Palworld - Model Swapping Guide for the Player Mesh

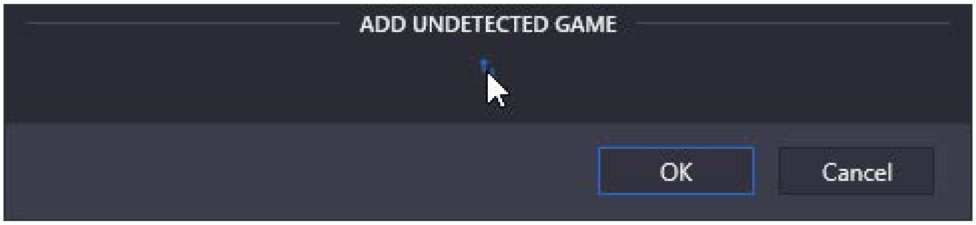
Requirements

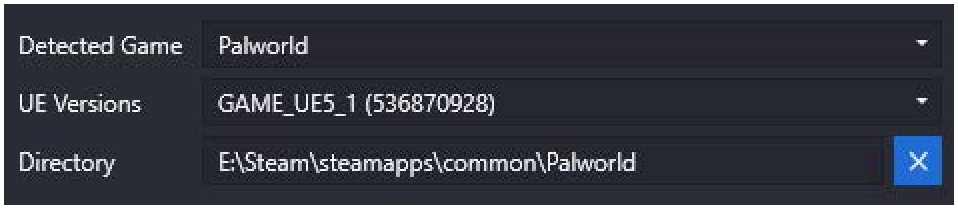
- FModel (https://fmodel.app/download) **note: FModel requires .NET 8.0 Desktop Runtime installed. - also download the mappings.usmap**
- Unreal Engine 5.1 - Palworld uses unreal engine 5.1.1 so any 5.1 version is required for this process.
- Blender PSK importer (https://github.com/matyalatte/blender3d_import_psk_psa)
- UnrealPak

Setting up FModel

The first step to creating our player model is going to be using FModel. FModel is the program we will be using to extract assets such as the character mesh and skeleton so that we can use it to line up our custom mesh to the original model. This could also be useful for beginners, as they can use this model to copy the template of the weight painting.

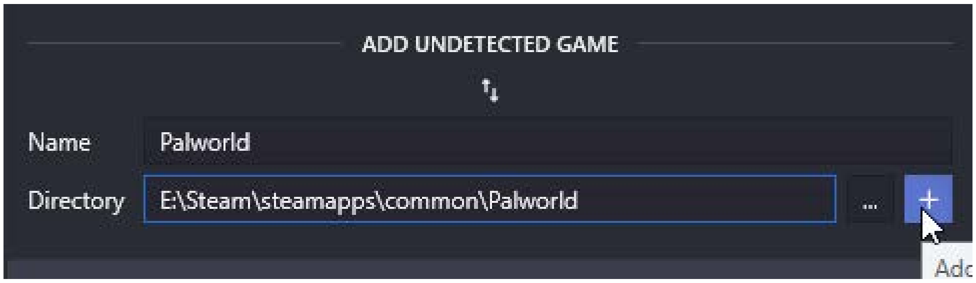
First of all, Run FModel and select the "**Add Undetected game**" option, in the window it opens navigate to your game directory of palworld and click the blue plus icon and next set the UE Versions box to GAME_UE5 setting, then hit OK. Finally, open the settings in the new window and enable "**Local Mapping File**" then set the file to the mappings.usmap file




Detected Game: Palworld

UE Versions: GAME_UE5_1 (536870928)

Directory: E:\Steam\steamapps\common\Palworld



Name: Palworld

Directory: E:\Steam\steamapps\common\Palworld

Now that we are inside of FModel, we can start browsing for the assets you want to swap. First, Double click on the Pal-Windows.pak. This will open the folder where all of palworlds assets are kept and stored. In order to find the directory for the player model, we must open the pal folder, go into the "content" folder, then the "Pal" folder again, then into the "Model" folder and "Character". If at this point you are here for swapping Pal models instead, you should enter the monster directory and find the name of the pal you want to be swapping. If you do not know the name of the pal you want to swap due to the model files having different names, you can visit the google docs spreadsheet made by Noxatris and the modding community. (https://docs.google.com/spreadsheets/d/1D7HF3vG8ECIpmJHthsvGHZfEGIGfW9LzDv3N3zMon9I/edit#gid=0)

However for the player, we go inside of the "Player" Directory. At first glance you may think to open the "Body" folder and use the model from there, however unfortunately it is not that easy and each clothing type has its own model. So for this tutorial, **we have to go inside of the Outfit folder**. To find the skeletal meshes for the assets, you should click on the folder you want and then click on the "Packages" button at the top of fmodel. This will show you the skeletal meshes for this folder. Do note that at this stage the best way to be matching your model up is to grab the Body model, as this includes the player head aswell and lets you correctly get the scale of the whole mesh. At this stage, please again take note of the Entire directory path we have went in. **Pal\Content\Pal\Model\Character\Player\Outfits\SK_Player_Female_Outfit_OldCloth001**

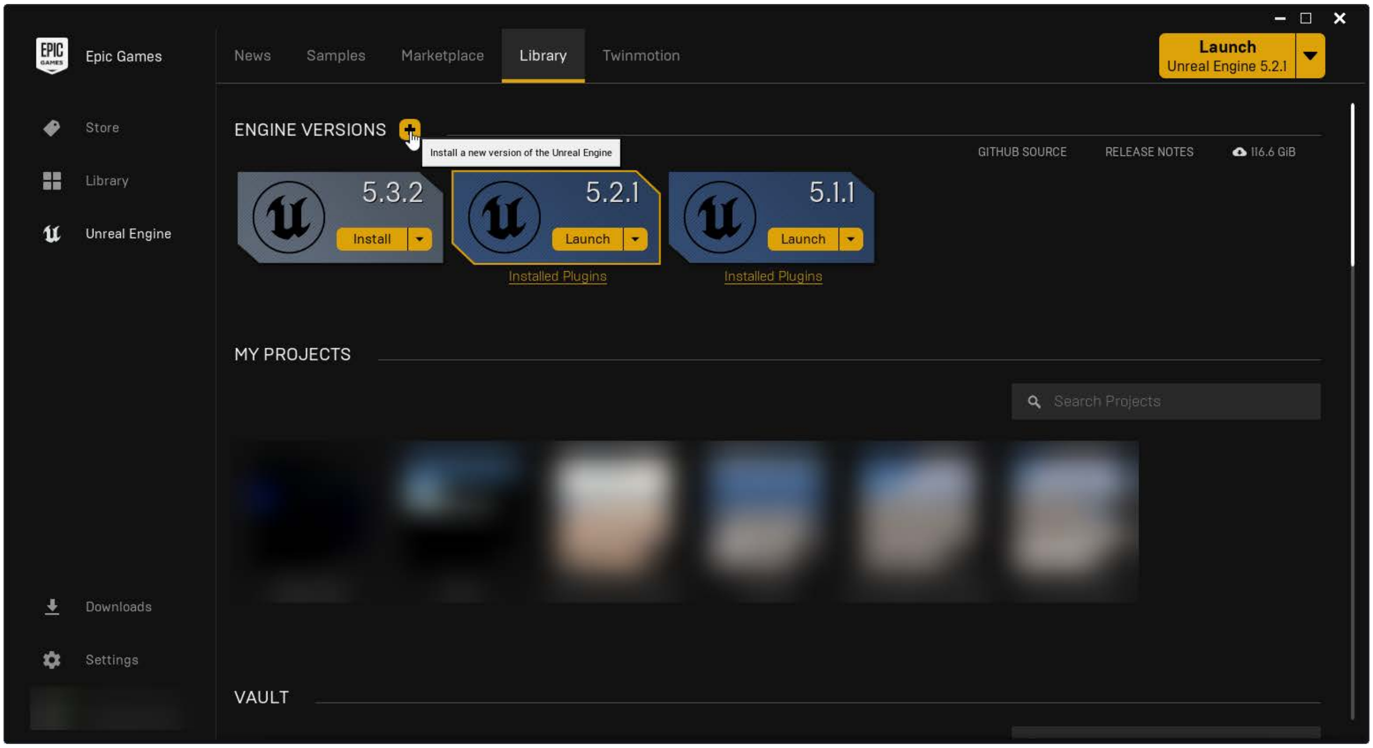
In order to swap the model that you first see in character creation, you must use the SK_Player_Female_Outfit_OldCloth001 folder (or male, depending on what character you want to swap)

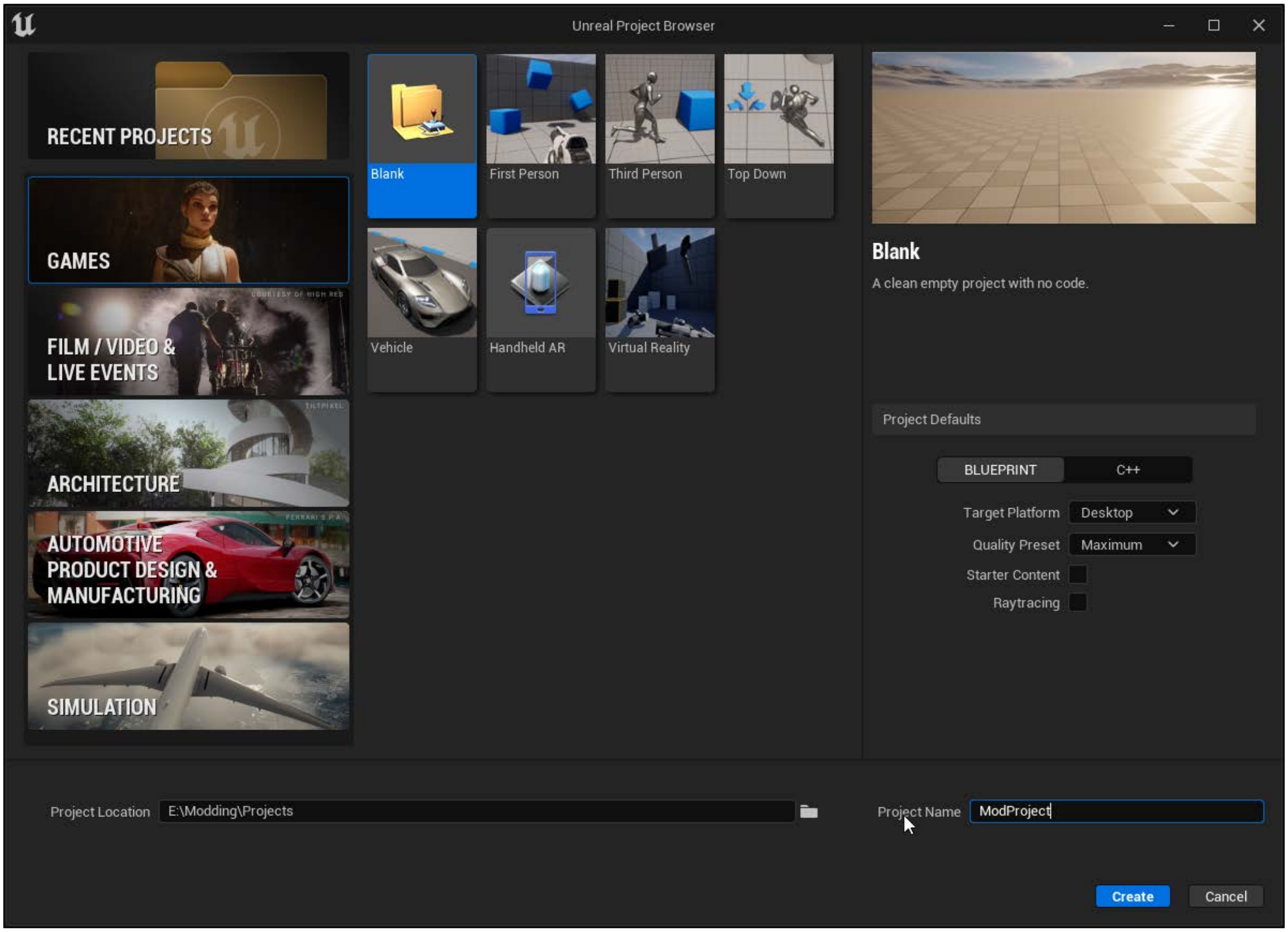
You must also take note of the **Pal\Content\Pal\Model\Character\Skeleton\Human** this is the folder that you must recreate later on in Unreal Engine and place the Skeleton of the model inside of, in order to make sure Unreal and the game knows where our model gets the skeleton from.

Creating our Unreal Project

At this stage, we can now begin creating our Unreal Engine project settings and project structure. Note that for this stage, you must be on any version of Unreal Engine 5.1, I used 5.1.1 as thats the exact version palworld uses. To select your version, you can open the epic games launcher, go to the unreal engine library and click on the yellow plus icon to add a new engine version. Next, click the arrow right above the name of the version and change it to 5.1.1

There are a few very specific settings that need to be enabled for this stage. First of all, boot up unreal engine and create a new project using the blank template. Make sure that starter content is disabled, raytracing is disabled, the project type is set to "Blueprint" and the quality preset is Maximum. The project name and location do not matter, however you can name your project <project name>_P for convenience.

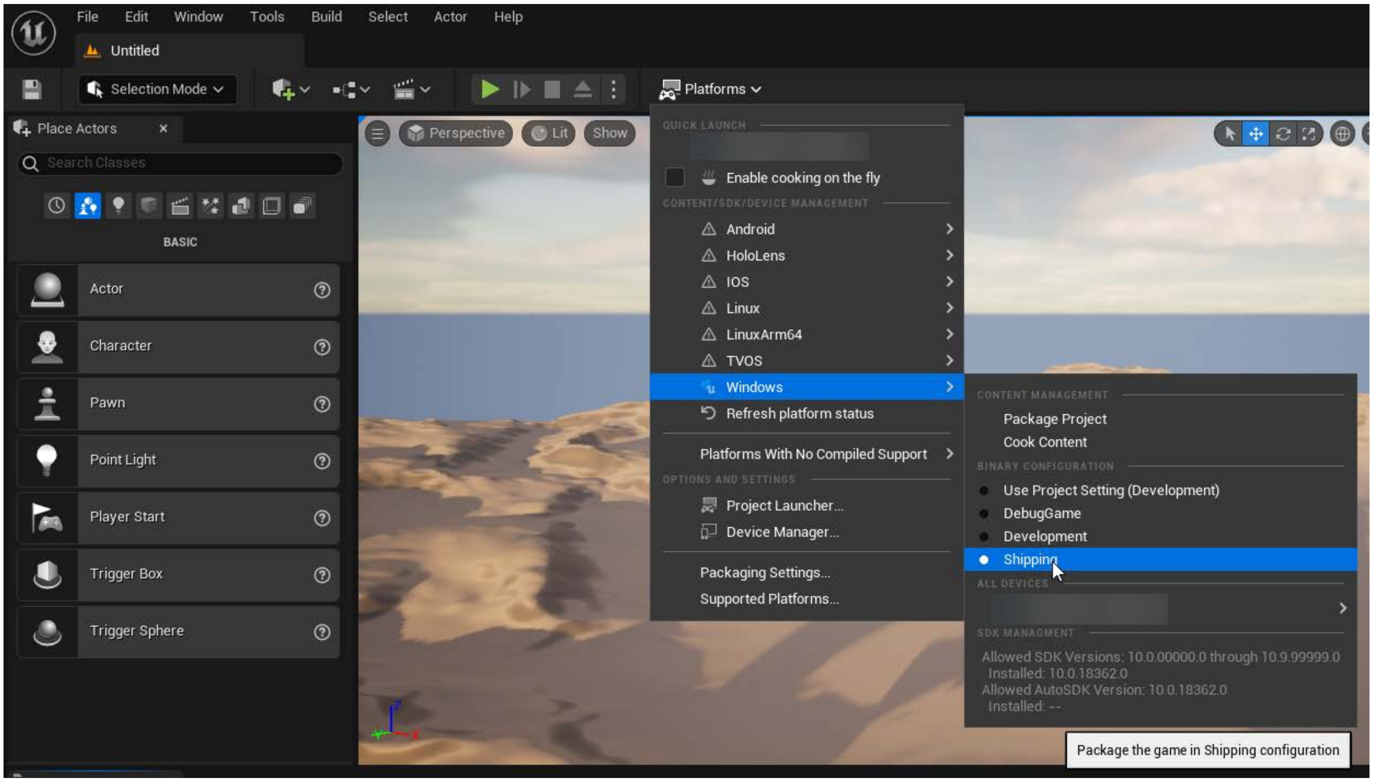




Target Platform: Desktop
Quality Preset: Maximum
Starter Content: □
Raytracing: □

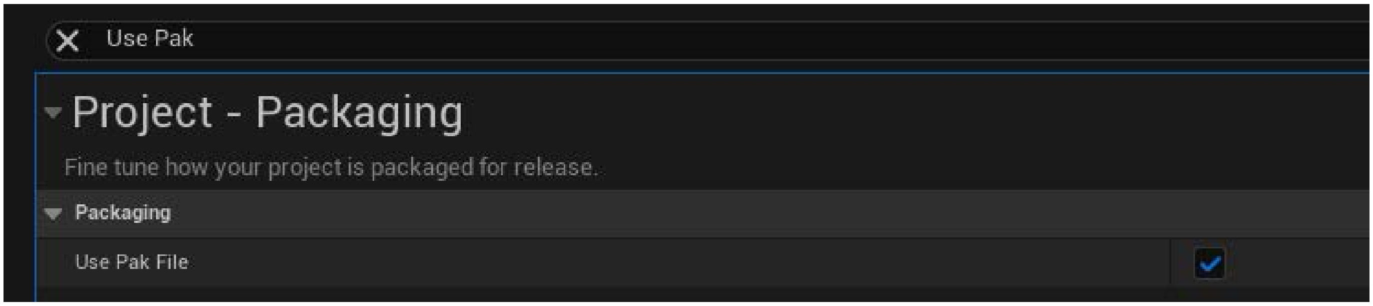
Project Location: E:\Modding\Projects
Project Name: ModProject

After creating your project, you will see unreal open with a blank landscape. Now, we want to click on "Platforms" at the top of Unreal Engine and click "Windows" then "Shipping"



Allowed SDK Versions: 10.0.00000.0 through 10.9.99999.0
Installed: 10.0.18362.0
Allowed AutoSDK Version: 10.0.18362.0
Installed: --

Next, open the project settings and search for "**Cook everything in the project content directory**" and make sure that is set to true aswell as "**Use Pak File**" and set that to true aswell.
After all that is done, you can recreate the folder structure mentioned earlier.



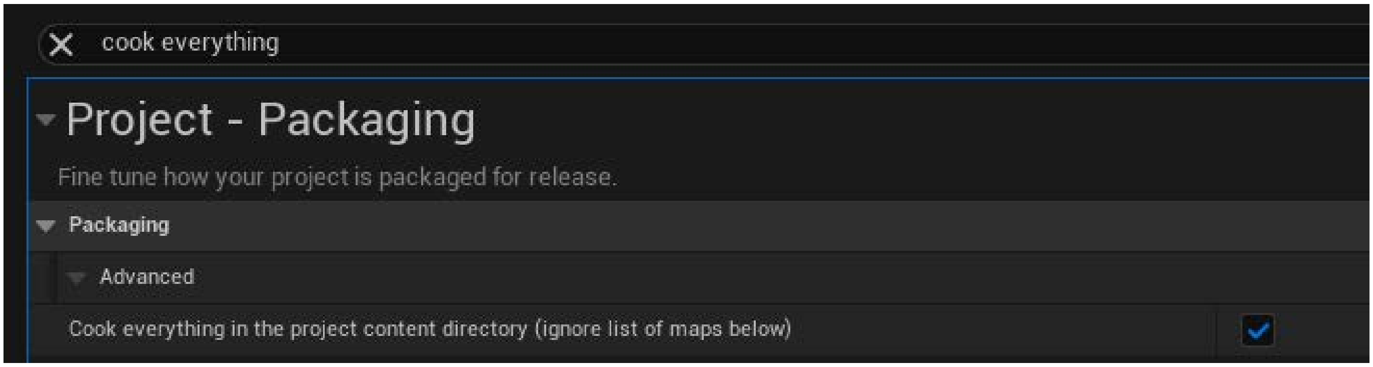
Use Pak

Project - Packaging

Fine tune how your project is packaged for release.

Packaging

Use Pak File ☑



cook everything

Project - Packaging

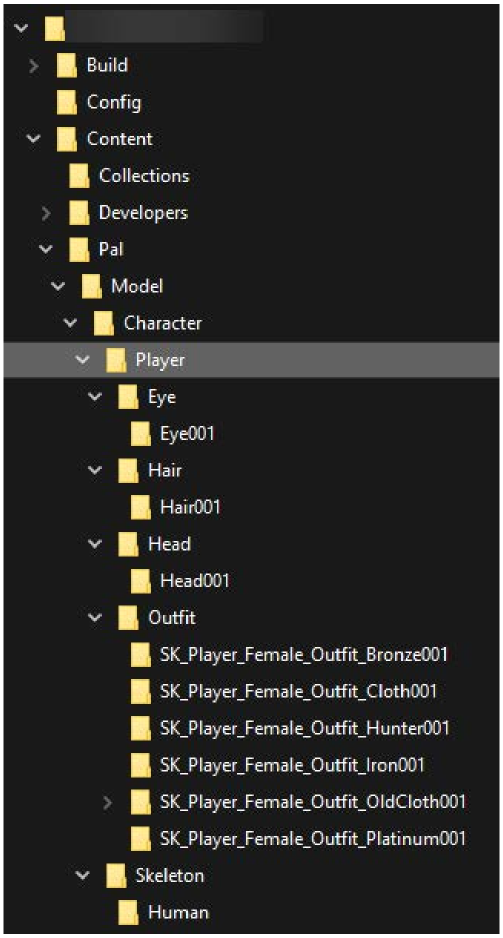
Fine tune how your project is packaged for release.

Packaging

Advanced

Cook everything in the project content directory (ignore list of maps below) ☑

Below is an example of what your unreal folder should look like, with the blurred line being the project name.



- [redacted]
  - Build
  - Config
  - Content
    - Collections
    - Developers
    - Pal
      - Model
        - Character
          - Player
            - Eye
              - Eye001
            - Hair
              - Hair001
            - Head
              - Head001
            - Outfit
              - SK_Player_Female_Outfit_Bronze001
              - SK_Player_Female_Outfit_Cloth001
              - SK_Player_Female_Outfit_Hunter001
              - SK_Player_Female_Outfit_Iron001
              - SK_Player_Female_Outfit_OldCloth001
              - SK_Player_Female_Outfit_Platinum001
          - Skeleton
            - Human

Lyãʎ̊ŘŜ ƨŦ ŭKŜ 9èŜ૪ KI·ʎ̊ǋ I·yŘ KŜI·Ř Ŧƨf̆ŘŜǋă I·ǋŜ ǿKŜǋŜ ǿŜ I·ǋŜ ∃ƨʎ̊y∃ ŭƨ ōŜ Ǉf̆I·Ŏʎ̊y∃ I· ŦŜǿ ŜϒǇŭè ŭŜȅŭdzǋŜ Ŧʎ̊f̆Ŝă૪ ōI·ăʎ̊ŎI·ffè ǿKŜyʃèƨdz ăǿI·Ǉ ŭKŜ ƨdzŭŦʎ̊ŭ ϒƨŘŜfă Ŧƨǋ èƨdzǋ ƨǿy૪ ŭKŜ ∃I·ϒŜ ǿʎ̊ff ăŭʎ̊ff ŜyŘ dzǇ ǋŜyŘŜǋʎ̊y∃ ŭKŜ ƨǋʎ̊∃ʎ̊yl·f KŜI·ŘĸŜèŜă I·yŘ KI·ʎ̊ǋɸ /dzǋǋŜyŭf̆è૪ ŭKŜ ăʎ̊ϒǇf̆Ŝăŭ ǿI·è ŭƨ Kʎ̊ŘŜ ŭKʎ̊ă ŭKI·ŭ L ǇŜǋăƨyl·ffè ʅyƨǿ ƨŦ ʎ̊ă ŭƨ ǋŜǇf̆I·ŎŜ ŭKŜ ŭŜȅŭdzǋŜă ǿʎ̊ŭK ŭKŜ ƨf̆Ř ƨyŜăɸ ¸ƨdz ǿI·yŭ ŭƨ ǋŜǇf̆I·ŎŜ I·ff ƨŦ ŭKŜ ŭŜȅŭdzǋŜă ʎ̊yʃ ŭKŜ Ŧƨf̆ŘŜǋ૪ ăƨ yƨŭKʎ̊y∃ ǋŜyŘŜǋă ʎ̊y∃I·ϒŜ ƨǋ Ŝf̆ăŜ èƨdz ϒI·è ŜyŘ dzǇ ǿʎ̊ŭK ăƨϒŜŭKʎ̊y∃ ăŭʎ̊ff ǋŜyŘŜǋʎ̊y∃ I·yŘ f̆ƨƨʅʎ̊y∃ ƨŘŘ ƨǋ ōǋƨʅŜyɸ

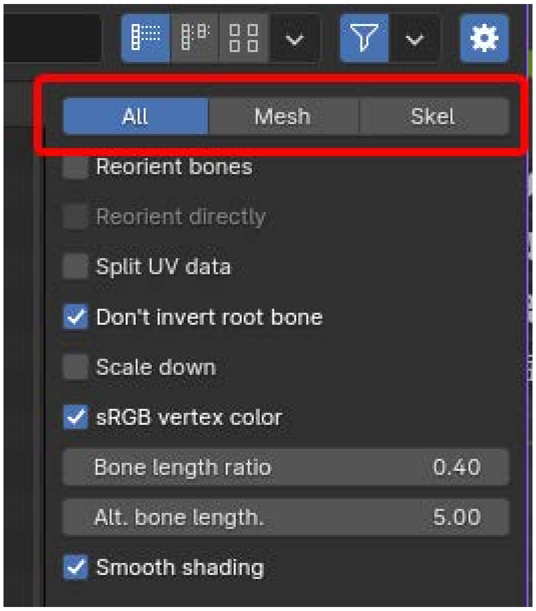
aI·ʅʎ̊y∃ èƨdzǋ ϒƨŘŜf ǿƨǋʅ

yƨǿ ŭKI·ŭ ǿŜ KI·ǾŜ ƨdzǋ Ŧƨf̆ŘŜǋ ăŜŭdzǇ ŎǋŜI·ŭŜŘ I·yŘ ǋŜI·Řè૪ ǿŜ ŎI·yʃăŭI·ǋŭ preparing our model. Here I use Blender, so for that you are going to need the Blender PSK model importer plugin mentioned earlier to import the models. You also need to have extract the player model using Fmodel from Pal\Content\Pal\Model\Character \Player\Body\<female/male> and then the packages tab.

Naming guidelines

In palworld and unreal engine, assets have naming conventions. Here is an important time to mention these. Remember these model names or note them down for later.
SK_ - Skeleton/Skeletal Mesh (skeletons are in their own folder in palworld, you will know which is which.
SK_<modelname>_PhysicsAsset - Used for Physics assets
MI_ - Stands for "Material Instance" But will be used for our normal materials later.
T_ - Texture files. T_<texturename_B are the colour textures. T_<texturename_M = MRAO (Metallic, Roughness, Occlusion and Specular texture - you can make these in substance painter when exporting by choosing Unreal Engine 4 as the type, or use this link to conver textures to mrao using gimp https://dev.epicgames.com/community/learning/tutorials/6Gn5/creating-mrao-textures-using-free-software)
T_<texturename>_N is for normal maps.

Anyhow. Now you can export the psk from the right folder in fmodel by right clicking and save model as psk, where you can then open blender and import this player model in order to copy the skeleton bones and position/sizing. When importing into blender using the plugin, a common error is that the model will only import the skeleton, which can be fixed by clicking "Mesh" on the import settings to just import the mesh



□ Reorient bones

□ Reorient directly

□ Split UV data

☑ Don't invert root bone

□ Scale down

☑ sRGB vertex color

Bone length ratio 0.40

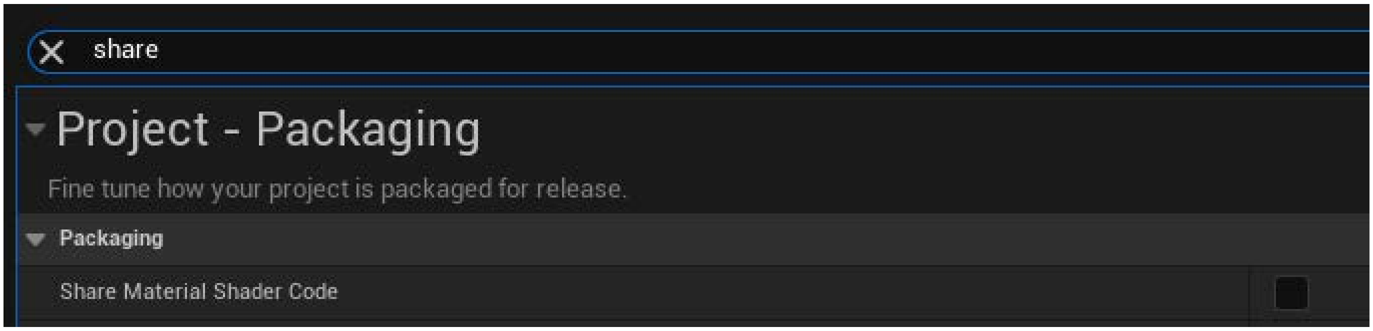
Alt. bone length. 5.00

☑ Smooth shading

At this stage, if you do not know how to weight paint and rename the bones correctly, I would recommend following another tutorial online. These are plentiful and often tutorials from modding other games can be a great resource for beginners here. But basically; there are a few ways to do it. One way is to just use the original armature imported from the PSK, enter weight paint mode and copy the weights (making sure any original vertex groups were removed) and then adjust the weights where needed. However if your mesh is a bit different, this will not work. You could also use the secondary option of using your original meshes bones, but renaming them to the same as the original skeletons. You also must ensure the armature object is called "Armature". Another point here that could be used to save time is to use the material combiner addon, so once you have all your materials set simply combine them into one texture file then you can export your FBX. This is not mandatory however it will make the material process much easier

Now that we have our mesh setup, you must import it into unreal and put the skeleton and skeletal meshes into the right folders. The skeleton goes in the human folder that I mentioned earlier, and the name will be "**SK_PalHuman_Skeleton**" and for the skeletal mesh itself, the name will be "**SK_Player_<Female/ Male>_Outfit_OldCloth001**"

Now we will move onto one of the last stages for model swapping the player mesh, here we have to make an ABP (Animation blueprint) To setup our materials and also 100% make sure that the head is hidden ingame. Before doing this, we have to again open our project settings and turn off the setting called "**Share Material Shader Code**"



share

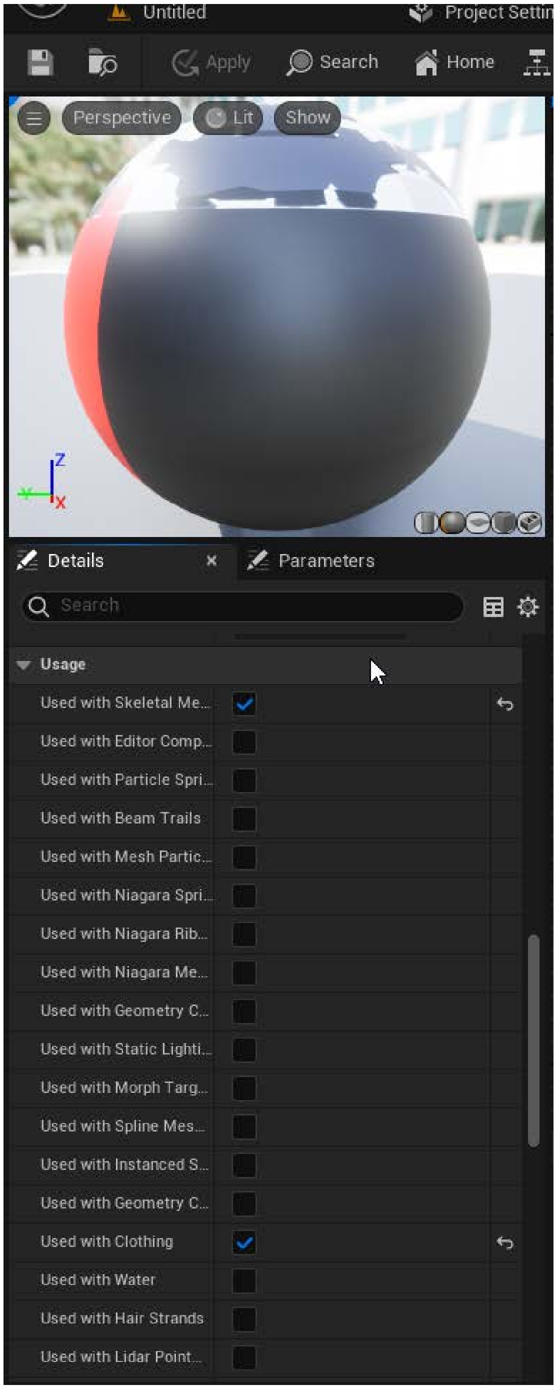
Project - Packaging

Fine tune how your project is packaged for release.

Packaging

Share Material Shader Code ☐

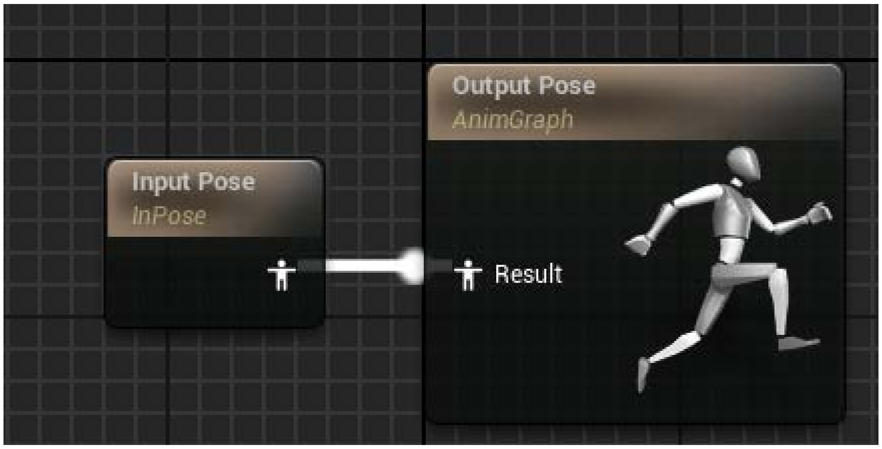
Now you can open your material settings, scroll down and make sure it has "**Use with Skeletal Mesh**" and "**Use with Clothing**" enabled. I'm unsure if use with clothing is needed, however I had it on mine and it worked flawlessly, so for this im enabling that too. Your material and texture don't need specific names, since we will be setting the material manually it does not need to match the games name here.



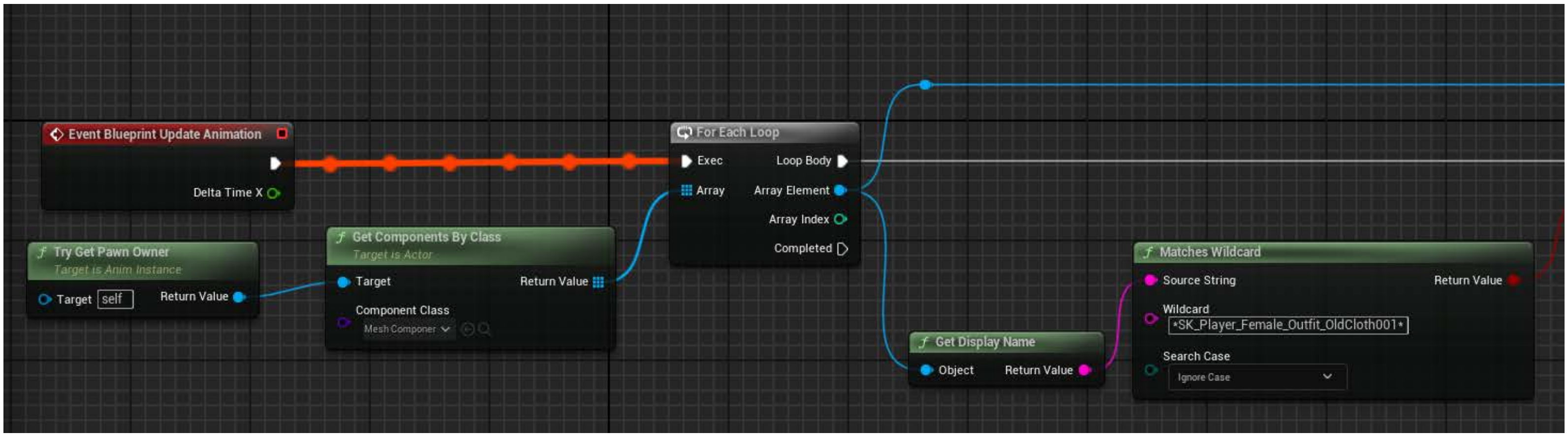
Usage

Used with Skeletal Me... ☑
Used with Editor Comp... ☐
Used with Particle Spri... ☐
Used with Beam Trails ☐
Used with Mesh Partic... ☐
Used with Niagara Spri... ☐
Used with Niagara Rib... ☐
Used with Niagara Me... ☐
Used with Geometry C... ☐
Used with Static Lighti... ☐
Used with Morph Targ... ☐
Used with Spline Mes... ☐
Used with Instanced S... ☐
Used with Geometry C... ☐
Used with Clothing ☑
Used with Water ☐
Used with Hair Strands ☐
Used with Lidar Point... ☐

Now, create an animation blueprint (called whatever you want, however I recommend using the clothing model name if you plan on swapping the other clothing models too) In the anim graph, we simply have to add one node which is an input pose node, connected to the output pose node.



After that, open the Event graph where we are going to make the code to set our material and hide the other meshes if they were to show for some reason.
From the "**Try Get Pawn Owner**" Node, you should drag off the return value and attach a "**Get Components By Class**" Node, making sure to set the component class to "**Mesh Component**". The array value return node from this is then hooked into a "for each loop" that is attached to the "**Event Blueprint Update Animation**" node at the start, from the array element of the For each loop, we must attach it to a "**Get Display Name**" Node, then hook the return value there into a "Matches wildcard" Node. The **wildcard has to be the name of the mesh you have surrounded by a * on either side**. E.G ***SK_Player_Female_Outfit_OldCloth001*** and have search case as ignore case. The boolean from this is hooked into a branch, where after the false, we get the array element node again from the for each loop and drag that out to a "**set hidden in game"** node with the new hidden set to true. From the false, we have our set material nodes. Here is where when I said earlier it would be easier with one material, since you have one material node with the element index of 0 and the material being the once you set. If your mesh has multiple however, its a fairly similar process where you just have multiple set material nodes and have each as the element index of whatever slot is on the model. So for example if your model has the eyes in slot 0, you set the material to be the eyes and index on the node to be 0, then if the face is slot 1 you set index to 1 and material to the eyes, etc.

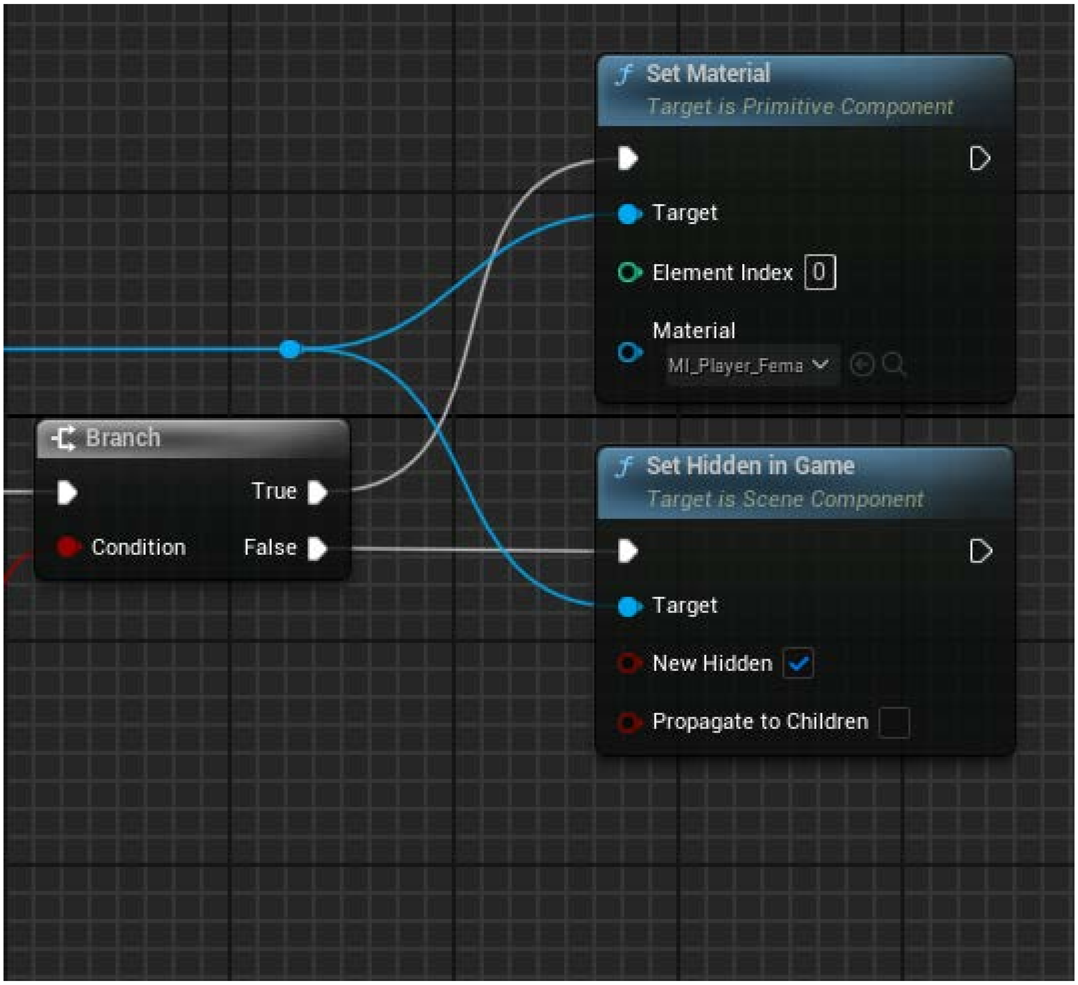


Target self

Component Class Mesh Componer

Wildcard *SK_Player_Female_Outfit_OldCloth001*

Search Case Ignore Case



Element Index 0

Material MI_Player_Fema

New Hidden ☑

Propagate to Children □

To end off the ABP section, open up your mesh file in Unreal and search for "**Post Process Anim Blueprint**" where you then assign the new ABP that we have just created.

Finishing up and Packaging our mod

Now we are getting to the final stage! After everything above is setup, you can now try packaging your mod. The first thing to do is to cook your project, from the window where we went earlier to set the project type to shipping, instead you now will click the "Cook Content" button. On the first time doing this, it will take a long time as Unreal is packing the engine files too, unless you have already cooked. Do note that each time you change assets, you have to recook the content!

Next, you want to find the Unreal project directory, go to the "saved" folder, the "cooked" folder, then "windows" then whichever folder is your mod name, then "content" and go through until you see the player and skeleton folder. When packaging the mod, we do not need to pak the skeleton file. It is simply used for unreal to know where to find the skeleton for our mesh. You do however want to pak any of the assets we changed earlier, such as the eye/hair/head textures, the player model in the outfit folder, the materials, the textures, the physics asset, and our animation blueprint. However if your setup is the same as mine, you should be able to just copy the "Player" Folder. After copying it, you want to back out of this directory and find somewhere on your drive where you want to make the mod folder. In here, we have to **make sure the folder is called <modname>_P** as having _P in the name makes the game know its a patch file that will overwrite game files. In here, your folder directory should be **<modname>_P\Pal\Content\Pal\Model\Character** and then paste your player folder from the cooked unreal folder into here. Now, you can back out again and find the Unreal Pak files that you downloaded, then open a window and drag the mod folder into "UnrealPak-With-Compression.bat" which should give you a window telling you how many files it patched. The program will then place the finished .pak file in the directory just before where you saved your mod and **should be called <modname>_P.pak**

now you can install the mod!
Steam:
- Right click palworld, manage > browse local files then go inside the pal folder, go inside content, paks then drop your pak file in there.

Gamepass:
- Right click palworld game title in the xbox app, click "Manage", click the files tab, click the browse button and the root folder of the xbox library will open the palworld directory is **B0E91CD1-8929-451B-A14D-8CE236A4964F\Content\Pal\Content\Paks**

Guide Credits

NorskPL - FModel/UE setup/folder structure/pak creation/ gamepass install directions

Sleepyhead08/Smoovers - Original Model Swap tutorial designed for pals

Mythical - ABP instructions and setup

Noxatris and anyone else involed - Pal name spreadsheet

Amadeus (me) - writing of guide and putting all steps into a PDF

if any steps are wrong or perhaps need to be improved/changed, please tell me!