

# RPG BOX DOCUMENTATION



# Table of Content

---

Subject	Page
Free Version VS Pro Version	4
Items	5
Attributes	7
Loot Tables	9
Inventory System	10
Shop System	12
Equipment System	14
Items Tooltip	16
DatabaseManager	17

Subject	Page
Error Messages System	18
Bars System	19
PRO Only: Save and Load	20
PRO Only: Enemy System	21
PRO Only: Floating Numbers	23
PRO Only: Items Creator	24
PRO Only: Loot Tables Creator	24
PRO Only: Enemies Creator	24
Contact Information	25

# What is new in RPG Box V1.2

---

- **Pro Version Only:** Added Save and Load system (Page 20)
  - **Giveaway:** Since no feature has been added to the Free version. We will give away 5 Pro Version keys to 5 random users who Rate and Write a review about RPG Box – Free Version in Unity AssetStore (Please send us your email (contact page 25) and your Unity AssetStore username when you submit a review so we can communicate with you if you win). Giveaway Ends in 9/1/2017 or when at least 10 reviews are submitted
  - **Last Chance:** Pro Version price is discounted by 50%. Get it before the price increases to 19.99\$ in few days
- 

Please write a review and rate RPG Box in Unity Asset Store so we can add more features

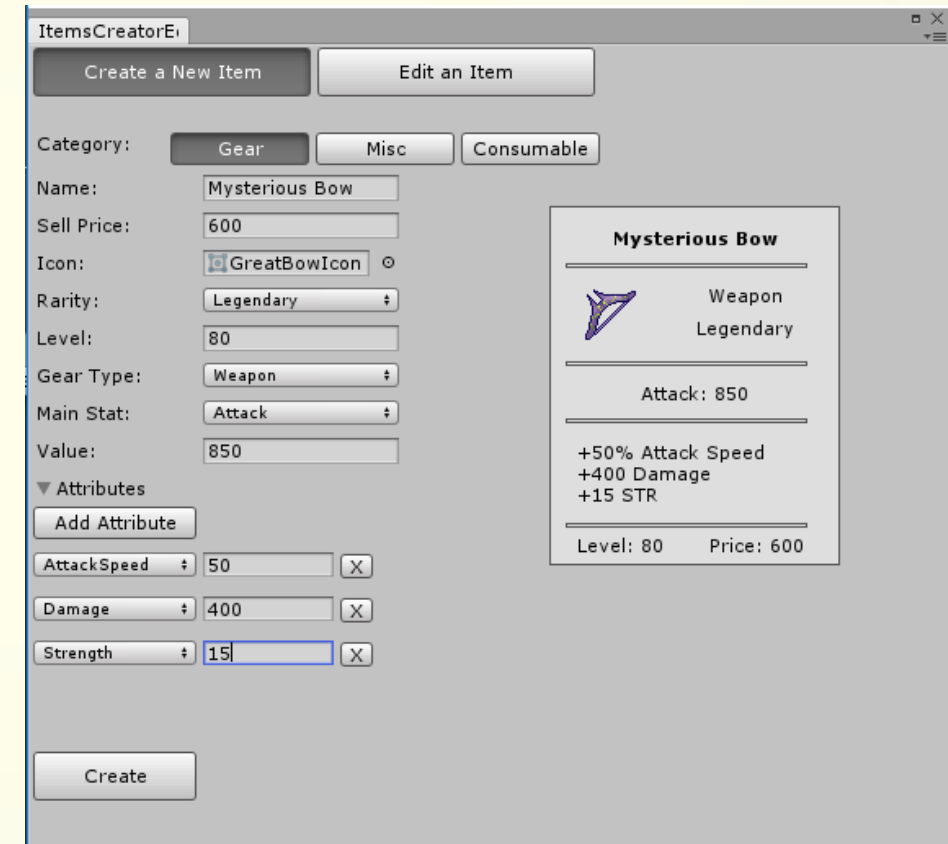
**Pro Version:** <https://www.assetstore.unity3d.com/en/#!/content/94554>

**Free Version:** <https://www.assetstore.unity3d.com/en/#!/content/94542>

# What are the differences between Free and Pro versions

Pro version has all the feature of the Free version plus:

- Item Creator Editor.
- Loot Tables Creator Editor.
- Enemy System
- Enemy Creator Editor.
- Loot Explosion When a Monster Dies
- Floating Number System
- Save and Load System
- Attribute Manager (Example: Damage Attribute makes player deal more damage)



Item Creator Editor.

# Items: How to Create

---

- Item is a ScriptableObject which contains all the information of the item.
- Right click on the asset folder and create “Item”.
- To let the DatabaseManager loads the Item, it needs to be under “ItemsBases” folder
- Main scripts are Item and ItemProps.

# Items: Brief Introduction About Each Script

---

## ❖Item Script:

- Item script contains many enums for item information (such as ItemType and MainStat)
- It contains all the needed information of items.

## ❖ItemProps Script:

- This script is for processing the item information
- ConsumeItem() method is responsible for consuming items of type Consumable
- MouseClicked() is called when the player right clicked any item.
- TakeInfo() must be called first when creating a new item during run time.
- PRO Only:** There is a section of variables for when the monster dies and drops loot

## Attributes: How to set up

---

- Attribute is a ScriptableObject which contains all the information of the attribute.
- Right click on the asset folder and create “Attribute”.
- To let the DatabaseManager load the Item, it needs to be under “Attributes” folder
- For the display text, you have to write #VALUE# instead of the exact value, so the item tooltip can change that string to an actual value. Example: +#VALUE# Health

# Attributes: Brief Introduction About The Script

---

❖ **PRO Only:** AttributeManager Script:

This class is the one which is responsible for applying Attributes to your Player Character when an item is equipped and removes the Attribute when the item is unequipped.

For every Attribute you have created in the game, create an IEnumerator method with name equals to the attribute name (just copy the example Method and add your own variable)



# Loot Tables

---

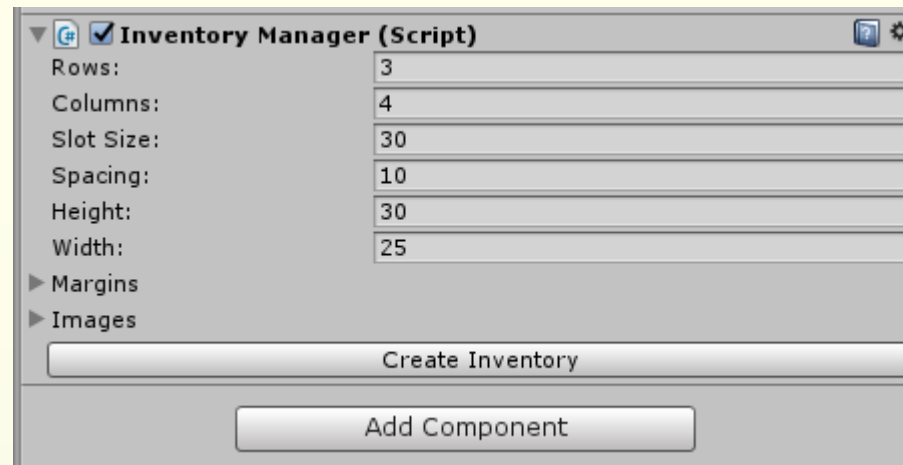
- LootTable is a ScriptableObject which contains all the information of the LootTable.
- Right click on the asset folder and create “Loot Table ”.
- To let the DatabaseManager loads the Item, it needs to be under “LootTables” folder
- Main scripts is LootTable which contains all the information of the Loot Table as well as two methods: GetAllItems() which returns all the items of the loot table and GenerateItemsFromLootTable() which returns items based on their drop chance

# Inventory System: How to set up

---

-Inventory is a place where the player stores his items in.

-To set up an inventory: go to InventoryWindow GameObject and change values of the script InventoryManager then press Create Inventory.



# Inventory System: Brief Introduction About The Script

---

## ❖ InventoryManager Script:

### 1- Methods:

- AddItemToInventory() is called when you want to add an item to the inventory
- RemoveItemFromInventory() is called when you want to remove an item from the inventory

### 2- Variables

- Inventory is the dictionary of items and which slot has which item.
- MaxNumberOfStacks is the maximum number of stacks a player can have of any consumable
- ListOfStackables list for all of the stackable items in the inventory

# Shop System: How to set up

---

- Shop is a place where the player buys items from or sells items for.
- To set up an Shop : go to Shop Window GameObject and change values of the script Shop Manager then press Create Shop.
- Reroll button is when the player wants to reroll the items of the shop (You can delete it if you want)
- Sell button is a button which allows the player to sell items from his inventory (You can delete the button and call its function when the shop is opened for example)

# Shop System: Brief Introduction About The Script

---

## ❖ ShopManager Script:

### 1- Methods:

- CreateItemsInTheShop() is called when you want to add items from a LootTable to the shop
- RerollItemsInShop() removes all items and generates others from a LootTable
- RemoveItemFromShop() is called when you want to remove an item from the shop
- BuyFromShop() is called when the player wants to buy an item from the shop

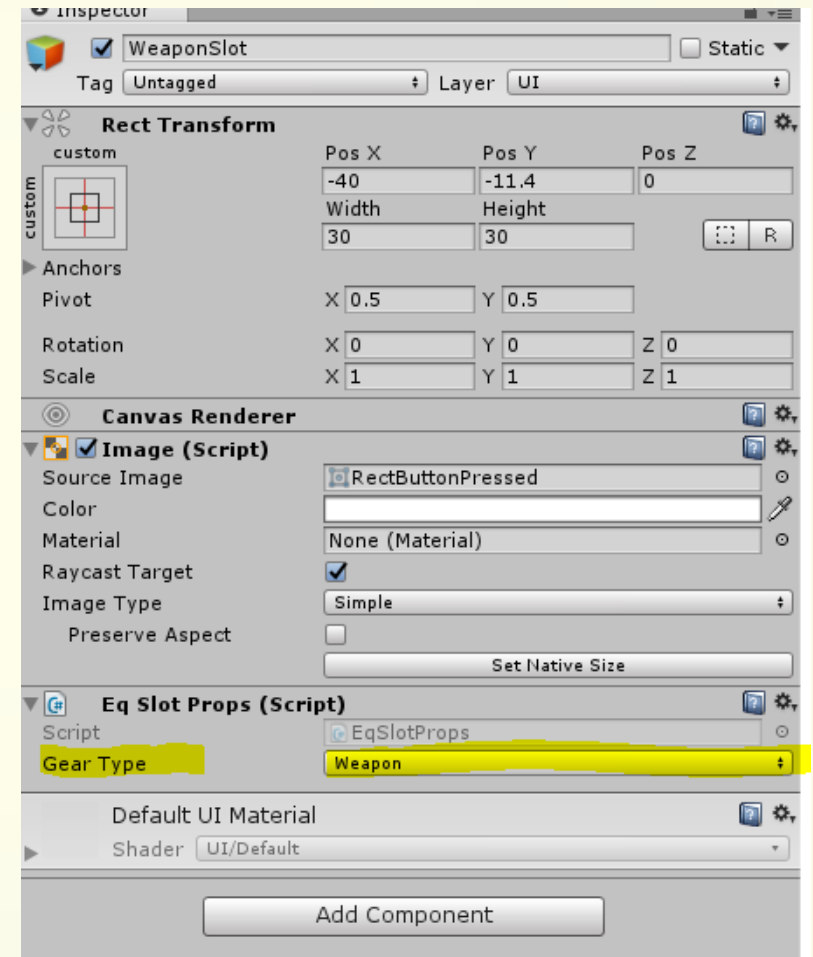
### 2- Variables

- PlayerBuyInventory is the dictionary of items and which slot has which item.

# Equipment System: How to set up

---

- Equipment Inventory is the place where the player equip items from the inventory.
- To set up an Equipment slot, add a EqSlotPrefab from Prefabs folder into EquipmentSlots of the EquipmentWindow.
- Change the Slot type by changing it from the “Gear Type” drop down menu of EqSlotProps component



# Equipment System: Brief Introduction About The Script

---

## ❖ EquipmentManager Script:

### 1- Methods:

- AddItemToEquipmentInventory() Equips an item
- RemoveItemFromEquipmentInventory() Unequips an item
- **PRO Only:** OnEquip() called when a player equips an item (it applies all of the item Attributes to the character)
- **PRO Only:** OnUnequip() called when a player unequips an item (it removes all of the item Attributes from the character)

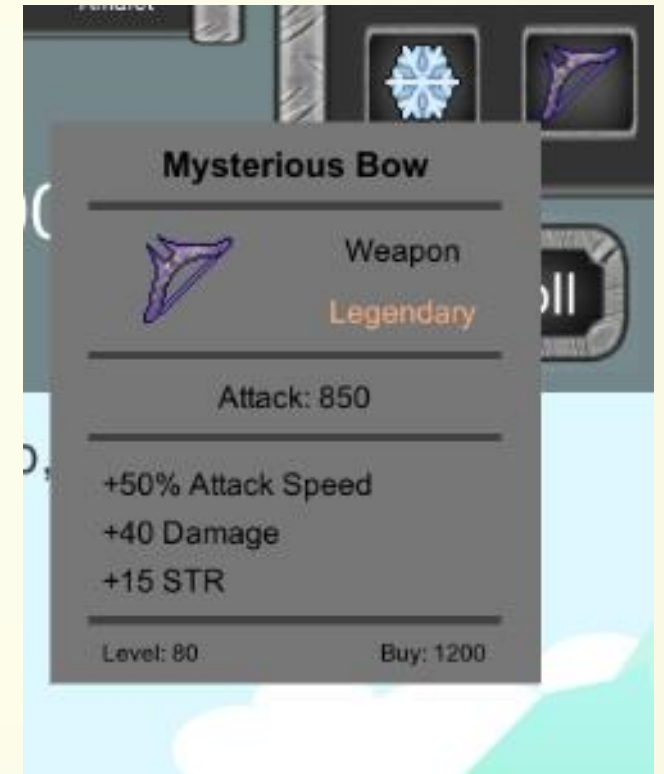
### 2- Variables

- EquipmentInventory is the dictionary of items and which slot has which item.

# Items Tooltip

---

- Items tooltip is the window which shows the item properties and information
- The window is available under ItemsTooltip GameObject, and you can change its size when you click on ItemHoverer GameObject which is a child of ItemsTooltip GameObject.





# DatabaseManager

---

- DatabaseManager script holds all the database of RPG Box.
- Available databases are:
  - Items
  - Loot Tables
  - Attributes
  - **PRO Only:** Enemies
- To call a database simply write: DatabaseManager. then the list you want. Example DatabaseManager.Attributes[0];

# Error Messages System

---

- ErrorMessageText script is responsible for Error messages which are shown in the middle of the screen.
- To show an error message write:  
`ErrorMessageText.instance.ShowMessage("Message here")`
- To change the properties of the text, check the GameObject MessageBoxText which is a parent of MustManagers GameObject

# Bars System

---

- To create a bar, simply create one of the prefabs in the Prefabs folder.
- Free version has only Horizontal Bars while Pro has Vertical, Radial and Horizontal.



- Also Pro Version has the ability to change the sprite and image of the bar.
- To change the size and number of the bar, simply Call the function `ChangeBarSize()` of the `BarManager` of the bar itself and pass the values

## PRO Only: Save and Load System

---

-Save and Load system allow the player to keep the data of Inventory, Shop and Equipment saved and not lost when the game is closed (even you as the programmer when you close Unity Editor).

-How To Save: Simply call, `SaveLoadManager.Save();`

-How To Load: Simply call, `SaveLoadManager.Load();`

-No code is needed to call the loaded Inventory, Shop and Equipment. They will be loaded immediately when you call `SaveLoadManager.Load();`

## PRO Only: Enemies: How to Create

---

- Enemy is a ScriptableObject which contains all the information of the Enemy.
- Right click on the asset folder and create “Enemy”.
- To let the DatabaseManager loads the Item, it needs to be under “Enemies” folder
- To add more tribes, check Enemy script.

## PRO Only: Enemies: Brief Introduction About Each Script

---

### ❖EnemyManager Script:

- This script is responsible for spawning enemies, the main method is SpawnAnEnemy().
- Enemies spawned will be children of EnemyParent.

### ❖EnemySelfer Script:

- This script is for processing the enemy information
- TakeDamage() method is responsible for taking damage.
- GenerateFloatingNumber() is called when the enemy takes damage to show it.
- DropLoot() is called when the enemy dies and drops loot.
- KillMe() Called when the enemy Health Points reach 0.

## PRO Only: Floating Numbers.

---

❖ To create a floating number system in your project, simply follow these steps:

1- Create an empty script and attach it to a GameObject, and write the following codes:

```
public GameObject TextFloating;|
public void GenerateFloatingNumber(string NumberAsString)//floating numbers generator.
{
    GameObject NewTextFloating = GameObject.Instantiate(TextFloating, transform);//choose the parent here
    NewTextFloating.transform.position = transform.position; // change the exact position where the numbers come from
    TextSelfFloating AccT = NewTextFloating.GetComponent<TextSelfFloating>();
    AccT.TakeInfo(NumberAsString); // this shows the number
}
```

2- In Prefabs folder, there is a prefab called “FloatingText”. Duplicate it and modify it if you want

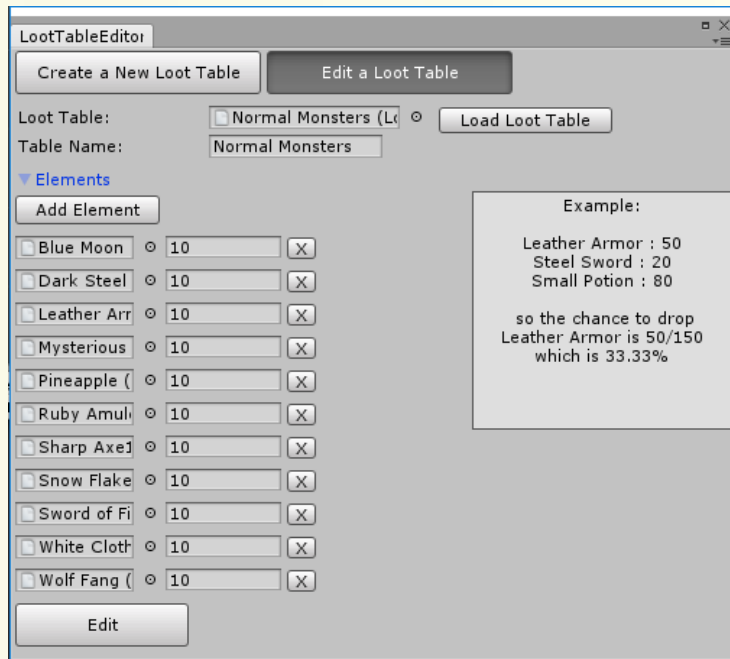
3- Now attach this new created prefab to the empty place in the GameObject you have created.

4- To call the floating number, now just call GenerateFloatingNumber() method and pass your number there

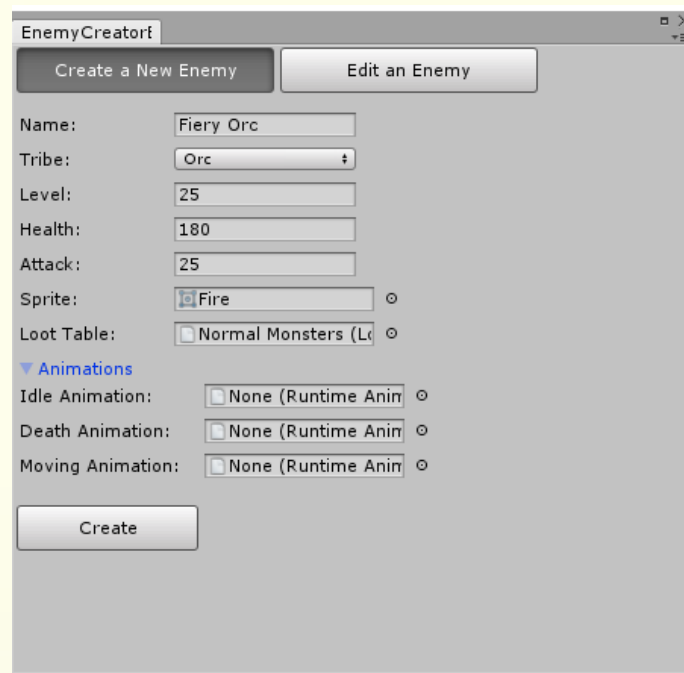
\*Enemy has the exact Floating system, so you can check it play with it.

# PRO Only: Items Creator. Loot Tables Creator. Enemies Creator.

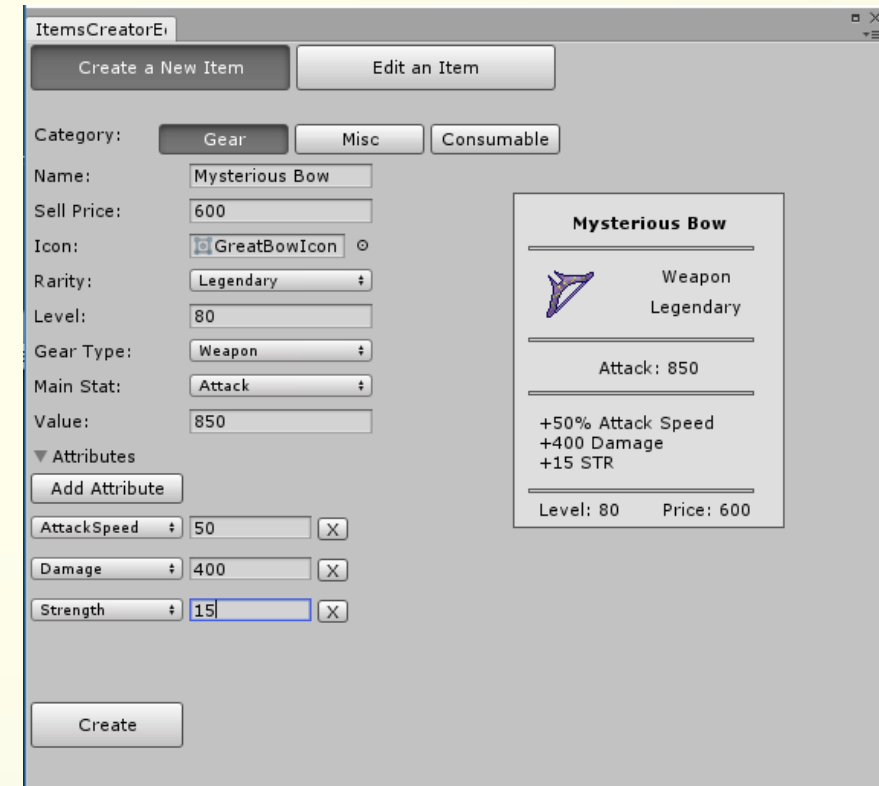
- ❖ To access these creators go to the upper menus in Unity and go to Tools, then RPG Box then click on the creator you want



Loot Tables Creator



Enemies Creator



Items Creator



# Contact Information

---

- If you have any question, do not hesitate to email: [yazun.shn@gmail.com](mailto:yazun.shn@gmail.com)
- Twitter: @Wurrad
- Please note that we require 1-2 business days for processing email support requests.