-ZERIN LABS-

Character set

Modular fantasy RPG characters

Welcome!
...and thanks for buying this outstanding modular characters pack :)

On this small tutorial you will find all the necessary details to configure the characters and prepare your project for those amazing video-game graphics.

For any doubt feel free to contact us at: zerinlabs@gmail.com

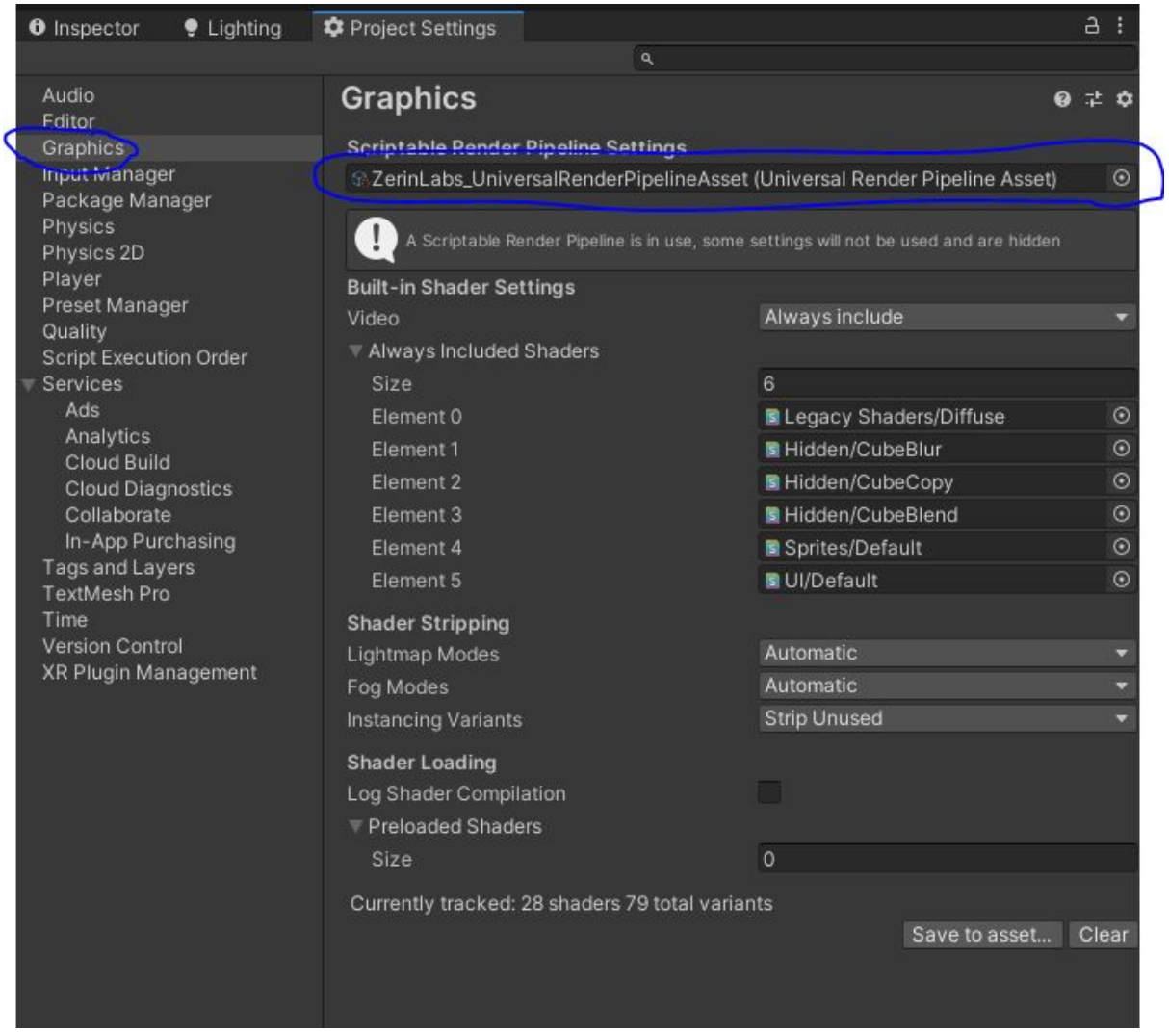
HOW TO USE THE CHARACTERS INSIDE UNITY

IMPORTANT: In order to display your characters properly it is important for you to **install and configure URP or use the LWRP/HDRP** as your main render pipeline.

It is recommended you install **Shader Graph** in case you would like to tweak or modify the included “sh_characterMultiColoring” shader (but, in theory, it is actually not necessary). However, if for some reason the shader doesn’t work on your project try to install ShaderGraph before messing the things up ;)

Remember that it is required to change the “default” render pipeline

To do this, add the custom URP render pipeline file on the unity **project_settings/graphics section:**



Alternatively you could use the default unity render pipeline, but in that case you will not be able to use the include special “sh_characterMultiColoring” shader

However, in case you want to configure your own standard materials, we're including all the necessary textures.

IMPORTANT: We strongly recommend to update your project to the newer and more reliable URP system not only because the old standard pipeline will be eventually (soon) get deprecated but because this way you will be able to use ALL the functionalities included with this outstanding package

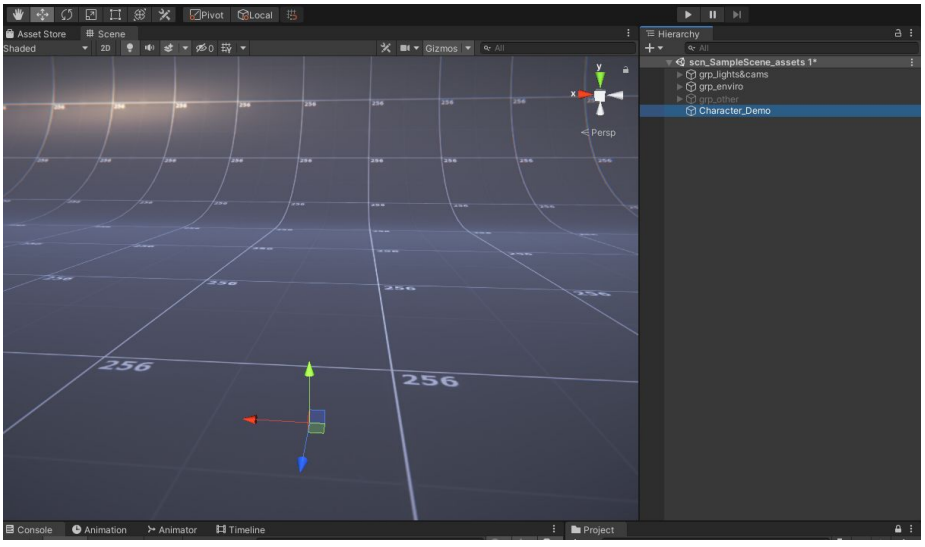
Configuring a "modular character"

The characters are meant to be set on "costume layers". This means that you can "dress" a single character with several pieces coming from different "costumes". This may sound a bit confusing now but it is actually quite simple to configure.

BTW, if you're concerned about the performance of this technique, we can tell you that after several tests we can confirm that if you configure (and hide) properly all character pieces there is no extra performance impact beyond the basic "skinned mesh" cost

Step 1: Create an empty game object

This game object will work as a container for all the character costumes and pieces. In this case we're calling it "Character_Demo"

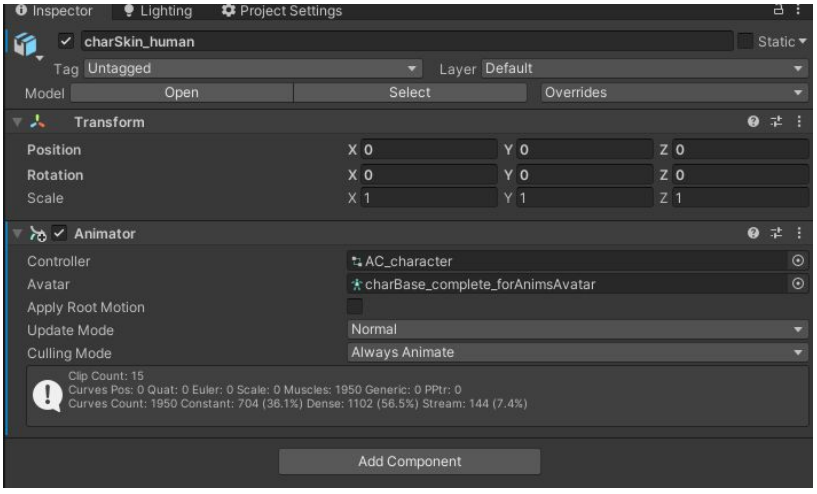


Easy peasy, right? ;)

Let's move then to the following step

Step 2: Add (drag n drop) a “charSkin” mesh inside the recently created game object

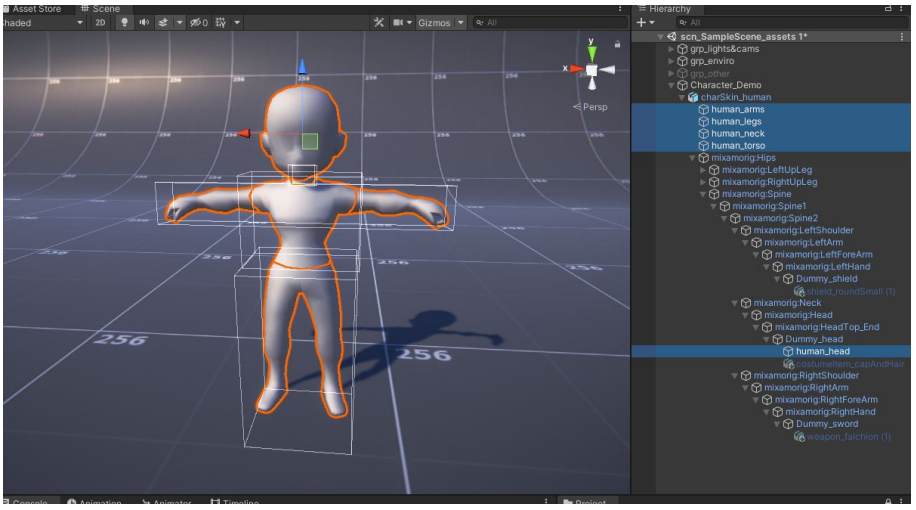
Place it on the [0,0,0] position inside the empty game object so it shares the position with it. At same time, we highly recommend you add now the desired animation controller you would like to use with the character.



IMPORTANT: the same animation controller should be shared among all the “character costumes” or “skins” of the same character

Now it would be a good moment to add the materials to the meshes.

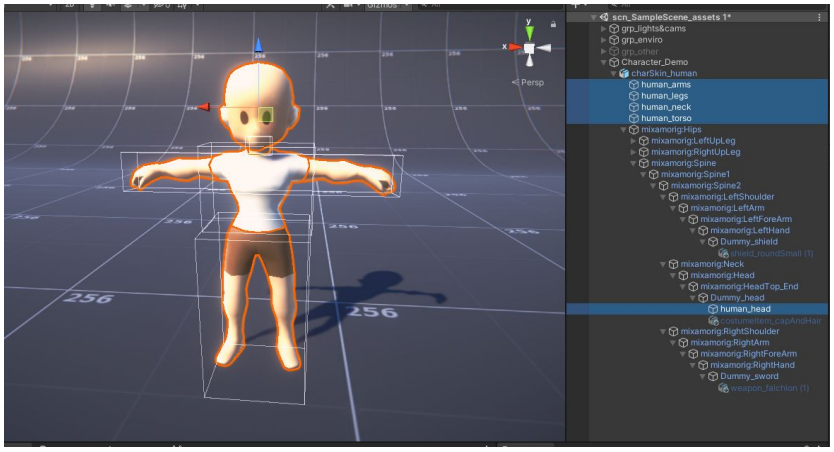
In order to do this take in account that: for the **charSkins** and the costumes, all the “mesh objects” are stored inside the “skinned mesh” like is shown on the following image



On the included material folder, remember that all the character materials are named according to the meshes they belong to.

For instance, here we will be using the “**Mat_CHAR_human**” material that belongs to the “**charSkin_human**” mesh

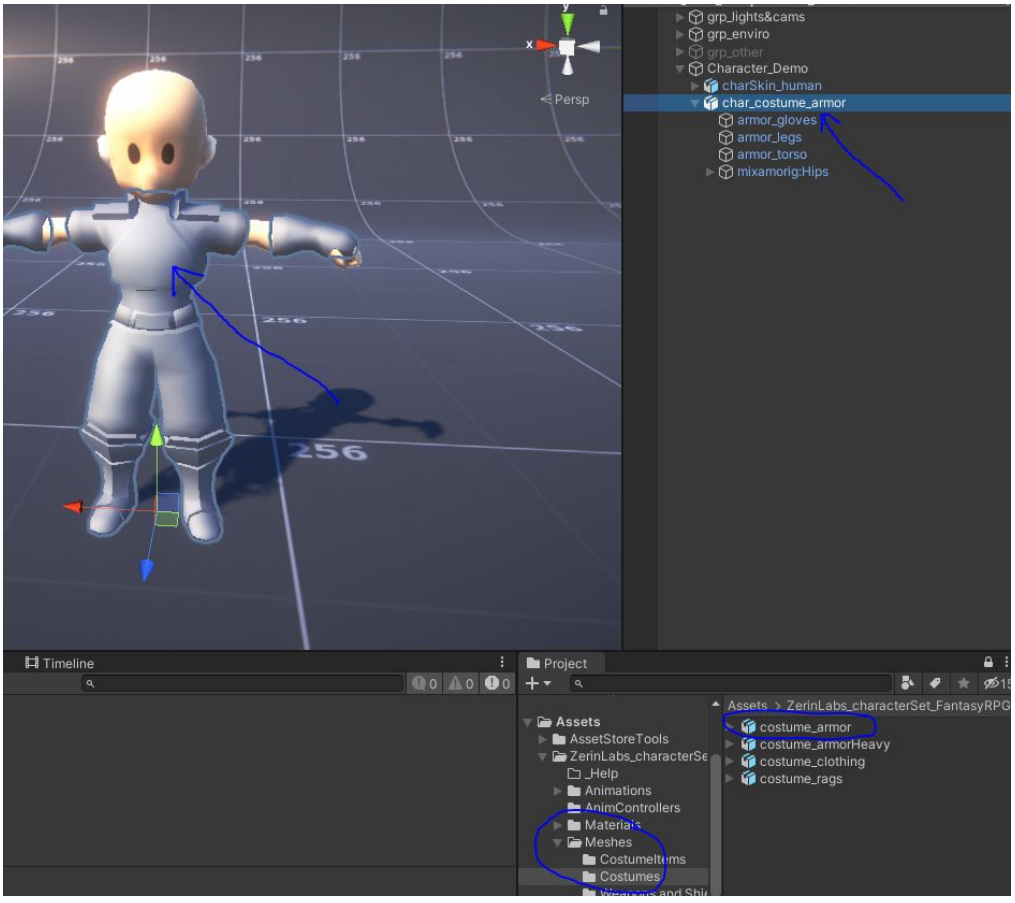
IMPORTANT: The textures are atlased so you should use the same material for the entire character



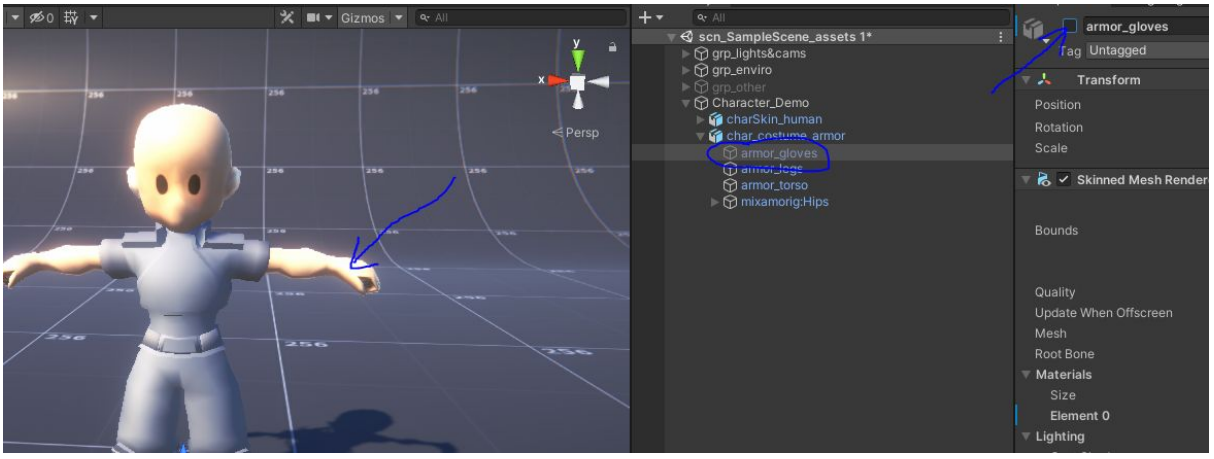
Step 3: adding the desired costume layers

Now it's time to dress up our character so, from the costumes folder add the costumes you would like to sue for your character

In this case we're using the "**char_costume_armor**" costume



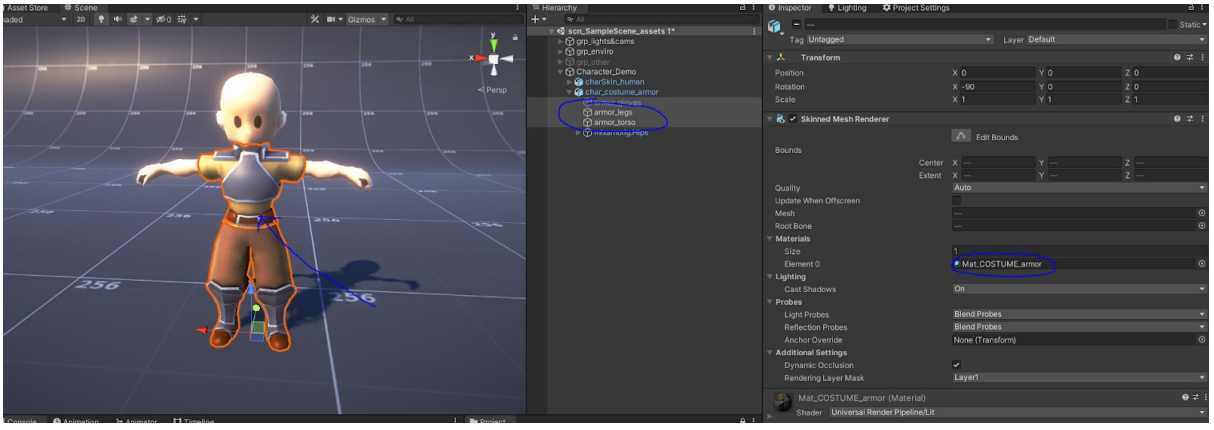
If you like you can disable some mesh elements from the costume, for instance here we will be disabling the “**armor_gloves**” due we don’t want them for character setup



Sometimes it could be recommended to “disable” some of the charSkin parts in order to avoid overlappings with other meshes and optimize the overall character.
however, as a rule of thumb, always consider “disabling” the meshes that are not visible on your character configuration

IMPORTANT: Alternatively you could include **more than one costume on the same character** in case you would like to include some pieces from other “costumes”.
For instance, you could be using the pants from an specific costume and the torso or gloves from another one.
This feature makes this overall technique and pack VERY versatile and powerful. (But remember to disable always all those pieces that are not visible)

On the other side, just like we did with the “charSkin” mesh, you can apply here the materials too in the same way…



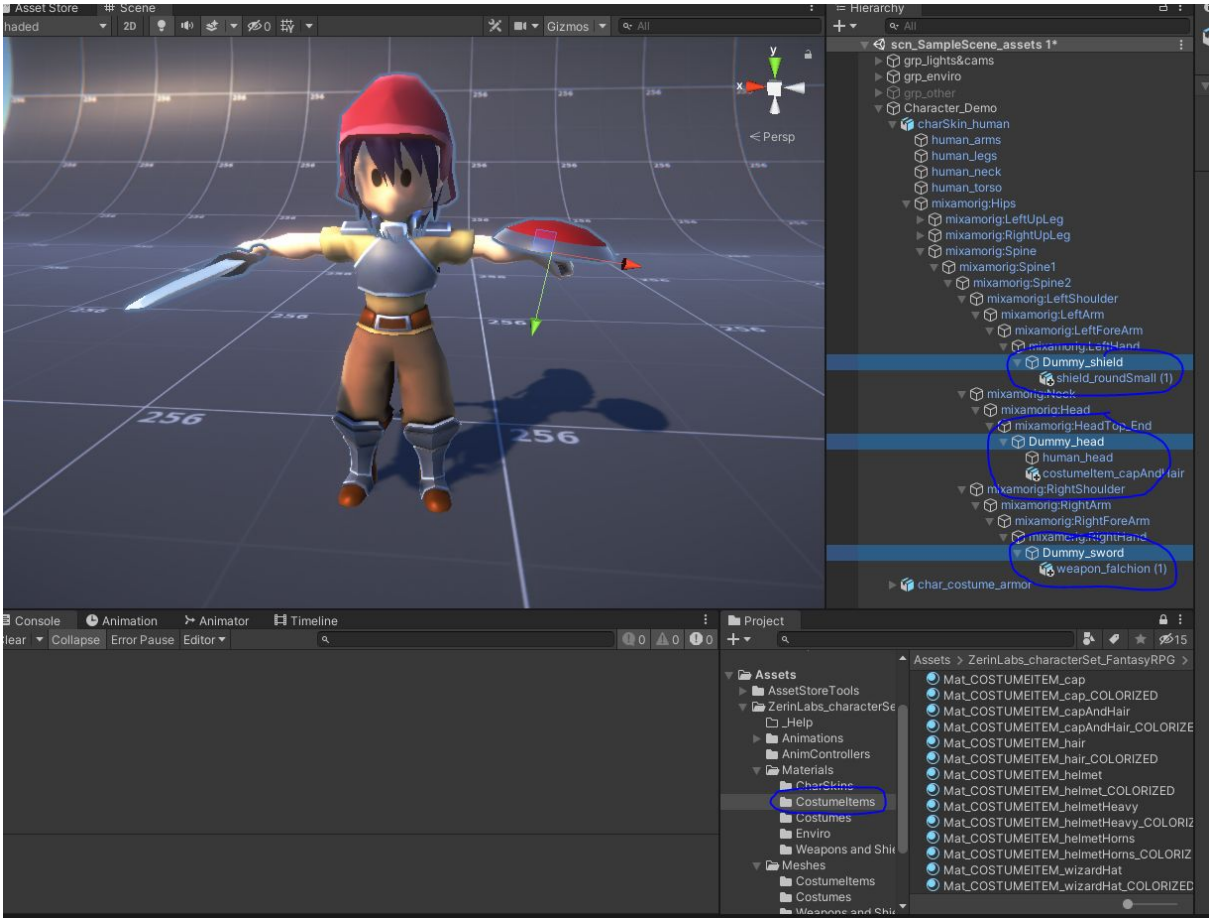
Step 4: Adding weapons and other items

Included on this pack you have mainly 3 kinds of items you can add to your characters:

- Weapons
- Shields
- Costume items (hats wigs, helmets, etc.)

Adding any of those to your character is extremely easy.
In order to do that you simply open the charSkin object and navigate it to find the proper dummies we will use to “link” our character items

Then, drag and drop the desired item to the proper dummy so the dummy becomes the “parent” of the item aso shown on the following image:



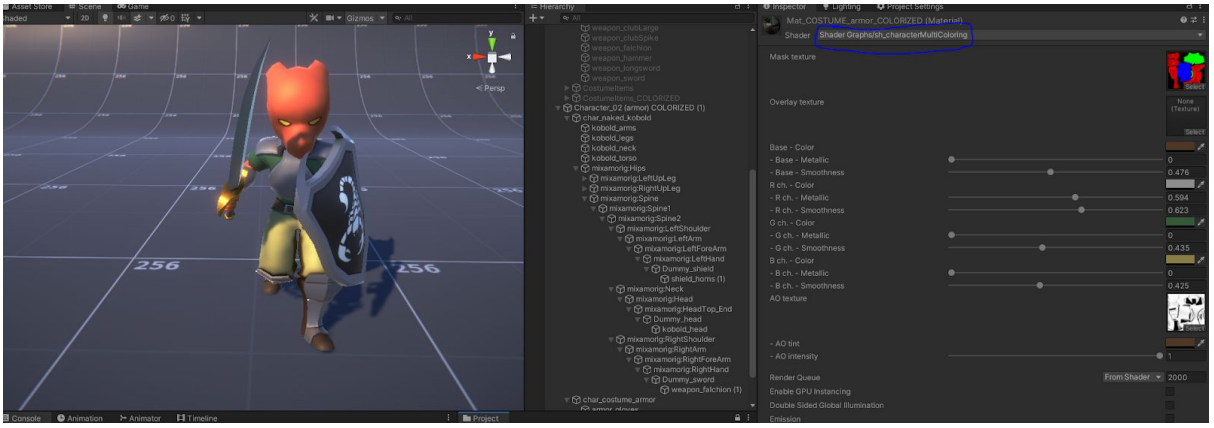
IMPORTANT: The dummys are already properly placed and oriented so the objects will themselves become placed and aligned properly with the character hands and head. However, on some of the items it may be necessary a simple rotation of 90º on some of the axis.

Using the “sh_characterMultiColoring” shader

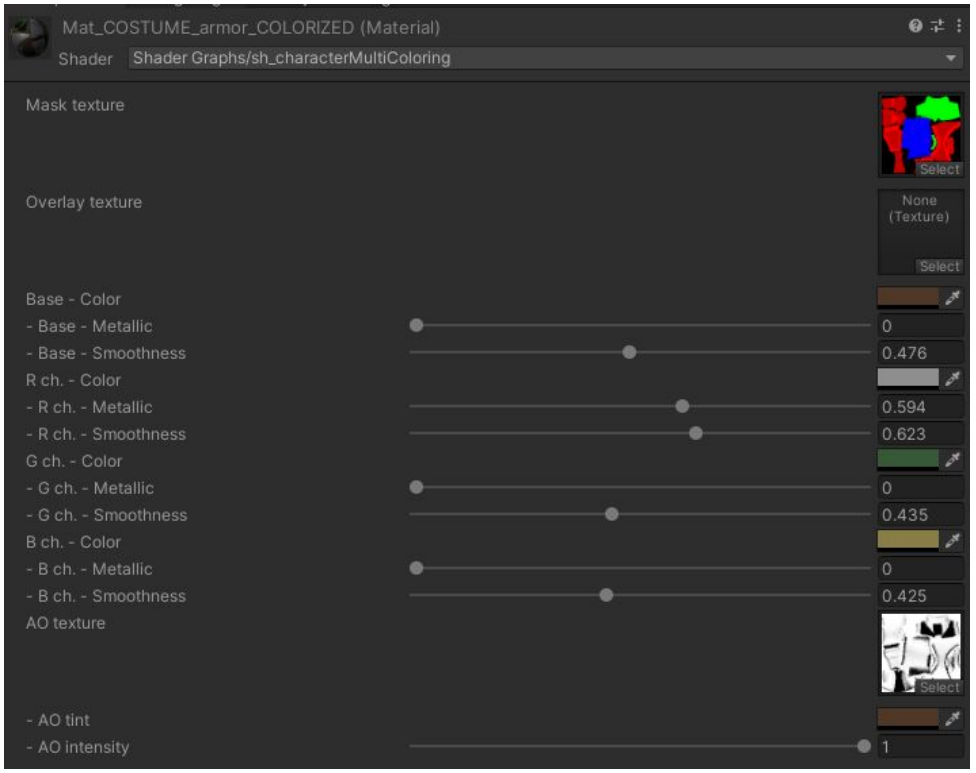
The “sh_characterMultiColoring” is a specially designed shader meant to make the overall process of character coloring creation and iteration easy and flexible so you can obtain great results without messing much around with the textures.

The technique of the shader is based on a special “mask texture” that encodes different costume and body parts so the can be colored and shaded independently.

The shader includes an “ambient occlusion” slot for AO shading and a special “overlay slot” for decals textures and other stuff you may like to the character texture just over the color layer



These are the shader properties and their description.



Properties

- **Mask texture**
  - It's the texture that encodes each one of the mesh parts in order to be colored independently.
  - TIP: assign this texture as an albedo texture to identify the different parts of the model and how they can be colored
- **Overlay texture**
  - This texture it's optional and may contain some extra details that you want to maintain "un-tinted" over the whole colored material.
  - This is specially useful for instance to add the faces of the characters (mouth eyes, etc) that will appear over the "skin color", the "coat of arms" that may appear on the shields and in order to add some extra detailed "texture" to the fabric or chain-mail like on the included example texture.
- **Base - Color (black/default channel color)**
  - This is the color that will appear on all the "black" sections of the "mask texture"
- **Base - Metallic**
  - The usual PBR metallic component. Works the same way as the standard metallic component but dedicated and masked to the "base" channel
- **R ch. - Color (red channel color)**
  - This is the color that will appear on all the "red" sections of the "mask texture"
- **R ch. - Metallic**
  - The usual PBR metallic component. Works the same way as the standard metallic component but dedicated and masked to the "red" channel
- **G ch. - Color (green channel color)**
  - This is the color that will appear on all the "green" sections of the "mask texture"
- **G ch. - Metallic**
  - The usual PBR metallic component. Works the same way as the standard metallic component but dedicated and masked to the "green" channel
- **B ch. - Color (blue channel color)**
  - This is the color that will appear on all the "green" sections of the "mask texture"
- **B ch. - Metallic**
  - The usual PBR metallic component. Works the same way as the standard metallic component but dedicated and masked to the "blue" channel
- **AO texture (ambient occlusion)**
  - This B/W texture will be multiplied over the model for AO shading purposes.
- **AO tint**
  - Color that will be applied to dark areas of the AO texture
- **AO intensity**
  - VAlue that modulates the opacity of the AO affectation

CONTACT

- Mail
  - zerinlabs@gmail.com
- Twitter
  - @zerinlabs
- Site
  - https://zerinlabs.blogspot.com/
- Youtube channel
  - https://www.youtube.com/channel/UC-u0QyXSJUS60hAfc-UnF-A