FLOW - UI Toolkit Extended

Flow is a Unity asset designed to extend the capabilities of the UI Toolkit. It provides a set of extension functions and custom visual elements that allow developers to build complex and responsive UI layouts using a fluent and chainable API. The asset's design promotes a clean, readable, and maintainable codebase by abstracting common styling and layout operations into concise, expressive method calls.
Key Features

- FLOW API: Chainable methods such as Expand(), Border(), Margin(), and BGColor() enable quick styling adjustments.
- Custom Visual Elements: Includes elements like Div, Row, Column, Circle, and Rectangle that extend Unity's native VisualElement for specialized layouts.
- Responsive Layouts: Easily create dynamic and responsive editor and runtime UI with built-in methods for spacing, alignment, and scaling.
- Easy Integration: Designed to work seamlessly with Unity Editor scripts, enabling rapid prototyping and efficient UI development.

API Documentation

Namespace: SABI.Flow

Alignment

- AlignContent<T>(this T element, StyleEnum<Align> alignContent)
  Description: Sets the alignContent style property of a VisualElement.

- AlignItems<T>(this T element, StyleEnum<Align> alignItems)
  Description: Applies the alignItems style property to a VisualElement
- AlignSelf<T>(this T element, StyleEnum<Align> alignSelf)
  Description: Sets the alignSelf property, allowing an element to override its container's alignment settings.
- JustifyContent<T>(this T element, Justify justify)
  Description: Configures the justifyContent style property, determining how space is distributed among child elements.
- Flex<T>(this T element)
  Description: Sets the display style of the element to Flex, enabling flexbox layout.
- FlexBasis<T>(this T element, Length flexBasis)
  Description: Defines the initial main size of an element before any flex-grow or flex-shrink adjustments.
- FlexDirection<T>(this T element, FlexDirection direction)
  Description: Specifies the direction of the flex layout (e.g., row or column).
- FlexGrow<T>(this T element, float grow = 1)
  Description: Determines how much the element will grow relative to its siblings.
- FlexShrink<T>(this T element, float shrink)
  Description: Defines the shrink factor of the element when there is insufficient space.
- FlexWrap<T>(this T element, Wrap wrap)
  Description: Sets the flexWrap property to control whether flex items should wrap onto multiple lines.
- CenterF<T>(this T element)
  Description: Centers the element using flexbox properties.
- CenterT<T>(this T element)
  Description: Centers the element using absolute positioning.

Animations

- TransitionDelay<T>(this T element, StyleList<TimeValue>? delay = null)
  Description: Sets the transition delay for a VisualElement using a provided StyleList<TimeValue>. If no delay is specified, a default delay of 1 second is used.
- TransitionDelay<T>(this T element, float delay)
  Description: Overload that accepts a float value for delay in seconds.
- Transition DurationTransitionDuration<T>(this T element, StyleList<TimeValue>? duration = null)
  Description: Sets the transition duration using a provided StyleList<TimeValue>. Defaults to 0.2 seconds if no duration is provided.
- TransitionDuration<T>(this T element, float duration)
  Description: Overload that accepts a float value for duration.
- TransitionProperty<T>(this T element, StyleList<StylePropertyName>? property = null)
  Description: Sets the transition property for the element using a StyleList<StylePropertyName>. Defaults to transitioning all properties if not provided.
- TransitionProperty<T>(this T element, string property)
  Description: Overload that accepts a string for the transition property.
- TransitionTimingFunction<T>(this T element)
  Description: Sets the transition timing function to the default easing mode (EaseInOutSine).
- TransitionTimingFunction<T>(this T element, EasingMode timingFunction)
  Description: Overload that accepts a custom easing mode.
- Animate<T>(this T element, float duration = AnimationDuration, EasingMode easingMode = AnimationTimingFunction, float delay = AnimationDelay, string property = AnimationTransitionProperty)
  Description: A convenience method that chains the transition delay, duration, property, and timing function methods to apply a complete animation transition.

Border

- BorderWidth<T>(this T element, float value = DefaultBorderWidth)
  Description: Sets the same border width on all four sides.
- BorderWidth<T>(this T element, float topBottomValue, float leftRightValue)
  Description: Sets a specific width for top/bottom and left/right.
- BorderWidth<T>(this T element, float top, float bottom, float left, float right)
  Description: Allows separate values for each side.
- Methods like BorderWidthTop, BorderWidthBottom, BorderWidthLeft, and BorderWidthRight allow setting border width on specific sides.
- BorderWidthTopBottom and BorderWidthLeftRight provide an easy way to set both top & bottom or left & right simultaneously.
- BorderRadius<T>(this T element, Length? value = null)
  Description: Sets the same radius for all four corners.
- BorderRadius<T>(this T element, Length topBottomValue, Length leftRightValue)
  Description: Sets one radius for the top and bottom and another for the left and right.
- BorderRadius<T>(this T element, Length topLeft, Length topRight, Length bottomRight, Length bottomLeft)
  Description: Assigns different radius values to each corner.
- Methods like BorderRadiusTopLeft, BorderRadiusTopRight, BorderRadiusBottomLeft, and BorderRadiusBottomRight let you set the radius for one specific corner.
- BorderRadiusTop, BorderRadiusBottom, BorderRadiusLeft, and BorderRadiusRight are used to set the radius for paired corners.
- BorderColor<T>(this T element, StyleColor? color = null)
  Description: Sets the same color for all sides; defaults to white.
- BorderColor<T>(this T element, StyleColor topBottomValue, StyleColor leftRightValue)
  Description: Sets different colors for top/bottom and left/right.

- BorderColor<T>(this T element, StyleColor bottomColor, StyleColor topColor, StyleColor leftColor, StyleColor rightColor)
  Description: Allows for distinct colors for each side.
- BorderColorRandom<T>(this T element) sets a random color.
- NoBorderColor<T>(this T element) removes the border color by setting it to clear.
- Similar to width, methods like BorderColorTop, BorderColorBottom, BorderColorLeft, and BorderColorRight let you set colors on specific sides.
- BorderColorTopBottom and BorderColorLeftRight let you set colors for the top & bottom or left & right sides simultaneously.

Box

- public static T Enable<T>(this T element) where T : VisualElement
  Description: Sets the VisualElement to an enabled state.
- public static T Disable<T>(this T element) where T : VisualElement
  Description: Disables the VisualElement, making it non-interactive.
- public static T Cursor<T>(this T element, StyleCursor cursor) where T : VisualElement
  Description: Sets the cursor style using a StyleCursor.
- public static T Cursor<T>(this T element, UnityEngine.UIElements.Cursor cursor) where T : VisualElement
  Description: Sets the cursor style using a
- public static T Display<T>(this T element, DisplayStyle display) where T : VisualElement
  Description: Sets the element's display style.
- public static T Show<T>(this T element) where T : VisualElement
  Description: Makes the element visible by setting its display to Flex.
- public static T Hide<T>(this T element) where T : VisualElement
  Description: Hides the element by setting its display style to None.
- public static T Overflow<T>(this T element, StyleEnum<Overflow> overflow) where T : VisualElement

Description: Sets the overflow style, controlling how content that exceeds bounds is handled.

- public static T OverflowHidden<T>(this T element) where T : VisualElement
  Description: Sets the overflow to Hidden.
- public static T OverflowVisible<T>(this T element) where T : VisualElement
  Description: Sets the overflow to Visible.
- public static T Tooltip<T>(this T element, string tooltip) where T : VisualElement
  Description: Assigns a tooltip string to the element for providing additional information.
- public static T SetStyle<T>(this T element, T transferer) where T : VisualElement
  Description: Copies style properties from one VisualElement to another using the FlowUtil.CopyStyle method.

Color

- BGColor<T>(this T element, StyleColor? color = null)
  Description: Sets the background color of the element to the specified StyleColor, or to the default if none is provided.
- BGColor<T>(this T element, float r, float g, float b, float a = 1)
  Description: Sets the background color using individual red, green, blue, and optional alpha values.
- BGColor<T>(this T element, float rgb, float a = 1)
  Description: Sets a grayscale background color by applying the same value for red, green, and blue.
- BGColorEditorDefault<T>(this T element)
  Description: Applies the default background color defined for the Unity editor by retrieving it via FlowUtil.GetDefaultEditorBGColor().
- BGColorRandom<T>(this T element, StyleColor? color = null)
  Description: Sets a random background color using Random.ColorHSV(), ensuring a unique appearance.

- NoBGColor<T>(this T element)
  Description: Clears the background color by setting it to transparent (Color.clear).
- BGColor<T>(this T element, Color color1, Color color2, GradientDirection gradientDirection = GradientDirection.Horizontal, float gradientInfluence = 0.5f)
  Description: Applies a gradient background to the element. The gradient is created by blending color1 and color2 along the specified direction with a defined influence.
- BGGradientHorizontal<T>(this T element, Color color1, Color color2, float gradientInfluence = 0.5f)
  Description: Convenience method for applying a horizontal gradient.
- BGGradientVertical<T>(this T element, Color color1, Color color2, float gradientInfluence = 0.5f)
  Description: Convenience method for applying a vertical gradient.
- Opacity<T>(this T element, float value = 0.5f)
  Description: Sets the opacity of the element to the specified value, allowing for transparency effects.
- TextColor<T>(this T element, StyleColor? color = null)
  Description: Sets the text color of the element. If no color is provided, it uses the default text color.

Composition

- public static T Container<T>(this T element) where T : VisualElement
  Description:Applies a combination of border styling, margin, padding, and expansion to create a container.
- Internally, it calls Border(), Margin(), Padding(), and Expand() in sequence.
- public static T Border<T>(this T element) where T : VisualElement
  Description:Applies border styling to the element by setting border radius, border width, and border color.

- public static T Row<T>(this T element) where T : VisualElement
  Description:Configures the element to display its children in a horizontal layout by enabling flex and setting the flex direction to Row.
- public static T Column<T>(this T element) where T : VisualElement
  Description:Sets the element to arrange its children vertically by setting the flex direction to Column.
- public static T ColumnReverse<T>(this T element) where T : VisualElement
  Description:
- Arranges the children of the element in a reversed vertical order by setting the flex direction to ColumnReverse.
- public static T RowReverse<T>(this T element) where T : VisualElement
  Description:Arranges the children of the element in a reversed horizontal order by setting the flex direction to RowReverse.
- public static T Expand<T>(this T element, float grow = 1) where T : VisualElement
  Description:Sets the flex grow property of the element, allowing it to expand and fill available space.

Events

- public static T OnClick<T>(this T element, Action callback) where T : VisualElement
  Description: Registers a callback to be invoked on a click event.
- public static T OnMouseEnter<T>(this T element, Action callback) where T : VisualElement
  Description: Invokes the callback when the mouse enters the element's bounds.
- public static T OnMouseLeave<T>(this T element, Action callback) where T : VisualElement
  Description: Invokes the callback when the mouse leaves the element.

- public static T OnMouseDown<T>(this T element, Action callback) where T : VisualElement
- public static T OnMouseUp<T>(this T element, Action callback) where T : VisualElement
- Description: Registers callbacks for when the mouse button is pressed or released.
- public static T OnHover<T>(this T element, Action<T> onEnter, Action<T> onExit) where T : VisualElement
  Description: Combines mouse enter and leave events to handle hover effects. The onEnter callback is triggered when the mouse enters, and the onExit callback is triggered when the mouse leaves. Note: This method first invokes onExit to ensure any pre-hover state is handled before applying the animation and registering the callbacks.

Font

- public static T FontSize<T>(this T element, float size = DefaultFontSize) where T : VisualElement
  Description: Sets the font size of the element.
- public static T FontColor<T>(this T element, Color color) where T : VisualElement
  Description: Sets the font color.
- public static T Italic<T>(this T element) where T : VisualElement
  Description: Applies italic styling by setting the font style to Italic.
- public static T Bold<T>(this T element) where T : VisualElement
  Description: Applies bold styling to the text.
- public static T BoldAndItalic<T>(this T element) where T : VisualElement
  Description: Applies both bold and italic styling.
- public static T NoFontStyle<T>(this T element) where T : VisualElement
  Description: Resets font style to normal.

- public static T FontStyleAndWeight<T>(this T element, FontStyle fontStyle) where T : VisualElement
  Description: Sets a custom font style and weight.
- public static T FontAsset<T>(this T element, Font font) where T : VisualElement
- Description: Applies a specific font asset to the element.
- public static T UnityTextAlign<T>(this T element, StyleEnum<TextAnchor> align) where T : VisualElement
  Description: Sets the text alignment using a StyleEnum<TextAnchor>.
- public static T UnityTextAlignCenter<T>(this T element) where T : VisualElement
  Description: Centers the text by default.
- public static T TextOverflow<T>(this T element, StyleEnum<TextOverflow>? overflow = null) where T : VisualElement
  Description: Sets how text overflow is handled, defaulting to ellipsis if no value is provided.
- public static T TextShadow<T>(this T element, StyleTextShadow shadow) where T : VisualElement
  Description: Applies a shadow effect to the text.
- public static T TextOutlineColor<T>(this T element, StyleColor? color = null) where T : VisualElement
  Description: Sets the outline color for the text; defaults to red if not specified.
- public static T TextOutlineWidth<T>(this T element, float width = 1) where T : VisualElement
  Description: Sets the outline width.
- public static T LetterSpacing<T>(this T element, float spacing) where T : VisualElementpublic static T WordSpacing<T>(this T element, float spacing) where T : VisualElement
  Description: Adjust the spacing between letters and words, respectively.
- public static T UnityParagraphSpacing<T>(this T element, float spacing) where T : VisualElement
  Description: Sets the spacing between paragraphs.

- public static T UnityTextOverflowPosition<T>(this T element, TextOverflowPosition position) where T : VisualElement
  Description: Sets the position where text overflow occurs.
- public static T H1<T>(this T element) where T : VisualElement
  Description: Applies a bold font style, sets the font size to 32, and applies a top-bottom margin of 21.4f.
- public static T H2<T>(this T element) where T : VisualElement
  Description: Applies a bold font style with a font size of 24 and a margin of 18.
- H3, H4, H5, H6
  Description: Each method applies a descending scale for font size and margin, ensuring a visual hierarchy. For example, H3 uses 18.72f font size with a margin of 15.6f, H4 uses 16 with 12, H5 uses 13.28f with 9, and H6 uses 10.72f with 7.

Background Image

- public static T BackgroundImage<T>(this T element, Background background) where T : VisualElement
  Description:Sets the background image of the element using a provided Background object.
- public static T BackgroundPositionX<T>(this T element, StyleBackgroundPosition positionX) where T : VisualElement
  Description:Sets the horizontal position of the background image.
- public static T BackgroundPositionY<T>(this T element, StyleBackgroundPosition positionY) where T : VisualElement
  Description:Sets the vertical position of the background image.

Margine

- Margin<T>(this T element, Length? value = null)
  Description:Applies the same margin on all sides of the element. If no value is provided, it defaults to 15.

- Margin<T>(this T element, Length topBottomValue, Length leftRightValue)
  Description:Sets the top and bottom margins to topBottomValue and the left and right margins to leftRightValue.
- Margin<T>(this T element, Length top, Length bottom, Length left, Length right)
  Description:Allows specifying individual margin values for the top, bottom, left, and right sides.
- MarginTop<T>(this T element, Length? value = null)
  Description:Sets the top margin of the element. Defaults to 15 if no value is provided.
- MarginBottom<T>(this T element, Length? value = null)
  Description:Sets the bottom margin of the element, using the default value if none is specified.
- MarginLeft<T>(this T element, Length? value = null)
  Description:Sets the left margin of the element with an optional custom value.
- MarginRight<T>(this T element, Length? value = null)
  Description:Sets the right margin of the element, using the default value if not provided.
- MarginTopBottom<T>(this T element, Length? value = null)
  Description:Applies the same margin to both the top and bottom sides.
- MarginLeftRight<T>(this T element, Length? value = null)
  Description:Applies the same margin to both the left and right sides.

Padding

- public static T Padding<T>(this T element, Length? value = null) where T : VisualElement
  Description:Sets the same padding on all four sides (top, bottom, left, right).If no value is provided, the default padding of 15 is applied.
- public static T Padding<T>(this T element, Length topBottomValue, Length leftRightValue) where T : VisualElement

Description:Sets the top and bottom padding to topBottomValue and the left and right padding to leftRightValue.

- public static T Padding<T>(this T element, Length top, Length bottom, Length left, Length right) where T : VisualElement
  Description:
- Allows specifying separate padding values for the top, bottom, left, and right sides.
- public static T PaddingTop<T>(this T element, Length? value = null) where T : VisualElement
  Description:Sets the padding for the top side, defaulting to 15 if no value is provided.
- public static T PaddingBottom<T>(this T element, Length? value = null) where T : VisualElement
  Description:Sets the padding for the bottom side with an optional custom value.
- public static T PaddingLeft<T>(this T element, Length? value = null) where T : VisualElement
  Description:Sets the left padding of the element.
- public static T PaddingRight<T>(this T element, Length? value = null) where T : VisualElement
  Description:Sets the right padding of the element.
- public static T PaddingTopBottom<T>(this T element, Length? value = null) where T : VisualElement
  Description:Applies the same padding to both the top and bottom sides.
- public static T PaddingLeftRight<T>(this T element, Length? value = null) where T : VisualElement
  Description:
- Applies the same padding to both the left and right sides.

Size

- public static T Size<T>(this T element, Length value) where T : VisualElement

Description:Sets both the height and width of the element to the same value.

- public static T FixedSize<T>(this T element, Length width, Length height) where T : VisualElement
  Description:Fixes the element's size by setting both its minimum and maximum sizes to the specified width and height.
- public static T FixedSize<T>(this T element, Length size) where T : VisualElement
  Description:Sets a square fixed size by applying the same value to both width and height constraints.
- public static T Size<T>(this T element, Length width, Length height) where T : VisualElement
  Description:Sets the element’s width and height individually.
- public static T Width<T>(this T element, Length width) where T : VisualElement
  Description:Sets the element’s width.
- public static T Height<T>(this T element, Length height) where T : VisualElement
  Description:Sets the element’s height.
- public static T FixedWidth<T>(this T element, Length width) where T : VisualElement
  Description:Sets a fixed width by applying the same value to both the element’s minimum and maximum widths.
- public static T FixedHeight<T>(this T element, Length height) where T : VisualElement
  Description:Sets a fixed height by applying the same value to both the element’s minimum and maximum heights.
- public static T MinSize<T>(this T element, Length value) where T : VisualElement
  Description:Sets both the minimum width and height to the specified value.
- public static T MinSize<T>(this T element, Length width, Length height) where T : VisualElement
  Description:Sets the minimum width and height individually.

- public static T MaxSize<T>(this T element, Length value) where T : VisualElement
  Description:Sets both the maximum width and height to the specified value.
- public static T MaxSize<T>(this T element, Length width, Length height) where T : VisualElement
  Description:Sets the maximum width and height individually.
- public static T MinHeight<T>(this T element, Length minHeight) where T : VisualElement   public static T MaxHeight<T>(this T element, Length maxHeight) where T : VisualElement
  Description:Sets the minimum or maximum height respectively.
- public static T MinWidth<T>(this T element, Length minWidth) where T : VisualElement
- public static T MaxWidth<T>(this T element, Length maxWidth) where T : VisualElement
  Description:Sets the minimum or maximum width respectively.

Transform

- public static T Rotate<T>(this T element, StyleRotate rotate) where T : VisualElement
  Description:Sets the rotation of the element using a StyleRotate object.
- public static T Rotate<T>(this T element, float rotate) where T : VisualElement
- public static T Scale<T>(this T element, StyleScale scale) where T : VisualElement
  Description:Applies a scale transformation to the element.
- Overloads: public static T Scale<T>(this T element, Vector3 scale) where T : VisualElement
- public static T Scale<T>(this T element, float scale) where T : VisualElement

- public static T Position<T>(this T element, Position position) where T : VisualElement
  Description:Sets the CSS-like positioning of the element.
- AbsolutePosition<T>(): Sets position to Absolute.
- RelativePosition<T>(): Sets position to Relative.
- public static T Translate<T>(this T element, StyleTranslate translate) where T : VisualElement
  Description:Applies a translation (movement) transformation to the element.
- Overloads:public static T Translate<T>(this T element, Vector3 translate) where T : VisualElement
- public static T Translate<T>(this T element, float x, float y, float z) where T : VisualElement
- public static T TransformOrigin<T>(this T element, Length x, Length y) where T : VisualElement
  Description:Sets the origin point for transform operations, affecting rotation and scaling.
- public static T Bottom<T>(this T element, Length value) where T : VisualElement
- public static T Left<T>(this T element, Length value) where T : VisualElement
- public static T Right<T>(this T element, Length value) where T : VisualElement
- public static T Top<T>(this T element, Length value) where T : VisualElement
  Description:Each method sets the corresponding directional property (bottom, left, right, or top) to a specified value.
- public static T StretchTopToBottom<T>(this T element, Length value) where T : VisualElement
  Description:Applies the same value to both the top and bottom properties.
- public static T StretchLeftToRight<T>(this T element, Length value) where T : VisualElement
  Description:Applies the same value to both the left and right properties.

- public static T Stretch<T>(this T element, Length value) where T : VisualElement
  Description:Sets the same value for top, bottom, left, and right, effectively stretching the element's boundary.