# Basic Documentation for Modular Anime Character Package

### Models and customization

#### Method 1

The simplest way to get a custom character ready is to use the "F\_FullBody AllEquipments" or "M\_FullBody AllEquipments" prefabs. These prefab contains all the customizable pieces inside them, so you can just disable or remove the ones you dont want from the prefab.

#### Method 2

You can also set the desired mesh and material to the SkinnedMeshRenderer manually. This method is how the "F\_FullBody AllEquipments" prefabs were created.

Just duplicate one of the body parts and change the Mesh and Materials in the SkinnedMeshRenderer component.

Duplicating an existing body part is important to make sure both of them are using the armature and bone hierarchy. Else the bone hierarchy will need to be initialized through code.

	⊕ F_body_hips 分 F_body_legs_knee	🔻 🗞 🗹 Skinned Mesh Rendere	er	07‡ :
		Bounds	Pb Edit Bounds	
	F_body_torso	Cer	nter X 0 Y 0	Z 0.7999999
• )		Exte	ent X 0.5 Y 0.4999999	Z 0.8085654
		Quality	Auto	
	Ga F_Head3	Update When Offscreen		
	Ga F_Head4	Mesh	F_outfit.003	
	G Outfit1	Root Bone	ARL_BoneRoot (Transform)	0
	🙀 Outfit2	▼ Materials	KL_BONEROOT (Transform)	
	G Outfit3			
	ିନ୍ତ Top1 ନ୍ୟୁ Top2	Element 0	mat_outfit.003	
	Ga Top3			+ -
	Ga Top4	▼ Lighting		
	G Top5	Cast Shadows	On	
	<b>б</b> аў Тор6			
	Controp7-A	▼ Probes		
	ନ୍ତି Top7-B ନ୍ତ୍ରି Bot1		Blend Probes	
	Ga Bot2	Light Probes		
	G Bot3		Simple	v 
	🗟 Bot4	Anchor Override	None (Transform)	
	କ୍ଟି Bot5 କ୍ଟି Bot6	In Deferred Shading, all of	bjects receive shadows and get per-pixel reflecti	
	Shoes1	Additional Settings		
	G Shoes2	Skinned Motion Vectors	~	
	ନ୍ତି Shoes3 ନ୍ତ୍ରି Shoes4	Dynamic Occlusion	~	
	Ga Shoes5	-		
	G Shoes6 G Shoes7	Mat_outfit.003 (Mater		07‡ ÷
		Shader Shader Graphs	/Rukha93/ShaderGraph_CharacterLit	▼ Edit

#### Method 3

The last method is the same as the second but done through script. You instantiate a new GameObject, add a SkinnedMeshRenderer component, set the new mesh and materials and copy the existing SkinnedMeshRenderer propertiers.

This method is used for the Customization Sample and can be seen in the CustomizationDemo.cs:

174	//instantiate new meshes, init properties, par	ent to character
175	GameObject go = null;	
176	SkinnedMeshRenderer skinnedMesh = null;	
177	foreach(var mesh in item.meshes)	
178		
179	//instantiate new gameobject	
180	<pre>go = new GameObject(mesh.name);</pre>	
181	<pre>go.transform.SetParent(m_Character.transfo</pre>	rm, false);
182	<pre>m_Equiped[cat].instantiatedObjects.Add(go)</pre>	;
183		
184	//add the renderer	
185	<pre>skinnedMesh = go.AddComponent<skinnedmeshr< pre=""></skinnedmeshr<></pre>	enderer>();
186	<pre>skinnedMesh.rootBone = m_ReferenceMesh.roo</pre>	tBone;
187	<pre>skinnedMesh.bones = m_ReferenceMesh.bones;</pre>	
188	<pre>skinnedMesh.bounds = m_ReferenceMesh.bound</pre>	s;
189	<pre>skinnedMesh.sharedMesh = mesh.sharedMesh;</pre>	
190	<pre>skinnedMesh.sharedMaterials = mesh.sharedM</pre>	aterials;
191	}	
102		

#### Hair

The hair prefab must be added as a child of the head bone (CC\_Base\_Head).



## Shaders

#### Overview

The main shader, "ShaderGraph\_CharacterLit", uses a Mask texture where each channel defines an area of the model that can be colorized:



The ShaderGraph implementation for the color customization is in the subgraph file "SubGraph\_ColorCustomization".

It also has a customizable rim light, implemented in the subgraph "SubGraph\_Rimlight".

#### Variations

Besides the standard Lit version, there are 3 extra toon shaders. Though they are unlit shaders, they support one directional light to control the shadows. The implementation is found in the "SubGraph\_ToonShading" subgraph.

The "SubGraph\_ToonShading" subgraph uses a Custom Function to get the direction and color of the main directional light. The implementation of this custom function may need to be changed in future versions of unity. If you need help to update this, feel free to send me an email.

Basic (ShaderGraph\_ToonBasic)

The Basic version is meant only for the simple areas with one single color, like the eyebrows.

Rim Light (ShaderGraph\_CharacterToon)

This is version used the mask texture used for color customization. It also adds a customizable Rim Ligh to make the model stand out.

Rim Light + Fake Outline (ShaderGraph\_CharacterToon+FakeOutline)

This version simulates an outline using a dark rimlight.