

Customizer

Lovatto Studio

1.3

Content:

Tutorial

- Introduction
- Getting started
 - 1.1.1. How add a new object for customize
 - 1.1.2. How add a new object in play scene
- Example

Class list

- bl_Cmanager class reference
- bl_Customizer
- bl_ReadCustom

Character 1

1.1 Introduction

Customizer is a system personalization of objects, intuitive, modern, stylish and functional, that lets you give a variety of ways to suit the player.

- Modern
- Elegant
- functional

1.2 Getting started.

1.1. How add a new object for customization

- In scene "Customizer", drag the prefabs "Marke" in the example folder and place as the son of "CustomizerManager".
- then remove the Marker prefabs son's son "MarkerModel", and place the new model (make sure it is in the same position the "MarkerModel").
- then if you want to replace the "attachments" (Barrel, Optic, etc ..) do the same: remove these and place new ones.
- once replaced everything you want, change the name for the new Prefabs name (also in the variable "GO_Name" (make sure that these two have the same name – GameObject and Go_Name).
- now bl_Customizer.cs list called "Customizer", add new attachments depending on their type, make sure that in each type, each attachment has a different ID
- on bl_Customizer.cs in var called "target", place CustomizerManager from scene.

1.2 How add a new object in play scene

- in the scene where the player takes the "object" that is personalized, if not already set the same object / weapon in scena of customization place this.
- add the script bl_ReadCustom.cs
- in the variable "Go_Name" set the name of the object / weapon (must be the same as the name of the scena "Customizer").
- on the list of bl_ReadCustom.cs called Customizer, add attachments in the same order as the "Customizer" scena (each attachment must have the same id in the scena Customizer).

1.3 example

in scena "Customizer" We can see, as the customization system works with a example of weapon called "Marker" and some attachments example.

When you click the Button "Separate" the ajuntos present a distance to stay away, to give an effect of separation, when mouse over one of these attachments, a button, which if clicked, it will show us show list attachments available to join the object / weapon.

when we again click the button will be the effect of assembly, and can see a preview of how the object / weapon when customizing this list, we give click on the "Save" button to save our configuration will look like.

then to return to the scene of the game, we click on the return button if the configuration has been saved, the scena is caragara, if not a prevention message

in scena "ReadExample" an example of how personalized weapon before making new attachments are displayed.

PRO TIPS:

GET POSITION OF ATTACHMENT IN INSPECTOR:

Fuction:

Get Auto the default position of each attachment (normalpositionbarrel,etc...)

-Go to the Component Menu of bl_Customizer and select the option "Get Default Position".

CALCULATE THE CUSTOM POSITION IN INSPECTOR

Function:

Calculate the position where attachments when it is positioned in edit mode.

-Go to the Component Menu of bl_Customizer and select the option "Calculate Custom Position".

Class List

bl_Cmanager class reference:

```
public Transform m_Manager;
```

the transform that contains all children objects / weapons

```
public GUI Skin Skin;
```

the skin for UI

```
public string Level ToLoad;
```

the scene to which one you see return completed customization.

```
public bool Have_Name = false;
```

is equivalent to whether the PlayerPrefs contains a Key Value.

```
public List<bl_Customizer> AllCustom = new List<bl_Customizer>();
```

contains all objects / weapons that can be customized

bl_Customizer

```
public Transform target;
```

the root parent object of this object (usually the same as the bl_Cmanager.cs).

```
public string Go_Name;
```

the name of this object / weapon has to be the same as the name of the GameObject

```
public float speed;
```

the speed at which the object will rotate

```
public float AutoRotSpeed = -60F;
```

the speed at which the object will Auto-rotate

```
public AudioClip SeperateEffect;
```

the sound effect when the gun is assembled

```
public List<Button> Button;
```

contains the reference to the sub-class that contains buttons weapon.

```
public List<Customizer> Customizer;
```

contains the reference to the sub-class that contains Attachments.

```
public List<ObjVector> ObjVector;
```

contains the reference to the sub-class that contains transform root of attachments

bl_ReadCustom

```
public string Go_Name;
```

the name of this object / gun, must be the same as that of the scena customizer

```
public List<Customizer> Customizer;
```

contains the list of the sub-class customizer, which contains all attachments, each attachment for each type must have the same ID as the scena "Customizer".