Table of Contents

Namespace UtilitySystem

This is The Namespace for the UtilitySystem, Assembly-CSharp

UtilitySystem.Extension

UtilitySystem.Tool

UtilitySystem.UI

Namespace UtilitySystem.Extension

Classes

FloatExtension

This Static Class extends the float Variable.

IntExtension

This Static Class extends the integer Variable.

TransformExtension

This Static Class extends the Transform type.

VisualElementExtension

Class FloatExtension

This Static Class extends the float Variable.

Inheritance

System.Object
FloatExtension

Namespace: UtilitySystem.Extension

Assembly: Assembly-CSharp.dll

Syntax

```
public static class FloatExtension
```

Methods

FlipValue(Single)

This Method will change an negative number to positive and positive to negative.

Declaration

```
public static float FlipValue(this float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	the float that will be flipped.

Returns

TYPE	DESCRIPTION
System.Single	return the flipped float.

RoundDown(Single)

This Method will round down the float to an interger.

Declaration

```
public static int RoundDown(this float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	the float that will be round down.

Returns

TYPE	DESCRIPTION
System.Int32	Return an integer of the float round down.

RoundUp(Single)

This Method will round up the float to an interger.

Declaration

```
public static int RoundUp(this float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	the float that will be round up.

Returns

TYPE	DESCRIPTION
System.Int32	Return an integer of the float round up.

Class IntExtension

This Static Class extends the integer Variable.

Inheritance

System.Object
IntExtension

Namespace: UtilitySystem.Extension

Assembly: Assembly-CSharp.dll

Syntax

```
public static class IntExtension
```

Methods

FlipValue(Int32)

This Method will change an negative number to positive and positive to negative.

Declaration

```
public static int FlipValue(this int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	the integer that will be flipped.

Returns

TYPE	DESCRIPTION
System.Int32	return the flipped integer.

HighestValue(Int32, Int32)

This Method will take the highest of 2 integers.

Declaration

```
public static int HighestValue(this int value, int compare)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	The integer that will check.
System.Int32	compare	The integer that will be compare.

Returns

TYPE	DESCRIPTION
System.Int32	Return the highest integer.

LowestValue(Int32, Int32)

This Method will take the lowest of 2 integers.

Declaration

```
public static int LowestValue(this int value, int compare)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	The integer that will check.
System.Int32	compare	The integer that will be compare.

Returns

TYPE	DESCRIPTION
System.Int32	Return the lowest integer.

Class TransformExtension

This Static Class extends the Transform type.

Inheritance

System.Object
TransformExtension

Namespace: UtilitySystem.Extension

Assembly: Assembly-CSharp.dll

Syntax

```
public static class TransformExtension
```

Methods

GetChildren(Transform, Boolean)

This Method will get a list of the children from a gameobject.

Declaration

```
public static List<GameObject> GetChildren(this Transform transform, bool includeInactive = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	This is the transform that will get it's children.
System.Boolean	includeInactive	If the param is true it will also includeInactive gameobject. else ignore.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<UnityEngine.GameObject>	return a list of GameObjects.

ResetTransformation(Transform)

This Method will reset the transform.

Declaration

```
public static void ResetTransformation(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	This is the transform that will get reset.

Class VisualElementExtension

Inheritance

System.Object
VisualElementExtension

Namespace: UtilitySystem.Extension

Assembly: Assembly-CSharp.dll

Syntax

```
public static class VisualElementExtension
```

Methods

BorderRadius(VisualElement, LengthUnit, Single[])

This Method will set the border radius of the element.

Declaration

```
public static void BorderRadius(this VisualElement element, LengthUnit unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit	unit	This will set that values to either pixel or percentage.
System.Single[]	args	These are the values for the styling.

BorderRadius(VisualElement, LengthUnit[], Single[])

This Method will set the border radius of the element.

Declaration

```
public static void BorderRadius(this VisualElement element, LengthUnit[] unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit[]	unit	These will set that values to either pixel or percentage.
System.Single[]	args	These are the values for the styling.

BorderStyleColor(VisualElement, Color[])

This Method will set the border color of the element.

Declaration

```
public static void BorderStyleColor(this VisualElement element, params Color[] color)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.Color[]	color	These will set the color of the border.

BorderWidth(VisualElement, Single[])

This Method will set the border width of the element.

Declaration

```
public static void BorderWidth(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
System.Single[]	args	These are the values for the styling.

Child(VisualElement, VisualElement, Boolean)

This Method will add, remove an element in this parent or clear the children from the parent.

Declaration

```
public static void Child(this VisualElement parent, VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	The parent that will add, remove a child element or clear all children.
UnityEngine.UIElements.VisualElement	content	This is an optional param if not null will allow for adding or remove of the content element. if null them clear all of the children from the element.
System.Boolean	isAdd	this is an optional param the will determine of the stylesheet will be added or remove from the element.

PickModeOFF(VisualElement)

This Method will not allow this element to be interactable

Declaration

```
public static void PickModeOFF(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's pickmode set to ignore.

PickModeON(VisualElement)

This Method will allow this element to be interactable.

Declaration

```
public static void PickModeON(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's pickmode set to position.

PositionBL(VisualElement, Single, Single)

This Method will set the position of the UI element on the bottom and left

Declaration

```
public static void PositionBL(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
System.Single	x	This is to set the left position.
System.Single	y	This is to set the bottom position.

PositionBL(VisualElement, Vector2)

This Method will set the position of the UI element on the bottom and left

Declaration

```
public static void PositionBL(this VisualElement element, Vector2 position)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
UnityEngine.Vector2	position	This is to set the bottom and left position.

PositionBR(VisualElement, Single, Single)

This Method will set the position of the UI element on the bottom and right

Declaration

```
public static void PositionBR(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
System.Single	x	This is to set the right position.
System.Single	y	This is to set the bottom position.

PositionBR(VisualElement, Vector2)

This Method will set the position of the UI element on the bottom and right

Declaration

```
public static void PositionBR(this VisualElement element, Vector2 position)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
UnityEngine.Vector2	position	This is to set the bottom and right position.

PositionTL(VisualElement, Single, Single)

This Method will set the position of the UI element on the top and left

Declaration

```
public static void PositionTL(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
System.Single	x	This is to set the left position.
System.Single	y	This is to set the top position.

PositionTL(VisualElement, Vector2)

This Method will set the position of the UI element on the top and left

Declaration

```
public static void PositionTL(this VisualElement element, Vector2 position)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
UnityEngine.Vector2	position	This is to set the top and left position.

PositionTR(VisualElement, Single, Single)

This Method will set the position of the UI element on the top and right

Declaration

```
public static void PositionTR(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
System.Single	x	This is to set the right position.

TYPE	NAME	DESCRIPTION
System.Single	y	This is to set the top position.

PositionTR(VisualElement, Vector2)

This Method will set the position of the UI element on the top and right

Declaration

```
public static void PositionTR(this VisualElement element, Vector2 position)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's position set.
UnityEngine.Vector2	position	This is to set the top and right position.

ResetStyleFloat(StyleFloat)

This Method will reset a stylefloat value to it's Initial value

Declaration

```
public static void ResetStyleFloat(this StyleFloat styleFloat)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleFloat	styleFloat	This is the style float that will be set to Initial.

ResetStyleLength(StyleLength)

This Method will reset stylelength value to it's Initial value

Declaration

```
public static void ResetStyleLength(this StyleLength styleLength)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleLength	styleLength	This is the style length that will be set to Initial.

SetBorderStyle(VisualElement, Single[])

This Method will set the border of the element.

Declaration

```
public static void SetBorderStyle(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
System.Single[]	args	These are the values for the styling.

SetBorderStyle(VisualElement, Color, Single[])

This Method will set the border of the element.

Declaration

```
public static void SetBorderStyle(this VisualElement element, Color color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.Color	color	This will set the color of the border.
System.Single[]	args	These are the values for the styling.

SetBorderStyle(VisualElement, Color[], Single[])

This Method will set the border of the element.

Declaration

```
public static void SetBorderStyle(this VisualElement element, Color[] color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.Color[]	color	These will set the color of the border.
System.Single[]	args	These are the values for the styling.

SetBorderStyle(VisualElement, LengthUnit, Color, Single[])

This Method will set the border of the element.

Declaration

```
public static void SetBorderStyle(this VisualElement element, LengthUnit unit, Color color, params float[]
args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit	unit	This will set that values to either pixel or percentage.
UnityEngine.Color	color	This will set the color of the border.
System.Single[]	args	These are the values for the styling.

SetBorderStyle(VisualElement, LengthUnit, Color[], Single[])

This Method will set the border of the element.

Declaration

```
public static void SetBorderStyle(this VisualElement element, LengthUnit unit, Color[] color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit	unit	This will set that values to either pixel or percentage.
UnityEngine.Color[]	color	These will set the color of the border.
System.Single[]	args	These are the values for the styling.

SetBorderStyle(VisualElement, LengthUnit[], Color, Single[])

This Method will set the border of the element.

Declaration

```
public static void SetBorderStyle(this VisualElement element, LengthUnit[] unit, Color color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit[]	unit	These will set that values to either pixel or percentage.
UnityEngine.Color	color	This will set the color of the border.
System.Single[]	args	These are the values for the styling.

SetBorderStyle(VisualElement, LengthUnit[], Color[], Single[])

This Method will set the border of the element.

Declaration

```
public static void SetBorderStyle(this VisualElement element, LengthUnit[] unit, Color[] color, params
float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit[]	unit	These will set that values to either pixel or percentage.
UnityEngine.Color[]	color	These will set the color of the border.
System.Single[]	args	These are the values for the styling.

SetFont(VisualElement, Font)

This Method will apply a font to this element.

Declaration

```
public static void SetFont(this VisualElement element, Font font)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have a font apply to it.
UnityEngine.Font	font	This is the font that will be set in the element.

SetMarginStyle(VisualElement, Single[])

This Method will set the margin and high of the element.

Declaration

```
public static void SetMarginStyle(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
System.Single[]	args	These are the values for the styling.

SetMarginStyle(VisualElement, LengthUnit, Single[])

This Method will set the margin and high of the element.

Declaration

```
public static void SetMarginStyle(this VisualElement element, LengthUnit unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit	unit	This will set that values to either pixel or percentage.
System.Single[]	args	These are the values for the styling.

SetMarginStyle(VisualElement, LengthUnit[], Single[])

This Method will set the margin and high of the element.

Declaration

```
public static void SetMarginStyle(this VisualElement element, LengthUnit[] unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit[]	unit	These will set that values to either pixel or percentage.
System.Single[]	args	These are the values for the styling.

SetPaddingStyle(VisualElement, LengthUnit, Single[])

This Method will set the padding and high of the element.

Declaration

```
public static void SetPaddingStyle(this VisualElement element, LengthUnit unit = LengthUnit.Pixel, params
float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit	unit	This will set that values to either pixel or percentage.
System.Single[]	args	These are the values for the styling.

SetPaddingStyle(VisualElement, LengthUnit[], Single[])

This Method will set the padding and high of the element.

Declaration

```
public static void SetPaddingStyle(this VisualElement element, LengthUnit[] unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit[]	unit	These will set that values to either pixel or percentage.
System.Single[]	args	These are the values for the styling.

SetSizeStyle(VisualElement, LengthUnit, Single[])

This Method will set the width and high of the element.

Declaration

```
public static void SetSizeStyle(this VisualElement element, LengthUnit unit = LengthUnit.Pixel, params
float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit	unit	This will set that values to either pixel or percentage.
System.Single[]	args	These are the values for the styling.

SetSizeStyle(VisualElement, LengthUnit, LengthUnit, Single[])

This Method will set the width and high of the element.

Declaration

```
public static void SetSizeStyle(this VisualElement element, LengthUnit unitWidth, LengthUnit unitHeight,
params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will .
UnityEngine.UIElements.LengthUnit	unitWidth	This will set that values to either pixel or percentage for the width.
UnityEngine.UIElements.LengthUnit	unitHeight	This will set that values to either pixel or percentage for the height.
System.Single[]	args	These are the values for the styling.

SetStyleColor(StyleColor, Color)

This Method will set the style color of the element.

Declaration

```
public static void SetStyleColor(this StyleColor styleColor, Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleColor	styleColor	
UnityEngine.Color	color	This will set the color of the style color.

SetTogglePick(VisualElement)

This Method will change the pickmode from position to ignore and vis vera.

Declaration

```
public static void SetTogglePick(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's pickmode changed.

StyleSheet(VisualElement, StyleSheet, Boolean)

This Method will add, remove a stylesheet in this parent or clear the stylesheets in the element.

Declaration

```
public static void StyleSheet(this VisualElement element, StyleSheet style = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will a style sheet added, removed or cleared.
UnityEngine.UIElements.StyleSheet	style	This is an optional param if not null will allow for adding or remove of this stylesheet. /// if null them clear all of the stylesheets from the element.
System.Boolean	isAdd	this is an optional param the will determine of the stylesheet will be added or remove from the element. this is an optional param the will determine of the stylesheet will be added or remove from the element.

Namespace UtilitySystem.Script

Classes

Assets

RichText

This static class will simplify setting up rich text for "Text Mesh Pro" or "UI ToolKit"

Class Assets

Inheritance

System.Object
Assets

Namespace: UtilitySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public static class Assets
```

Methods

GetAllInstances<T>()

This Generic Method is used to get all of the type T from the project.

Declaration

```
public static T[] GetAllInstances<T>()
    where T : ScriptableObject
```

Returns

TYPE	DESCRIPTION
T[]	return an array of T.

Type Parameters

NAME	DESCRIPTION
T	The type of ScriptableObject.

Class RichText

This static class will simplify setting up rich text for "Text Mesh Pro" or "UI ToolKit"

Inheritance

System.Object
RichText

Namespace: UtilitySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public static class RichText
```

Properties

NewLine

This Method will place an object and create a new line string.

Declaration

```
public static string NewLine { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Tab

This Method will place an object and create a tab gap string.

Declaration

```
public static string Tab { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

BoldText(Object)

This Method will place an object and create a string that is bold.

Declaration

```
public static string BoldText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is set to bold.

BooleanText(Boolean)

This Method will place an object and create a string that is either colored green for true or red for false.

Declaration

```
public static string BooleanText(this bool arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is either colored green for true or red for false.

ColoredText(Object, String)

This Method will place an object and create a string that will color the text

Declaration

```
public static string ColoredText(this object arg, string color)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string. This is the object that will be passed in and will be added into a string.
System.String	color	this is the color that will be used to color the text.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is colored from the color param.

ColoredText(Object, Color)

This Method will place an object and create a string that will color the text.

Declaration

```
public static string ColoredText(this object arg, Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
UnityEngine.Color	color	this is the color that will be used to color the text.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is colored from the color param.

ItalicText(Object)

This Method will place an object and create a string that is italic.

Declaration

```
public static string ItalicText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is given an italic lean.

LowerCaseText(Object)

This Method will return a string to an lowercase

Declaration

```
public static string LowerCaseText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is all lowercase.

PositionText(Object, Single)

This Method will place an object and create a string that is a set position for the text on the line.

Declaration

```
public static string PositionText(this object arg, float pos)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
System.Single	pos	This is a float that will be used to set the position of the text in the line.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is set at a position.

SetText(List<Func<String>>)

This Method will place an object and create a string that is either colored green for true or red for false.

Declaration

```
public static string SetText(List<Func<string>> messages)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.Func<System.String>>	messages	this is a list of Func strings that will be run and appended to a StringBuilder

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string from the StringBuilder.

SizeText(Object, Int32)

This Method will place an object and create a string that is the size for the text.

Declaration

```
public static string SizeText(this object arg, int size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
System.Int32	size	This is an integer for the size of the text.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that has it's size changed to the size param.

StrikeText(Object)

This Method will place an object and create a string that is strikethough.

Declaration

```
public static string StrikeText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that has a line though it.

SubText(Object)

This Method will place an object and create a string that is lowered.

Declaration

```
public static string SubText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is lower then other text.

SupText(Object)

This Method will place an object and create a string that is raised.

Declaration

```
public static string SupText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is higher then other text .

Text(Object)

This Method will place an object and create a string.

Declaration

```
public static string Text(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is a generic text string.

UnderLineText(Object)

This Method will place an object and create a string that is underline.

Declaration

```
public static string UnderLineText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is underlined.

UpperCaseText(Object)

This Method will return a string to an uppercase

Declaration

```
public static string UpperCaseText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This Method will return a string that is all uppercase.

Namespace UtilitySystem.Tool

Classes

CustomLogger

Timer

This Class is a Timer that can be used to activate an action after an amount of time.

Class CustomLogger

Inheritance

System.Object
CustomLogger

Namespace: UtilitySystem.Tool

Assembly: Assembly-CSharp.dll

Syntax

```
public static class CustomLogger
```

Properties

logEnabled

To runtime toggle debug logging [ON/OFF].

Declaration

```
public static bool logEnabled { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Break()

Pauses the editor.

Declaration

```
public static void Break()
```

Error(String)

A variant of Log that logs an error message to the console.

Declaration

```
public static void Error(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	String or object to be converted to string representation for display.

Error(String, Object)

A variant of Log that logs an error message to the console.

Declaration

```
public static void Error(string message, Object context)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	String to be converted to string representation for display.
UnityEngine.Object	context	Object to which the message applies.

Log(String)

Logs a message to the Unity Console By Type.

Declaration

```
public static void Log(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	String or object to be converted to string representation for display.

Log(String, Object)

Logs a message to the Unity Console.

Declaration

```
public static void Log(string message, Object context)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	String or object to be converted to string representation for display.
UnityEngine.Object	context	Object to which the message applies.

LogException(Exception)

A variant of Log that logs an exception message to the console.

Declaration

```
public static void LogException(Exception exception)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Exception	exception	Runtime Exception.

LogException(Exception, Object)

A variant of Log that logs an exception message to the console.

Declaration

```
public static void LogException(Exception exception, Object context)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Exception	exception	Runtime Exception.
UnityEngine.Object	context	Object to which the message applies.

Warning(String)

A variant of Log that logs a warning message to the console.

Declaration

```
public static void Warning(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	String or object to be converted to string representation for display.

Warning(String, Object)

A variant of Log that logs a warning message to the console.

Declaration

```
public static void Warning(string message, Object context)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	String or object to be converted to string representation for display.
UnityEngine.Object	context	Object to which the message applies.

Class Timer

This Class is a Timer that can be used to activate an action after an amount of time.

Inheritance

System.Object
Timer

Namespace: UtilitySystem.Tool

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class Timer
```

Constructors

Timer(Action, Single, Boolean)

This is a Constructor that will setup the timer.

Declaration

```
public Timer(Action action, float timer, bool isStopped = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	the Action that will be run when the timer finishes.
System.Single	timer	the amount of time it will take to finish.
System.Boolean	isStopped	This param if true will set the timer to stop from the start. (default false).

Properties

CurrentTime

Declaration

```
public int CurrentTime { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

ResetTimer()

This Method will reset the timer.

Declaration

```
public void ResetTimer()
```

StartTimer()

This Method will start the timer.

Declaration

```
public void StartTimer()
```

StopTimer()

This Method will stop the timer.

Declaration

```
public void StopTimer()
```

UpdateTimer()

This Method will update the timer.

Declaration

```
public void UpdateTimer()
```

Extension Methods

RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)

RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Namespace UtilitySystem.UI

Classes

DialogBox

DialogBox.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Icon

This Class is a VisualElement that is created for the Inventory.

Icon.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Slot

This Class is a VisualElement that is created for the Inventory.

Slot.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

SpriteElement

This Class is an VisualElement that is created for the Inventory Editor.

SpriteElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

ToolTip

ToolTip.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Class DialogBox

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
DialogBox

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class DialogBox : VisualElement, IEventHandler, ITransform, ITransitionAnimations,
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

DialogBox()

This is a Constructor and will create the dialog box.

Declaration

```
public DialogBox()
```

DialogBox(String, Vector2, StyleSheet)

This is a Constructor and will create the dialog box.

Declaration

```
public DialogBox(string titleText, Vector2 position, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	titleText	The text for the label that will be the title of the element.
UnityEngine.Vector2	position	The position to set the element at.
UnityEngine.UIElements.StyleSheet	style	The stylesheet for the element.

Methods

Close()

This Method will remove this element from it's parent.

Declaration

```
public void Close()
```

Collapse()

This Method will collapse the content box

Declaration

```
public void Collapse()
```

Content(VisualElement, Boolean)

This Method will add, remove an element or clear the whole content box.

Declaration

```
public void Content(VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	content	Is an optional element to be added to the content box.
System.Boolean	isAdd	Is an optional bool to determine of added or removed.

Footer(VisualElement, Boolean)

This Method will add, remove an element or clear the whole foot bar.

Declaration

```
public void Footer(VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	content	Is an optional element to be added to the foot bar.

TYPE	NAME	DESCRIPTION
System.Boolean	isAdd	Is an optional bool to determine of added or removed.

Maximize()

This Method will expand the element to cover the whole screen.

Declaration

```
public void Maximize()
```

Title(VisualElement, Boolean)

This Method will add, remove an element or clear the whole title bar.

Declaration

```
public void Title(VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	content	Is an optional element to be added to the title bar.
System.Boolean	isAdd	Is an optional bool to determine of added or removed.

UpdatePosition(Vector2)

This Method will update the position of the element on screen and maintain its bound within the screens bounds.

Declaration

```
public void UpdatePosition(Vector2 anchoredPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	anchoredPosition	This is the vector2 position it is trying to move to.

WindowDragInit()

This Method when run will allow for the element to be moved by the player.

Declaration

```
public void WindowDragInit()
```

Remarks

Parent Element needed and cover the whole screen.

Extension Methods

VisualElementExtension.BorderStyleColor(VisualElement, Color[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit, Single[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit[], Single[])
VisualElementExtension.BorderWidth(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetMarginStyle(VisualElement, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.SetTogglePick(VisualElement)
VisualElementExtension.PickModeOFF(VisualElement)
VisualElementExtension.PickModeON(VisualElement)
VisualElementExtension.SetFont(VisualElement, Font)
VisualElementExtension.PositionTL(VisualElement, Vector2)
VisualElementExtension.PositionTL(VisualElement, Single, Single)
VisualElementExtension.PositionTR(VisualElement, Vector2)
VisualElementExtension.PositionTR(VisualElement, Single, Single)
VisualElementExtension.PositionBL(VisualElement, Vector2)
VisualElementExtension.PositionBL(VisualElement, Single, Single)
VisualElementExtension.PositionBR(VisualElement, Vector2)
VisualElementExtension.PositionBR(VisualElement, Single, Single)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)

RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class DialogBox.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object
UnityEngine.UIElements.UxmlFactory<DialogBox, UnityEngine.UIElements.VisualElement.UxmlTraits>
DialogBox.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<DialogBox, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class Icon

This Class is a VisualElement that is created for the Inventory.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
Icon

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class Icon : VisualElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures,
IVisualElementScheduler, IResolvedStyle
```

Constructors

Icon()

This is a Constructor and will set the style for the Icon.

Declaration

```
public Icon()
```

Icon(Sprite, Int32, Boolean, StyleSheet)

Declaration

```
public Icon(Sprite image, int amount, bool canStack, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Sprite	image	
System.Int32	amount	
System.Boolean	canStack	
UnityEngine.UIElements.StyleSheet	style	

Methods

ItemAmount(Int32)

This method will create a Label for the amount that the item currently has. Set the style of the Label and adds to the Icon.

Declaration

```
public void ItemAmount(int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amount	the amount that the item currently has

Style()

This Method will set the style of the Icon.

Declaration

```
public void Style()
```

UpdateIcon(Int32)

This Method will update the Icon with a new amount.

Declaration

```
public bool UpdateIcon(int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amount	The amount to update item amount.

Returns

TYPE	DESCRIPTION
System.Boolean	Return true when updated.

Extension Methods

VisualElementExtension.BorderStyleColor(VisualElement, Color[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit, Single[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit[], Single[])

VisualElementExtension.BorderWidth(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetMarginStyle(VisualElement, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.SetTogglePick(VisualElement)
VisualElementExtension.PickModeOFF(VisualElement)
VisualElementExtension.PickModeON(VisualElement)
VisualElementExtension.SetFont(VisualElement, Font)
VisualElementExtension.PositionTL(VisualElement, Vector2)
VisualElementExtension.PositionTL(VisualElement, Single, Single)
VisualElementExtension.PositionTR(VisualElement, Vector2)
VisualElementExtension.PositionTR(VisualElement, Single, Single)
VisualElementExtension.PositionBL(VisualElement, Vector2)
VisualElementExtension.PositionBL(VisualElement, Single, Single)
VisualElementExtension.PositionBR(VisualElement, Vector2)
VisualElementExtension.PositionBR(VisualElement, Single, Single)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class Icon.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object
UnityEngine.UIElements.UxmlFactory<Icon, UnityEngine.UIElements.VisualElement.UxmlTraits>
Icon.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<Icon, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class Slot

This Class is a VisualElement that is created for the Inventory.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
Slot

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class Slot : VisualElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures,
IVisualElementScheduler, IResolvedStyle
```

Constructors

Slot()

This is a Constructor and will set the style for the Slot.

Declaration

```
public Slot()
```

Slot(Int32, Icon, StyleSheet)

This Method will set the style of the Slot.

Declaration

```
public Slot(int key, Icon icon = null, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	
Icon	icon	
UnityEngine.UIElements.StyleSheet	style	

Fields

key

This Variable is the key that corresponds to the dictionary entry's key.

Declaration

```
public int key
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

AddslotLabel(String)

This Method will create a slat label to this Slot if it is a slot no the Hotbar.

Declaration

```
public void AddslotLabel(string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	the text that will be added to the slot label.

ClearIcon()

This Method will remove the icon from this slot.

Declaration

```
public void ClearIcon()
```

GetIcon()

This Method will get Icon attached to this Slot.

Declaration

```
public Icon GetIcon()
```

Returns

TYPE	DESCRIPTION
Icon	Return the Icon.

SetIcon(Icon)

This Method will add an Icon to this slot.

Declaration

```
public void SetIcon(Icon icon)
```

Parameters

TYPE	NAME	DESCRIPTION
Icon	icon	the Icon that will be added to this slot.

SetIcon(Sprite, Int32, Boolean)

Declaration

```
public void SetIcon(Sprite image, int amount, bool canstack)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Sprite	image	
System.Int32	amount	
System.Boolean	canstack	

Style()

This Method will set the style of the Icon.

Declaration

```
public virtual void Style()
```

TryGetIcon(out Icon)

This Method will try to get Icon attached to this Slot.

Declaration

```
public bool TryGetIcon(out Icon icon)
```

Parameters

TYPE	NAME	DESCRIPTION
Icon	icon	out Icon when found.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if icon is found.

Extension Methods

VisualElementExtension.BorderStyleColor(VisualElement, Color[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit, Single[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit[], Single[])
VisualElementExtension.BorderWidth(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetMarginStyle(VisualElement, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.SetTogglePick(VisualElement)
VisualElementExtension.PickModeOFF(VisualElement)
VisualElementExtension.PickModeON(VisualElement)
VisualElementExtension.SetFont(VisualElement, Font)
VisualElementExtension.PositionTL(VisualElement, Vector2)
VisualElementExtension.PositionTL(VisualElement, Single, Single)
VisualElementExtension.PositionTR(VisualElement, Vector2)
VisualElementExtension.PositionTR(VisualElement, Single, Single)
VisualElementExtension.PositionBL(VisualElement, Vector2)
VisualElementExtension.PositionBL(VisualElement, Single, Single)
VisualElementExtension.PositionBR(VisualElement, Vector2)

VisualElementExtension.PositionBR(VisualElement, Single, Single)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class Slot.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object
UnityEngine.UIElements.UxmlFactory<Slot, UnityEngine.UIElements.VisualElement.UxmlTraits>
Slot.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<Slot, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class SpriteElement

This Class is an VisualElement that is created for the Inventory Editor.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
UnityEngine.UIElements.Image
SpriteElement

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class SpriteElement : Image, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures,
IVisualElementScheduler, IResolvedStyle
```

Constructors

SpriteElement()

This is a Constructor and will set the style for the SpriteElement.

Declaration

```
public SpriteElement()
```

SpriteElement(Sprite, StyleSheet)

This is a Constructor and will set the style for the SpriteElement.

Declaration

```
public SpriteElement(Sprite sprite, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Sprite	sprite	
UnityEngine.UIElements.StyleSheet	style	

Methods

RemoveSprite()

This Method will remove the sprite for this element.

Declaration

```
public void RemoveSprite()
```

SetSprite(Sprite)

This Method will set the sprite for this element.

Declaration

```
public void SetSprite(Sprite sprite)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Sprite	sprite	

Extension Methods

VisualElementExtension.BorderStyleColor(VisualElement, Color[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit, Single[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit[], Single[])
VisualElementExtension.BorderWidth(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetMarginStyle(VisualElement, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.SetTogglePick(VisualElement)
VisualElementExtension.PickModeOFF(VisualElement)
VisualElementExtension.PickModeON(VisualElement)
VisualElementExtension.SetFont(VisualElement, Font)
VisualElementExtension.PositionTL(VisualElement, Vector2)
VisualElementExtension.PositionTL(VisualElement, Single, Single)
VisualElementExtension.PositionTR(VisualElement, Vector2)
VisualElementExtension.PositionTR(VisualElement, Single, Single)
VisualElementExtension.PositionBL(VisualElement, Vector2)
VisualElementExtension.PositionBL(VisualElement, Single, Single)

VisualElementExtension.PositionBR(VisualElement, Vector2)
VisualElementExtension.PositionBR(VisualElement, Single, Single)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class SpriteElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object
UnityEngine.UIElements.UxmlFactory<SpriteElement, UnityEngine.UIElements.Image.UxmlTraits>
SpriteElement.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<SpriteElement, Image.UxmlTraits>, IUxmlFactory
```

Extension Methods

RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class ToolTip

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
ToolTip

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class ToolTip : VisualElement, IEventHandler, ITransform, ITransitionAnimations,
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

ToolTip()

This is a Constructor for the tool tip.

Declaration

```
public ToolTip()
```

ToolTip(Font, Int32, StyleSheet)

This is a Constructor for the tool tip.

Declaration

```
public ToolTip(Font font = null, int textSize = 0, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Font	font	Is an optional font that will be added to the label.
System.Int32	textSize	Is an optional size of the text in the label.
UnityEngine.UIElements.StyleSheet	style	Is an optional stylesheet for the element.

Methods

AddText(List<Func<String>>)

This Method will add a new message to the Func string list.

Declaration

```
public void AddText(List<Func<string>> messages)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.Func<System.String>>	messages	The list of func strings that will be added to the func string list for the message.

AddText(Func<String>)

This Method will add a new message to the Func string list.

Declaration

```
public void AddText(Func<string> message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.String>	message	The func string that will be added to the func string list for the message.

AddText(String)

This Method will add a new message to the Func string list.

Declaration

```
public void AddText(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	The string that will be added to the func string list for the message.

HideToolTip()

This Method will conceal the tool tip.

Declaration

```
public void HideToolTip()
```

RemoveText()

this will clear the message.

Declaration

```
public void RemoveText()
```

ShowToolTip()

This Method will reveal the tool tip.

Declaration

```
public void ShowToolTip()
```

Style()

This Method will set the style of the Icon.

Declaration

```
public void Style()
```

TextSize(Int32)

This Method will allow for the text size to be changed.

Declaration

```
public void TextSize(int textSize)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	textSize	the int used to update the text size.

UpdatePosition(Vector2)

This Method will update the position of the element on screen and maintain its bound within the screens bounds.

Declaration

```
public void UpdatePosition(Vector2 anchoredPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	anchoredPosition	This is the vector2 position it is trying to move to.

UpdateText()

This Method will update the message that is displayed.

Declaration

```
public void UpdateText()
```

Extension Methods

VisualElementExtension.BorderStyleColor(VisualElement, Color[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit, Single[])
VisualElementExtension.BorderRadius(VisualElement, LengthUnit[], Single[])
VisualElementExtension.BorderWidth(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit, Color[], Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color, Single[])
VisualElementExtension.SetBorderStyle(VisualElement, LengthUnit[], Color[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetMarginStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetMarginStyle(VisualElement, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetPaddingStyle(VisualElement, LengthUnit[], Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, Single[])
VisualElementExtension.SetSizeStyle(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.SetTogglePick(VisualElement)
VisualElementExtension.PickModeOFF(VisualElement)
VisualElementExtension.PickModeON(VisualElement)
VisualElementExtension.SetFont(VisualElement, Font)
VisualElementExtension.PositionTL(VisualElement, Vector2)
VisualElementExtension.PositionTL(VisualElement, Single, Single)
VisualElementExtension.PositionTR(VisualElement, Vector2)
VisualElementExtension.PositionTR(VisualElement, Single, Single)
VisualElementExtension.PositionBL(VisualElement, Vector2)
VisualElementExtension.PositionBL(VisualElement, Single, Single)
VisualElementExtension.PositionBR(VisualElement, Vector2)
VisualElementExtension.PositionBR(VisualElement, Single, Single)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)

RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)

Class ToolTip.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object
UnityEngine.UIElements.UxmlFactory<ToolTip, UnityEngine.UIElements.VisualElement.UxmlTraits>
ToolTip.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<ToolTip, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)