Table of Contents

- Inventory System
  - Data
    - Inventory
    - InventoryItem
    - InventoryType
    - KeyItem
    - SortByEnum
    - SpawnItem
    - TransferItem
  - Editor
    - InventoryWindow
  - Script
    - ContextMenuGenerator
    - CreateElements
    - DemoSceneManager
    - DictionaryActions
    - Enemy
    - EquippedItem
    - GenerateData
    - InventorySpawner
    - InventoryTarget
    - InventoryTrader
    - InventoryTraderManager
    - PickupItem
    - Player
    - Storage
    - Trader

WorldItem
UI
IconElement
IconElement.UxmlFactory
InventoryElement
InventoryElement.UxmlFactory
SlotElement
SlotElement.UxmlFactory
SlotPopupElement
SlotPopupElement.UxmlFactory
TransferPopupElement
TransferPopupElement.UxmlFactory

Namespace InventorySystem

This is The Namespace for the InventorySystem, Assembly-CSharp

InventorySystem.Data

InventorySystem.Editor

InventorySystem.Script

InventorySystem.UI

Namespace InventorySystem.Data

Classes

Inventory

This class is a Scriptable Object that stores the values from an inventory.

InventoryItem

This class is the Item for the Inventory.

KeyItem

This class is the KeyItem it is used for storage the InventoryItem and its key for the Dictionary. and is used to convert for saving string

SpawnItem

This class is the SpawnItem that is used to create an InventoryItem for the inventory system.

TransferItem

Enums

InventoryType

This is used to determine how an Inventory should be Interacted with. To use this enum in UI Builder

SortByEnum

This is used to determine how an Inventory will be sorted. To use this enum in UI Builder

Class Inventory

This class is a Scriptable Object that stores the values from an inventory.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
Inventory

Namespace: InventorySystem.Data

Assembly: Assembly-CSharp.dll

Syntax

```
[CreateAssetMenu]
[Serializable]
public class Inventory : ScriptableObject
```

Fields

currencyChanged

This action will be invoked when the inventory's currency is changed.

Declaration

```
public Action<int, int> currencyChanged
```

Field Value

TYPE	DESCRIPTION
System.Action<System.Int32, System.Int32>	

Remarks

Pass the currencyValue variable & the value it was changed by.

maxWeight

This is the maximum weight of the inventory.

Declaration

```
public float maxWeight
```

Field Value

TYPE	DESCRIPTION
System.Single	

size

This is the size of the inventory.

Declaration

```
public int size
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Storage

This is where the contents of the inventory is held.

Declaration

```
[SerializeField]
public List<KeyItem> Storage
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<KeyItem>	

Properties

CurrencyValue

This is the amount of gold that is held.

Declaration

```
[SerializeField]
public int CurrencyValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	When the value changes it will Invoke the currencyChanged Action. & pass the currencyValue variable & the value it was changed by.

FreeSlots

This will show how many slots are left in this inventory.

Declaration

```
public int FreeSlots { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

IsTrader

This will check if this Inventory is of type 'InventoryType.Trader'.

Declaration

```
public bool IsTrader { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

TitleName

This is the name that is used in te UI title.

Declaration

```
public string TitleName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Type

This is the type of inventory this is.

Declaration

```
public InventoryType Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
InventoryType	

UID

This is the unique id for the inventory.

Declaration

```
public string UID { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ClearCurrency()

This will clear the currency of the attached inventory.

Declaration

```
public void ClearCurrency()
```

DecreaseCurrency(Int32)

This will decrease the currency of the attached inventory.

Declaration

```
public void DecreaseCurrency(int money)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	money	This is how much will be removed to the inventory's currency.

GetCurrency()

This will get the currency of the attached inventory.

Declaration

```
public int GetCurrency()
```

Returns

TYPE	DESCRIPTION
System.Int32	The return is the amount of currency the inventory has.

IncreaseCurrency(Int32)

This will increase the currency of the attached inventory.

Declaration

```
public void IncreaseCurrency(int money)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	money	This is how much will be added to the inventory's currency.

Init()

This is a an Init and will set a new UID;

Declaration

```
public void Init()
```

Init(Inventory)

This is an Init that will populates from another Inventory File.

Declaration

```
public void Init(Inventory inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	inventory	This param Inventory will add it's contents to this Inventory

Init(String)

This is a an Init and will set a new UID; and the TitleName

Declaration

```
public void Init(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	This is the TitleName of the inventory.

Init(String, Int32, InventoryType)

This is a an Init and will set a new UID; and the TitleName. and the inventory size. and the type of inventory it is.

Declaration

```
public void Init(string name, int size, InventoryType type)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	This is the TitleName of the inventory.
System.Int32	size	This is the size of the inventory capacity.
InventoryType	type	This is the type of inventory it is.

InventoryWeight()

This will get the current weight of the inventory items.

Declaration

```
public float InventoryWeight()
```

Returns

TYPE	DESCRIPTION
System.Single	The return is the current weight of all of the items in the inventory.

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class InventoryItem

This class is the Item for the Inventory.

Inheritance

System.Object
InventoryItem

Namespace: InventorySystem.Data

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class InventoryItem
```

Constructors

InventoryItem(InventoryItem)

This constructor Populates the InventoryItem File with data from another InventoryItem File

Declaration

```
public InventoryItem(InventoryItem itemInventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	itemInventory	Copy the value from the param to this InventoryItem.

InventoryItem(ItemData, Int32)

This populates the InventoryItem File with data from an ItemData File.

Declaration

```
public InventoryItem(ItemData itemData, int currentStack = 1)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	itemData	Copy the value from the param to this InventoryItem.

TYPE	NAME	DESCRIPTION
System.Int32	currentStack	Copy the param to this InventoryItem.currentStack.

Fields

currentStack

This is the amount of the item.

Declaration

```
public int currentStack
```

Field Value

TYPE	DESCRIPTION
System.Int32	

isEquipped

This is used to determine if this item is equipped.

Declaration

```
public bool isEquipped
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

itemData

The data of the item.

Declaration

```
public ItemData itemData
```

Field Value

TYPE	DESCRIPTION
ItemData	

Properties

ItemFreeSpace

This is how much freeSpace this item has. (how many more items that can be added to the currentStack)

Declaration

```
public int ItemFreeSpace { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

ItemWeight

This is how much the item weights times the number of the current stack size.

Declaration

```
public float ItemWeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

TotalCost

This is the totalCost of all of the item(s) in the currentStack.

Declaration

```
public int TotalCost { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

AmountCanBuy(Int32)

This will see how may of this item can be purchased.

Declaration

```
public int AmountCanBuy(int currencyValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	currencyValue	This is the avaliable currency that will see how many can be purchased.

Returns

TYPE	DESCRIPTION
System.Int32	The return is the amount that can be purchased. (Round Down)

CombineIntoItem(InventoryItem)

This will combine this item's current stack into the param item's currentstack.

Declaration

```
public bool CombineIntoItem(InventoryItem itemInventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	itemInventory	This param item will be fill with the item that called this method.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if any amount has been transferred.

DropItem(Transform)

This will create a World Object from the ItemData foreach item in currentStack.

Declaration

```
public void DropItem(Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	

DropItem(Transform, Int32)

This will create a World Object from the ItemData for amount.

Declaration

```
public void DropItem(Transform transform, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	
System.Int32	amount	how many of the item to drop clamped between 1 and currentStack.

EquipItem()

This will create a World Object from the ItemData foreach item in currentStack.

Declaration

```
public void EquipItem()
```

SplitItem(Int32)

This will create a new item and give the amount from this item's currentStack to the new item.

Declaration

```
public InventoryItem SplitItem(int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amount	This is the amount that will be split into the new item.

Returns

TYPE	DESCRIPTION
InventoryItem	The return is the new item.

TradeStackAmount(InventoryItem, Int32)

This will trade the stack from this item to the param item, by param amount.

Declaration

```
public void TradeStackAmount(InventoryItem target, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	target	This item will receive the amount.
System.Int32	amount	This is the amount to be traded.

UnEquipItem()

This will unequipped item and destroy the World Object.

Declaration

```
public void UnEquipItem()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Enum InventoryType

This is used to determine how an Inventory should be Interacted with. To use this enum in UI Builder

Namespace: InventorySystem.Data

Assembly: Assembly-CSharp.dll

Syntax

```
public enum InventoryType
```

Remarks

InventorySystem.Data.InventoryType, Assembly-CSharp

Fields

NAME	DESCRIPTION
Loot	
Player	
Storage	
Trader	

Extension Methods

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

Class KeyItem

This class is the KeyItem it is used for storage the InventoryItem and its key for the Dictionary. and is used to convert for saving string

Inheritance

System.Object
KeyItem

Namespace: InventorySystem.Data

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class KeyItem
```

Constructors

KeyItem(Int32, InventoryItem)

This will take a Key and InventoryItem.

Declaration

```
public KeyItem(int key, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This integer will be the key for the dictionary Key.
InventoryItem	item	this InventoryItem will be the value for the dictionary Value.

KeyItem(String)

This will accept a string with the 'ConvertToStorage' format "Key:Item.IID:ItemCurrentStack". and convert the values into the Key and Item

Declaration

```
public KeyItem(string storage)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	storage	This param is the string that should be formatted correctly.

Properties

ConvertToStorage

This will convert the KeyItem into a strong for saving.

Declaration

```
public string ConvertToStorage { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	The return will be a string for saving and storage.

Item

This is the Item that is the pair value of the key in the dictionary.

Declaration

```
public InventoryItem Item { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryItem	

Key

This is the Key of the item in its owner's dictionary

Declaration

```
public int Key { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Enum SortByEnum

This is used to determine how an Inventory will be sorted. To use this enum in UI Builder

Namespace: InventorySystem.Data

Assembly: Assembly-CSharp.dll

Syntax

```
public enum SortByEnum
```

Remarks

InventorySystem.Data.SortByEnum, Assembly-CSharp

Fields

NAME	DESCRIPTION
None	
SortByAmount	
SortByCost	
SortByName	
SortByType	
SortByWeight	

Extension Methods

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

Class SpawnItem

This class is the SpawnItem that is used to create an InventoryItem for the inventory system.

Inheritance

System.Object
SpawnItem

Namespace: InventorySystem.Data

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class SpawnItem
```

Constructors

SpawnItem(ItemData)

This constructor will create a SpawnItem that will be used to create an InventoryItem.

Declaration

```
public SpawnItem(ItemData item)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is the ItemData that will give the InventoryItem its data.

SpawnItem(ItemData, Int32, Boolean)

This constructor will create a SpawnItem that will be used to create an InventoryItem.

Declaration

```
public SpawnItem(ItemData item, int amount, bool isRandom = false)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is the ItemData that will give the InventoryItem its data.
System.Int32	amount	This is the Amount that will be added to the InventoryItem.

TYPE	NAME	DESCRIPTION
System.Boolean	isRandom	This will determine if the Amount will be used or a random number.

Properties

Amount

This is the Amount that will be added to the InventoryItem.

Declaration

```
public int Amount { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CreateItem

This will Create an InventoryItem from it's data.

Declaration

```
public InventoryItem CreateItem { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryItem	The return will be the InventoryItem that will be created.

IsRandom

This will determine if the amount will be used for a random number from (1 to the Item.maxStack) is used.

Declaration

```
public bool IsRandom { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Item

This is the Data that is going to be used to create the InventoryItem.

Declaration

```
public ItemData Item { get; }
```

Property Value

TYPE	DESCRIPTION
ItemData	

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class TransferItem

Inheritance

System.Object
TransferItem

Namespace: InventorySystem.Data

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class TransferItem
```

Constructors

TransferItem(Int32, InventoryItem)

This is the constructor that will create this TransferItem.

Declaration

```
public TransferItem(int key, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This is the Key of the item in its owner's dictionary.
InventoryItem	item	This is the Item that is going to be transferred. and it's currentStack will set Amount.

TransferItem(Int32, InventoryItem, Int32)

This is the constructor that will create this TransferItem.

Declaration

```
public TransferItem(int key, InventoryItem item, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This is the Key of the item in its owner's dictionary.

TYPE	NAME	DESCRIPTION
InventoryItem	item	This is the Item that is going to be transferred.
System.Int32	amount	This is the Amount that is going to be transferred.

Properties

Amount

This is the Amount of the item that will transferred.

Declaration

```
public int Amount { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Cost

This is the Cost of all of the item(s) that will be transferred.

Declaration

```
public int Cost { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Item

This is the Item that is going to be transferred.

Declaration

```
public InventoryItem Item { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryItem	

Key

This is the Key of the item in its owner's dictionary and if the whole item is transferred it will be removed from it's owner.

Declaration

```
public int Key { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Namespace InventorySystem.Editor

Classes

InventoryWindow

This class is an Editor Window for the creation & editing of the inventory and it's content.

Class InventoryWindow

This class is an Editor Window for the creation & editing of the inventory and it's content.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
UnityEditor.EditorWindow
InventoryWindow

Namespace: InventorySystem.Editor

Assembly: Assembly-CSharp-Editor.dll

Syntax

```
public class InventoryWindow : EditorWindow
```

Properties

SelectedInventory

This is connected to the selectedInventory Variable. If selectedInventory is different from the incoming value then set the value & refresh the UI.

Declaration

```
public Inventory SelectedInventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	

Methods

AddItemData(ItemData)

This will add a new ItemData File to items list Then Add that ItemData to the UI.

Declaration

```
public void AddItemData(ItemData item)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is a new ItemData.

AddItemInToDictionary(InventoryItem)

This will add the item to the dictionary.

Declaration

```
public bool AddItemInToDictionary(InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	this is the item that will be added to the dictionary.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if added to dictionary.

ChangeItemData(ItemData)

This gets invoked from an event when the OpenItemUpdate clicks the update or Duplacate button

Declaration

```
public void ChangeItemData(ItemData currentItem)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	currentItem	This is the that is changed ItemData.

CloseWindow()

This static will close the InventoryWindow. The window can be open either by double clicking an Inventory file, Though the UnityEditor Toolbar: Tools/DownUnder Studios/Inventory System/Tools/Close Inventory Editor Or by the shortcut key CTRl + SHIFT + ALT + O

Declaration

```
[MenuItem("Tools/DownUnder Studios/Inventory System/Tools/Close Inventory Editor %#&O")]
public static void CloseWindow()
```

CreateGUI()

This is a unity method, that will be run when the window is created.

Declaration

```
public void CreateGUI()
```

GenerateItemDataFiles()

This will get the ItemData file from the Resources folder and add them to the items list. Then Create UI for each ItemData.

Declaration

```
public void GenerateItemDataFiles()
```

GetInventoryFiles()

This will get all Inventory files in the Resources.

Declaration

```
public void GetInventoryFiles()
```

NewInventory(String)

This will create a new Inventory File And add it to the *inventories* List & Dropdown Field

Declaration

```
public void NewInventory(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	This is the name of the new inventory.

OnOpenAsset(Int32, Int32)

This static will open the InventoryWindow when an Inventory file is double clicked.

Declaration

```
[OnOpenAsset]
public static bool OnOpenAsset(int instanceID, int line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	instanceID	not used.
System.Int32	line	not used.

Returns

TYPE	DESCRIPTION
System.Boolean	

OpenEditItem(ItemData)

This will open the EditItem Window to edit the selected ItemData.

Declaration

```
public void OpenEditItem(ItemData item)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is the ItemData that will be changed.

OpenWindow()

This static will open the InventoryWindow. The window can be open either by double clicking an Inventory file, Though the UnityEditor Toolbar: Tools/DownUnder Studios/Inventory System/Tools/Open Inventory Editor Or by the shortcut key CTRl + SHIFT + ALT + I

Declaration

```
[MenuItem("Tools/DownUnder Studios/Inventory System/Tools/Open Inventory Editor %#&I")]
public static void OpenWindow()
```

Save()

This will Save the inventory's changes. Only if the editor is not in play mode.

Declaration

```
public void Save()
```

SetWindowSize(Int32, Int32)

This will set the Editor Window min & max size to a fix size. With both params being the width & height respectively.

Declaration

```
public void SetWindowSize(int width, int height)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	width	The width of the Window.
System.Int32	height	The height of the Window.

UpdateUI()

This will check if SelectedInventory is null or not. If not null set the displays to flex and window size to 1200-500 else set the displays to none and window size to 570-30

Declaration

```
public void UpdateUI()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Namespace InventorySystem.Script

Classes

ContextMenuGenerator

This static class will generate Context Menu for inventory.

CreateElements

This static class will allow the creation of UI Toolkit elements.

DemoSceneManager

DictionaryActions

This static class is used to perform actions on Inventory Dictionary.

Enemy

This class is the script thats applied to a Enemy Character.

EquippedItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

GenerateData

This static Class is used to Generate Inventories & InventoryItems from ItemData files.

InventorySpawner

This class is the InventorySpawner it will create an inventory scriptableobject on awake. and generate its Storage from the list of SpawnItem(s).

InventoryTarget

This class is the base for all classes with with inventory. (Enemy, Player, Storage, Trader)

InventoryTrader

This static Class is used to Trade Inventory item(s).

InventoryTraderManager

This class handles the inventories and logic for transfering of item(s) from 1 inventory to another.

PickupItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

Player

This class is the script thats applied to a player Character.

Storage

This class is the script thats applied to a storage inventory.

Trader

This class is the script thats applied to a Trader Character.

WorldItem

This abstract class is the base of any item in the world. such as the "PickupItem" or the "EquippedItem".

Class ContextMenuGenerator

This static class will generate Context Menu for inventory.

Inheritance

System.Object
ContextMenuGenerator

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public static class ContextMenuGenerator
```

Methods

CombineItem(InventoryElement, Int32, List<ContextMenuItemData>)

This will create ContextMenuItemButtonData(s) for combining the item with same item(s) in this inventory.

Declaration

```
public static void CombineItem(InventoryElement From, int key, List<ContextMenuItemData> parentMenu)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Int32	key	The slot key for the dictionary entry.
System.Collections.Generic.List<ContextMenuItemData>	parentMenu	This is the menu that will hold the ContextMenuItemParentData.

ContextItem_Button_Slot(InventoryElement, String, InventoryElement, Int32, Boolean)

This will create ContextMenuItemButtonData for the TransferItem.

Declaration

```
public static ContextMenuItemButtonData ContextItem_Button_Slot(InventoryElement From, string itemName,
InventoryElement To, int key, bool isFull = false)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.String	itemName	This is the name of the ContextMenuItemButtonData.
InventoryElement	To	
System.Int32	key	The slot key for the dictionary entry.
System.Boolean	isFull	This boolean is true then invoke 'TransferItem'. else invoke 'OpenTransferPopup'.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

ContextItem_Button_TransferItems(InventoryElement, String, InventoryElement, List<TransferItem>)

This will create ContextMenuItemButtonData for all of the TransferItem(s).

Declaration

```
public static ContextMenuItemButtonData ContextItem_Button_TransferItems(InventoryElement From, string itemName, InventoryElement To, List<TransferItem> transferItems)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.String	itemName	This is the name of the ContextMenuItemButtonData.
InventoryElement	To	The inventory that will be used to create ContextMenuItemData.
System.Collections.Generic.List<TransferItem>	transferItems	The list of TransferItem(s) to be transferred.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

ContextItem_Menu_InventoryType(InventoryElement, List<ContextMenuItemData>, String, InventoryElement[])

This will create ContextMenuItemParentData for all of the inventories.

Declaration

```
public static void ContextItem_Menu_InventoryType(InventoryElement From, List<ContextMenuItemData>
parentMenu, string itemName, params InventoryElement[] inventories)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Collections.Generic.List<ContextMenuItemData>	parentMenu	This is the menu that will hold the ContextMenuItemParentData.
System.String	itemName	This is the name of the ContextMenuItemParentData.
InventoryElement[]	inventories	The params of inventories that will be used to create ContextMenuItemData.

ContextItem_Menu_ItemType(InventoryElement, List<ContextMenuItemData>, InventoryElement)

This will create ContextMenuItemParentData for all of the TransferItem(s).

Declaration

```
public static void ContextItem_Menu_ItemType(InventoryElement From, List<ContextMenuItemData> parentMenu,
InventoryElement To)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<ContextMenuItemData>	parentMenu	This is the menu that will hold the ContextMenuItemParentData.
InventoryElement	To	

ContextItem_Menu_Sort(InventoryElement)

This will create a ContextMenuItemParentData for sorting the inventory storage.

Declaration

```
public static ContextMenuItemParentData ContextItem_Menu_Sort(InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory that will be sortted.

Returns

TYPE	DESCRIPTION
ContextMenuItemParentData	The return is the ContextMenu Item Parent Data.

ContextItemMenu_AllInventoryTypes(InventoryElement, List<ContextMenuItemData>)

This will create ContextMenuItemParentData for all of the different Types of inventories.

Declaration

```
public static void ContextItemMenu_AllInventoryTypes(InventoryElement From, List<ContextMenuItemData>
parentMenu)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Collections.Generic.List<ContextMenuItemData>	parentMenu	This is the menu that will hold the ContextMenuItemParentData.

ContextItemMenu_ConsolidateInventory(InventoryElement)

This will create a ContextMenuItemParentData for consolidate the inventory storage.

Declaration

```
public static ContextMenuItemButtonData ContextItemMenu_ConsolidateInventory(InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory that will be sortted.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

ContextItemMenu_Slot_InventoryType(InventoryElement, List<ContextMenuItemData>, String, Int32, InventoryElement[])

This will create ContextMenuItemParentData for all of the inventories.

Declaration

```
public static void ContextItemMenu_Slot_InventoryType(InventoryElement From, List<ContextMenuItemData>
parentMenu, string itemName, int key, params InventoryElement[] inventories)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Collections.Generic.List<ContextMenuItemData>	parentMenu	This is the menu that will hold the ContextMenuItemParentData.
System.String	itemName	This is the name of the ContextMenuItemButtonData.
System.Int32	key	The slot key for the dictionary entry.

TYPE	NAME	DESCRIPTION
InventoryElement[]	inventories	The params of inventories that will be used to create ContextMenuItemData.

ContextItemMenu_TradeItem(InventoryElement, List<ContextMenuItemData>, Int32)

This will create ContextMenuItemParentData for all of the different Types of inventories.

Declaration

```
public static void ContextItemMenu_TradeItem(InventoryElement From, List<ContextMenuItemData> parentMenu, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Collections.Generic.List<ContextMenuItemData>	parentMenu	This is the menu that will hold the ContextMenuItemParentData.
System.Int32	key	The slot key for the dictionary entry.

ContextMenu(InventoryElement, out List<ContextMenuItemData>)

This will create a list of ContextMenuItemData for a ContextMenu.

Declaration

```
public static bool ContextMenu(InventoryElement From, out List<ContextMenuItemData> items)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data. This is the inventory of the data.
System.Collections.Generic.List<ContextMenuItemData>	items	The out param is the ContextMenuItemData that was made.

Returns

TYPE	DESCRIPTION
System.Boolean	The return is true.

ContextMenuSlot(InventoryElement, SlotElement, out List<ContextMenuItemData>)

This will create a list of ContextMenuItemData for a ContextMenu.

Declaration

```
public static bool ContextMenuSlot(InventoryElement From, SlotElement slot, out List<ContextMenuItemData>
items)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
SlotElement	slot	This slot will be the From of the data.
System.Collections.Generic.List<ContextMenuItemData>	items	The out param is the ContextMenuItemData that was made.

Returns

TYPE	DESCRIPTION
System.Boolean	The return is true if items has at list 1 item.

DetailBoxItem(InventoryElement, Int32)

This will create ContextMenuItemButtonData for a DetailBox for the slot's item.

Declaration

```
public static ContextMenuItemButtonData DetailBoxItem(InventoryElement From, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.

TYPE	NAME	DESCRIPTION
System.Int32	key	The slot key for the dictionary entry.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

DropItem(InventoryElement, Int32)

This will create ContextMenuItemButtonData for dropping an item.

Declaration

```
public static ContextMenuItemButtonData DropItem(InventoryElement From, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Int32	key	The slot key for the dictionary entry.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

EquipItem(InventoryElement, Int32)

This will create ContextMenuItemButtonData for equipping an item.

Declaration

```
public static ContextMenuItemButtonData EquipItem(InventoryElement From, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Int32	key	The slot key for the dictionary entry.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

SplitItem(InventoryElement, Int32)

This will create ContextMenuItemButtonData for Splitting an item.

Declaration

```
public static ContextMenuItemButtonData SplitItem(InventoryElement From, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Int32	key	The slot key for the dictionary entry.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

UnEquipItem(InventoryElement, Int32)

This will create ContextMenuItemButtonData for unequipped an item.

Declaration

```
public static ContextMenuItemButtonData UnEquipItem(InventoryElement From, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
System.Int32	key	The slot key for the dictionary entry.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonData	The return is the ContextMenu Item Button Data.

Class CreateElements

This static class will allow the creation of UI Toolkit elements.

Inheritance

System.Object
CreateElements

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public static class CreateElements
```

Methods

CreateElement_Icon(VisualElement, InventoryItem)

This will create a IconElement and add to the element that invokes this.

Declaration

```
public static IconElement CreateElement_Icon(this VisualElement parent, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
InventoryItem	item	This is the item that will be used to create the icon.

Returns

TYPE	DESCRIPTION
IconElement	The return is the IconElement that is created.

CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)

This will create an InventoryElement and add to the element that invokes this.

Declaration

```
public static InventoryElement CreateElement_Inventory(this VisualElement parent, int numberOfSlots, int
slotsPerRow, Inventory inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Int32	numberOfSlots	This is how many slot(s)
System.Int32	slotsPerRow	This how many slot(s) per row.
Inventory	inventory	This is the Inventory to be added to the InventoryElement.

Returns

TYPE	DESCRIPTION
InventoryElement	The return is the InventoryElement that is created.

CreateElement_Slot(VisualElement, Int32)

This will create a SlotElement and add to the element that invokes this.

Declaration

```
public static SlotElement CreateElement_Slot(this VisualElement parent, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Int32	key	This is the key of the slot.

Returns

TYPE	DESCRIPTION
SlotElement	The return is the SlotElement that is created.

CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)

This will create a SlotPopupElement and add to the element that invokes this.

Declaration

```
public static SlotPopupElement CreateElement_SlotPopup(this VisualElement parent, Vector2 size, int key,
InventoryElement owner, bool isSplit)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
UnityEngine.Vector2	size	This is the size of the popup.
System.Int32	key	This is the key of the slot.
InventoryElement	owner	This is the InventoryElement that owns the item.
System.Boolean	isSplit	This is the boolean that will determine if it is split or drop.

Returns

TYPE	DESCRIPTION
SlotPopupElement	The return is the SlotPopupElement that is created.

CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

This will create a TransferPopupElement and add to the element that invokes this.

Declaration

```
public static TransferPopupElement CreateElement_TransferPopup(this VisualElement parent, Vector2 size,
TransferItem item, InventoryElement To, InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	This is the size of the popup.
TransferItem	item	This is the item that will be transferred.
InventoryElement	To	This is the InventoryElement that owns the item.
InventoryElement	From	This is the InventoryElement that will receive the item.

Returns

TYPE	DESCRIPTION
TransferPopupElement	The return is the TransferPopupElement that is created.

Class DemoSceneManager

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
DemoSceneManager

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class DemoSceneManager : MonoBehaviour
```

Fields

currencyDefault

Declaration

```
public Button currencyDefault
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

currencyExample

Declaration

```
public Label currencyExample
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Label	

currencyGold

Declaration

```
public Button currencyGold
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

currencyState

Declaration

```
public bool currencyState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

document

Declaration

```
public UIDocument document
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.UIDocument	

Instance

Declaration

```
public static DemoSceneManager Instance
```

Field Value

TYPE	DESCRIPTION
DemoSceneManager	

startDemo

Declaration

```
public Button startDemo
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

weightExample

Declaration

```
public Label weightExample
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Label	

weightImperial

Declaration

```
public Button weightImperial
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

weightMetric

Declaration

```
public Button weightMetric
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

weightState

Declaration

```
public bool weightState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class DictionaryActions

This static class is used to perform actions on Inventory Dictionary.

Inheritance

System.Object
DictionaryActions

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public static class DictionaryActions
```

Methods

AddItemInToDictionary(InventoryElement, Boolean, InventoryItem[])

This will add an item to the dictionary.

Declaration

```
public static bool AddItemInToDictionary(InventoryElement target, bool canMerge, params InventoryItem[]
items)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the InventoryElement who's dictionary the item will be added to.
System.Boolean	canMerge	The boolean will determine if the item will marge with other dictionary entries of the same item.
InventoryItem[]	items	

Returns

TYPE	DESCRIPTION
System.Boolean	Return true if the item is Added.

AddItemInToDictionary(Dictionary<Int32, InventoryItem>, Int32, Boolean, InventoryItem[])

This will add an item to the dictionary.

Declaration

```
public static bool AddItemInToDictionary(Dictionary<int, InventoryItem> dictionary, int size, bool canMerge,
params InventoryItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that the item will be added to.
System.Int32	size	
System.Boolean	canMerge	The boolean will determine if the item will marge with other dictionary entries of the same item.
InventoryItem[]	items	

Returns

TYPE	DESCRIPTION
System.Boolean	Return true if the item is Added.

AmountCanFillOfItem(Dictionary<Int32, InventoryItem>, InventoryItem)

This will check the dictionary for the same item as the param item. and count the free space in each item.

Declaration

```
public static int AmountCanFillOfItem(Dictionary<int, InventoryItem> dictionary, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be used to get the amount.
InventoryItem	item	This is the Item that will be used to search.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Int32	The Return is the amount of free space on all of the param item in dictionary.

ConsolidateInventory(InventoryElement)

This will Consolidate InventoryItem(s) into the smallest number of stacks.

Declaration

```
public static void ConsolidateInventory(InventoryElement target)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	this is the InventoryElement who's dictionary will be consolidated.

ConsolidateInventory(Dictionary<Int32, InventoryItem>, Int32)

This will Consolidate InventoryItem(s) into the smallest number of stacks.

Declaration

```
public static void ConsolidateInventory(Dictionary<int, InventoryItem> dictionary, int size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be consolidated.
System.Int32	size	The size of the dictionary.

ConsolidateInventoryItems(InventoryElement)

This will Consolidate InventoryItem(s) into the smallest number of stacks without moving slots.

Declaration

```
public static void ConsolidateInventoryItems(InventoryElement target)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	this is the InventoryElement who's dictionary will be consolidated.

ConvertDictionaryToStorage(Dictionary<Int32, InventoryItem>)

This will add dictionary into the KeyItem list.

Declaration

```
public static List<KeyItem> ConvertDictionaryToStorage(Dictionary<int, InventoryItem> dictionary)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will added into the list of KeyItem(s).

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<KeyItem>	The return is the list of KeyItem(s).

ConvertStorageToDictionary(Dictionary<Int32, InventoryItem>, List<KeyItem>)

This will add KeyItem list into the dictionary.

Declaration

```
public static void ConvertStorageToDictionary(Dictionary<int, InventoryItem> dictionary, List<KeyItem> items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will take the KeyItem(s).
System.Collections.Generic.List<KeyItem>	items	This is the list of KeyItem(s) that will be added into the dictionary.

GetTransferItem(Dictionary<Int32, InventoryItem>, Int32)

This will create a TransferItem for the dictionary value at the param key.

Declaration

```
public static TransferItem GetTransferItem(Dictionary<int, InventoryItem> dictionary, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be used to get the TransferItem's Item.
System.Int32	key	This is the key that will be used to located the item in the dictionary.

Returns

TYPE	DESCRIPTION
TransferItem	The return is the TransferItem that is create.

MergeItemAtDictionaryKey(Dictionary<Int32, InventoryItem>, InventoryItem, Int32)

This will merge the Item into the dictionary at the index.

Declaration

```
public static bool MergeItemAtDictionaryKey(Dictionary<int, InventoryItem> dictionary, InventoryItem item,
int index)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be used to merge an item at an index.
InventoryItem	item	This is the item to be added.
System.Int32	index	This is the index of the dictionary to be added to.

Returns

TYPE	DESCRIPTION
System.Boolean	if the item has been merged fully or partially return true.

RemoveItems(InventoryElement, Int32[])

This will remove item(s) from the dictionary.

Declaration

```
public static void RemoveItems(InventoryElement target, params int[] keys)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the InventoryElement who's dictionary the item will be removed from.
System.Int32[]	keys	these params will be the keys pointing to the dictionary's entries to be removed.

RemoveItems(Dictionary<Int32, InventoryItem>, Int32[])

This will remove item(s) from the dictionary.

Declaration

```
public static void RemoveItems(Dictionary<int, InventoryItem> dictionary, params int[] keys)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that the item will be removed from.
System.Int32[]	keys	these params will be the keys pointing to the dictionary's entries to be removed.

SortByAmount(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item current stack, either Ascending or Desending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByAmount(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be sortted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Desending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	Return the dictionary after it has been sortted.

SortByCost(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item cost, either Ascending or Desending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByCost(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be sortted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Desending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	Return the dictionary after it has been sortted.

SortByName(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item name, either Ascending or Desending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByName(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be sortted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Desending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	Return the dictionary after it has been sortted.

SortByType(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item type, either Ascending or Desending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByType(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be sortted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Desending.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	Return the dictionary after it has been sortted.

SortByWeight(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item weight, either Ascending or Desending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByWeight(Dictionary<int, InventoryItem> dictionary, bool
isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	dictionary	This is the dictionary that will be sortted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Desending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	Return the dictionary after it has been sortted.

Class Enemy

This class is the script thats applied to a Enemy Character.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryTarget
Enemy

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
[RequireComponent(typeof(InventorySpawner))]
public class Enemy : InventoryTarget
```

Fields

health

This is the heath of the enemy.

Declaration

```
public int health
```

Field Value

TYPE	DESCRIPTION
System.Int32	

onDeath

When invoked will invoke all method subscribed to it.

Declaration

```
public Action onDeath
```

Field Value

TYPE	DESCRIPTION
System.Action	

Properties

Health

This gets health & sets health If health == 0 & isDead == false then set isDead to true & invoke Death Method.

Declaration

```
public int Health { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

isDead

This if true will stop Death Action from being invoke.

Declaration

```
public bool isDead { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Start()

This is a unity method, will set the decay timer. and subscribe the 'decay.StartTimer' and 'InitInventory' methods to the onDeath action

Declaration

```
public void Start()
```

Update()

This is a unity method, and will UpdateTimer on the decay timer unless the inventory is open or the decay timer hasn't started yet.

Declaration

```
public void Update()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class EquippedItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
WorldItem
EquippedItem

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class EquippedItem : WorldItem
```

Methods

Use()

This will be invoked when the player invokes "useItem" action. This will use the item wether as a weapon or other.

Declaration

```
public virtual void Use()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class GenerateData

This static Class is used to Generate Inventories & InventoryItems from ItemData files.

Inheritance

System.Object
GenerateData

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public static class GenerateData
```

Properties

NewInventory

This Property will create an instance of the scriptableObject 'Inventory'.

Declaration

```
public static Inventory NewInventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	The return is the Inventory that was created.

Methods

CreateInventory(Inventory, Int32, List<ItemData>, Int32, Boolean)

This will create an Inventory add items to an inventory's storageData.

Declaration

```
public static void CreateInventory(Inventory inventory, int slotLine, List<ItemData> itemDataList, int
inventoryGold, bool isRandom = false)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	inventory	Inventory file after it is created.

TYPE	NAME	DESCRIPTION
System.Int32	slotLine	how many slot will the UI have per row. so the inventory's can be determined along with number of items to be added.
System.Collections.Generic.List<ItemData>	itemDataList	list of ItemDat files that will be added to the inventory. between(1 - Max Stack).
System.Int32	inventoryGold	the amount of gold the inventory will have or will be the max random range if isRandom is true.
System.Boolean	isRandom	if true each item will be given a random key. else 1 by 1.

CreateInventoryItem(String, Int32)

This will create an InventoryItem and add an amount to it.

Declaration

```
public static InventoryItem CreateInventoryItem(string targetIID, int amount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	targetIID	the Item IID.
System.Int32	amount	the amount to be added to the InventoryItem.

Returns

TYPE	DESCRIPTION
InventoryItem	Return the InventoryItem that was created.

CreateRandomInventoryItem()

This will create a random InventoryItem and add a random amount to it.

Declaration

```
public static InventoryItem CreateRandomInventoryItem()
```

Returns

TYPE	DESCRIPTION
InventoryItem	Return the InventoryItem that was created.

FindItemData(String)

This will to located the ItemData by IID.

Declaration

```
public static ItemData FindItemData(string targetIID)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	targetIID	the IID of the ItemData to be found.

Returns

TYPE	DESCRIPTION
ItemData	Return ItemData file after it is found.

SetItems()

This will get all ItemData from the Resources folder and add them to the items list.

Declaration

```
public static void SetItems()
```

TryFindItemData(String, out ItemData)

This will try to located the ItemData by IID.

Declaration

```
public static bool TryFindItemData(string targetIID, out ItemData itemData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	targetIID	the IID of the ItemData to be found.
ItemData	itemData	out ItemData file after it is found.

Returns

TYPE	DESCRIPTION
System.Boolean	Return true if ItemData file is found.

Class InventorySpawner

This class is the InventorySpawner it will create an inventory scriptableobject on awake. and generate its Storage from the list of SpawnItem(s).

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventorySpawner

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class InventorySpawner : MonoBehaviour
```

Properties

Inventory

This is the Inventory that will be create.

Declaration

```
[SerializeField]
public Inventory Inventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	

Methods

GenerateInventoryStorage()

This will create a InventoryItem(s) from the spawnList and add them to a list of KeyItem(s).

Declaration

```
public List<KeyItem> GenerateInventoryStorage()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<KeyItem>	The return is the list of KeyItem(s).

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class InventoryTarget

This class is the base for all classes with with inventory. (Enemy, Player, Storage, Trader)

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryTarget
Enemy
Player
Storage
Trader

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class InventoryTarget : MonoBehaviour
```

Fields

inventory

This is the character's inventory.

Declaration

```
[SerializeField]
protected Inventory inventory
```

Field Value

TYPE	DESCRIPTION
Inventory	

inventorySlotLength

This is how many slots will be on each row of the inventoryUI;

Declaration

```
[SerializeField]
protected int inventorySlotLength
```

Field Value

TYPE	DESCRIPTION
System.Int32	

inventoryUI

This is the DialogBox that contains the inventory UI.

Declaration

```
protected DialogBoxElement inventoryUI
```

Field Value

TYPE	DESCRIPTION
DialogBoxElement	

inventoryUIPosition

This will be that value used to set the position of the UI on the screen.

Declaration

```
[SerializeField]
protected Vector2 inventoryUIPosition
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Vector2	

Properties

GetContent

This is the InventoryElement for this character.

Declaration

```
public InventoryElement GetContent { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement	

HasInventory

This is a property check that will see of 'inventoryUI' isn't null.

Declaration

```
public bool HasInventory { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsInventoryOpen

Declaration

```
public bool IsInventoryOpen { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

CloseInventory()

This will close the inventoryUI

Declaration

```
public void CloseInventory()
```

InitInventory()

This will initialises the inventoryUI for this character.

Declaration

```
public void InitInventory()
```

OnDestroy()

This is a unity method, and will remove the UI from its parent element.

Declaration

```
public void OnDestroy()
```

OpenInventory()

This will open the inventory

Declaration

```
public void OpenInventory()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class InventoryTrader

This static Class is used to Trade Inventory item(s).

Inheritance

System.Object
InventoryTrader

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public static class InventoryTrader
```

Methods

Cost(TransferItem[])

This will get the cost of the purchase.

Declaration

```
public static int Cost(params TransferItem[] transferItems)
```

Parameters

TYPE	NAME	DESCRIPTION
TransferItem[]	transferItems	params of TransferItem(s) that have their cost added together.

Returns

TYPE	DESCRIPTION
System.Int32	return the cost of the purchase.

ItemsToBuy((Inventory, Dictionary<Int32, InventoryItem>, Action), TransferItem[])

This will check a list of TransferItem(s) how many of each item will be purchase.

Declaration

```
public static List<TransferItem> ItemsToBuy((Inventory, Dictionary<int, InventoryItem>, Action) To, params
TransferItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ValueTuple<Inventory, System.Collections.Generic.Dictionary<System.Int32, InventoryItem>, System.Action>	To	The Inventory that will be the purchaser of the items.
TransferItem[]	items	The list of the TransferItem that will be purchased.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	This will return the list of TransferItem(s).

ItemsToTransfer((Inventory, Dictionary<Int32, InventoryItem>, Action), TransferItem[])

This will check if their is enough freeslots for the amount of items.

Declaration

```
public static List<TransferItem> ItemsToTransfer((Inventory, Dictionary<int, InventoryItem>, Action) To,
params TransferItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ValueTuple<Inventory, System.Collections.Generic.Dictionary<System.Int32, InventoryItem>, System.Action>	To	The Inventory that will be the purchaser of the item.
TransferItem[]	items	The list of the TransferItem that will be Transferred.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	This will return the list of TransferItem(s).

ItemToPurchase(TransferItem, Int32)

Declaration

```
public static (TransferItem, int) ItemToPurchase(TransferItem item, int currencyValue)
```

Parameters

TYPE	NAME	DESCRIPTION
TransferItem	item	
System.Int32	currencyValue	

Returns

TYPE	DESCRIPTION
System.ValueTuple<TransferItem, System.Int32>	

TransferItems((Inventory, Dictionary<Int32, InventoryItem>, Action), (Inventory, Dictionary<Int32, InventoryItem>, Action), TransferItem[])

This will transfer all of the TransferItem(s).

Declaration

```
public static void TransferItems((Inventory, Dictionary<int, InventoryItem>, Action) From, (Inventory, Dictionary<int, InventoryItem>, Action) To, params TransferItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ValueTuple<Inventory, System.Collections.Generic.Dictionary<System.Int32, InventoryItem>, System.Action>	From	The Inventory that will be the Seller of the item.
System.ValueTuple<Inventory, System.Collections.Generic.Dictionary<System.Int32, InventoryItem>, System.Action>	To	The Inventory that will be the purchaser of the item.
TransferItem[]	items	These TransferItem(s) that will be traded.

TransferMoney(Inventory, Inventory)

This will transfer all of the currency From an inventory To another.

Declaration

```
public static void TransferMoney(Inventory From, Inventory To)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	From	The Inventory that will be the Seller of the item.
Inventory	To	The Inventory that will be the purchaser of the item.

TransferMoney(Inventory, Inventory, Int32)

This will transfer an amount of currency From an inventory To another.

Declaration

```
public static void TransferMoney(Inventory To, Inventory From, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	To	The Inventory that will be the Seller of the item.
Inventory	From	The Inventory that will be the purchaser of the item.
System.Int32	amount	The amount of currency that will be traded.

Class InventoryTraderManager

This class handles the inventories and logic for transfering of item(s) from 1 inventory to another.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryTraderManager

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
[RequireComponent(typeof(UIDocument))]
public class InventoryTraderManager : MonoBehaviour
```

Fields

contextMenu

Declaration

```
public ContextMenuElement contextMenu
```

Field Value

TYPE	DESCRIPTION
ContextMenuElement	

CurrencyState

Declaration

```
[SerializeField]
public bool CurrencyState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Instance

This is a singleton Instance variable.

Declaration

```
public static InventoryTraderManager Instance
```

Field Value

TYPE	DESCRIPTION
InventoryTraderManager	

WeightState

Declaration

```
[SerializeField]
public bool WeightState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

CountAll

This Property is the count of the inventories.

Declaration

```
public int CountAll { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountLoots

This Property is the count of the inventories of type Loot.

Declaration

```
public int CountLoots { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountPlayers

This Property is the count of the inventories of type Player.

Declaration

```
public int CountPlayers { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountStorages

This Property is the count of the inventories of type Storage.

Declaration

```
public int CountStorages { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountTraders

This Property is the count of the inventories of type Trader.

Declaration

```
public int CountTraders { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

dialogboxes

This is the list of DialogBoxes that hold an InventoryElement.

Declaration

```
public List<DialogBoxElement> dialogboxes { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<DialogBoxElement>	

FindAll

This Property is all of the InventoryElements in inventories.

Declaration

```
public InventoryElement[] FindAll { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement[]	

FindLoots

This Property is all of the InventoryElements in inventories of type Loot.

Declaration

```
public InventoryElement[] FindLoots { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement[]	

FindPlayer

This Property is all of the InventoryElements in inventories of type Player.

Declaration

```
public InventoryElement FindPlayer { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement	

FindStorages

This Property is all of the InventoryElements in inventories of type Storage.

Declaration

```
public InventoryElement[] FindStorages { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement[]	

FindTraders

This Property is all of the InventoryElements in inventories of type Trader.

Declaration

```
public InventoryElement[] FindTraders { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement[]	

Methods

AddInventory(InventoryElement)

This will add an InventoryElement to the inventories list.

Declaration

```
public void AddInventory(InventoryElement inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	inventory	This is the inventory that will added.

ContextMenu(InventoryElement, SlotElement)

This will Create a context menu from the button on the inventory.

Declaration

```
public void ContextMenu(InventoryElement target, SlotElement slot)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the target InventoryElement that will be used to get the Context Menu Item Data from.
SlotElement	slot	This is the SlotElement that will be used to set the position of the context menu on the screen.

ContextMenu(InventoryElement, Button)

This will Create a context menu from the button on the inventory.

Declaration

```
public void ContextMenu(InventoryElement target, Button button)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the target InventoryElement that will be used to get the Context Menu Item Data from.
UnityEngine.UIElements.Button	button	This is the button that will be used to set the position of the context menu on the screen.

CreateInventoryElement(Inventory, Int32, Vector2, out DialogBoxElement)

This will create the InventoryElement for the inventory.

Declaration

```
public bool CreateInventoryElement(Inventory inventory, int size, Vector2 position, out DialogBoxElement
dialogBox)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	inventory	This is the inventory that requires a UI for interaction.
System.Int32	size	This is the number of Slots that the UI will have.
UnityEngine.Vector2	position	This is the position of the DialogBox UI will be set at.
DialogBoxElement	dialogBox	This out is the DialogBoxElement so it can be returned for the created script.

Returns

TYPE	DESCRIPTION
System.Boolean	This will return true if the dialogbox is created for the UI.

DetailBox(Int32, InventoryElement)

This will create a DialogBoxElement for the item details.

Declaration

```
public void DetailBox(int key, InventoryElement inventoryElement)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This is the key for the dictionary's key.
InventoryElement	inventoryElement	This param is the target that holds the dictionary.

OpenSlotPopup(Int32, InventoryElement, Boolean)

Declaration

```
public void OpenSlotPopup(int key, InventoryElement From, bool isSplit)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	
InventoryElement	From	
System.Boolean	isSplit	

OpenTransferPopup(TransferItem, InventoryElement, InventoryElement)

This will open a popup UI for transfer an item from an inventory to another.

Declaration

```
public void OpenTransferPopup(TransferItem item, InventoryElement To, InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
TransferItem	item	This is the TransferItem that will be the target of transfer.
InventoryElement	To	This is the inventory that will own that item when traded.
InventoryElement	From	This is the owner of the item.

RemoveContextMenu()

This will remove the context menu from the UI if it exist.

Declaration

```
public void RemoveContextMenu()
```

RemoveInventory(InventoryElement)

This will remove an InventoryElement to the inventories list.

Declaration

```
public void RemoveInventory(InventoryElement inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	inventory	This is the inventory that will removed.

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class PickupItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
WorldItem
PickupItem

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class PickupItem : WorldItem
```

Methods

PickUp()

This will be invoked when the player invokes "useItem" action. This will pickup the item and add it to the player's inventory.

Declaration

```
public void PickUp()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class Player

This class is the script thats applied to a player Character.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryTarget
Player

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class Player : InventoryTarget
```

Fields

equippedItem

This is the gameobject for the item in the playerHand.

Declaration

```
public GameObject equippedItem
```

Field Value

TYPE	DESCRIPTION
UnityEngine.GameObject	

Instance

This is a singleton instance.

Declaration

```
public static Player Instance
```

Field Value

TYPE	DESCRIPTION
Player	

playerHand

This is the gameobject for the hand. When item is equipped it will be apply to the hand.

Declaration

```
public GameObject playerHand
```

Field Value

TYPE	DESCRIPTION
UnityEngine.GameObject	

useItem

This is for when the equippedItem is used.

Declaration

```
public Action useItem
```

Field Value

TYPE	DESCRIPTION
System.Action	

Methods

DestroyHeldObjects()

This will destory the items in the player's hand.

Declaration

```
public void DestroyHeldObjects()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class Storage

This class is the script thats applied to a storage inventory.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryTarget
Storage

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class Storage : InventoryTarget
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class Trader

This class is the script thats applied to a Trader Character.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryTarget
Trader

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public class Trader : InventoryTarget
```

Fields

currency

This is the currency of the trader when it gets a restock.

Declaration

```
public int currency
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

Restock()

This is the method that will be called for restocking of the inventory. it will also reset the timer so when it the inventoryUI is open again it will start the timer again.

Declaration

```
public void Restock()
```

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class WorldItem

This abstract class is the base of any item in the world. such as the "PickupItem" or the "EquippedItem".

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
WorldItem
EquippedItem
PickupItem

Namespace: InventorySystem.Script

Assembly: Assembly-CSharp.dll

Syntax

```
public abstract class WorldItem : MonoBehaviour
```

Fields

item

This is the item that is related to this world item. The "InventoryItem" is used over the "ItemData" as the amount will be needed for pickup as it could be a stack of the item. Or could be a consumable equipped item.

Declaration

```
[SerializeField]
public InventoryItem item
```

Field Value

TYPE	DESCRIPTION
InventoryItem	

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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Namespace InventorySystem.UI

Classes

IconElement

This class is an element that is created for the Inventory.

IconElement.UxmlFactory

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

InventoryElement

this class is an element for the inventory.

InventoryElement.UxmlFactory

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

SlotElement

This class is an element that is created for the Inventory.

SlotElement.UxmlFactory

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

SlotPopupElement

This class is the popup for the Slot, for the splitting and dropping.

SlotPopupElement.UxmlFactory

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

TransferPopupElement

This class is the popup for the Slot, for the selling and trading.

TransferPopupElement.UxmlFactory

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

Class IconElement

This class is an element that is created for the Inventory.

Inheritance

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

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class IconElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations,
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

IconElement()

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

Declaration

```
public IconElement()
```

IconElement(InventoryItem, StyleSheet)

Declaration

```
public IconElement(InventoryItem item, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	
UnityEngine.UIElements.StyleSheet	style	

Methods

ItemAmount(Int32)

This 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

Select()

This will set the border color to green to show the user it is selected.

Declaration

```
public void Select()
```

Style()

This will set the style of the Icon.

Declaration

```
public void Style()
```

UnSelect()

This will set the border color to auto to show the user it is unselected.

Declaration

```
public void UnSelect()
```

UpdateIcon(Int32)

This 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

VisualElementStyleExtension.BackGround_Color(VisualElement, Color)
VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)

VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)

VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)

VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)

VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)

CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class IconElement.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<IconElement, UnityEngine.UIElements.VisualElement.UxmlTraits>
IconElement.UxmlFactory

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<IconElement, 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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class InventoryElement

this class is an element for the inventory.

Inheritance

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

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class InventoryElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations,
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

InventoryElement()

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

Declaration

```
public InventoryElement()
```

InventoryElement(Int32, Int32, Inventory, StyleSheet)

This is the constructor for this class.

Declaration

```
public InventoryElement(int numberOfSlots, int slotsPerRow, Inventory Inventory, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	numberOfSlots	This is the number of slots that will be added to the UI.
System.Int32	slotsPerRow	
Inventory	Inventory	This is the Inventory for this UI.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleSheet	style	This is an optional StyleSheet.

Fields

currentSortOrder

This enum is used for sorting the UI item.

Declaration

```
public SortByEnum currentSortOrder
```

Field Value

TYPE	DESCRIPTION
SortByEnum	

dictionary

This is the Dictionary that will store the InventoryItem(s).

Declaration

```
public Dictionary<int, InventoryItem> dictionary
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem>	

footer

This is the footer section.

Declaration

```
public VisualElement footer
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

header

This is the header section.

Declaration

```
public VisualElement header
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

isDescending

this boolean is used in sort and if it should be ascending or desending.

Declaration

```
public bool isDescending
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

onUpdate

This is the Action that will be called when the ui will be updated.

Declaration

```
public Action onUpdate
```

Field Value

TYPE	DESCRIPTION
System.Action	

slotParent

This is the parent element that hold the slot(s).

Declaration

```
public VisualElement slotParent
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

slots

This is the list of slots on the UI.

Declaration

```
[SerializeField]
public List<SlotElement> slots
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<SlotElement>	

slotsToUpdate

This is a list of slots that need to be update.

Declaration

```
public List<int> slotsToUpdate
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

titleName

This is the string that will be used on the UI title.

Declaration

```
public string titleName
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

AllTransferItems

This will return a list of all TransferItem(s) from the dictionary.

Declaration

```
public List<TransferItem> AllTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	

ArmourTransferItems

This will return a list of TransferItem(s) from the dictionary of type Armour.

Declaration

```
public List<TransferItem> ArmourTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	

ConsumableTransferItems

This will return a list of TransferItem(s) from the dictionary of type Consumable.

Declaration

```
public List<TransferItem> ConsumableTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	

Inventory

This property is the Inventory of the character.

Declaration

```
public Inventory Inventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	

MaterialsTransferItems

This will return a list of TransferItem(s) from the dictionary of type Materials.

Declaration

```
public List<TransferItem> MaterialsTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	

MiscTransferItems

This will return a list of TransferItem(s) from the dictionary of type Misc.

Declaration

```
public List<TransferItem> MiscTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	

Type

This property is the Inventory's Type.

Declaration

```
[SerializeField]
public InventoryType Type { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryType	

UID

This property is the Unique id from the Inventory.

Declaration

```
public string UID { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

WeaponTransferItems

This will return a list of TransferItem(s) from the dictionary of type Weapon.

Declaration

```
public List<TransferItem> WeaponTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List<TransferItem>	

Methods

GenerateSlots()

This will create and set up the slotElements.

Declaration

```
public void GenerateSlots()
```

OnClose()

This should be invoked when closing the UI.

Declaration

```
public void OnClose()
```

OnOpen()

This will be invoked in constructor.

Declaration

```
public void OnOpen()
```

Remarks

should be invoked when opening UI. (if the onClose is used.)

SetCurrencyValue(Int32, Int32)

This will set the currencyValue from when the inventory's currency is changed.

Declaration

```
public void SetCurrencyValue(int gold, int valueChanged)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	gold	This is the Currency Value.
System.Int32	valueChanged	This is how much it is changed by.

SetWeightValue()

This will set the weightValue from when the inventory's updated.

Declaration

```
public void SetWeightValue()
```

Sortby(SortByEnum)

This will sort the dictionary by enum.

Declaration

```
public void Sortby(SortByEnum sortBy)
```

Parameters

TYPE	NAME	DESCRIPTION
SortByEnum	sortBy	This param is an enum that will be used to choose what the dictionary will be sorted by.

ToggleInteractable(Boolean)

This will allow or disallow the user to interact with any of the slots.

Declaration

```
public void ToggleInteractable(bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	value	This value will determine of interactions will be allowed or disallowed.

UpdateAllSlots()

This will update all of the slots with corresponding dictionary item or null.

Declaration

```
public void UpdateAllSlots()
```

UpdateSlots()

This will update only the slot(s) that were added to the slotsToUpdate list. then clear the slotsToUpdate list.

Declaration

```
public void UpdateSlots()
```

Extension Methods

VisualElementStyleExtension.BackGround_Color(VisualElement, Color)
VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)

VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])

VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)

VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)

VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class InventoryElement.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<InventoryElement, UnityEngine.UIElements.VisualElement.UxmlTraits>
InventoryElement.UxmlFactory

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<InventoryElement, 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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class SlotElement

This class is an element that is created for the Inventory.

Inheritance

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

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class SlotElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations,
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

SlotElement()

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

Declaration

```
public SlotElement()
```

SlotElement(Int32, StyleSheet)

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

Declaration

```
public SlotElement(int key, StyleSheet style = null)
```

Parameters

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

Properties

GetIcon

This property will get Icon attached to this Slot.

Declaration

```
[SerializeField]
public IconElement GetIcon { get; }
```

Property Value

TYPE	DESCRIPTION
IconElement	Return the Icon.

HasIcon

This property will check if the slot has an icon.

Declaration

```
[SerializeField]
public bool HasIcon { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Key

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

Declaration

```
public int Key { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

ClearIcon()

This will remove the icon from this slot.

Declaration

```
public void ClearIcon()
```

SetIcon(InventoryItem)

This 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 on the Icon.

SetIcon(IconElement)

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

Declaration

```
public void SetIcon(IconElement icon)
```

Parameters

TYPE	NAME	DESCRIPTION
IconElement	icon	

Style()

This will set the style of the Icon.

Declaration

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

UpdateSlot(InventoryItem)

This will update the slot's icon with the new item.

Declaration

```
public void UpdateSlot(InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	this is the item that will update the slot's icon.

Extension Methods

VisualElementStyleExtension.BackGround_Color(VisualElement, Color)
VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)

VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)

VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)

VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)

VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)

CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class SlotElement.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<SlotElement, UnityEngine.UIElements.VisualElement.UxmlTraits>
SlotElement.UxmlFactory

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<SlotElement, 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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class SlotPopupElement

This class is the popup for the Slot, for the splitting and dropping.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
PopupBaseElement
SlotPopupElement

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class SlotPopupElement : PopupBaseElement, IEventHandler, ITransform, ITransitionAnimations,
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

SlotPopupElement()

This is the constructor

Declaration

```
public SlotPopupElement()
```

SlotPopupElement(Vector2, Int32, InventoryElement, Boolean)

This is the constructor

Declaration

```
public SlotPopupElement(Vector2 size, int key, InventoryElement From, bool isSplit)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	This is the size of the popup.
System.Int32	key	This the key of the item in the dictionary.
InventoryElement	From	This is the From of the dictionary.

TYPE	NAME	DESCRIPTION
System.Boolean	isSplit	This bool will determine if it will split or drop the item.

Methods

GenerateElements()

This will generate the elements for the popup

Declaration

```
public override void GenerateElements()
```

Overrides

PopupBaseElement.GenerateElements()

Style(Vector2)

This will style the element.

Declaration

```
public override void Style(Vector2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	

Overrides

PopupBaseElement.Style(Vector2)

Submit()

This will be invoke by the submit button and will either drop or split the item.

Declaration

```
public override void Submit()
```

Overrides

PopupBaseElement.Submit()

Extension Methods

VisualElementStyleExtension.BackGround_Color(VisualElement, Color)
VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])

VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)

VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)

VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class SlotPopupElement.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<SlotPopupElement, UnityEngine.UIElements.VisualElement.UxmlTraits>
SlotPopupElement.UxmlFactory

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<SlotPopupElement, 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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class TransferPopupElement

This class is the popup for the Slot, for the selling and trading.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
PopupBaseElement
TransferPopupElement

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class TransferPopupElement : PopupBaseElement, IEventHandler, ITransform, ITransitionAnimations,
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

TransferPopupElement()

This is the constructor

Declaration

```
public TransferPopupElement()
```

TransferPopupElement(Vector2, TransferItem, InventoryElement, InventoryElement)

This is the constructor

Declaration

```
public TransferPopupElement(Vector2 size, TransferItem item, InventoryElement To, InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	This is the size of the popup.
TransferItem	item	This is the item that will be split or dropped.
InventoryElement	To	This is the purchaser of the item.

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the owner of the item.

Methods

GenerateElements()

This will generate the elements for the popup

Declaration

```
public override void GenerateElements()
```

Overrides

PopupBaseElement.GenerateElements()

Style(Vector2)

This will style the element.

Declaration

```
public override void Style(Vector2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	

Overrides

PopupBaseElement.Style(Vector2)

Submit()

This will be invoke by the submit button and will either drop or split the item.

Declaration

```
public override void Submit()
```

Overrides

PopupBaseElement.Submit()

Extension Methods

VisualElementStyleExtension.BackGround_Color(VisualElement, Color)
VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])

VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)

VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)

VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class TransferPopupElement.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<TransferPopupElement, UnityEngine.UIElements.VisualElement.UxmlTraits>
TransferPopupElement.UxmlFactory

Namespace: InventorySystem.UI

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<TransferPopupElement, 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)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)