Scripting API – Inventory Master

Support: buchheim.sander@yahoo.de

Youtube tutorial series:

1. How to create inventory elements
2. Scripting examples(HP/Mana bar, random loot system, change inventory settings and so on)

Table of content

Item.class

Variables

string itemName	The name of the item.
int itemID	Every item has his own unique ID.
string itemDesc	The description for the item.
Sprite itemIcon	The sprite for the item.
GameObject itemModel	The GameObject of the item.
int itemValue	The itemValue is 1 at default.
Itemtype itemtype	The ItemType of the item.
int maxStack	A cap for the stack.
int rarity	The rarity of this item when it getting spawned/dropped

Functions

getCopy()

Description

This function returns an Item copy.

ItemDataBase.class

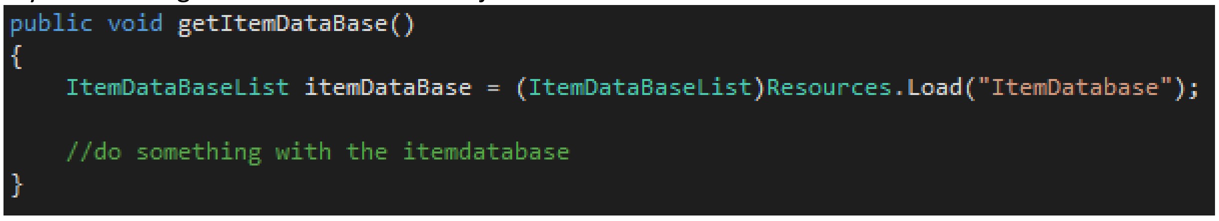
Variables

List<Item> itemList	A list with all the items in the itemdatabase.

Example

Description

If you want to get the itemdatabase list just do this:



```
public void getItemDataBase()
{
    ItemDataBaseList itemDataBase = (ItemDataBaseList)Resources.Load("ItemDatabase");

    //do something with the itemdatabase
}
```

Inventory.class

Description

Every storage, hotbar, craftsystem and charactersystem is using the inventory script.

Variables

bool mainInventory	Boolean, this defines if it is the main inventory of the player. If you got a hotbar you have to define one inventory as a mainInventory.
List<Item> ItemsInInventory	Items, stored in a list, which are currently in the inventory. Getting updated every time when you drag, drop and consume an item.
int height	The slot amount in vertical of the inventory.
int width	The slot amount in horizontal of the inventory.
bool stackable	When this Boolean is true, the number of the items will appear otherwise not.
int slotSize	The slotsize of each slot in px.
int iconSize	The size of the item icons.
int paddingBetweenX	padding between the slots in X-axis
int paddingBetweenY	padding between the slots in Y-axis
int paddingLeft	left padding from the slot to the border
int paddingRight	right padding from the slot to the border
int paddingBottom	bottom padding from the slot to the border
int paddingTop	top padding from the slot to the border
int positionNumberX	position of the number from the stackable items(X-axis)
int positionNumberY	position of the number from the stackable items(Y-axis)

Functions

adjustInventorySize()

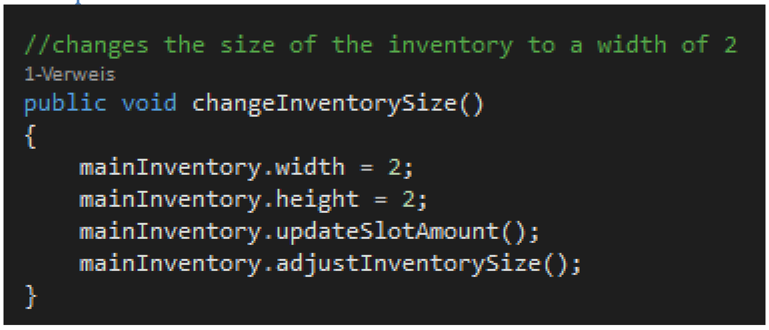
Description

This function adjusts the general size of the inventory.

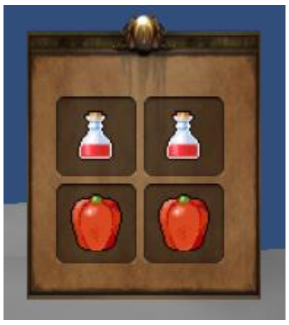
Used variables

- width
- height
- slotsize
- paddingbetweenX
- paddingbetweenY
- paddingLeft
- paddingRight
- paddingTop
- paddingBottom

Example



```
//changes the size of the inventory to a width of 2
1-Verweis
public void changeInventorySize()
{
    mainInventory.width = 2;
    mainInventory.height = 2;
    mainInventory.updateSlotAmount();
    mainInventory.adjustInventorySize();
}
```



updateSlotAmount()

Description

This function updates the size of the inventory in horizontal and vertical.

Used variables

- width
- height

Example

Check the example of function **adjustInventorySize()**.

updateSlotSize()

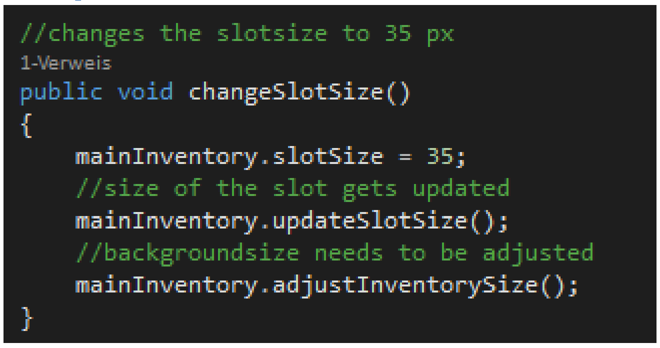
Description

This function updates the size of the slot.

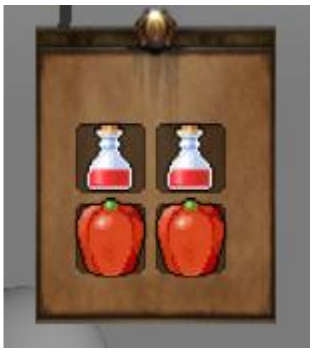
Used variables

- slotSize

Example



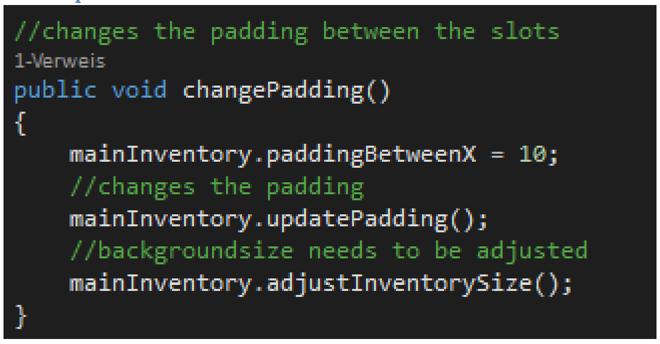
```
//changes the slotsize to 35 px
1-Verweis
public void changeSlotSize()
{
    mainInventory.slotSize = 35;
    //size of the slot gets updated
    mainInventory.updateSlotSize();
    //backgroundsize needs to be adjusted
    mainInventory.adjustInventorySize();
}
```



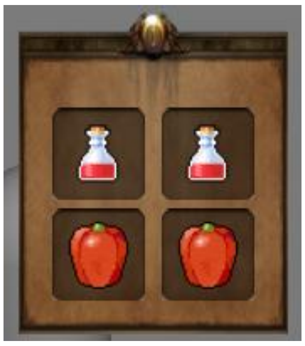
updatePadding()

Description
This function updates the padding of the inventory.

Used variables
- paddingBetweenX
- paddingBetweenY
- paddingRight
- paddingLeft
- paddingTop
- paddingBottom

Example


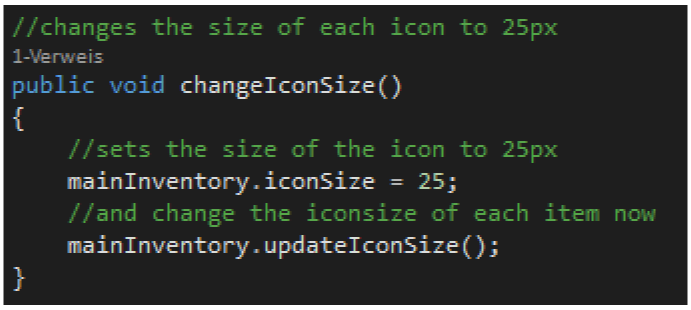
```
//changes the padding between the slots
1-Verweis
public void changePadding()
{
    mainInventory.paddingBetweenX = 10;
    //changes the padding
    mainInventory.updatePadding();
    //backgroundsize needs to be adjusted
    mainInventory.adjustInventorySize();
}
```



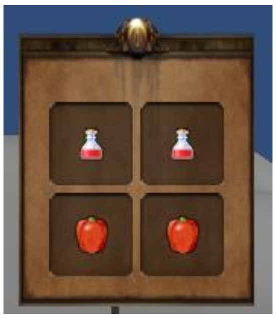
updateIconSize()

Description
This function changes the size of each item icon.

Used variables
- iconSize

Example


```
//changes the size of each icon to 25px
1-Verweis
public void changeIconSize()
{
    //sets the size of the icon to 25px
    mainInventory.iconSize = 25;
    //and change the iconsize of each item now
    mainInventory.updateIconSize();
}
```

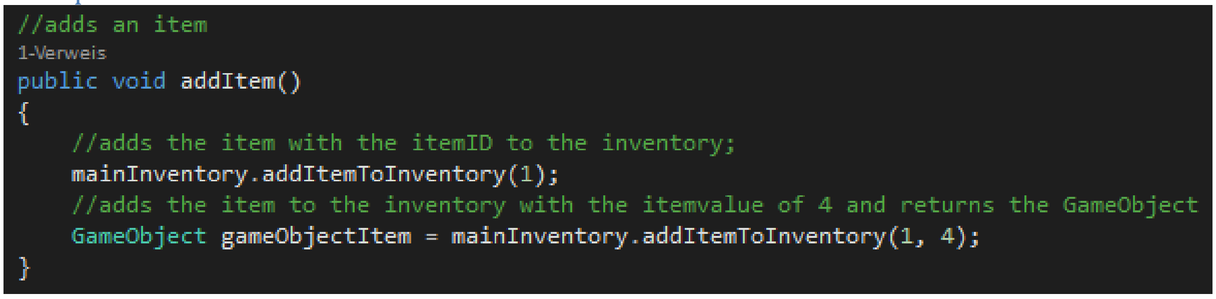


addItemToInventory(int ITEMID), GameObject addItemToInventory(int itemID, int itemVALUE)

Description

This function adds an item. This function needs an item-ID and possibly an item-value. If you do not use the function with a specific item-value, it will be 1 by default. addItemToInventory(int itemID, int itemValue) also returns the GameObject with the item.

Example



```
//adds an item
1-Verweis
public void addItem()
{
    //adds the item with the itemID to the inventory;
    mainInventory.addItemToInventory(1);
    //adds the item to the inventory with the itemvalue of 4 and returns the GameObject
    GameObject gameObjectItem = mainInventory.addItemToInventory(1, 4);
}
```

GameObject getItemGameObject (Item item)

Description

Returns the GameObject of the Item.

bool checkIfItemAllreadyExist(int itemID, int itemValue)

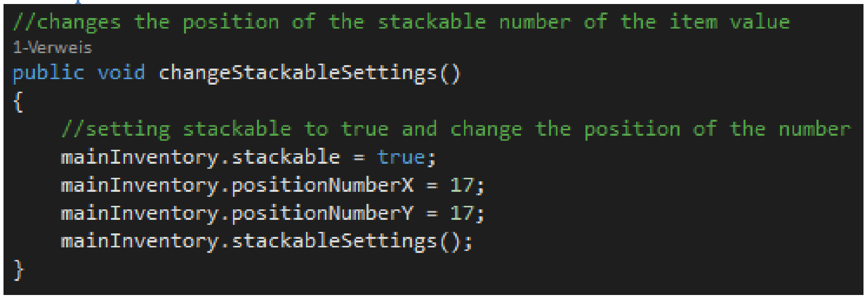
Description

This function returns a Boolean and also adds the item to a stack when one exists where it can be placed on. It returns true if it got stacked and false when there is no existing stack where you could place it on.

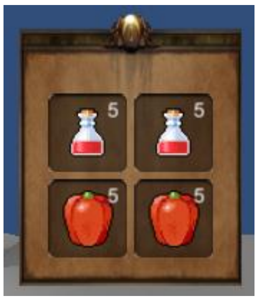
stackableSettings()

Description
This function changes the position of the number for the itemvalue.

Used variables
- positionNumberX
- positionNumberY
- stackable

Example


```
//changes the position of the stackable number of the item value
1-Verweis
public void changeStackableSettings()
{
    //setting stackable to true and change the position of the number
    mainInventory.stackable = true;
    mainInventory.positionNumberX = 17;
    mainInventory.positionNumberY = 17;
    mainInventory.stackableSettings();
}
```



deleteAllItems ()

Description
This function deletes all items in the inventory

deleteItem (Item item)

Description
This function deletes a specific item in the inventory.

sortItems(Item item)

Description
This function takes all items and places them in an order.

deleteItemFromInventoryWithGameObject (Item item)

Description
Deletes the item with the GameObject from the inventory.

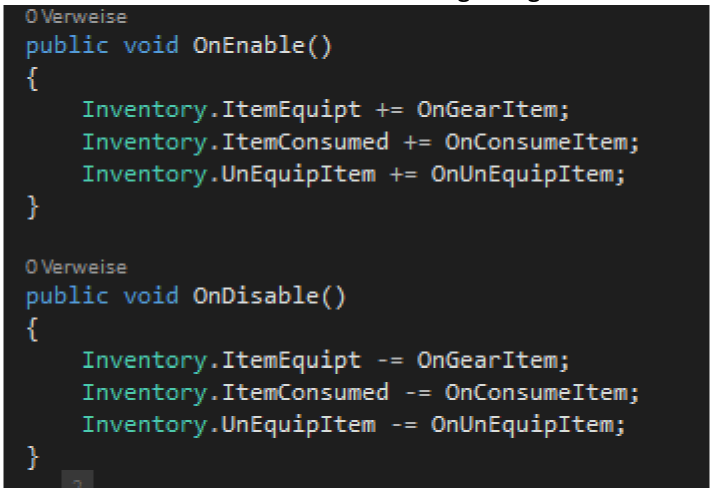
Events

Inventory	
Inventory.ItemEquipt	This event is getting called when you equip an item.
Inventory.ItemConsumed	This event is getting called when you consume an item
Inventory.UnEquipItem	This event is getting called when you unequip an item.
Inventory.InventoryOpened	This event is getting called when you open an inventory
Inventory.AllInventoriesClosed	This event is getting called when ALL inventories are closed

Example

First step:

First thing which you have to do, when you use the events, you have to add the functions which the event has to called when this event is getting called.

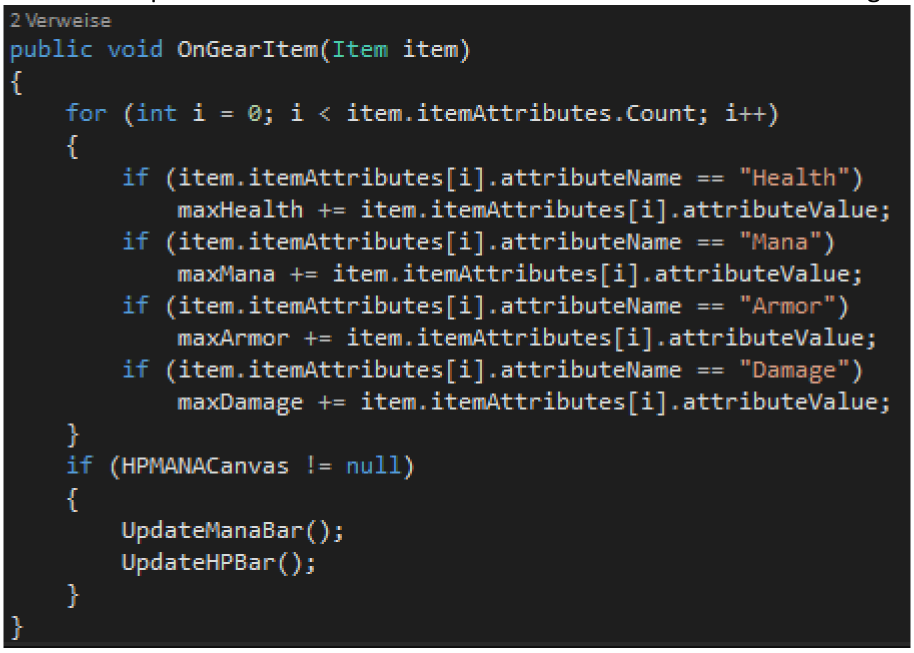


```
0 Verweise
public void OnEnable()
{
    Inventory.ItemEquipt += OnGearItem;
    Inventory.ItemConsumed += OnConsumeItem;
    Inventory.UnEquipItem += OnUnEquipItem;
}

0 Verweise
public void OnDisable()
{
    Inventory.ItemEquipt -= OnGearItem;
    Inventory.ItemConsumed -= OnConsumeItem;
    Inventory.UnEquipItem -= OnUnEquipItem;
}
```

Second step:

The next step is to define what the function has to do when the event is getting called.



```
2 Verweise
public void OnGearItem(Item item)
{
    for (int i = 0; i < item.itemAttributes.Count; i++)
    {
        if (item.itemAttributes[i].attributeName == "Health")
            maxHealth += item.itemAttributes[i].attributeValue;
        if (item.itemAttributes[i].attributeName == "Mana")
            maxMana += item.itemAttributes[i].attributeValue;
        if (item.itemAttributes[i].attributeName == "Armor")
            maxArmor += item.itemAttributes[i].attributeValue;
        if (item.itemAttributes[i].attributeName == "Damage")
            maxDamage += item.itemAttributes[i].attributeValue;
    }
    if (HPMANACanvas != null)
    {
        UpdateManaBar();
        UpdateHPBar();
    }
}
```

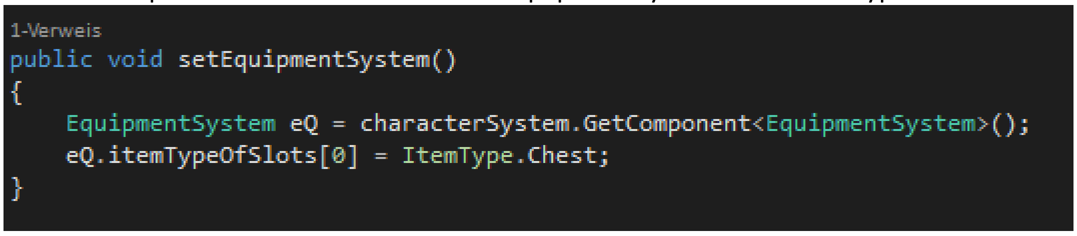
EquipmentSystem.class

Variables

int slotsInTotal	The amount of slots in the EquipmentSystem.
Itemtype[] itemTypeOfSlots	An array which stores the ItemTypes for each slot.

Example

In this example we set the first slot of the EquipmentSystem to an ItemType of chest.



```
1-Verweis
public void setEquipmentSystem()
{
    EquipmentSystem eQ = characterSystem.GetComponent<EquipmentSystem>();
    eQ.itemTypeOfSlots[0] = ItemType.Chest;
}
```

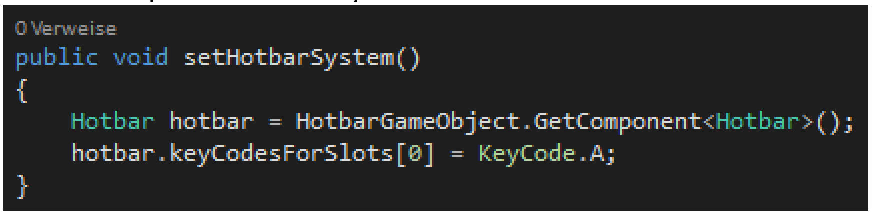
Hotbar.class

Variables

int slotsInTotal	The amount of slots in the Hotbar.
Itemtype[] keyCodesForSlots	An array which stores the keycodes for each slot.

Example

In this example we set the keycode of the first slot to “a”.



```
0 Verweise
public void setHotbarSystem()
{
    Hotbar hotbar = HotbarGameObject.GetComponent<Hotbar>();
    hotbar.keyCodesForSlots[0] = KeyCode.A;
}
```