

Control Behaviors 3.7

To keep backwards compatibility for any interface change in the public API the old interface has been kept and marked as deprecated. The deprecated interfaces will be removed in future versions. Compiler warnings will be generated when using a deprecated interface through the usage of Candra Macro `CANDERA_DEPRECATED_3_7_0`.

- [Behaviors CMake Build System Changes](#)
- [Behaviors Changes](#)

Behaviors CMake Build System Changes

Added flags:

Name	Description	Possible values
CONTROLS_HOVER_EVENT_ENABLED	Activate hover events from Control Touch Session. An explicit call to ControlTouchSession::SetHoverEnable <u>d</u> is not required in application.	ON/OFF

Behaviors Changes

Converted 2D/3D behaviors to "mixed" behaviors

Several behaviors had a dedicated 2D and 3D variant. Such behaviors have been cleaned up and replaced by a mixed behavior providing a 2D or 3D implementation based on the node attached to. The 2D and 3D specific behaviors have been deprecated, a solution conversion will automatically convert existing Scene Composer solutions.

Affected behaviors:

- ArcAlignBehaviorBase2D deprecated, replaced by [ArcAlignBehavior](#)
- TouchIndicationBehavior2D and TouchIndicationBehavior3D deprecated, replaced by [TouchIndicationBehavior](#)
- PositionProcessValueBehavior2D and PositionProcessValueBehavior3D deprecated, replaced by PositionProcessValueBehavior
 - property TranslationAxis has been renamed to TranslationAxis2D, a new property TranslationAxis3D has been introduced
- RotateProcessValueBehavior2D and RotateProcessValueBehavior3D deprecated, replaced by RotateProcessValueBehavior
 - property RotationAxis added
- ScaleProcessValueBehavior2D deprecated, replaced by ScaleProcessValueBehavior
 - property IsScalingAlongXEnabled has been renamed to ScaleAlongX
 - property IsScalingAlongYEnabled has been renamed to ScaleAlongY
 - new property ScaleAlongZ added
 - property VectorXY has been renamed to ScaleVector2D
 - new property ScaleVector3D added
- TranslateProcessValueBehavior2D and TranslateProcessValueBehavior3D deprecated, replaced by TranslateProcessValueBehavior
 - property VectorXY (Vector2) has been replaced by MultipliedVector (Vector3)
- TrackPositionProcessValueBehavior2D and TrackPositionProcessValueBehavior3D deprecated, replaced by [TrackPositionProcessValueBehavior](#)
- AlphaProcessValueBehavior2D and AlphaProcessValueBehavior3D deprecated, replaced by [AlphaProcessValueBehavior](#)
- ColorActionBehavior2D deprecated, replaced by [ColorActionBehavior](#)
- EffectColorBehavior2D deprecated, replaced by [EffectColorBehavior](#)
- ImageActionBehavior2D deprecated, replaced by [ImageActionBehavior](#)
 - property Texture has been added
- SetImageBehavior2D deprecated, replaced by [SetImageBehavior](#)
 - new property Textures added

- TextBehavior2D deprecated, replaced [TextBehavior](#)
- ZoomProcessValueBehavior2D deprecated, replaced by [ZoomProcessValueBehavior](#).
- ScrollbarBehavior2D and ScrollbarBehavior3D deprecated, replaced by [ScrollbarBehavior](#)
- SwitchActionBehavior2D deprecated, replaced by [SwitchActionBehavior](#)
- SwitchImageProcessValueBehavior2D deprecated, replaced by [SwitchImageProcessValueBehavior](#)
 - property RenderNode has been renamed to TargetNode
 - new property Textures added
- SwitchRangeProcessValueBehavior2D deprecated, replaced by [SwitchRangeProcessValueBehavior](#)

Further changes:

- Constructor ClippingSetterTraverser(const Candra::Rectangle& clippingArea, Candra::Node2D& node) has been deprecated
 - Please use ClippingSetterTraverser(const Candra::Rectangle& clippingArea, Candra::AbstractNodePointer& node) instead
 - [PropertyProviderBehaviorBase](#): properties Texture, Appearance, Value have been added
-

ControlTouchSession considers Camera for sorting

In the past the camera was not considered by the touch handling. This resulted in ambiguous touched controls.

The [ControlTouchSession](#) has to sort behaviors according their "visual" order to assure, that touch inputs are forwarded to the "nearest" touchable behaviors. For this purpose a sub-struct [ControlTouchSession::TouchedBehavior](#) has been introduced. The camera will be used to retrieve the according Candra::RenderTarget. Consequently HW layer order, SW Layer order, surface id and camera sequence number are considered during sorting of behaviors.

Each behavior-camera pair will be treated separately, thus a behavior rendered by two cameras will be distinguished for each camera.

Beside the internal handling in the [ControlTouchSession](#) also the touch related events (TouchEvent, ClickEvent, PressedEvent, DragDropEvent and HoverEvent) have been extended by a touched camera. The old constructors have been deprecated. Please use the one with the additional camera instead.

For the same reason also [KeyboardFocusManager](#) has been enhanced by new methods [KeyboardFocusManager::SetFocused\(const Candra::AbstractNodePointer& node, const](#)

[Candera::AbstractCameraPointer& camera\)](#) and [KeyboardFocusManager::GetFocusedCamera\(\)](#).

Please also use the SetFocused with the extra camera parameter in future.

Implementation of geometric and tab order focus navigation

A new event class [KeyboardFocusNavigationEvent](#) has been added to reflect navigation input like a rotary input to navigate between the visual controls. Furthermore the controls will use the new RegisterNavigation and DeregisterNavigation interface of the [KeyboardFocusManager](#) to register themselves for the navigation.

An interface has been added to extend the ControlTouchSession by gesture detectors

The detection of Click, press/release and drag&drop has been transformed into such a gesture detector. Furthermore, a gesture detector for swipe/flick and a gesture detector for transformations (rotate, pan and pinch) has been added.

Further minor changes

- The return value of EventToMessageBridge::PostAsMessage has been changed to an enum type to reflect the exact outcome of the Post and to reduce the logging to errors only.
 - A Getter/Setter interface has been added to the [ControlTouchSession](#) to enable/disable the consumption of touch events by the [ControlTouchSession](#). If the touch event is not consumed the [Courier](#) Touch Session will forward the touch messages to all widgets in the old touch handling style as message.
-

Property Provider supports Value

- Property Provider ([PropertyProviderBehavior](#)) supports Value variant type
 - Send Value As Event has now PropertyReference that can be set to Other and point to Property Provider
-

SelectionGroup memory leak patched

- Patched SelectionGroup memory leak when creating new [ChangeSelectionEvent](#)
-

Arithmetic operation behavior created

- Arithmetic operation behavior ([ArithmeticProcessValueBehavior](#)) can be placed in ProcessValue chain to perform simple arithmetic operations on provided values.
 - It allows to perform addition, subtraction, multiplication and division operations on the passed value.
-

Send ChangeValueEvent on ProcessValueEvent behavior created

- <Send [ChangeValueEvent](#) on ProcessValueEvent> ([SendChangeValueOnProcessValueEventBehavior](#)) can be used as a termination of ProcessValue chain to invoke value change on connected control
 - This Behavior sends a [ChangeValueEvent](#) which is meant for changing the value of an Value Behavior.
 - The value can be absolute or relative (value will be added).
-

ValueBehavior supports broadcast feature

- If enabled, [ValueBehavior](#) can now be used to forward input value even if it is the same as current value stored in the behavior, which is useful in broadcasting context
-

Provide Alignment adds Alignment property to the control

- Provide Alignment can be used to add a controllable Alignment property to the control
 - [AlignmentBehavior](#) enables setting the horizontal and vertical alignment of a node.
-

Tracks angle on touch with regard to control's base point

- Track angle behavior allows to obtain the angle (from the position of user interaction) with regard to control's base point, using the mathematical function atan2.
 - [TrackAngleProcessValueBehavior](#) listens to the touch of the selected node, returns the angle between pressing, moving and releasing.
-

DialBehavior supports Dial control

- [DialBehavior](#) uses the angle obtained to calculate the rotation of the knob and change the position of the twisted knob to the value which is passed by ProcessValue function and [ValueChangedEvent](#).
 - [DialBehavior](#) receives angle in ProcessValueEvent or value in [ChangeValueEvent](#). It changes to appropriate rotation for knob. Transmits the changed value through events ([ValueChangedEvent](#) and ProcessValueEvent).
-

CircularSlider behavior supports CircularSlider control

- [CircularSliderBehavior](#) draws a blending mask and rotates attached thumb.
 - Creates respective effects on runtime, according to [Candera](#) build options.
-

BreadcrumbBehavior supports breadcrumb control

- A breadcrumb is used as a navigational aid in user interfaces.
 - It allows users to keep track and maintain awareness of their location.
 - Child nodes like Text, TextValue, TextButton or other nodes can react on [TextEvent](#) which is sent to child nodes.
 - Also [BreadcrumbBehavior](#) sends [ValueChangedEvent](#) to the state machine with the last value that has been received.
 - [BreadcrumbActionBehavior](#) allows you to control the breadcrumb.
-

Provide Appearance adds Appearance property to the control

- Provide Appearance can be used to add a controllable 3D Appearance property to the control
 - Also extends Property Provider by Appearance.
-

Control State Behavior properly handles Disabled State

Previously it was possible to still interact disabled Control via Inputs. In 3.7 this is not happening anymore (e.g. a disabled button cannot be pressed).

Interpolate Value can be set to Instant Mode

Using the [ChangeInterpolationModeEvent](#) in a Behavior allows to control which Mode the Interpolate Value will operate.

Check Processed Value got extended by Almost Equal

Since comparing exact Float Values can lead to issues, the Check Processed Value Behavior now also supports checking if they are almost equal.

TrackPositionProcessValue can be set to only react on dragging the Knob

Normally the TrackPositionProcessValue would also react to Inputs outside of the Knob Node. By enabling the Drag On Knob Only property, this can be prevented.

Format String now supports all Variant Types

Format String did only support Integers and Float Values previously. This is now resolved.

DrawerActionBehavior / "Change Drawer State"

The [DrawerActionBehavior](#) has been created to support the new Drawer Control.

ParticleEmitterBehavior / "Particle Emitter"

The ParticleEmitterBehavior has been created to support the new Point Sprite Emitter Control.

Change Animation Mode by Event

The Animation Mode of [InterpolateProcessValueBehavior](#) can be changed by event now.

Input Range for MapProcessValueBehavior

The [MapProcessValueBehavior](#) has been enhanced by an input range for use cases where parts of the input range of the ProcessValueEvent shall be processed by several [MapProcessValueBehavior](#). The input range is optional. If not set, then the input range is taken from the ProcessValueEvent.

RenderingEnabled Property now bindable

The property RenderingEnabled of behavior [RenderNodeBehavior](#) / "Render Node" is now bindable.

See also:

[RenderNodeBehaviorBase::SetEnableRendering](#)
