Table of Contents

Namespace UtilitySystem

Extension

This is a Namespace in UtilitySystem Namespace, Assembly-CSharp

Tool

This is a Namespace in UtilitySystem Namespace, Assembly-CSharp

UI

This is a Namespace in UtilitySystem Namespace, Assembly-CSharp

Namespace Extension

This is a Namespace in UtilitySystem Namespace, Assembly-CSharp

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.

Class FloatExtension

This Static Class extends the float Variable.

Inheritance

System.Object
FloatExtension

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: UtilitySystem.Extension

Assembly: cs.temp.dll.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

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: UtilitySystem.Extension

Assembly: cs.temp.dll.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

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: UtilitySystem.Extension

Assembly: cs.temp.dll.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
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<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
Transform	transform	This is the transform that will get reset.

Namespace Tool

This is a Namespace in UtilitySystem Namespace, Assembly-CSharp

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

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: UtilitySystem.Tool

Assembly: cs.temp.dll.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, UnityEngine.Object)

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

Declaration

```
public static void Error(string message, UnityEngine.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, UnityEngine.Object)

Logs a message to the Unity Console.

Declaration

```
public static void Log(string message, UnityEngine.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, UnityEngine.Object)

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

Declaration

```
public static void LogException(Exception exception, UnityEngine.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, UnityEngine.Object)

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

Declaration

```
public static void Warning(string message, UnityEngine.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

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: UtilitySystem.Tool

Assembly: cs.temp.dll.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).

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()
```

Namespace UI

This is a Namespace in UtilitySystem Namespace, Assembly-CSharp

Classes

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.

Class Icon

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

Inheritance

System.Object
Icon

Namespace: UtilitySystem.UI

Assembly: cs.temp.dll.dll

Syntax

```
public class Icon : VisualElement
```

Constructors

Icon()

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

Declaration

```
public Icon()
```

Icon(InventoryItem)

This is the constructor that will accept an InventoryItem It will set the Image & Style and add the label for amount if the item canStack.

Declaration

```
public Icon(InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	an InventoryItem that will be used to add the image and label if canStack is true.

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.

Class Icon.UxmlFactory

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

Inheritance

System.Object
Icon.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: cs.temp.dll.dll

Syntax

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

Class Slot

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

Inheritance

System.Object
Slot

Namespace: UtilitySystem.UI

Assembly: cs.temp.dll.dll

Syntax

```
public class Slot : VisualElement
```

Constructors

Slot()

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

Declaration

```
public Slot()
```

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(InventoryItem)

This Method will create a new Icon to this slot using an InventoryItem.

Declaration

```
public void SetIcon(InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	the InventoryItem that will be used to make the Icon.

Style()

This Method will set the style of the Slot.

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.

Class Slot.UxmlFactory

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

Inheritance

System.Object
Slot.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: cs.temp.dll.dll

Syntax

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

Class SpriteElement

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

Inheritance

System.Object
SpriteElement

Namespace: UtilitySystem.UI

Assembly: cs.temp.dll.dll

Syntax

```
public class SpriteElement : Image
```

Constructors

SpriteElement()

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

Declaration

```
public SpriteElement()
```

SpriteElement(Sprite)

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

Declaration

```
public SpriteElement(Sprite sprite)
```

Parameters

TYPE	NAME	DESCRIPTION
Sprite	sprite	

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
Sprite	sprite	

Class SpriteElement.UxmlFactory

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

Inheritance

System.Object
SpriteElement.UxmlFactory

Namespace: UtilitySystem.UI

Assembly: cs.temp.dll.dll

Syntax

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