by calling this method: `CgiStudioControl::SwipeGestureDetector::GetInstance().SetSwipeDistanceThreshold(…)`.

Swiping

The swipe is in progress and the finger is moving over the screen.

Flick

In addition to the Swipe End a Swipe Flick will be sent if the velocity at the Swipe End exceeds the flick velocity threshold.

- The threshold is in pixel per second independent of the unit size of the control local coordinate system. You do not have to take care of that transformation.

- The default threshold is set to 0,01 pixel per second.
- The threshold can be changed by calling this method `CgiStudioControl::SwipeGestureDetector::GetInstance().SetFlickVelocityThreshold(…)`.

Abort

An external actor has aborted the swipe. The finger is still on the screen. But the gesture detector will no longer process the swipe gesture. This can be caused for example when a behavior is taking exclusive control over the touch event handling. Like the list behavior does it when it enters touch based scrolling mode.

To limit the condition to a certain direction just enable the direction. The default will be set to left (e.g. a next gesture in a menu screen or delete gesture in a message list). You can change it to Right, Up or Down. The direction is always in the local coordinate system of your control (e.g. if you have to implement a UI for a rotated display you will still receive the swipe up event even if it is swipe right in the display coordinate system).

Common Properties

Property Name	Type	Description
Type	FeatStd::Optional <SwipeType::Enum>	The type of the swipe gesture. *Possible values: Begin, Swiping, End, Flick, Abort*
Direction	FeatStd::Optional <SwipeDirection::Enum>	The direction of the swipe. *Possible values: Left, Right, Up, Down*

Transform Gesture

Use this condition to trigger actions in an event handler whenever a transform gesture has been detected (pan, pinch and rotate).

Common Properties

This behavior does not have any configurable properties.

Logic Condition Behaviors

Check Logic Condition

The logic condition combines multiple conditions with the configured logical operation.

Common Properties

Property Name	Type	Description
Operation	LogicConditionBehaviorOperator::Enum	Specifies logical operator that is used on values returned by the Conditions. The resulting value is returned. *Possible values: And, Or, XOr*
Conditions	Candera::ConditionBehaviorArrayProperty	ConditionBehaviors that are combined with the ConditionOperator.

Check Not Condition

Acts as a 'NOT' operand by inverting a value returned by a ConditionBehavior and returns it.

Common Properties

Property Name	Type	Description
Condition	Candera::ConditionBehavior*	ConditionBehavior whose value NotConditionBehavior inverts.

Scroll Condition Behaviors

Check Scroll Condition

Evaluates the configured ScrollableDirectionChanged event condition. The event gets sent from Scroll View Behavior. To be used within a trigger behavior to trigger a certain action.

Common Properties

Property Name	Type	Description
ScrollDirection	ScrollDirection::Enum	If the value that the ScrollDirectionConditionBehavior receives matches the set ScrollDirection, it will return a positive response. *Possible values: Vertical, Horizontal*

Check Culture Change

Class Reference

Checks if the culture has changed. This condition is satisfied when the culture matches the culture ID specified in the Local property, and an action behavior is subsequently invoked. Can be used to

change the text culture with a button press.

Common Properties

Property Name	Type	Description
Locale	FeatStd::String	Locale of a culture which will be checked.

Revision #24
Created 2023-02-27 08:53:18 UTC by Kanai Tomoaki
Updated 2025-12-26 06:44:34 UTC by Tsuyoshi.Kato