guiliani.de

Control Attributes

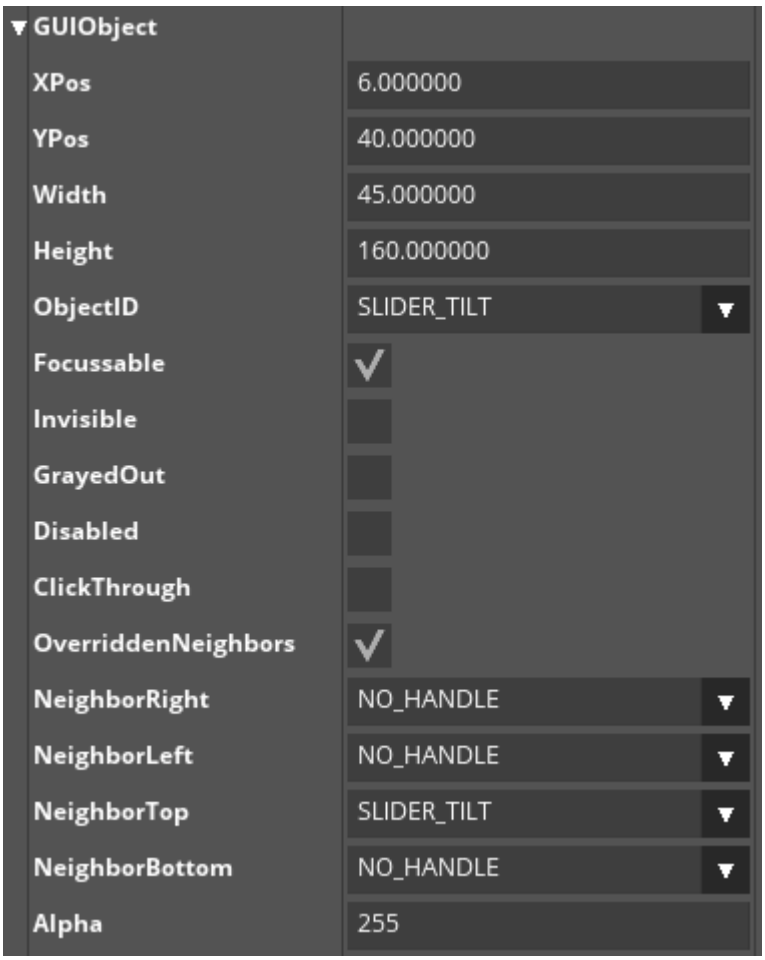
Product:	Guiliani Streaming Editor (GSE)
Release version:	2.2
Release date:	July 9, 2019

Table of contents

Common Information:

All IDs starting with “DUMMY_” mark that no resource is currently set for this attribute, e.g. DUMMY_IMAGE means that no ImageID is currently and so no image will be displayed.

GUIObject:



XPos: X-position of the object relative to the left side of the parent object (in pixels). If the object is moved around, this value also changes. Floating-point values are possible.

YPos: Y-position of the object relative to the top position of the parent object (in pixels). If the object is moved around, the value also changes. Floating-point values are possible.

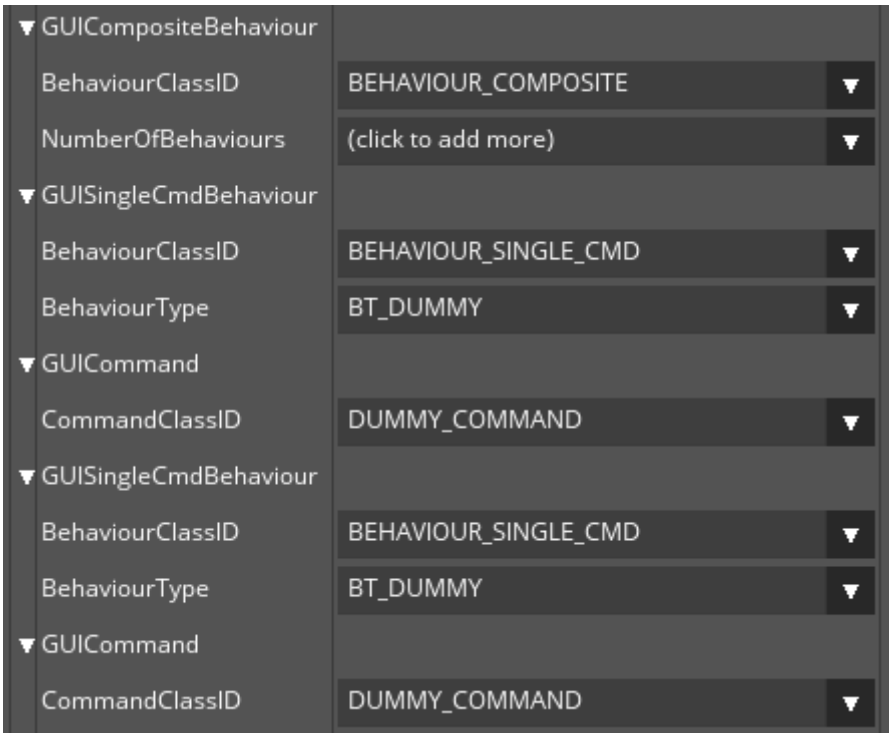
Width: width of the object. If the object is resized via mouse, this value also changes. Floating-point values are possible.
Height: height of the object. If the object is resized via mouse, this value also changes. Floating-point values are possible.
ObjectID: ObjectID for the object to access it from other parts (e.g. Commands, Behaviours, etc.) When a new object is created it receives an auto-generated ObjectID. This can be changed by selecting another ObjectID from the list or defining a new.
Focussable: If an object is focusable it can request or lose the focus depending on the application's workflow. Focussed objects can be controlled via keyboard.
Invisible: visible state of object
GrayedOut: Grayed out objects cannot receive any events and are displayed in a different way.
Disabled: Disabled objects like grayed out ones do not receive any events, but do not have a special visualization.
ClickThrough: If an object is click-through events occurring inside its bounding rectangle will be passed to objects "below" this object (z-order).
OverriddenNeighbors: if this setting is active the neighbours of the object – used for 4-way-navigation – are set
NeighborRight/NeighborLeft/NeighborTop/NeighborBottom: neighbours used by the 4-way focusing algorithm.
Alpha: alpha-value used to display the object. 0 means fully transparent, 255 fully opaque.

GUIBehaviour:

Behaviours are used to control the objects behaviour in response to events sent to the object. If an object has a behaviour attached to it, the behaviour receives the event first. The object itself only receives the event if the behaviour has not handled it (return false).

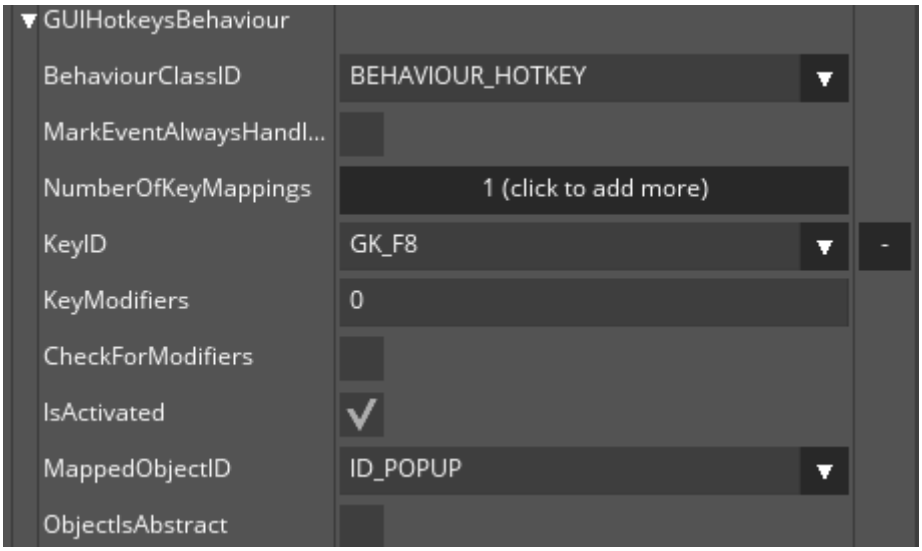
Note: The action of the behaviour is executed immediately and might block the running application.

GUICompositeBehaviour:



This Behaviour contains a number of other behaviours and forwards incoming events to them, until one of the behaviours has signalled that it has processed the event (return true).

GUIHotkeyBehaviour:



This behaviour can only be attached to a CompositeObject, otherwise processing will not be done.

MarkEventAlwaysHandled: if this is set, the behaviour always consumes the event (return true), even if the hotkey was not processed.

NumberOfKeyMappings: Adds a new key-mapping to the list of hot keys. Every time, you click on the value field, you get a new input in the list of hot keys. If you want delete some keys from this list, you can use “minus” Button on the right side of “KeyID” field.

KeyID: Key ID which triggers the action

KeyModifiers: modifiers which should be checked. Value for the modifiers are: 0 for NONE, 1 for SHIFT, 2 for CTRL, 4 for ALT and 8 for META (special implementation). Can be summed up, if multiple modifiers should be checked, e.g. 5 for SHIFT and ALT

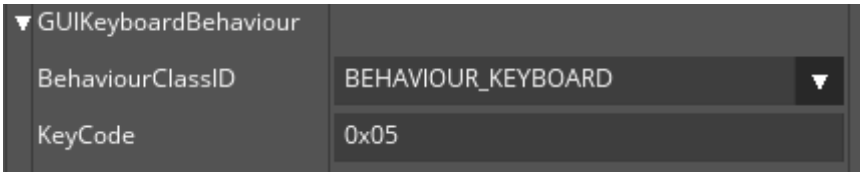
CheckForModifiers: if set, the modifiers are checked

IsActivated: if set, the keymapping will be processed

MappedObjectID: ObjectID of the object which will receive the resulting event. Only the container this behaviour is attached to will be searched for the given ObjectID.

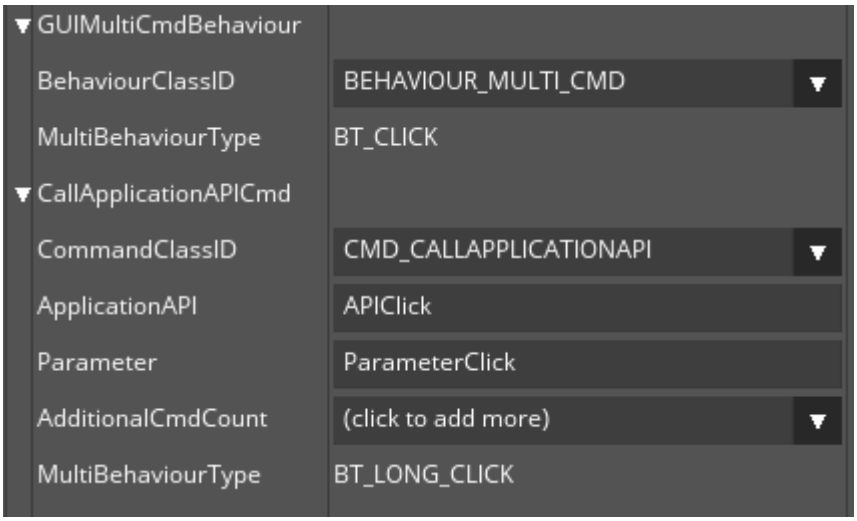
ObjectIsAbstract: If the object is abstract object, by activating the hot key, the DoClick method will be executed. Otherwise the object will be firstly checked whether it is clickable.

GUIKeyboardBehaviour:



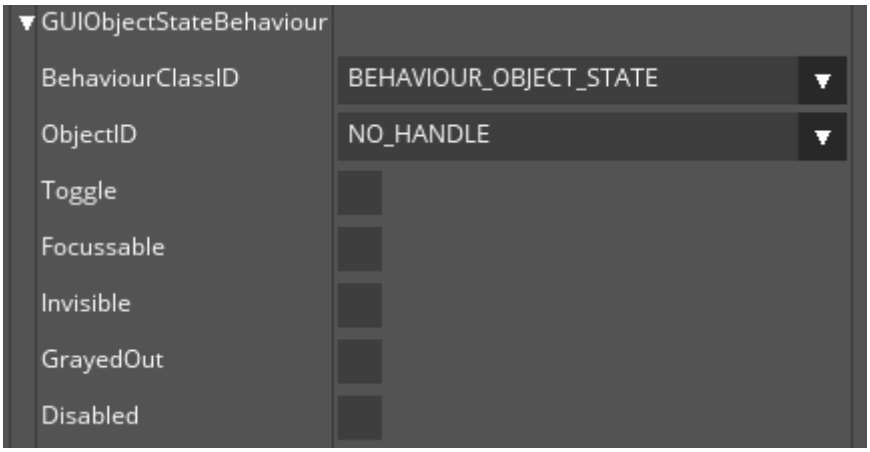
This behaviour is to be used with an OnScreen-Keyboard / Virtual Keyboard. It sends an UTF16 keycode to the Keyboard in reaction to a ButtonUp-Event. In the field KeyCode you can set the keycode.

GUIMultiCmdBehaviour:



This behaviour maps each available event to a specific action. Gives the possibility to execute a separate command for each event (e.g. click event, drag event, etc.). The field “MultiBehaviourType” shows the corresponding event-type for the mapping.

GUIObjectStateBehaviour:



This Behaviour is used for triggering changes of the states of the specified object

ObjectID: ID of the object to process

Toggle: if set the used attributes will be toggled each time this behaviour is executed. If not set, the attributes are set as they specified.

Focussable: processes the “Focussable”-attribute

Invisible: processes the “Invisible”-attribute

GrayedOut: processes the “GrayedOut”-attribute

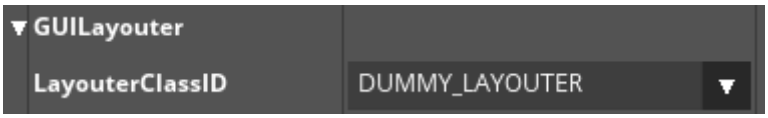
Disabled: processes the “Disabled”-attribute

GUISingleCmdBehaviour:

Execute a command in response to a specific event. Gives the possibility to execute a command when a specific event (e.g. click event, drag event, etc.) occurs.

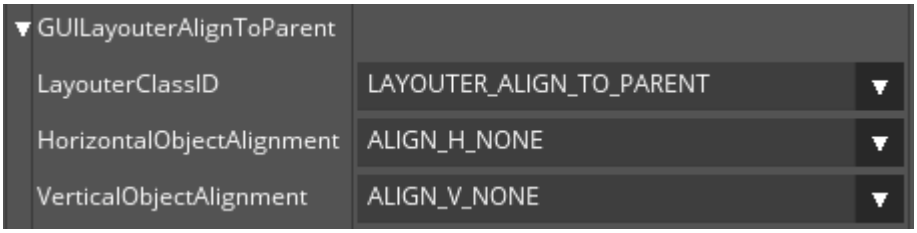
BehaviourType: specifies the type of event which should be linked to the command.

GUILayouter:



Layouters are used to arrange objects to a specific rule, if position or size of the attached object changes.

GUILayouterAlignToParent:



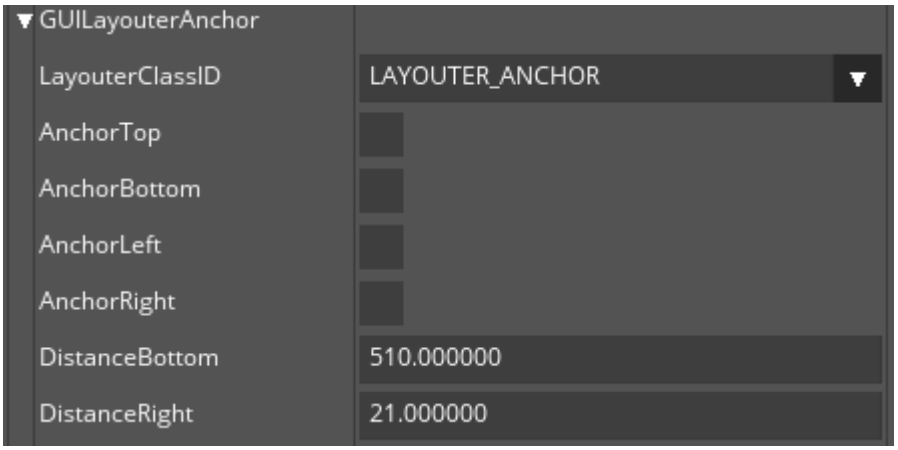
This Layouter aligns the attached object relative to its parent object.

HorizontalObjectAlignment: horizontal alignment (left, center, right)

VerticalObjectAlignment: vertical alignment (top, center, bottom)

Note: This will not directly refresh the layout until the attached object is resized.

GUILayouterAnchor:



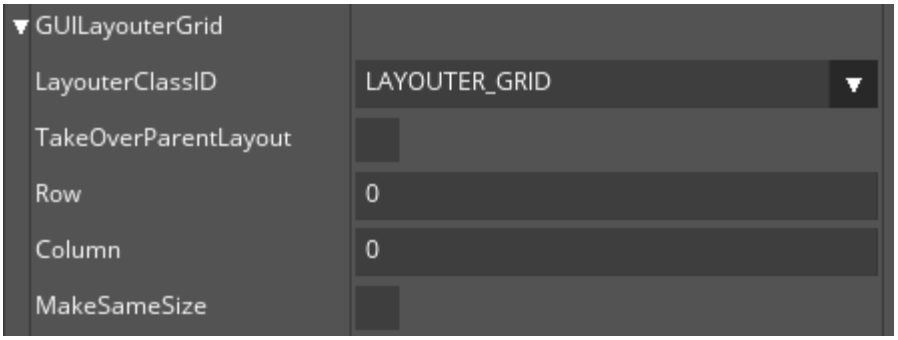
This Layouter can be used to 'fix' the edges of a widget to its parent. Anchor Top/Bottom/Left/Right. The following table gives you some examples on which behaviour will result from which anchor-settings:

AnchorTop/AnchorBottom/AnchorLeft/AnchorRight: shows /defines Anchor attributes. The following table gives you some examples on which behaviour will result from which anchor-settings:

top	botton	lef	righ	result
ON	ON	ON	ON	Object gets stretched in X/Y directions
off	off	off	off	Object will not get stretched, but will scale its position along with the change in size of its parent. i.e. If the width of the parent object gets doubled, the relative X position of the child object will double as well.
ON	ON	ON	off	Object gets stretched in Y direction and is locked to the left border of its parent
ON	ON	off	ON	Object gets stretched in Y direction and is locked to the right border of its parent
ON	ON	off	off	Object gets stretched in Y direction and will scale its X position along with the change in size of its parent/td>

DistanceBottom/DistanceRight: the distances of the associated object's right border to the parent's right border and of the associated object's bottom border to the parent's bottom border. This is useful for example when the height or width of the associated object changes so that when resizing the parent later on, the current anchor length are used and not those that were saved when the anchors were chosen.

GUILayouterGrid:



Note: This Layouter should only be attached to a CompositeObject, otherwise a warning-message will appear.

TakeOverParentLayout: if this is set, the attached object is moved to the relative position 0,0 and set to the size of the parent-object, before the actual layouting takes place.

Row/Column: number of rows/columns which define the size of the cells

MakeSameSize: if this is set, the sizes of each child-object is set to the size of one call of the specified grid.

GUILayouterList:

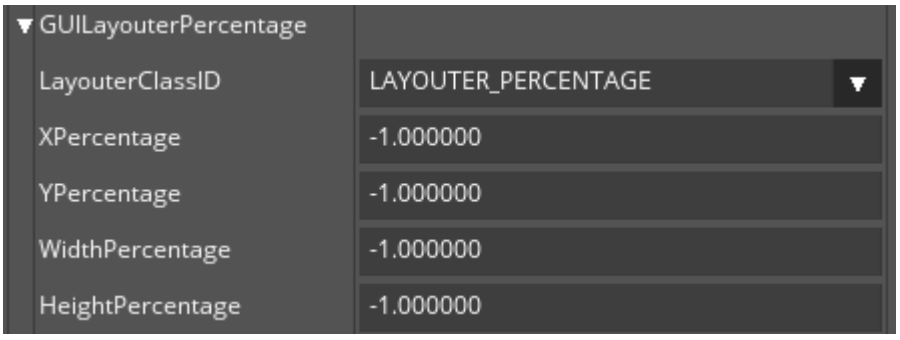
Note: This Layouter should only be attached to a CompositeObject, otherwise a warning-message will appear.

This Layouter aligns all child-objects in a list based on the ListBasePoint

TakeOverParentLayout: if this is set, the attached object is moved to the relative position 0,0 and set to the size of the parent-object, before the actual layouting takes place.

ListBasePoint: the base-point on which the list positioning is based (horizontal or vertical)

GUILayouterPercentage:

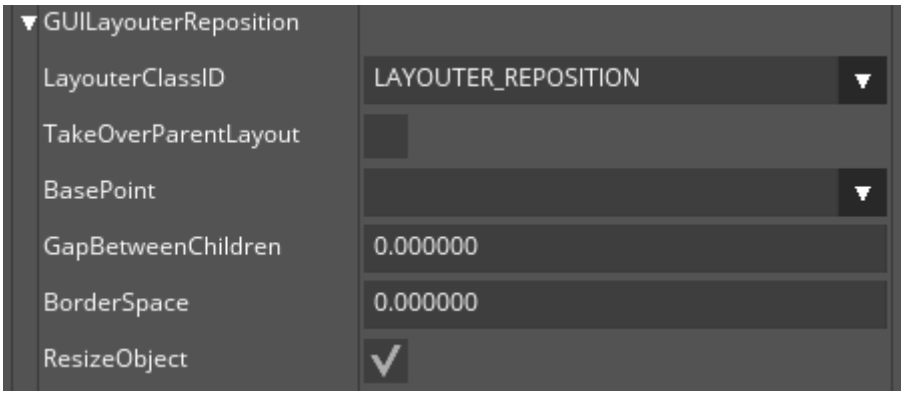


The Value -1.0 is used if the attribute should not be processed.

XPercentage/YPercentage: set object to a new relative position. Values are multiplied with the parent's width/height.

WidthPercentage/HeightPercentage: set object to a new size. Values are multiplied with the parent's width/height.

GUILayouterReposition:



Should only be attached to a CompositeObject, otherwise a warning-message will appear.

TakeOverParentLayout: if this is set, the attached object is moved to the relative position 0,0 and set to the size of the parent-object, before the actual layouting takes place.

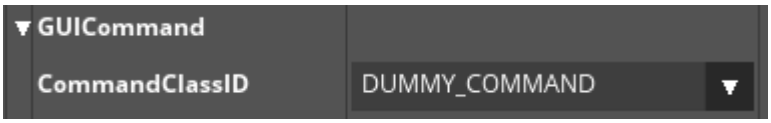
BasePoint: the rule on which the repositioning is based (horizontal/vertical)

GapBetweenChildren: the gap (in pixels) between the children used during layouting.

BorderSpace: border-space used during layouting.

ResizeObject: if this is set, the attached object is automatically resized to be exactly large enough to contain all of its children.

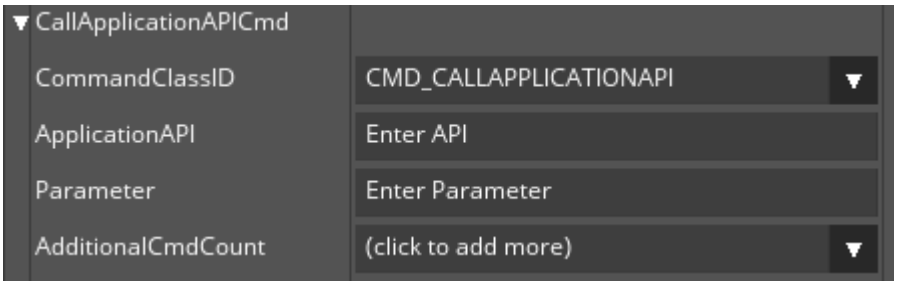
GUICommand:



Commands are used to execute specific actions asynchronously. Commands – when executed – are added to a queue and executed not before the next main-loop.

AdditionalCmdCount: Here additional commands can be selected, which will be executed after the current command is finished.

CallApplicationAPICmd:

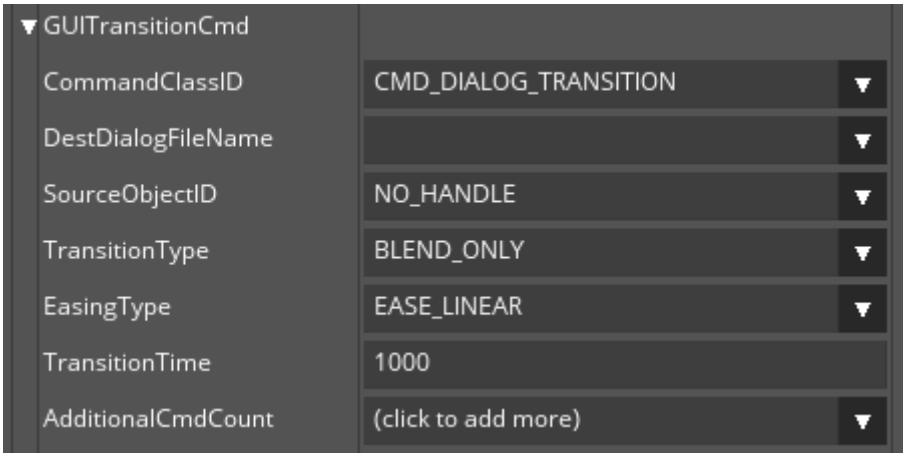


This Command can be used to execute various actions specified by the given attributes.

ApplicationAPI: string which is sent to the command and can be used to execute specific actions.

Parameter: string which is sent to the command which specifies the used parameter

GUITransitionCmd:



This Command executes a transition from one dialog to another. Various options can be set. This command first loads the destination-dialog and then traverses from the source-object to it using a specific type of transition.

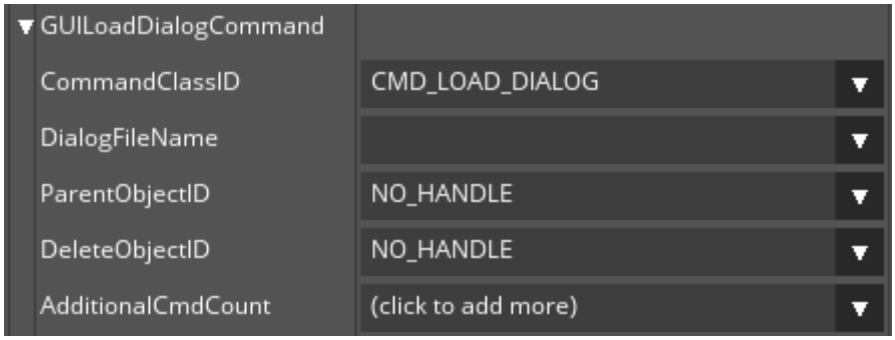
DestDialogFileName: The filename of the dialog which will be the destination-object.

SourceObjectID: ID of the source-object. This needs to be an object within the currently active dialog or the dialog itself to have the command work.

TransitionType: type of transition. This can be one of the following options:

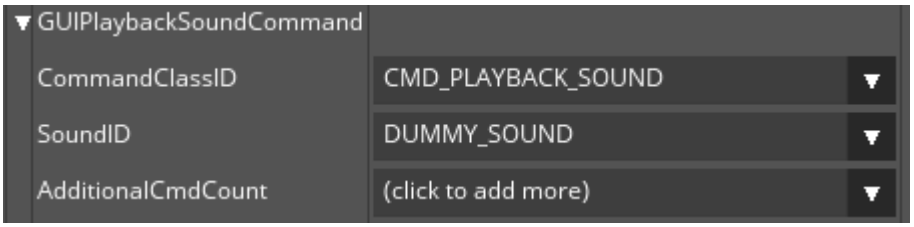
BLEND_ONLY/BLEND_FADE/BLEND_AND_SHRINK: blends to the destination dialog
PUSH_FROM_LEFT/PUSH_FROM_RIGHT/PUSH_FROM_TOP/PUSH_FROM_BOTTOM: moves the source-dialog out of the screen while pushing the destination-dialog into the screen
ROTATE_X_AXIS_CW/ROTATE_X_AXIS_CCW/ROTATE_Y_AXIS_CW/ROTATE_Y_AXIS_CCW: flips the screen as a 3D-panel, having the source-dialog on one side and the destination-dialog on the other side around the specified axis either clockwise or counter-clockwise.
EasingType: the easing type which is used for the transition
TransitionTime: duration in milliseconds for the transition.

GUILoadDialogCommand:



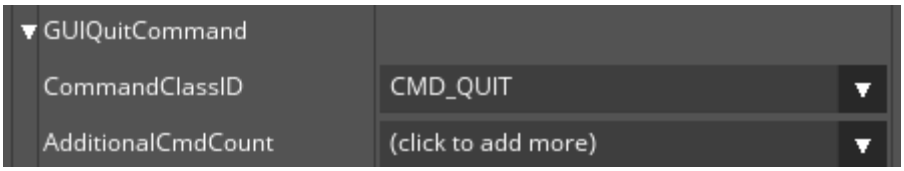
This command loads a new dialog and places it into the specified CompositeObject. Additionally a given object can be deleted.
DialogFileName: filename of the dialog which should be loaded. This can either be an xml-file or a binary file.
ParentObjectID: ID of the future parent object of the loaded dialog. If this attribute is “NO_HANDLE” the top-most object (i.e. the GUI itself) will be the new parent of the loaded dialog.
DeleteObjectID: ID of an object which should be deleted after the dialog was loaded.

GUIPlaybackSoundCommand:



SoundID: ID of the sound which should be played.

GUIQuitCmd:



This command cleanly ends the current application be shutting down the main-loop of Guiliani and destroys all created Wrapper-classes (e.g. Graphics-, Sound- or Font-Wrapper)

GUISetObjectStateCmd:

This command does the same as the GUISetObjectStateBehaviour, but will be executed asynchronously not before the next main-loop.

NinePatchBorders:

A Ninepatch is a smart way to scale bitmaps by defining 9 different regions of the original bitmap which will be scaled up independently.

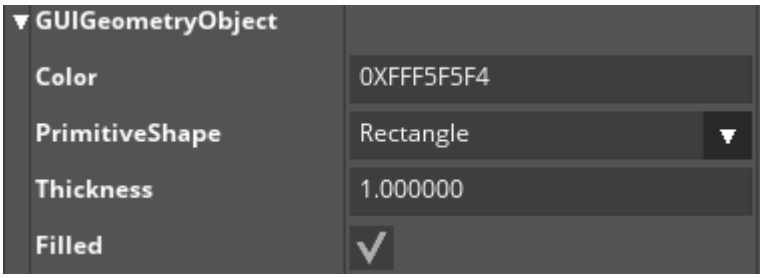
Top: number of pixels which define the top-stripe of the bitmap

Bottom: number of pixel which define the bottom-stripe of the bitmap

Left: number of pixels which define the left-stripe of the bitmap

Right: number of pixels which define the right-stripe of the bitmap

GUIGeometryObject:



Note: some combinations of PrimitiveShape and Filled may not work depending on the currently used Graphics-Wrapper in the application.

Color: color of the primitive which will be drawn.

PrimitiveShape: one of the following options:

PS_LINE1/PS_LINE2: a diagonal line top/left to bottom/right or top/right to bottom/left

PS_VERTICAL_LINE/PS_HORIZONTAL_LINE: vertical or horizontal line

PS_ELLIPSE/PS_RECTANGLE: an ellipse or rectangle

Thickness: the width of the primitive

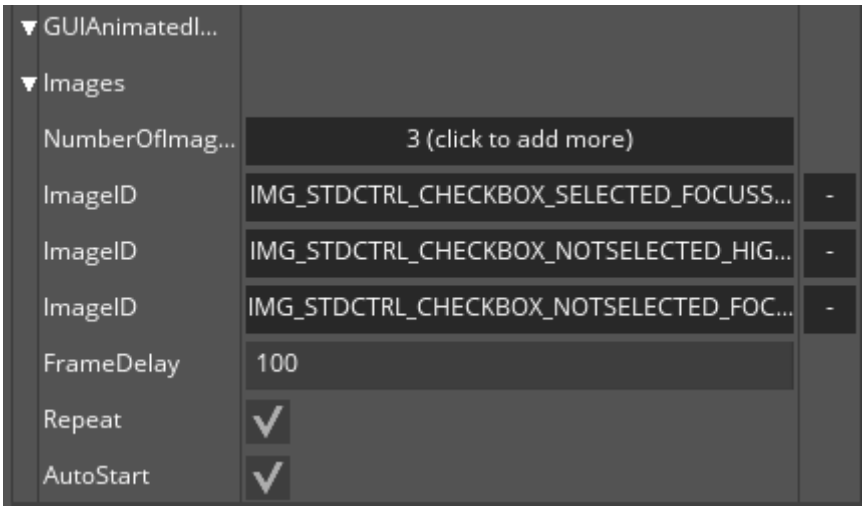
Filled: used for ellipses and rectangles for filling

GUIImage:

ImageID: used image for blitting. Can be set using manage-images-dialog.

StretchBlit: if this is set, the image will be stretched to the whole size of the object. This may decrease graphics-performance on large images. If this is not set the image will be drawn in the center of the object's bounding rectangle.

GUIAnimatedImage:



Note: at least two images should be used to see this control in action.

NumberOfImages: clicking on this button will add a new image. Clicking on the "-"-button in the right column will remove the image next to it.

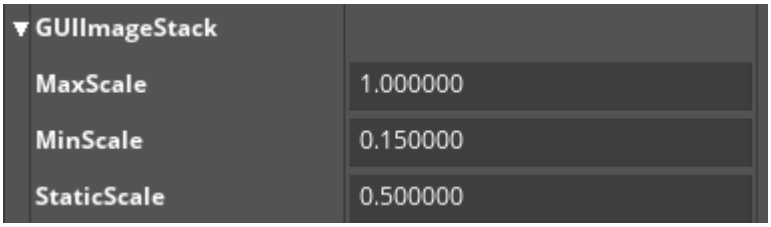
ImageID: image(s) which will be used for animation

FrameDelay: number of milliseconds between each transition.

Repeat: if this is set, the animation runs in an infinite loop.

AutoStart: If this is set, the animation automatically starts and can be viewed inside the editor.

GUIImageStack:

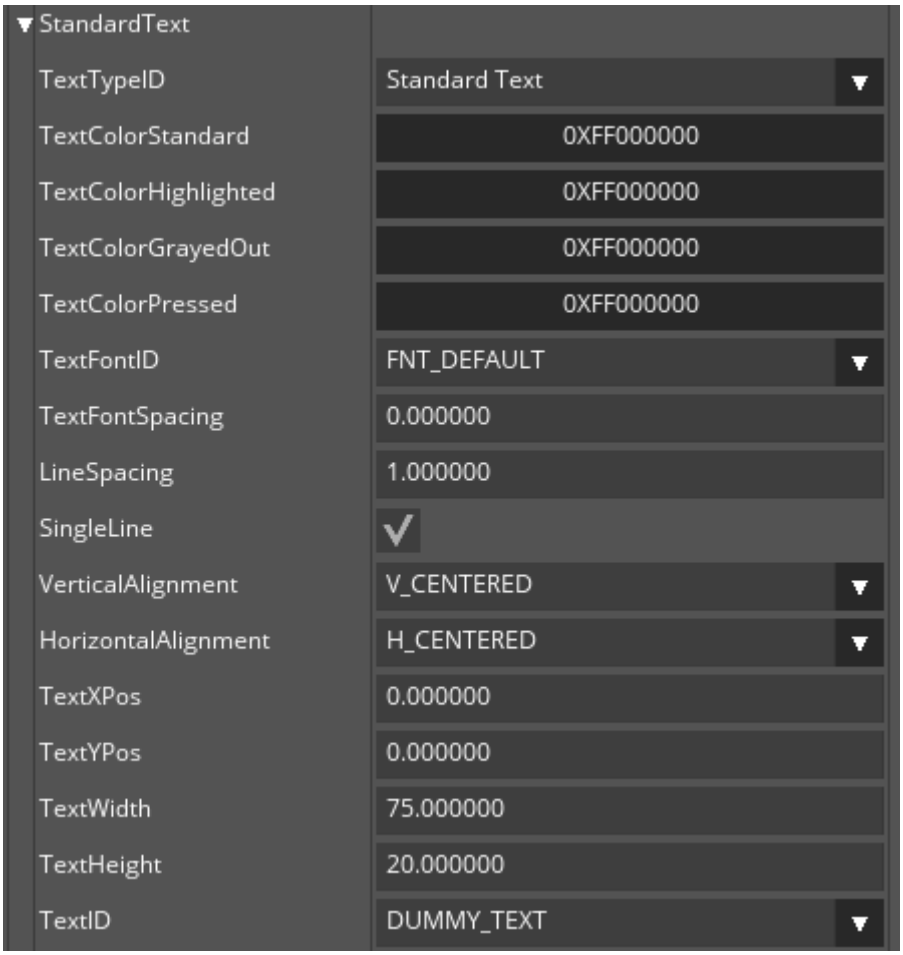


This control gives the possibility to swipe through a set of images which will be animated.

MaxScale/MinScale: maximum/minimum scaling factor the images during the animation

StaticScale: scaling factor for the image when not animated

StandardText: (applies for all texts)



TextTypeID: this changes the type of the text. This can be “Standard Text”, “Editable Text”, “Rich Text”, “Scrolling Text” or “No Text”.

TextColorStandard/TextColorHighlighted/TextColor/GrayedOut/TextColorPressed: colors used for the text according to the states of the object.

TextFontID: ID of the font which will be used to render the text. If “DUMMY_FONT” is used, no text will be displayed.

TextFontSpacing: spacing (in pixels) between neighbouring letters.

LineSpacing: spacing between lines. Only applies if “SingleLine”-attribute is not set.

SingleLine: indicates whether this text is single- or multi-line.

VerticalAlignment/HorizontalAlignment: horizontal/vertical alignment of the text. Can be left/centered/right or top-line/centered/bottom-line

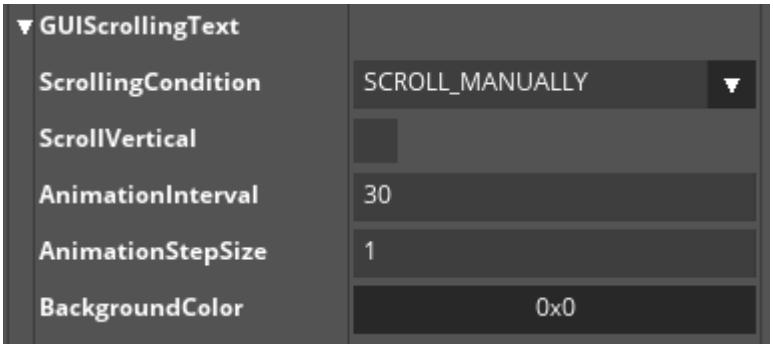
TextXPos/TextYPos: position of the textfield inside the object relative to the top/right-corner

TextWidth/TextHeight: width and height of the textfield inside the object

TextID: Choose the TextID which will be used for the text. If the currently active language-set is changed, this text will automatically be updated. If TextID is set to “text string”, then a static text will be used.

BackgroundColor: defines Background Color

GUIScrollingText:



ScrollingCondition: one of the following options:

SCROLL_ALWAYS: the text will always be scrolled

SCROLL_CUT_TEXT_ALWAYS: The text will be scrolled when the text is truncated

SCROLL_CUT_TEXT_FOCUSED: The text will be scrolled when the text is truncated and the parent object is focused

SCROLL_FOCUSED: The text will be scrolled when its parent object is focused

SCROLL_MANUALLY: The Animation can be started or stopped manually using API-calls.

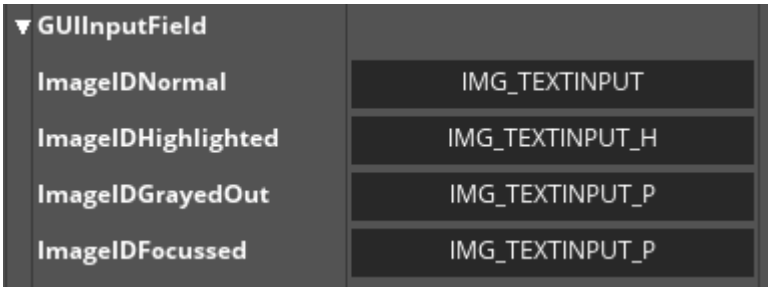
ScrollVertical: If active, then text scrolled vertically, if not is will be scrolled horizontally.

AnimationInterval: number of milliseconds between each animation-step

AnimationStepSize: number of pixels the current position of the text will be increased/decreased for each step. A value if 0 prevents moving

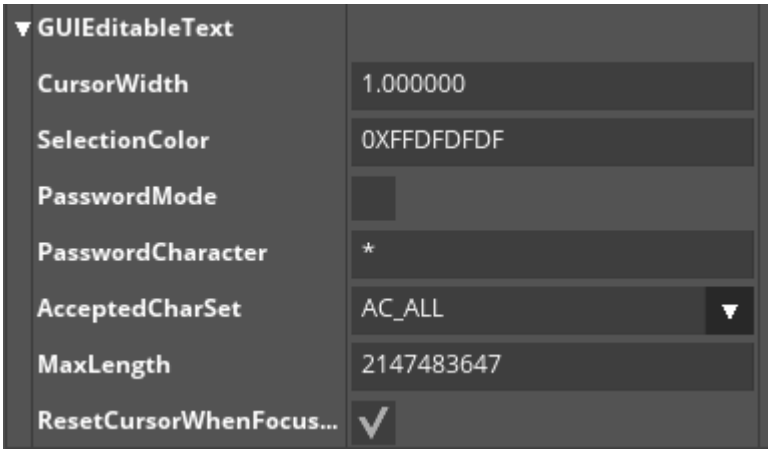
BackgroundColor: defines Background color

GUInputField:



ImageIDNormal/ImageIDHighlighted/ImageIDGrayedOut/ImageIDFocussed: image ids which will be used as the background of the input-field according to the current state of the object

GUIEditableText (O):



The input field consists of an editable text and a background object. To edit the attributes of the text input, select it in the Object Hierarchy window or in the Dialog window(CGUIEdit).

CursorWidth: shows/defines the width of the vertical cursor line.

SelectionColor: shows/defines the background color of selected text.

PasswordMode: shows/defines the password mode to obfuscate the input using a specified character that is displayed instead

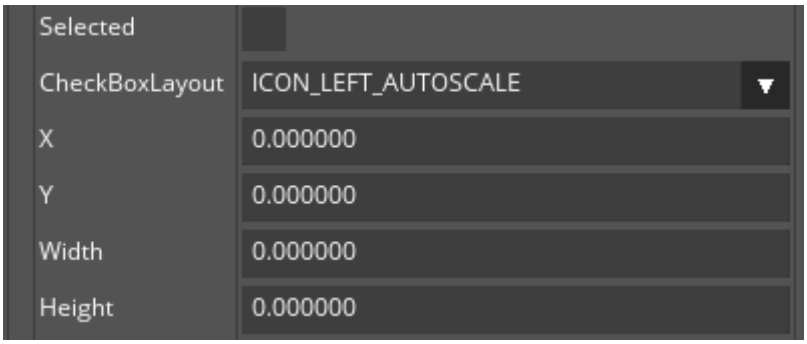
PasswordCharacter: shows/defines the character that shall be displayed in the input. If you will try with more as one character, only first character will be used

AcceptedCharSet: shows/defines accepted character set

MaxLength: shows/defines the maximal number of characters

ResetCursorWhenFocusslost: shows/defines cursor behaviour, when focus lost. If active: Cursor will be not seen after focus lost. In another case: you will steel see the cursor, after focus lost.

GUIBaseCheckBox:



Selected: selection-state of the control

CheckBoxLayout: chooses how icon and text are layouted. Can be one of the following options:

MANUAL_LAYOUT	No automatic layout. Text position and size are specified manuall
ICON_LEFT_AUTOSCALE	Icon left and scaled automatically, text will be drawn right to it
ICON_RIGHT_AUTOSCALE	Icon right and scaled automatically, text will be drawn left to it
ICON_LEFT_CENTERED	Icon left and not scaled, text will be drawn right to it
ICON_RIGHT_CENTERED	Icon right and not scaled, text will be drawn left to it
ICON_FILL_OBJECT	The Icon will fill the entire area of the object

X/Y: shows/defines position of the icon relative to the top/left-corner of the object.

Width/Height: size of the icon

GUICheckBox:

SelectedImageIDs: image-ids which are used for the selected-state according to the state of the object

NotSelectedImageIDs: image-ids which are used for the not-selected-state according to the state of the object

GUIRadioButton:

SelectedImageIDs: image-ids which are used for the selected-state according to the state of the object

NotSelectedImageIDs: image-ids which are used for the not-selected-state according to the state of the object

GUIButton:

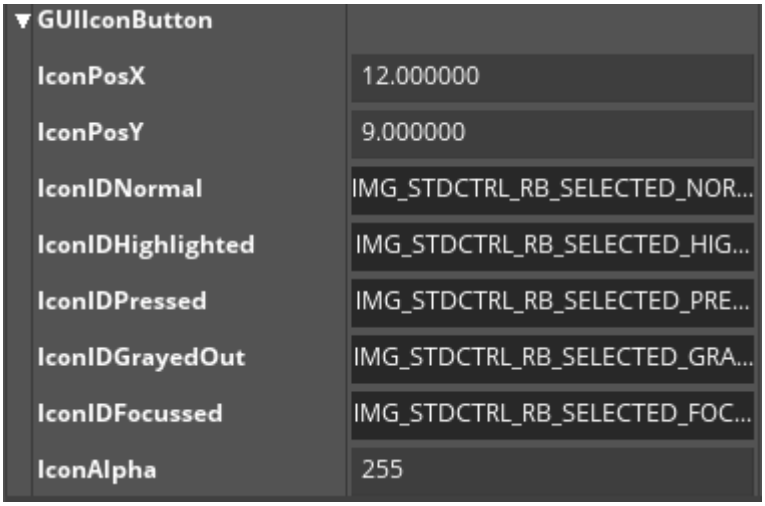
ImageIDs: image-ids which will be used for the object according to its state

GUIBlendButton:

BlendDuration: duration in milliseconds which will be used when the state of the object and thus the used image will change

CrossFade: if this is set a cross-fade will be used to switch to the new image according to the object's state

GUIIconButton:

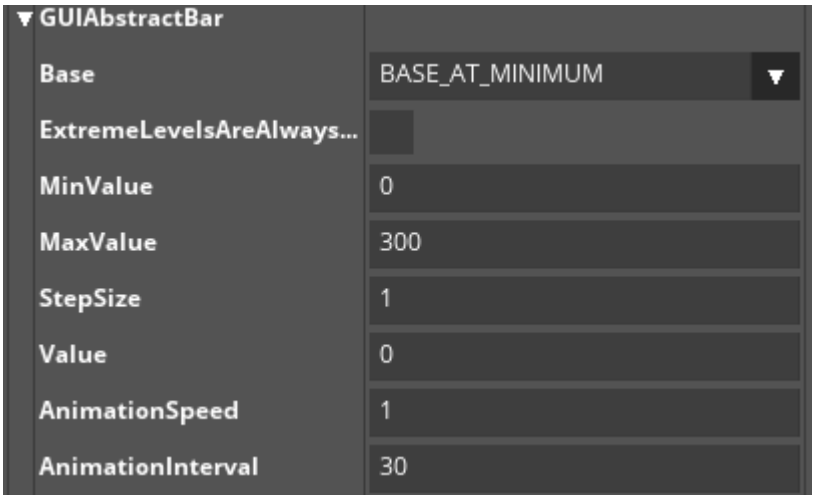


IconPosX/IconPosY: position of the icon-image relative to the object's top/left-

IconIDNormal/IconIDHighlighted/IconIDPressed/IconIDGrayedOut/IconIDFocussed: image-ids which will be used for the icon according to the object's state

IconAlpha: alpha-value used for the icon

GUIAbstractBar (O):



Base: shows/defines base of the level

ExtremeLevelsAreAlwaysValid: shows/defines True, if the maximum is also a valid current value, no matter what the step size is. False, else.

MinValue/MaxValue: shows/defines The minimum/maximum value of the range

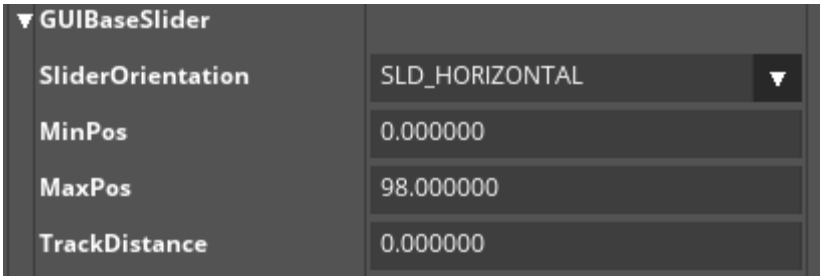
StepSize: step size

Value: currently set value

AnimationSpeed: speed for animated scrolling.

AnimationInterval: shows/defines timer interval in milliseconds for animation.

GUIBaseSlider (O):



SliderOrientation: sets the orientation of the slider either to horizontal or vertical

MinPos/MaxPos: position of the minimum/maximum-value of the slider

TrackDistance: the length of the slider track in pixel

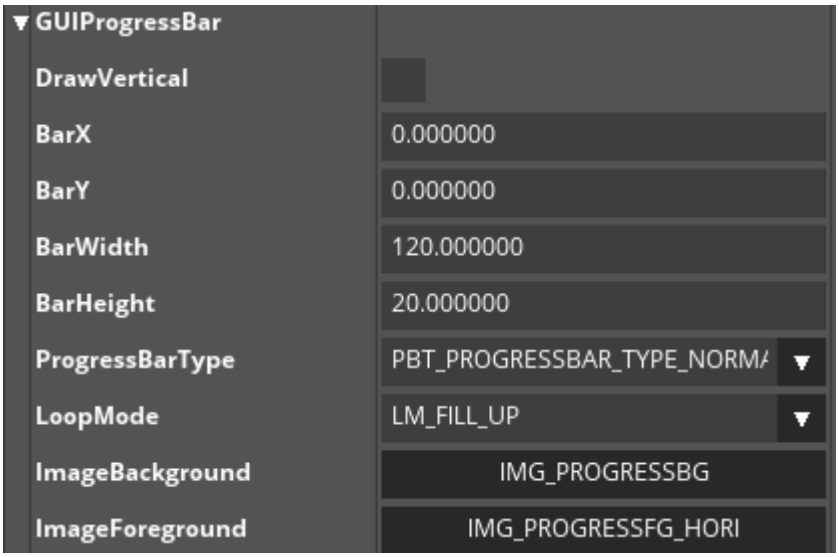
GUISlider:

ImageIDBackground: image-id of the background (i.e. the slider area)

ImageIDKnobNormal/ImageIDKnobHighlighted/ImageIDKnobPressed/ImageIDKnobGrayedOut: image-ids which are used for the knob according to the object's state

BackgroundMargin: number of pixels the background-image is moved relative to the top/left-corner of the object

GUIProgressBar:



DrawVertical: if this is set the progressbar will be drawn vertically, otherwise horizontally

BarX/BarY: position of the bar relative to the top/left-corner of the object

BarWidth/BarHeight: width/height of the bar

ProgressBarType: can be one of the following options:

PBT_PROGRESSBAR_TYPE_NORMAL: designed for full control of the fill state via the application

PBT_PROGRESSBAR_TYPE_ACTIVITY: designed for representation of unpredictable duration of operations

LoopMode: can be one of the following options:

LM_FILL_UP: fill up the bar until full and starts repeatedly again with an empty bar

LM_LOOP_MODE: repeatedly move the bar from start to end

LM_REVERTLOOP_MODE: move the bar from start to end and back to start again

ImageBackground/ImageForeground: image-ids which are used for the background and foreground of the bar

GUIRepositionCompositeObject:

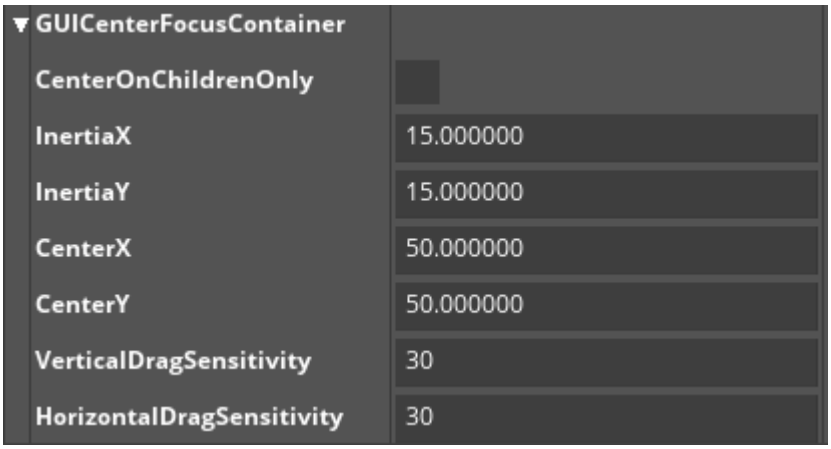
Note: The repositioning effect will only take place when resizing the container.

BasePoint: the base point on which the repositioning is based.

GapBetweenChildren: the gap between the children in pixel

BorderSpace: the border space to the composite border in pixel.

GUICenterFocusContainer (O):



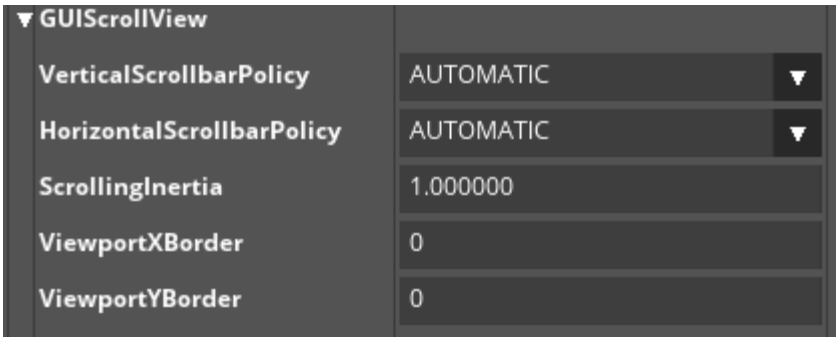
CenterOnChildrenOnly: shows/defines automatically centers the currently focused control at the supplied position. This centering can be animated with effects like speed-up or slow-down

InertiaX/InertiaY: shows/defines Inertia value in horizontal/vertical direction. Inertia value specifies how quick the container will move to a new position. Supplying bigger inertia values will slow down the speed.

CenterX/CenterY: shows/defines Horizontal/Vertical center position in relative coordinates, on which to center the focused object.

VerticalDragSensitivity/HorizontalDragSensitivity: shows/defines vertical/horizontal drag Sensitivity in pixel - distance after which the focus is moved to the next child.

GUIScrollView:



VerticalScrollbarPolicy/HorizontalScrollbarPolicy: can be one of the following options:

ALWAYS_VISIBLE: the scrollbars are always shown, even if the content might fit the available space

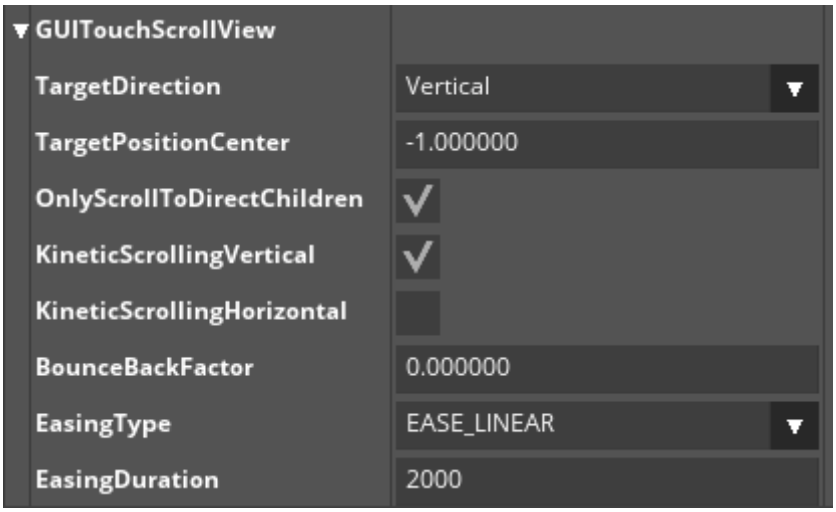
AUTOMATIC: the scrollbars are only shown if the available width/height is not enough to display the content

NEVER_VISIBLE: the scrollbars are never shown

ScrollingInertia: value for the scrolling-inertia. Valid values are >= 1

ViewportXBorder/ViewportYBorder: Horizontal/Vertical (in X/Y direction) gap between visible area and the area outside of which scrolling will occur.

GUITouchScrollView (O):



TargetDirection: shows/defines direction, used for scrolling, dragging and destination direction.

TargetPositionCenter: shows/defines the target position for the XPosCenter or the YPosCenter of the active object. When clicking on the scroll view, the nearest object to the click position (in destination direction) is chosen as the active object. When dragging, the nearest object to the target position is chosen as active object when the drag is finished. When kinetic scrolling is activated, the nearest object to the target position is chosen when the kinetic animation is about to fade out.

OnlyScrollToDirectChildren: Only direct children of the scroll view, or any descendant.

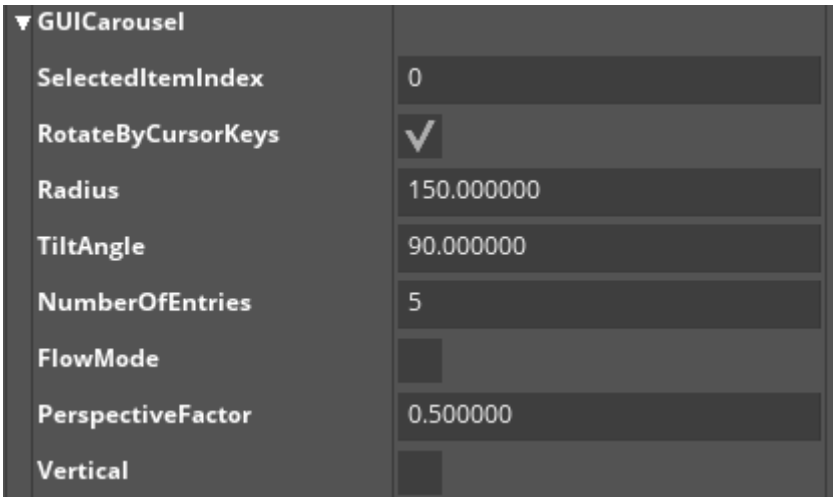
KineticScrollingVertical/KineticScrollingHorizontal: shows/defines if kinetic scrolling is activated in horizontal/vertical scroll direction.

BounceBackFactor: shows/defines when an edge of the scroll view is reached during an animation.

EasingType: shows/defines EasingType used for kinetic animation.

EasingDuration: shows/defines duration of easing animation for kinetic scrolling.

GUICarousel:



SelectedItemIndex: index of the child object which will be selected (shown in the middle)

RotateByCursorKeys: shows/defines whether left/right cursor keys should be used to rotate the carousel as well. If set, it might interfere with standard focusing behaviour

Radius: current radius (in pixel) of the virtual circle used to place the child-objects

TiltAngle: tilt-angle of the virtual circle used to place the child-objects

NumberOfEntries: shows/defines a virtual number of entries in the carousel, used e.g. for calculating the spacing between them. This setting is only relevant in FlowMode.

FlowMode: activates flow mode

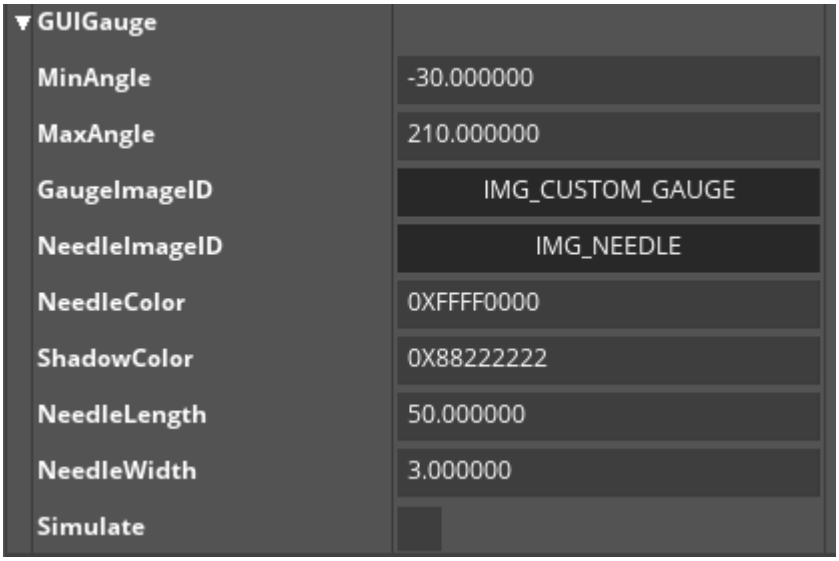
PerspectiveFactor: shows/defines perspective Factor. The width and height of the child-objects within the carousel get recalculated according to perspective. This setting specifies the strength of perspective distortion which is applied to objects inside tilted carousels. Objects in the background will appear smaller than those in the foreground. Objects in untilted carousels, and objects that reside on the X-Axis of tilted carousels, will be displayed using their original size. Sensible values are typically in the range of >0 to 1, where 0 means no perspective effect at all, and larger values result in stronger zooming / shrinking. Note that Z-Ordering will not work if perspective-Factor is 0.

Vertical: if this is set, the carousel is arranged vertically, otherwise horizontally

Note: Using the carousel in flow-mode:

When working with huge numbers of children in a carousel, you will notice that the carousel tends to appear very crowded. This is where the "flow mode" comes in handy. When used in "flow mode", the carousel can contain very large numbers of entries of which only a few are visible at any time. You can specify the number of virtual entries in the carousel, which will affect the spacing between the visual children, and therefore also number of simultaneously visible ones. If for instance you are having a total of 1000 entries in your carousel, and set the number of virtual entries to 10, the carousel will arrange its content as if it had only 10 children, by spreading out these 10 along the carousel's 360 degrees. But in fact you will only see the currently selected (=frontmost) child object, plus the two children to its left and the two children to i's right side.

GUIGauge:



MinAngle/MaxAngle: angle which is used for the needle for minimum/maximum values

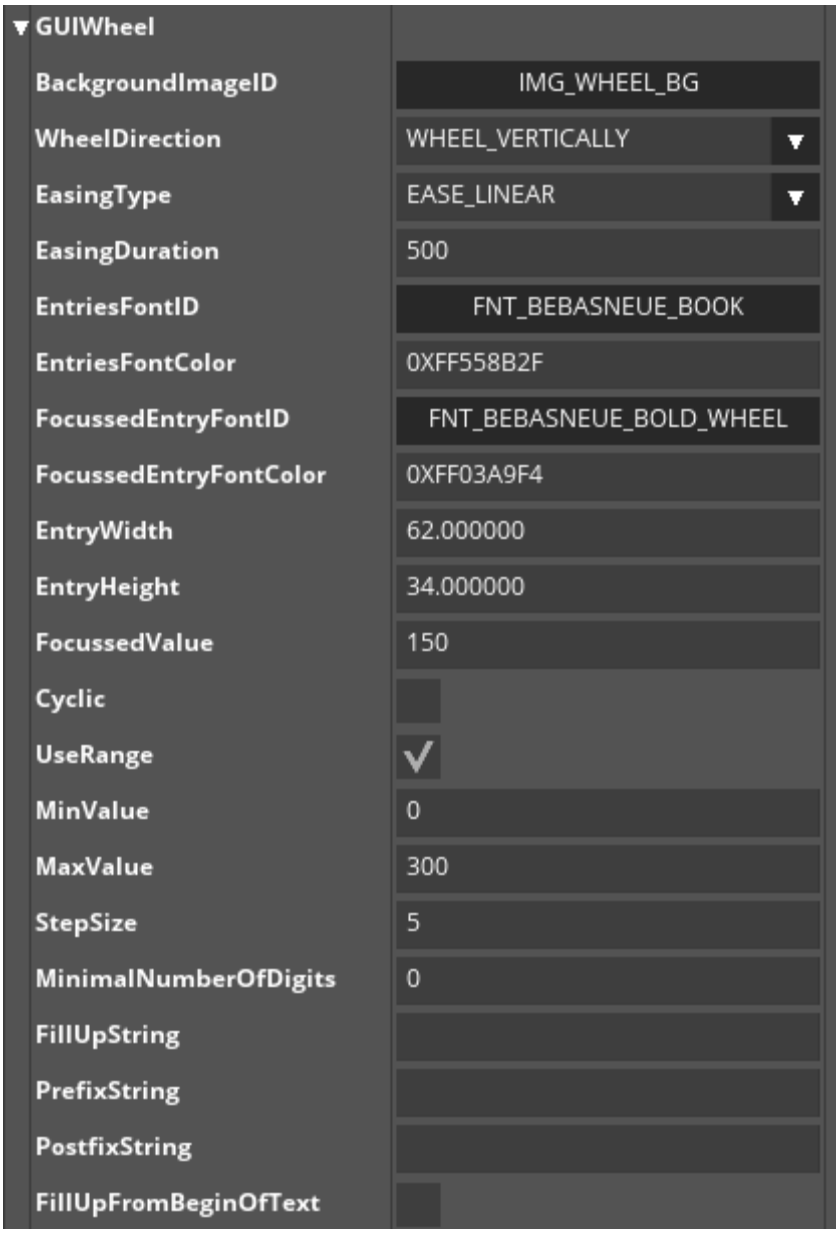
GaugeImageID/NeedleImageID: image used for gauge/needle

NeedleColor/ShadowColor: shows/defines color of the needle. This attribute is only used if “DUMMY_IMAGE” is set for NeedleImageID

NeedleLength/NeedleWidth: length and width of the needle in pixels (From center of gauge to its tip). This attribute is only used if “DUMMY_IMAGE” is set for NeedleImageID

Simulate: activate simulation-mode. In Simulation-mode the gauge will move repeatedly from minimum value to maximum value and back.

GUIWheel (O):



BackgroundImageID: image-id used to draw the background of the wheel. This image will always be stretched to fill the bounding rectangle of the object.

WheelDirection: direction of the wheel: horizontal or vertical.

EasingType: EasingType used for kinetic animation.

EasingDuration: Duration of easing animation for kinetic scrolling in milliseconds.

EntriesFontID/FocussedEntryFontID: Id of the font which is used for non-focused/focussed entries

EntriesFontColor/FocussedEntryFontColor: color of the text which is used for non-focused/focussed entries

EntryWidth/EntryHeight: width/height for one entry

FocussedValue: currently focused entry value.

Cyclic: if this is set, the wheel is cyclic, i.e. begins with the first element after the last element. If this is not set, the wheel has a lower and an upper end.

UseRange: if this is set, the entries of the wheel are auto-generated using the minimum/maximum-value and the step-size

MinValue/MaxValue: minimum/maximum-value

StepSize: size of one step

MinimalNumberOfDigits: shows/defines Minimal number of digits.

FillUpString: shows/defines String, will be used to fill up missing digits.

PrefixString: shows/defines String will be place in front of entry.

PostfixString: shows/defines String will be place behind of entry. Can be used for example for units etc.

FillUpFromBeginOfText: shows/defines If true entries will be filled in front of entry, if false at end.