



**TECHNICAL  
DOCUMENT**



# Index

<b>Description</b>	<b>3</b>
<b>Elements</b>	<b>3</b>
<b>Garments</b>	<b>3</b>
<b>Morpher characteristics</b>	<b>5</b>
<b>Eyes &amp; Skins colors</b>	<b>6</b>
<b>Animation</b>	<b>7</b>
<b>Unity Humanoid system and animation examples</b>	<b>7</b>
<b>Morpher expressions</b>	<b>7</b>
<b>System</b>	<b>8</b>
<b>Preconfigured prefabs</b>	<b>9</b>
<b>Runtime</b>	<b>11</b>
<b>Extend the package with my own elements</b>	<b>11</b>
<b>Skinned Meshes</b>	<b>11</b>
<b>Attaches normal meshes</b>	<b>15</b>
<b>Use the system with my own base model and not use the KEKOS one</b>	<b>17</b>
<b>Contact</b>	<b>19</b>

## Description

KEKOS is a customization tool focused on a fast iteration to create characters and save them as prefabs to use in your games and applications.

KEKOS characters are made of several different customization elements.

We provide several pre-customized character prefabs.

The package is not designed for runtime purposes but we have added an Example scene showing that functionality.

## Elements

We can split the KEKOS elements into three big groups:

### Garments

The garments have two subcategories. One kind of element with Skinned Mesh and others without it and directly attached to the head bone.

The garments follow a naming convention to work in the editor properly.

Garment name	Example image	Prefix_Subprefix_	Skinned / Attached
<b>Torso</b>		TOR_01_	Skinned
<b>Legs</b>		LEG_01_	Skinned

<b>Feet</b>		FEE_01_	Skinned
<b>Hands</b>		HAN_01_	Skinned
<b>Full Torso</b>		FTOR_01_	Skinned
<b>Backpack</b>		TORP_01_	Skinned
<b>Hair</b>		HAI_60_ <i>* Normal hair</i> HAI_61_ <i>* Sliced hair</i>	Attached
<b>Cap</b>		CAP_51_ <i>* With sliced hair</i> CAP_52_ <i>* With no hair</i>	Attached

<b>Glasses</b>		FACP_30_	Attached
<b>Eyebrows</b>		FACP_31_	Skinned

\* We use the XXX\_00\_ sub-prefix to indicate for us the default element.

### Morpher characteristics

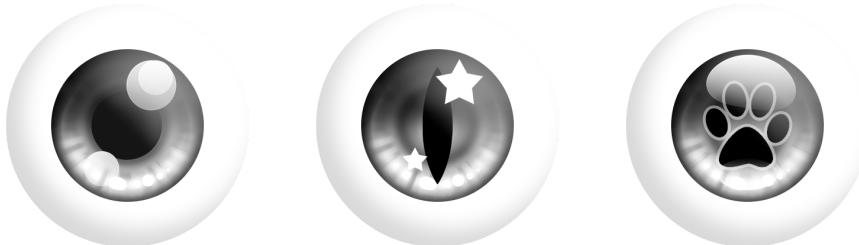
The head Skinned Mesh has several morphers (blendshapes) characteristics with the FACC\_ prefix on it.

You can use these morphers to make each character's face more unique with different shapes and sizes.

You can change the: nose, eye size and position, jaw shape, ears shape and size, and lips thickness.

## Eyes & Skins colors

You can customize the eye pupil shape and highlight.



Also, you can choose between several skin colors and combine them with some skin details.

Colors example:



Details example:



## Animation

KEKOS has two ways of adding expressivity to the characters. On one hand with body animations through the Unity humanoid system (remapping system for different rigs), and on the other hand with different morphers for a richer facial expression.

### Unity Humanoid system and animation examples

KEKOS use the Unity humanoid avatar, a system that has special features to work with humanoid characters. Due to the similarity in bone structure between different humanoid characters, it is possible to map animations from one humanoid character to another.

You can use almost any animation in the Unity Asset Store and animations with other skeletons (CAT, Biped, Blender armature, etc.) through this system.

You can find more information at this Unity documentation link: [Importing a model with humanoid animations](#).

Also, we provide some animations for testing purposes.

### Morpher expressions

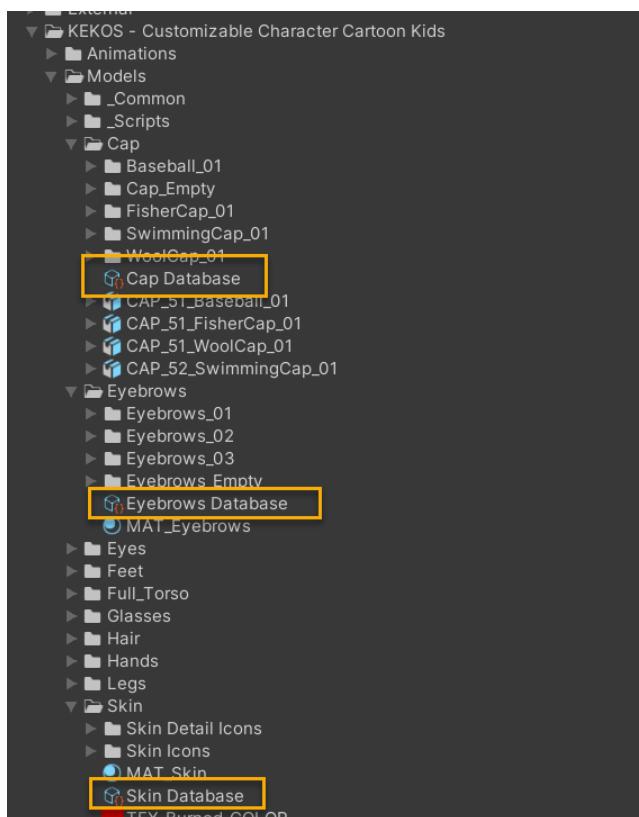
These blendshapes are identified in the head and eyebrows Skinned Meshes with the prefix **FACE\_**. There are 27 morphers in total for the control of the eyes, eyebrows, eyelids, mouth, jaw, cheeks, and tongue.

We provide some different setup examples like the following:



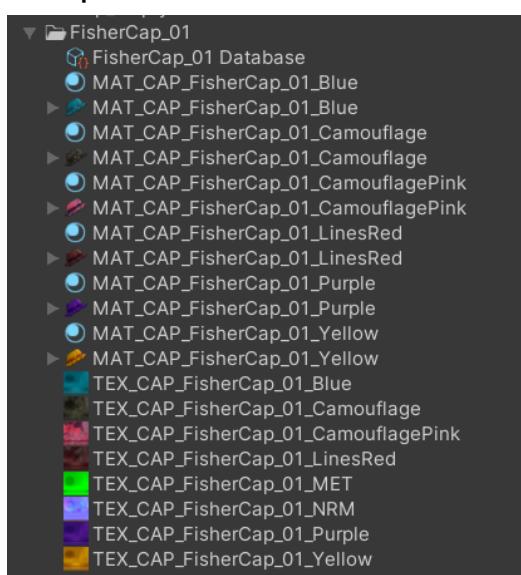
## System

Each kind of item has a database. These databases store the element's configuration such as sprite icon, element colors, materials, etc.



Also, there is a database for example facial expressions.

The sprite icons, materials, and textures follow a naming convention.

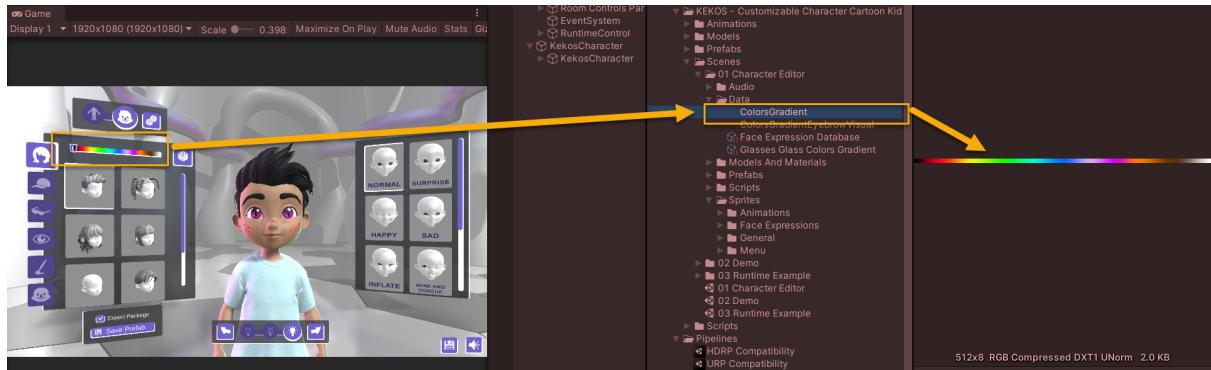


The material and sprite icon share the same name in the following format:

**MAT\_PREFIX\_Name\_XX\_Description**

The textures are the same with the prefix **TEX\_** except for the shared textures with the suffix **\_MET** (metallic) or **\_MET\_AO**, **NRM** (normal map) or **EMISSIVE** (emissive texture), etc.

Some element's color (hair, glasses, eyebrows, and eyes) can be edited inside the editor with a gradient defined in the editor.

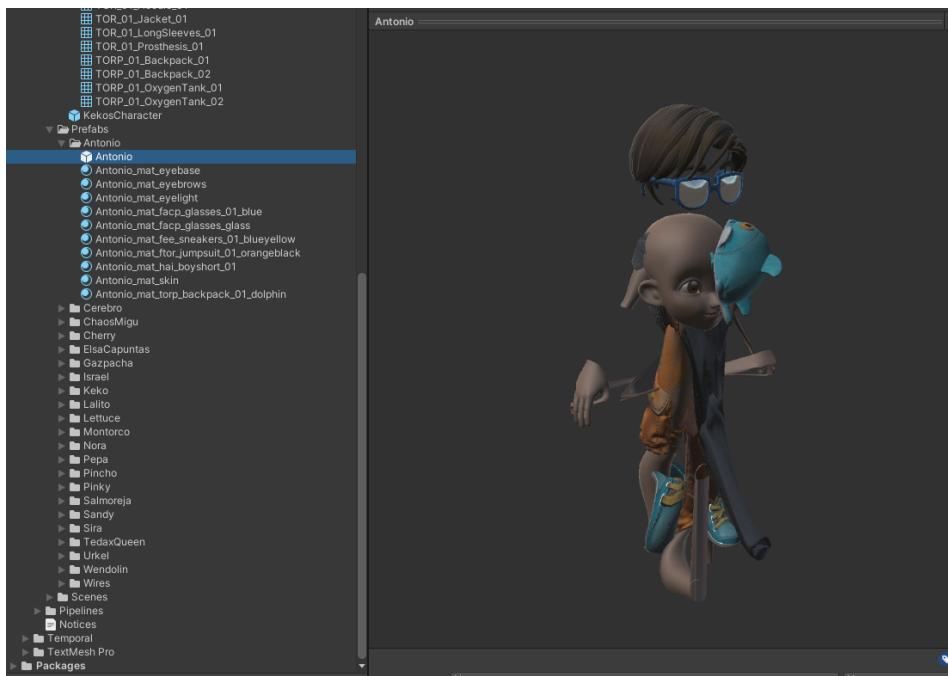


## Preconfigured prefabs

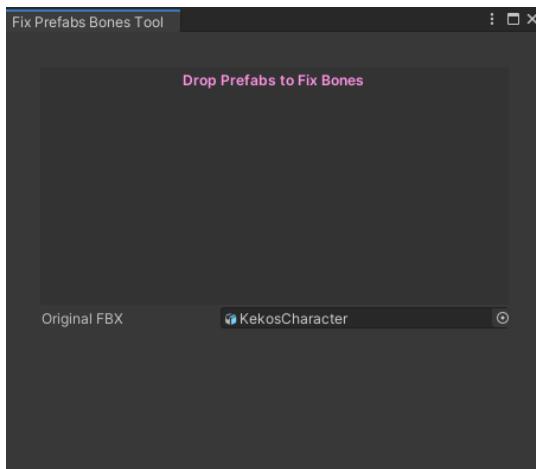
We provide several preconfigured prefabs for fast integration. You can go to the Prefabs folder and select one or more and export a package to integrate into your project, or drag and drop if you have imported the KEKOS package in your working project.



**IMPORTANT NOTE:** Due to an importation error in the Unity pipeline in the mesh bones order, your prefabs may appear to be broken.



We have prepared a tool to fix that problem. Go to Tools -> Mameshiba Games -> KEKOS -> Fix Prefabs Bones Tool window and drag your broken prefabs to fix them.



After that, you can see two “errors”:

- \* If you see in the prefab thumbnail it's still broken, change to some project element and come back. It will refresh the thumbnail with the fixed one.
- \*\* If you have in the scene the broken prefab, you have to save the scene to update the model with the fixed one.

## **Runtime**

The main purpose of the editor is to support the KEKOS character and prefab creation. Although there is an example scene named **03 Runtime Example** showing the runtime functionality with the editor and character spawner in the same scene.

We have added also the save and load system, to the **01 Character Editor** scene to show how the runtime system can work in two separate scenes (one for the editor only **01 Character Editor** and one for the runtime and the editor **03 Runtime Example** (you can have that runtime scene without the editor only with the spawner working)).

You can see all of that working in the [following video](#).

## **Extend the package with my own elements**

As we mention in the elements part, we have two kinds of meshes elements. The skinned meshes and the normal meshes that are attached to a bone. Each of them has a different workflow to extend in the system.

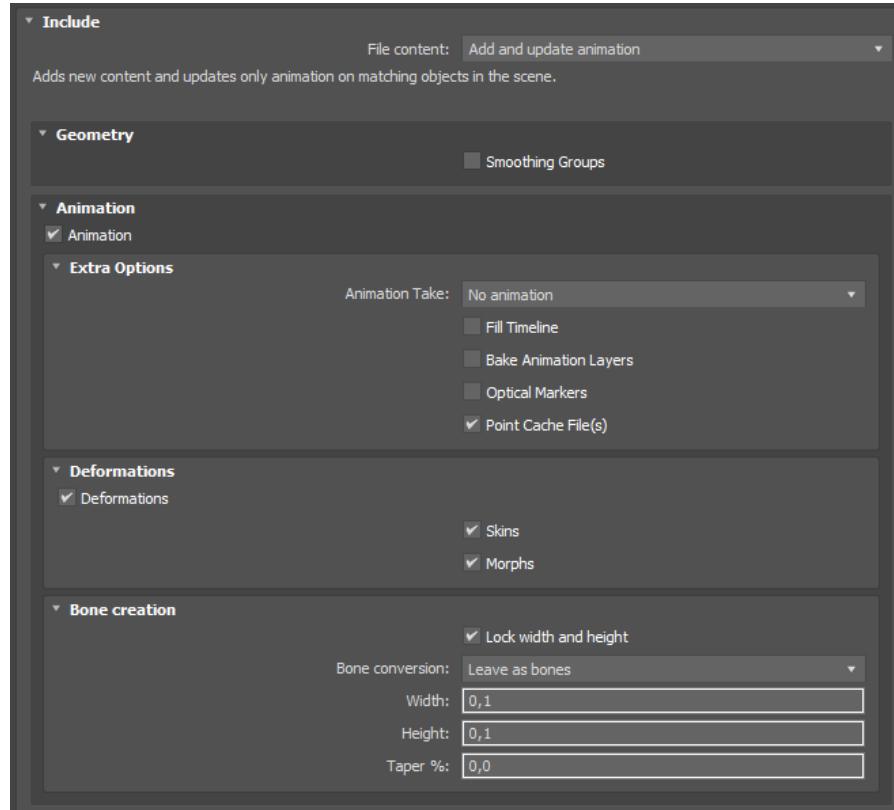
### Skinned Meshes

**IMPORTANT NOTE:** we are investigating a system to separate the elements in different model FBXs, but like right now there are in the same FBX if you change it and we update with new garments you have to repeat the process to add your elements again.

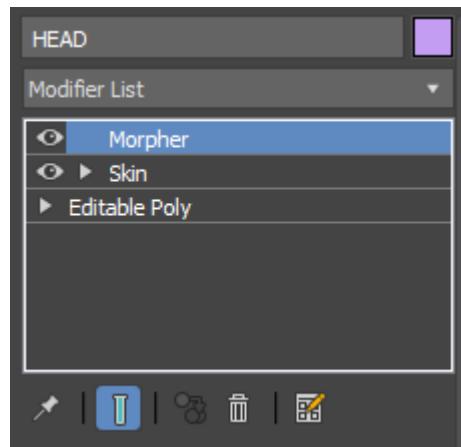
This is valid for **torso, legs, hands, feet, full torso, and backpack**. Follow these steps:

**Note:** this process is described with 3ds Max software, we can't assure it will work with the same steps in other programs.

1. Import the KekosCharacter.fbx to 3ds Max. Use the following settings.



2. We should change the order of the morpher and skin modifiers on the following objects: head and all the eyebrows. The morpher modifier should be on top of the stack.



3. We should fix some morphers names to the correct ones:

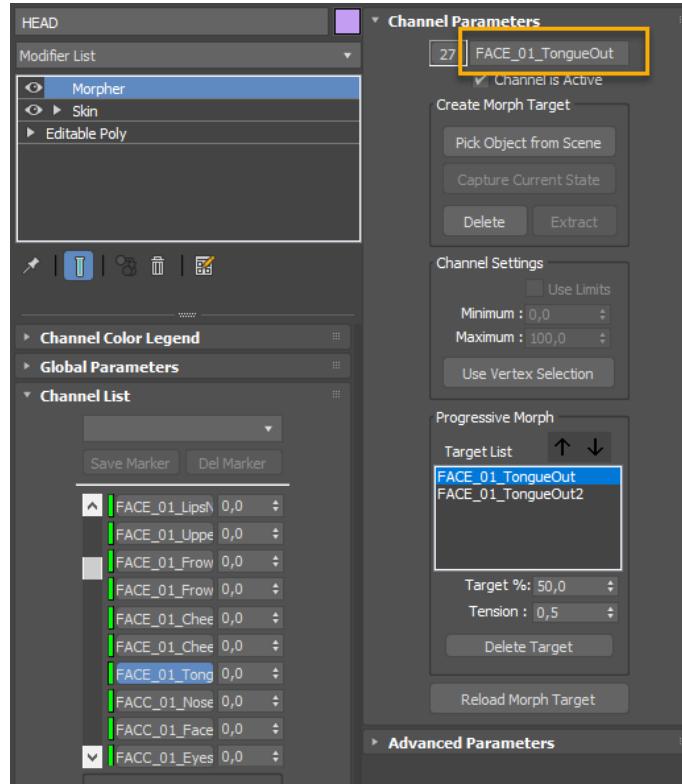
**FACE\_01\_TongueOut**

**FACE\_01\_LowerEyelidLUp**

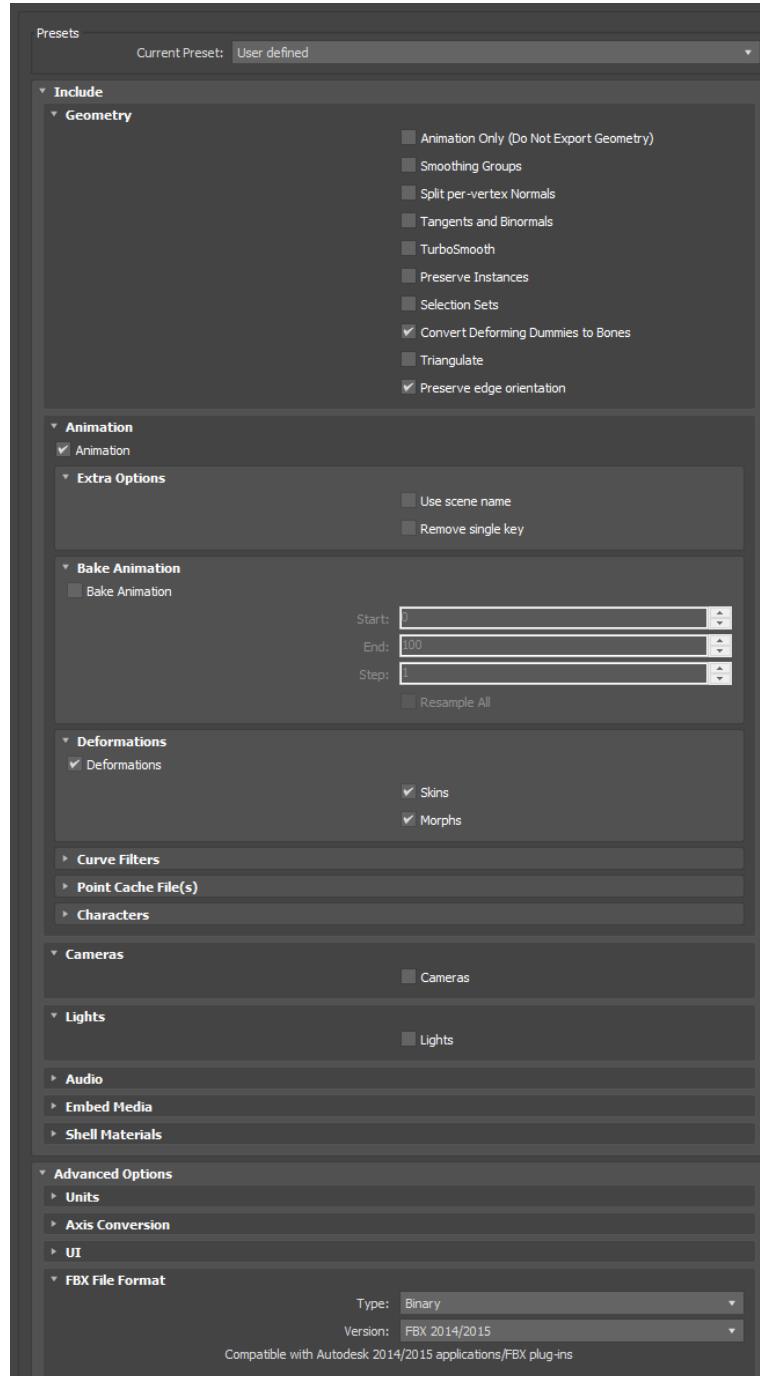
**FACE\_01\_LowerEyelidRUp**

**FACC\_01\_NoseShape**

**FACC\_01\_FaceShape**



4. Add or model your garments. Be sure the scale is 100% and the pivot position and rotation are the same as the other elements.  
*Note:* if your garment includes some skin you need two materials, one for the garment and one for the skin. The skin material should be **ID 1** and should be named: **MAT\_SkinColor**.
5. Include the model inside the **meshes** dummy.
6. Name it properly.
7. Skin the mesh using the bones inside the Skeleton dummy.
8. Export the FBX with the following settings:



9. Replace the KekosCharacter.FBX from Unity.
10. Create the element folder in the proper garment **Models** folder for the materials, textures, icons, and database.
11. Import the textures and sprite icons. Follow the naming conventions.
12. Create the materials. Follow the naming conventions.
13. Create the item database and assign the data (materials, icons, colors, etc.).
14. Add the item database and the models to the element database.

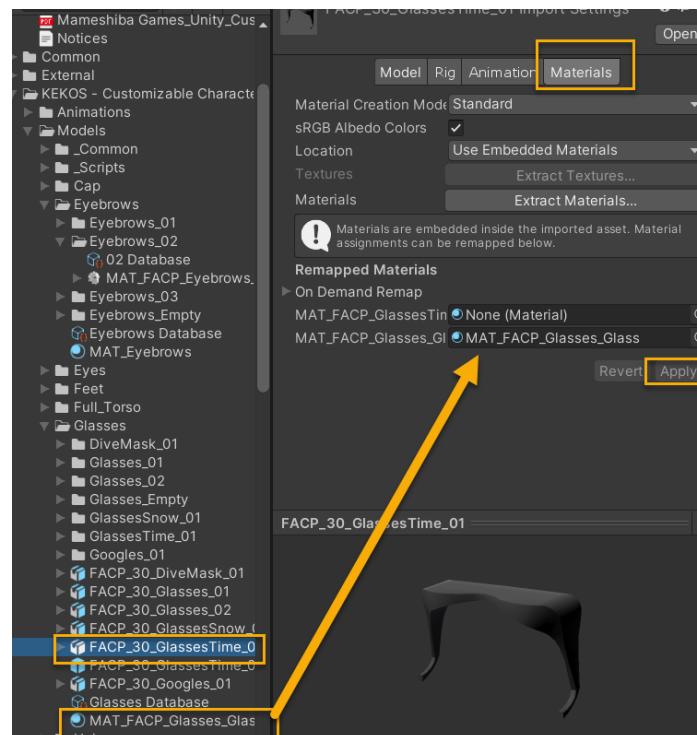
**Note:** in the process, your saved prefabs may be broken, read the [Preconfigured Prefabs](#) section in order to fix them.

## Attaches normal meshes

This is valid for **hair, cap, and glasses**. Follow these steps:

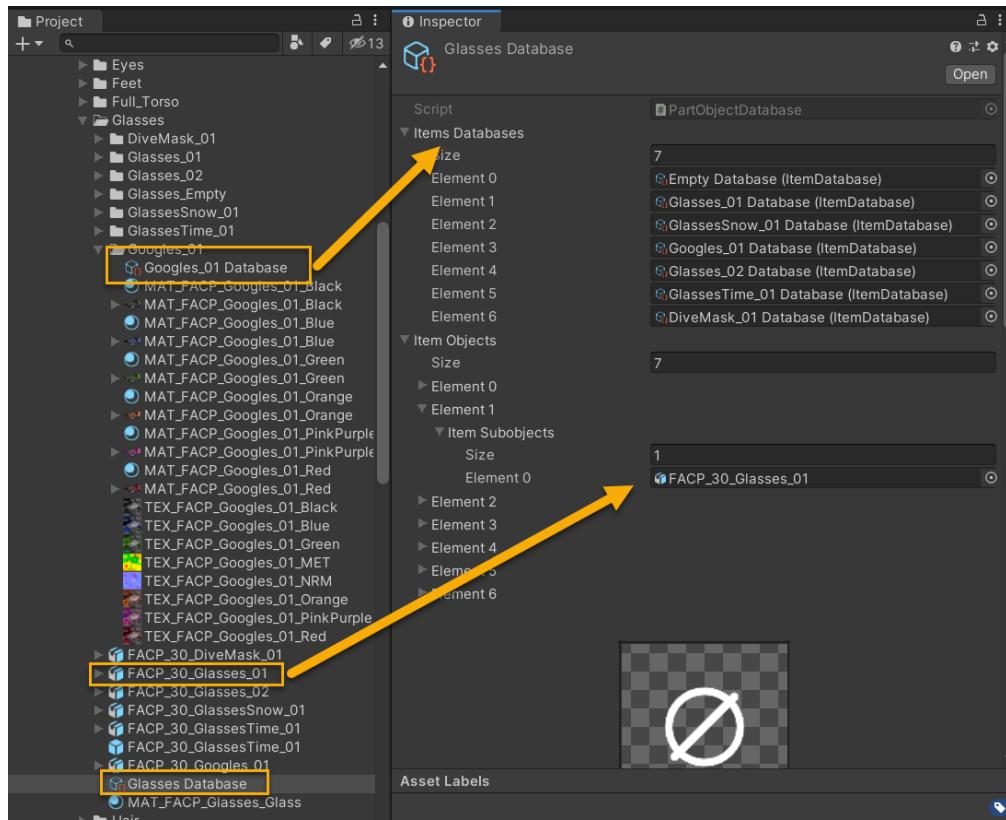
1. Model your **hair, cap, or glasses**, or open an existing one to have the scale and pivot reference.
- Note:** if you are making glasses you have to separate the main material from the glass material and put the main material first.
2. Copy or assign the pivot to your model. Note: the pivot has to be the same as the head bone. We highly recommend copying from an existing one.
3. Export the model in the FBX format with the proper naming convention.
4. Import the FBX into Unity inside the correct **Models** folder.

**Note:** if you are making glasses assign the **MAT\_FACP\_Glasses\_Glass** material to the glass model.

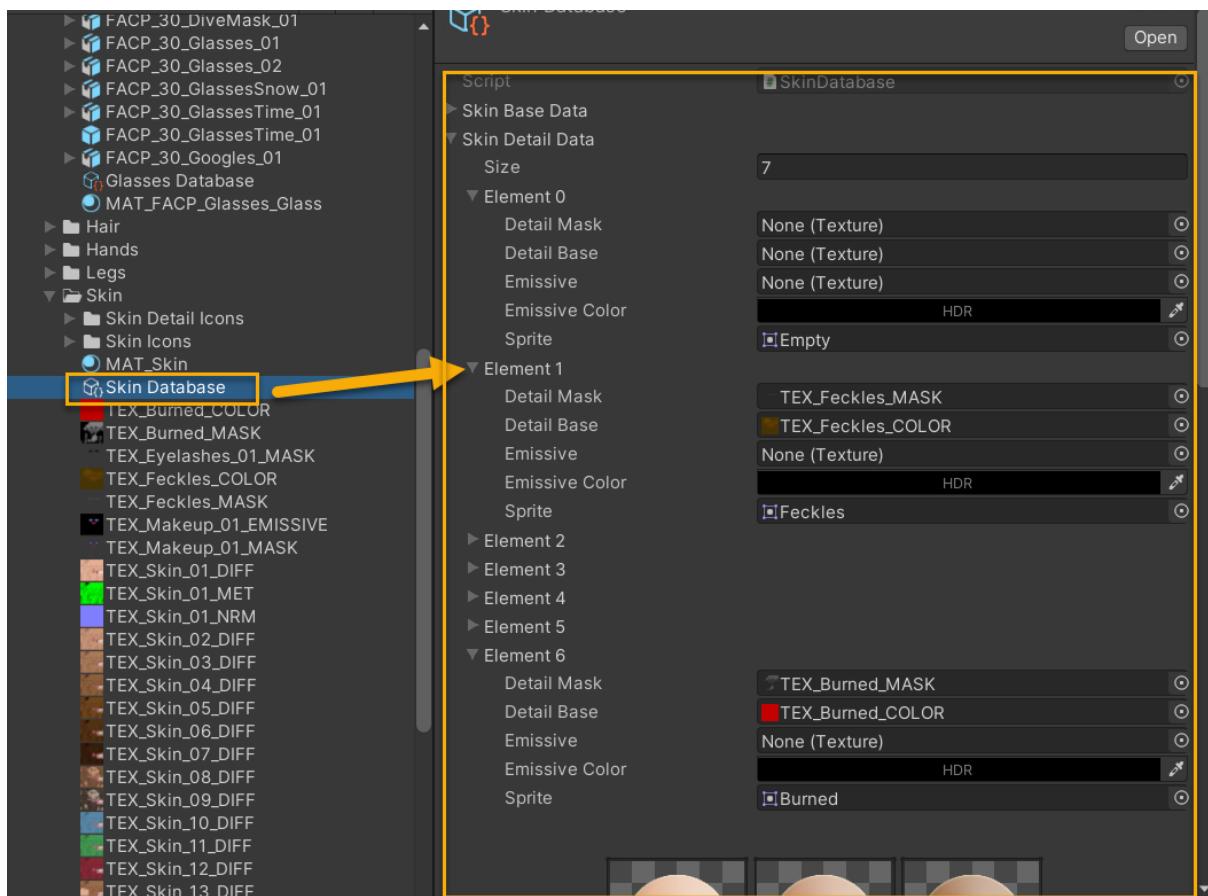


5. Import the textures and sprite icons. Follow the naming conventions.
6. Create the materials. Follow the naming conventions.

7. Create the item database and assign the data (materials, icons, colors, etc.).
8. Add the item database and the models to the element database.



In order to extend the skin colors and details or the eyes texture, you can add your own to the Skin or Eyes Database.

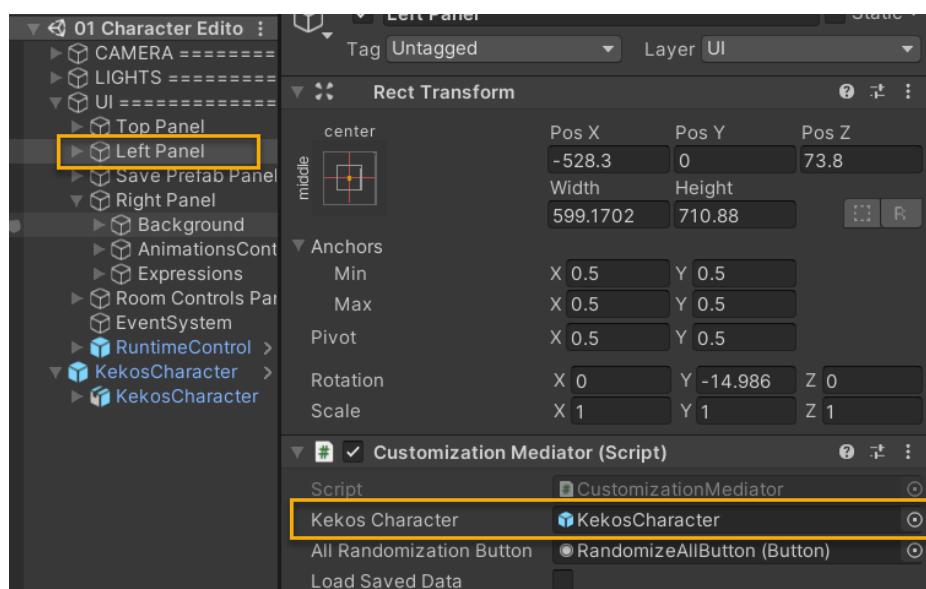
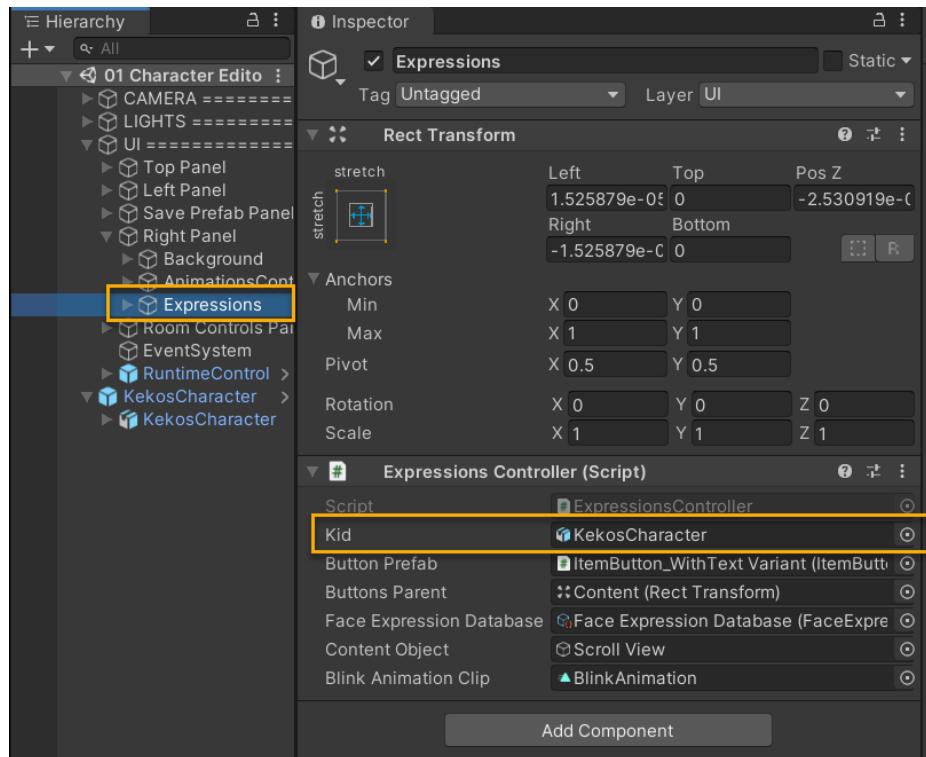


**Use the system with my own base model and not use the KEKOS one**

**NOTE:** KEKOS is not designed for that purpose. The editor is a tool to create KEKOS characters and save them as prefabs.

If you want to use the system with your own base model may demand a lot of work.

You have to follow the naming conventions described in this document. You need to configure the elements, animations, skin, etc. databases with your images and data. Also if you don't have any kind of elements (garment, skin, facial morphers, etc.) you probably will have to change the code in order to make it work. You have to assign your new model to some scripts like the following ones:



This is not a complete guide, only some help in the right direction. As we said, **using your base model may imply a lot of work.**

## Contact



MAMESHIBA  
GAMES

<https://mameshibagames.com/>

[MameshibaGames@gmail.com](mailto:MameshibaGames@gmail.com)

[KEKOS Forum link](#)