Interface modding - Hearts of Iron 4 Wiki

This is a community maintained wiki. If you spot a mistake then you are welcome to fix it.

In Hearts of Iron IV, the interface refers to the user interface the player uses to interact with the game.

Overview[编辑 | 编辑源代码]

The components of an interface file can be broken into two groups: containers and elements. For example, *containerWindowType* is a container, whereas *iconType* is an element.

An interface file uses the **.gui** filetype. All interface files are found in /Hearts of Iron IV/interface/.

When *internal* is used here, it is referring to the source code that only Paradox can edit. In short, it means the subject it refers to is not modifiable.

Macros[编辑 | 编辑源代码]

Macros can be used within interface files to make adjusting positions easier.

This for example would make a macro called **WINDOW_X_POS** with the value of 10.

```
@WINDOW_X_POS = 10
```

This can then be referred to in an element like so:

```
containerWindowType = {
  name = "my_container"
  position = { x = @WINDOW_X_POS y = -600 }

  # ...
}
```

Containers[编辑 | 编辑源代码]

Containers are used to group together elements and associate them with an internal function.

You cannot add new containers except through Scripted GUI. This is because the *name* attribute of a container is linked internally to whatever data the container requires. There are exceptions to this, a common one is that you can add new containers for custom technology folders in *countrytechtreeview.gui'*.

The order and nesting of container definitions affects how they are displayed.

containerWindowType[编辑 | 编辑源代码]

The most common container type, used to hold any group of elements.

The following attributes are commonly used:

name - The container name.

background - An element to use as the container background.

position - The screen position of the container.

size - The bounding box for the container. Can be in percentage (i.e. 100%%)

moveable - Sets whether the container can be dragged around.

fullscreen - Sets whether the container is considered to obscure the whole screen.

orientation - Sets the orientation origin for the *position* attribute.

clipping - Sets whether the container will clip other elements (i.e. cut their elements at boundary).

The following attributes are used for animated containers:

show_position - The screen position the container moves to in *show* mode.

hide_position - The screen position the container moves to in *hide* mode.

show_animation_type - The animation type used to show the container (*decelerated* or *linear*).

hide_animation_type - The animation type used to hide the container (*accelerated* or *linear*).

animation_type - The animation if both show and hide animations are supposed to be the same (*accelerated*, *decelerated*, or *linear*).

animation_time - The animation time in ms (i.e. 300 is 0.3 seconds).

upsound - The sound to play when container moves to *show* position.

downsound - The sound to play when container moves to *hide* position.

fade_time - The time to fade in.

fade_type - The type of fade in. (always *linear*).

The following attributes are rarely or never used:

dontRender

horizontalBorder

verticalBorder

scrollbarType[编辑 | 编辑源代码]

The **scrollbarType** is used to define which elements a scrollbar is composed of.

The following attributes are used:

name - The scrollbar name.

slider - The button element to use as the scrollbar slider.

track - The button element to use as the scrollbar tracker.

leftbutton - The button element to use as the scrollbar left increment button.

rightbutton - The button element to use as the scrollbar right increment button.

position - The screen position of the scrollbar.

size - The bounding box for the scrollbar.

priority - The priority the scrollbar has over other elements.

borderSize - The bounding box for border of the scrollbar.

maxValue - The maximum value the scrollbar moves to (used to control the increments).

minValue - The minimum value the scrollbar moves to (used to control the increments).

stepSize - The size of increments using the increment button.

startValue - The initial size the slider for the scrollbar starts at.

horizontal - Sets whether the scrollbar is horizontal (1) or vertical (0).

Elements[编辑 | 编辑源代码]

There are multiple element types used within containers. All elements must be used within containers, they will not work outside of one.

Elements will inherit the orientation of the containers they are located in unless the orientation is specified for the element itself.

The following elements can be freely added and are usable with scripted GUIs.

iconType - Used for static images.

instantTextBoxType - Used for text.

buttonType - Used for buttons

The following elements do rely on internal code. You can add new elements, they will not be populated with data.

smoothListboxType - Used for smooth scrollable lists.

listboxType - Used for scrollable lists.

checkboxType - Used for checkboxes.

OverlappingElementsBoxType - Used for overlapping many elements.

editBoxType - Used for editable textboxes.

shieldtype - Used for to display country flags.

The following elements are legacy:

guiButtonType - Same as *buttonType*.

textBoxType - Same as *instantTextBoxType*.

eu3dialogtype - Same as *windowType*.

shieldtype - Only used within *eu3dialogtype*. Different elements are used for flags in more recent files.

iconType[编辑 | 编辑源代码]

The *iconType* element is used to add images to the interface. It's usage overlaps with *buttonType*, which is similar but operates as a button.

The following attributes are used:

name - The icon name.

position - The screen position of the icon.

orientation - Sets the orientation origin for the *position* attribute.

spriteType - The image to use for the icon. Refers to a *spriteType* definition.

quadTextureSprite - The image to use for the icon. Refers to a dynamic *spriteType* definitions (i.e. flags) or multi-frame *spriteType* definitions.

frame - Which frame to use for the icon when using a multi-frame image.

alwaystransparent - Forces the icon to allow click through, i.e. clicking on an element behind another element.

hint_tag - Set the hint key the icon uses to display a hint tooltip with when hovered over.

pdx_tooltip - The tooltip that is shown when hovering over the button.

pdx_tooltip_delayed - Sets the delayed tooltip to display to the player. Takes a localization key.

centerposition - Sets whether the position is from the center of the icon.

instantTextBoxType[编辑 | 编辑源代码]

The *instantTextBoxType* element is used to add text to the interface. In some instances, the text for the element is generated internally (i.e. *regiment_count*). In these instances you cannot edit the text unless it is exposed in an localized string.

The following attributes are used:

name - The textbox name.

position - The screen position of the textbox.

orientation - Sets the orientation origin for the *position* attribute.

text - The text displayed by the textbox.

font - The font to use for the text.

maxWidth - The total width in pixels at which text is displayed.

maxHeight - The total height in pixels at which text is displayed.

format - How the text is aligned.

fixedsize - Whether the textbox should truncate text that exceeds its limits.

borderSize - The bounding box for the border of the textbox.

alwaystransparent - Forces the text to allow click through, i.e. clicking on an element behind another element.

pdx_tooltip - Sets the tooltip to display to the player. Takes a localization key.

pdx_tooltip_delayed - Sets the delayed tooltip to display to the player. Takes a localization key.

The following attributes are rarely or never used:

textureFile - Never used for anything.

Valid *format* values:

left

centre

right

buttonType[编辑 | 编辑源代码]

The *guiButtonType'* element is used to add buttons to the interface. Buttons are composed of an image and text, so they operate in a similar manner to *iconType* and *instantTextBoxType*.

The following attributes are used:

name - The button name.

position - The screen position of the button.

orientation - Sets the orientation origin for the *position* attribute.

spriteType - The image to use for the button. Refers to a *spriteType* definition.

quadTextureSprite - The image to use for the button. Refers to a dynamic *spriteType* definitions (i.e. flags) or multi-frame *spriteType* definitions.

frame - Which frame to use for the button when using a multi-frame image.

alwaystransparent - Forces the button to allow click through, i.e. clicking on an element behind another element.

buttonText - The text displayed by the button.

buttonFont - The font to display the button text in.

shortcut - The shortcut to add for this button.

clicksound - The sound to use when clicked.

oversound - The sound to play when hovered over.

hint_tag - Set the hint key the icon uses to display a hint tooltip with when hovered over.

pdx_tooltip - The tooltip that is shown when hovering over the button.

pdx_tooltip_delayed - Sets the delayed tooltip to display to the player. Takes a localization key.

scale - Scales the button size.

web_link - URL to open in browser

The following attributes are rarely or never used:

tooltip - Never used.

tooltipText - Never used.

delayedTooltipText - Never used.

smoothListboxType[编辑 | 编辑源代码]

The *smoothListboxType* element is used to define a listbox, which is a scrollable list that is populated with entries. Typically these elements are internally linked with another container, which composes the actual entry used in the list box.

The following attributes are used:

name - The listbox name.

position - The screen position of the listbox.

orientation - Sets the orientation origin for the *position* attribute.

size - The bounding box for the listbox.

spacing - The spacing to use between listbox entries.

'horizontal - Whether the listbox is horizontal (1) or vertical (0).

scrollbartype - The scrollbar to use for the listbox.

bordersize - The bounding box for the border for the listbox.

alwaystransparent - Forces the listbox to allow click through, i.e. clicking on an element behind another element.

The following attributes are rarely or never used:

background - Never used.

listboxType[编辑 | 编辑源代码]

The *listboxType* element is used to define a listbox, which is a scrollable list that is populated with entries. Typically these elements are internally linked with another container, which composes the actual entry used in the list box.

The following attributes are used:

name - The listbox name.

position - The screen position of the listbox.

orientation - Sets the orientation origin for the *position* attribute.

size - The bounding box for the listbox.

spacing - The spacing to use between listbox entries.

'*horizontal* - Whether the listbox is horizontal (1) or vertical (0).

scrollbartype - The scrollbar to use for the listbox.

bordersize - The bounding box for the border for the listbox.

alwaystransparent - Forces the listbox to allow click through, i.e. clicking on an element behind another element.

The following attributes are rarely or never used:

background - Never used.

checkboxType[编辑 | 编辑源代码]

The *checkboxType* element is used to add checkboxes to the interface. The actual effect of the checkbox is defined internally.

The following attributes are used:

name - The checkbox name.

position - The screen position of the checkbox.

orientation - Sets the orientation origin for the *position* attribute.

spriteType - The image to use for the checkbox. Refers to a *spriteType* definition.

quadTextureSprite - The image to use for the checkbox. Refers to a dynamic *spriteType* definitions (i.e. flags) or multi-frame *spriteType* definitions.

frame - Which frame to use for the checkbox when using a multi-frame image.

alwaystransparent - Forces the checkbox to allow click through, i.e. clicking on an element behind another element.

buttonText - The text displayed by the checkbox.

buttonFont - The font to display the checkbox text in.

shortcut - The shortcut to add for this checkbox.

clicksound - The sound to use when clicked.

hint_tag - Set the hint key the checkbox uses to display a hint tooltip with when hovered over.

pdx_tooltip - Set the short tooltip this checkbox uses.

pdx_tooltip_delayed - Set the full tooltip this checkbox uses.

scale - Scales the checkbox size.

The following attributes are rarely or never used:

tooltip - Never used.

tooltipText - Never used.

delayedTooltipText - Never used.

editBoxType[编辑 | 编辑源代码]

The *editBoxType* element is used to add editable textboxes to the interface.

The following attributes are used:

name - The textbox name.

position - The screen position of the textbox.

orientation - Sets the orientation origin for the *position* attribute.

text - The text displayed by the textbox.

font - The font to use for the text.

maxWidth - The total width in pixels at which text is displayed.

maxHeight - The total height in pixels at which text is displayed.

format - How the text is aligned.

fixedsize - Whether the textbox should truncate text that exceeds its limits.

borderSize - The bounding box for the border of the textbox.

alwaystransparent - Forces the text to allow click through, i.e. clicking on an element behind another element.

ignore_tab_navigation - Makes the element ignore tab navigation.

OverlappingElementsBoxType[编辑 | 编辑源代码]

The *OverlappingElementsBoxType* element is used to define a special kind of listbox that dynamically overlaps sub-elements within itself.

The following attributes are used:

name - The listbox name.

position - The screen position of the listbox.

orientation - Sets the orientation origin for the *position* attribute.

size - The bounding box for the listbox.

spacing - The spacing to use between listbox entries.

'*horizontal* - Whether the listbox is horizontal (1) or vertical (0).

bordersize - The bounding box for the border for the listbox.

alwaystransparent - Forces the listbox to allow click through, i.e. clicking on an element behind another element.

The following attributes are rarely or never used:

textureFile - Never used for anything.

Valid *format* values:

left

centre

right

Position and Orientation[编辑 | 编辑源代码]

Understanding how the position coordinates work is important to understanding how to edit interfaces correctly.

The *orientation* attribute used by an element (or container) informs you where the anchor point for the element is. For example, *UPPER_LEFT* means the *(0, 0)* position is in the top left corner of the screen/container, whereas *LOWER_RIGHT* is the bottom right.

The *position* attribute then works from this origin position. Therefore if you want an element to be on the opposite side of the screen, you'd want to change the *orientation* first.

This is system is used because it allows the interface to adapt to different resolutions. Because all positions are tied to a specific origin that is dynamically calculated from the player's screen resolution, the placement of containers and elements are kept consistent.

So, if you want to ensure your interface edits are consistent, you must use the *orientation* attribute properly.

An mistake would be to edit the *position* solely, for example `position = { x = 1800 y = 200 }`. This would look fine for anybody with a screen width greater than 1800px (i.e. 1900 x 1200 upwards), but the element would be off screen for those with smaller screens.

Valid *orientation* values (can be upper or lowercase):

CENTER

UPPER_LEFT

LOWER_LEFT

UPPER_RIGHT

LOWER_RIGHT

Fonts[编辑 | 编辑源代码]

See also: Font modding

The following fonts are usable in Hearts of Iron IV:

Arial12

Arial12_bold

cg_16b

cg_18b

garamond_12

garamond_14

garamond_14_bold

garamond_16

garamond_16_bold

garamond_24

hoi_16mbs

hoi_16tooltip3

hoi_16typewriter

hoi_18

hoi_18b

hoi_18mbs

hoi_20b

hoi_20bs

hoi_22tech

hoi_22typewriter

hoi_24header

hoi_26mbs

hoi_30header

hoi_33

hoi_36header

hoi4_typewriter16

hoi4_typewriter22

hoi_mapfont4

hoi_arrow_font

newsfeed_body

newsfeed_title

standard

standard_18

standard_22

tahoma_20_bold

vic_18

vic_18_grey

vic_18s

vic_22

vic_22_bl

vic_22s

vic_36

vic_36s