Virtual Currency

1. Introduction

Virtual Currency is crucial part of almost every game. While in development process it could be highly challenging to manage currency especially when you have a lot of currency items e.g. Coins, Cash, Gems, Gold, Dimond and many others.

This currency module is intended to make currency management highly generic and reusable in almost any sort of unity project.

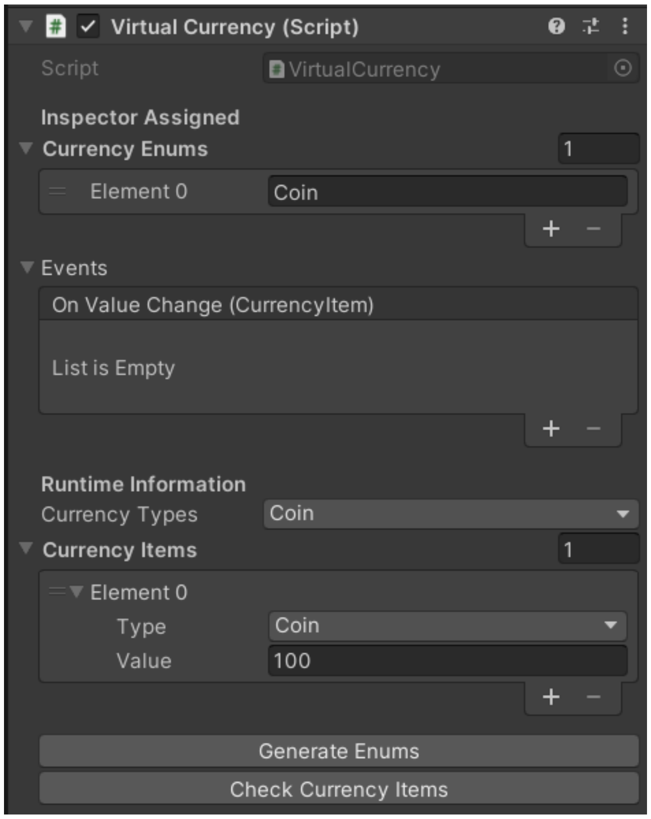
2. Dependencies

This module has following dependencies to work effectively.

- Folder structure in your project should be as follow:
    - *Assets/_CGS/Scripts/Virtual Currency*
- There should be a file in Virtual Currency folder named exactly as "CurrencyTypes.cs"

3. VirtualCurrency.cs

Once all the dependencies are resolved, all you have to do is to create an empty gameObject place it in the splash scene and attach the script "VirtualCurrency.cs" to it and you'll have the following inspector window of that gameObject.



Virtual Currency (Script)
Script: VirtualCurrency

Inspector Assigned
Currency Enums: 1
Element 0: Coin

Events
On Value Change (CurrencyItem)
List is Empty

Runtime Information
Currency Types: Coin
Currency Items: 1
Element 0
Type: Coin
Value: 100

First public field is a list of strings named "Currency Enums" under "Inspector Assigned" section. This list will contain the names of all of your currency items. You can delete an existing currency item and add more currency items, for now there's only one item i.e. Coin. Once you have inserted names of all of your currency items you need press "Generate Enums" button at the bottom. It will dynamically create an enum of type "CurrencyType" with all the values that present in "Currency Enums" list and it will be show under "Runtime Information" section in the inspector.

To manipulate values against the currency items that you've just added "VirtualCurrency.cs" exposes a public singleton instance variable to access the Virtual Currency gameObject in any script, three public methods and one event that you can use to get notified whenever a currency item get modified.

Singleton

public static VirtualCurrency instance;

Public Methods

public void AddCurrency(CurrencyType type, int amount);

This method takes the type of currency against which your are adding values. Don't you worry CurrencyType enum will be holding all the values against the currency items that you have added in "Currency Enums" list.

Example:

```
VirtualCurrency.instance.AddCurrency(CurrencyType.Coin, 100);
```

public bool SpendCurrency(CurrencyType type, int amount);

This method takes the type of currency that you want to spend and obviously the amount you want to spend. It will return true if spending is successful (i.e. amount to spend is less of equal to the amount we have stored against that currency item) and false otherwise.

Example:

```
bool result = VirtualCurrency.instance.SpendCurrency(CurrencyType.Coin, 50);
```

public int GetCurrency(CurrencyType type);

This method takes the type of currency you want to inquire about and returns the amount against that currency type.

Example:

```
int totalCoins = VirtualCurrency.instance.GetCurrency(CurrencyType.Coin);
```

Public Event

VirtualCurrency.Events.OnValueChange(CurrencyItem item);

You want to get notified when ever a currency item's value is changed then write a function that take parameter of type "CurrencyItem" and register it to above event.

Example:

```
Class MyClass : Monobehaviour {

        public Text text_Coin;

        private void Start() {

                VirtualCurrenc.instance.Events.OnValueChange.AddListener(Listen);
        }

        private void Listen(VirtualCurrency.Event.CurrencyItem item) {

                if (item.type == CurrencyType.Coin) {

                        text_Coin.text = item.value.ToString();

                }

        }

}
```

And lastly for debugging purposes at any time of development if you press the button named "Check Currency Items" it will populate the "Currency Items" list, each element of that list will contain the currency type and a value stored against that currency type.