CDDT'S Living Guide to Custom Stage Modding in KOF XV

This will be updated as new information is learned.

Pre-Requisites:

- Access to KOF Files
- UE4.26.2
- U4Pak
- Umodel
- Starter Material.7z

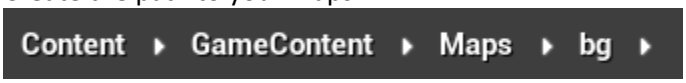
BLUF: In KOF you can use fully custom stages of your own design by replacing the existing bg umap. This guide will get you started on making your own custom maps.

Step 1: Setup

Follow the existing guides in the KOF Resources channel to setup your unreal engine

Step 2: Setup your Directories in UE4

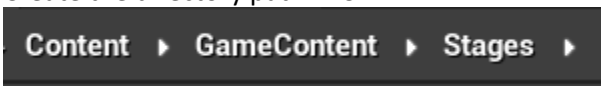
a. Create the path to your maps.



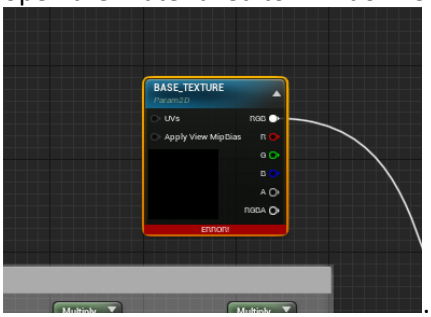
b. When you make a custom map you will save it here with the name of the stage you will be replacing
    a. In my first stage I used ST_000TRA for the Training Stage
c. Decide the location of your Custom Assets
    a. To make it easier to grab and pak I decided to make a folder called CUS_TRA (Custom Training Stage) within the bg folder but it's completely up to you.

Step 3: Import the KOF_Starter_Material.7z

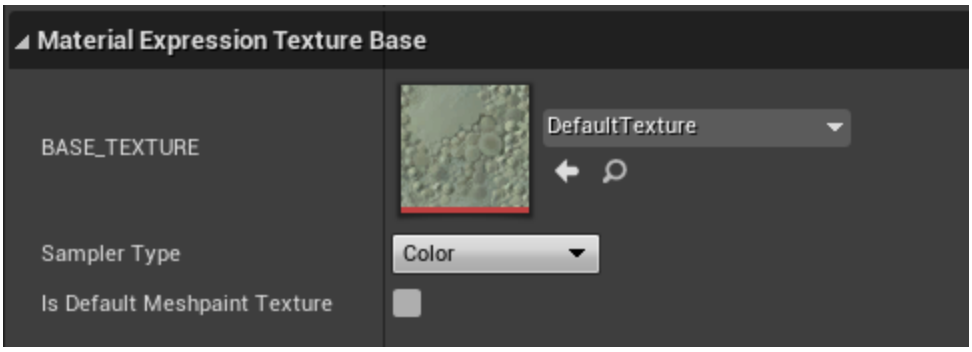
a. Create the directory path in UE:



b. Unzip and using file explore, navigate to to the equivelant of KOFXV\Content\GameContent on your PC.
c. Put the contents of the folder in here. All of the emulated materials are based on Stage materials which are in their respect folders.
d. Go back to UE and navigate to where you imported the Materials folder and double click on it to open the Material editor window. Several of the texture nodes will have a red warning below it.



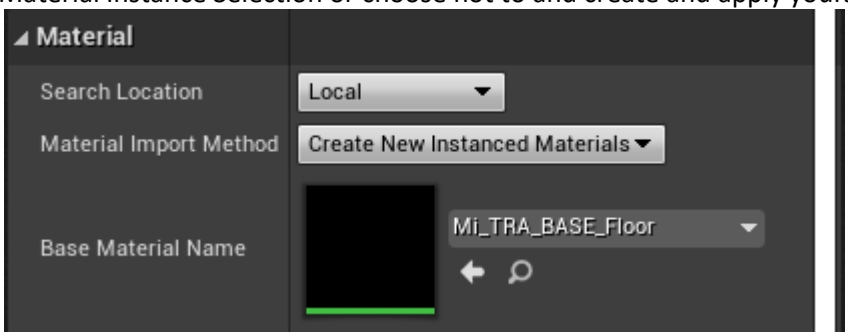
e. Select the Node and in the panel on the left select a default texture



f. Do this for all of them that show a warning then hit Save.

Step 4: Import your custom assets.

a. Import Settings
    a. Use default settings for Static or Skeletal
b. Material Instance Selection or choose not to and create and apply yourself.



c. Whatever textures are attached to your mesh will be loaded as best as ue can determine and use default if none are assigned when creating new Material Instances.

Step 5: Material Instancing/Stage/Map Setup

a. Material Instance Referencing
    a. When apply materials to your custom mesh you will need to reference the MI that corresponds to the desired appearance you are looking for. I've created a list of the emulated Materials, their Referenced MI's, and their attributes.
    b. You can make instances from the Referenced MI's to use on your assets and change parameters.
    c. PROTIP: Create a separate folder and make personal use MI's that you can use in future projects. So if you want to use say an emissive you can make a folder called "CDDT/MatInst/" and create a MI called MI_Emissive that is based on an emissive material per the chart.

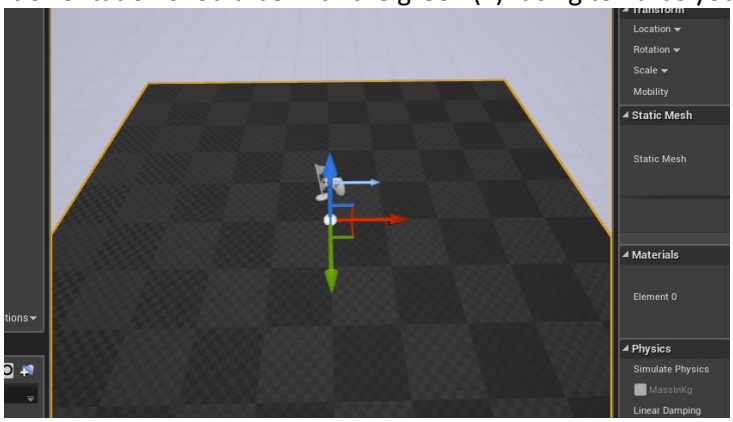
Material Referencing Chart

Emulated Material	Reference Material Instance	Property
M_TRA_Floor_Master	MI_TRA_BASE_Floor	Maps: Diffuse, Emissive, MRO, Normal Params: Emissive Power, Metallic Power, Metallic Value, Roughness Power, Roughness Value
M_LID_NoLIT	MI_LID_LIGHT	Maps: Color

		Vector: Emissive Power and Tint
M_CHN_PNO1M	M_CHN_PineNeedles_01	Maps: Color, Normal, Specular Params: Opacity (of SSS), R (Roughness), Specular, Wind Intensity, Wind Speed, Wind Weight Vector: SSS for Subsurface coloring.
	MI_SWC_LEDLIGHT_B	Maps: Color, Emissive Mask, Normal, MRO Vector: RGB_Emissive_Power

b. Stage Info:

a. Viewport orientation should be with the green (Y) facing towards you.

e 

b. A typical stage is 2400 in width so +1200 -1200 depth and height limitations currently unknown.

c. To keep objects out of view of the camera be sure to keep any mesh items that could potentially block the players view outside of the range between Y:0 and Y:-500

d. Using an extracted Training Stage model is a good tool as well to get a bearing on your overall stage dimensions.

c. Stage Lighting:

a. So far I just make 2-3 directional lights then put them all on Channels 0,1, and 2 but feel free to explore additional possibilities.

i. From what I can tell, Channel 0 is Background, Channel 1 is 1p, and Channel 2 is 2p …. generally speaking.

d. Stage Blueprints:

a. Work as normal.

Step 6: Finishing the Mod

a. Be sure to include all folders that contain your stage elements in your pak
Stage: umap, uexp,
BuiltData for Stage: uasset, ubulk
Textures/Custom MI: uasset, ubulk, uexp
Mesh: uasset, uexp
Blueprints: uasset, uexp

b. **DO NOT INCLUDE ANY REFERENCED ASSETS** like the Emulated Materials or the Referenced MI.
c. You can pak with u4pak or Uverum
    a. If not using Uverum just be sure to include a sig file which you can copy from the game pak folder then rename to match the mod pak.
    b. All mods and there sigs go in the ~mod folders as usual.