EasyInventory

Creating a Bank object

Index

A) Getting your script ready

1. Setting up your code

Now that we've imported the package successfully, we can now setup our scene to test a basic inventory.

2. Creating an ***Empty***

Next we will need to create an *Empty*. Right click on the hierarchy inspector and set Create Empty

Rename your *Empty* to "EasyInventoryTest". Your unity editor should look like the following figure.

See Figure - 1

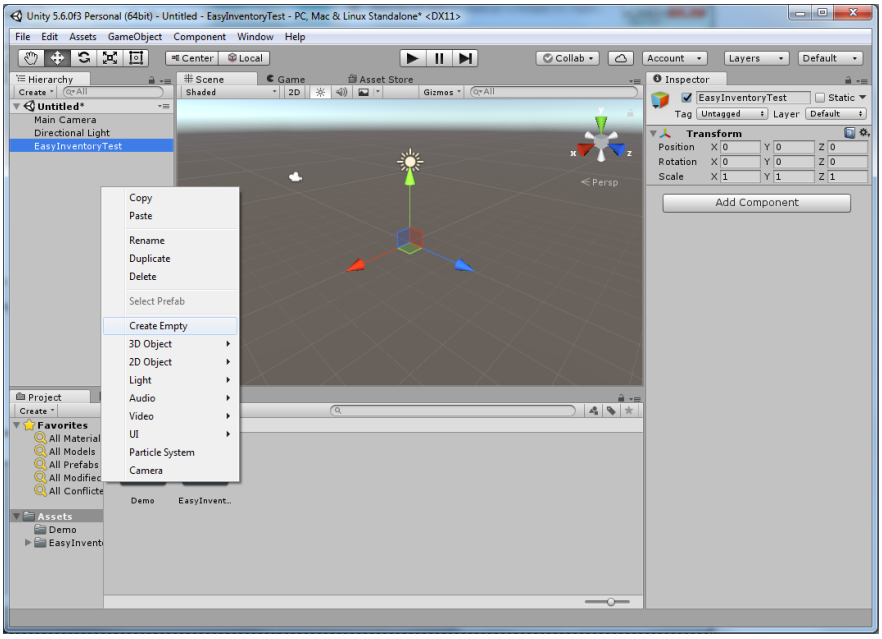

Figure - 1

3. Creating your first script

Next we will create our familiar MonoBehaviour. Right click on the project inspector and select Create -> C# Script. Rename this script to "EasyInventoryTest". Open your new script in one of your favorite *IDE's*.

See Figure - 2

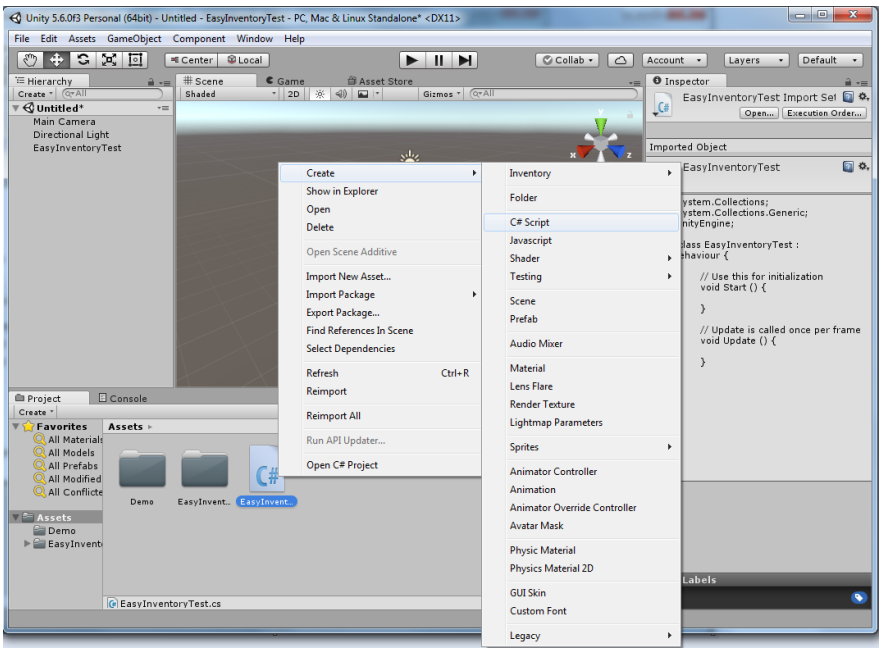

Figure - 2

4. Assigning your component

Finally the last step, we will apply our component to our empty object.

See Figure - 3

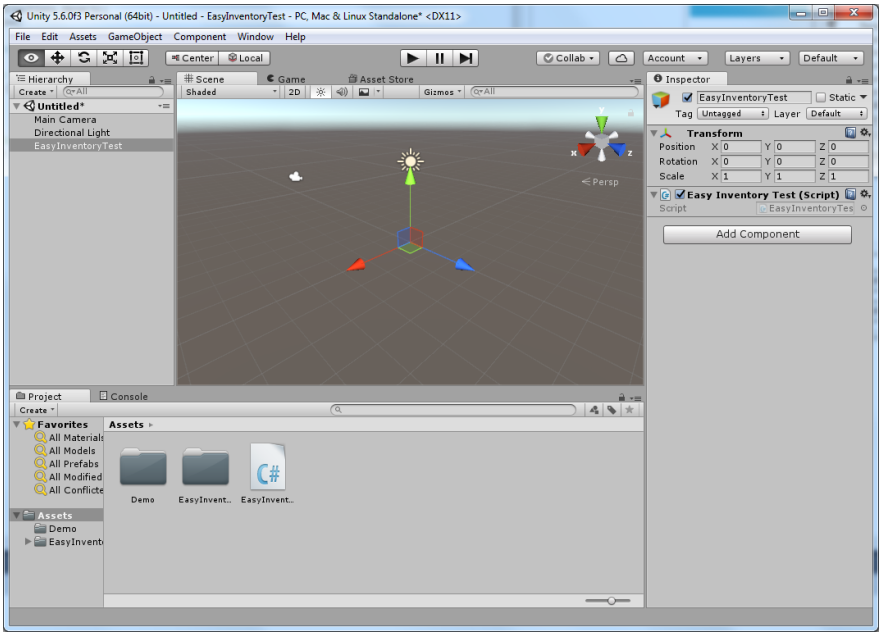

Figure - 3

Continue

B) Creating your first bank instance

1. Importing the correct modules

Before we begin coding, we will need to import a few things. Lets import the EasyInventory services interface.

```
using EasyInventory.ServicesInterface;
```

With this package, we will be able to create our first Inventory object.

2. The ***Bank*** service

There are two kinds of inventories that this package holds. We have an Inventory and a Bank. For those who aren't *Runescape* players, we will explain what a bank actually is.

3. What is a ***Bank***?

A bank inventory system is where you can hold all of your items. The bank size can be modified when instantiated. You cannot add more items to this inventory which exceeds the amount of bank slots. If you specify 28 bank slots, you cannot store more than 28 unique bank items.

A bank system can also store and stack items. Even if your item is applied as *non-stackable*, the bank system will force a stack. See the illustration below as an example.

Bank example

See Figure - 4

Figure - 4

Inventory example

See Figure - 5

Figure - 5

As you can see, even if an item is marked as *non-stackable*, the bank will force them together and begin a counter. The inventory cannot stack items, thus they reside within a separate slots individually.

4. Instantiating a bank object

You can create a bank with one single statement as seen below.

```
Bank myBank = new Bank(96);
```

Congratulations, you've created your first bank!

With this instance, you can add/remove items.

Continue

Up Next Creating and managing items->

Developed by Corey St-Jacques
Questions please contact Corey_stjacques@hotmail.com