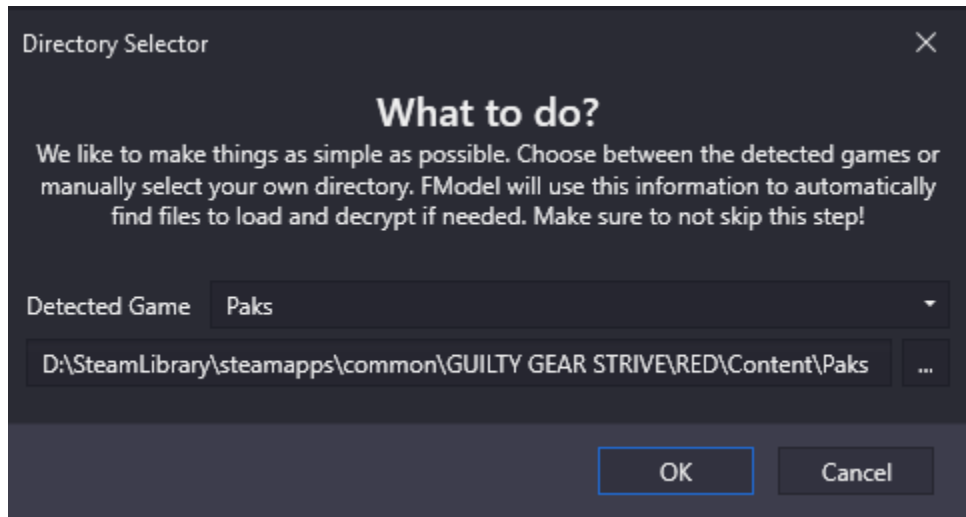


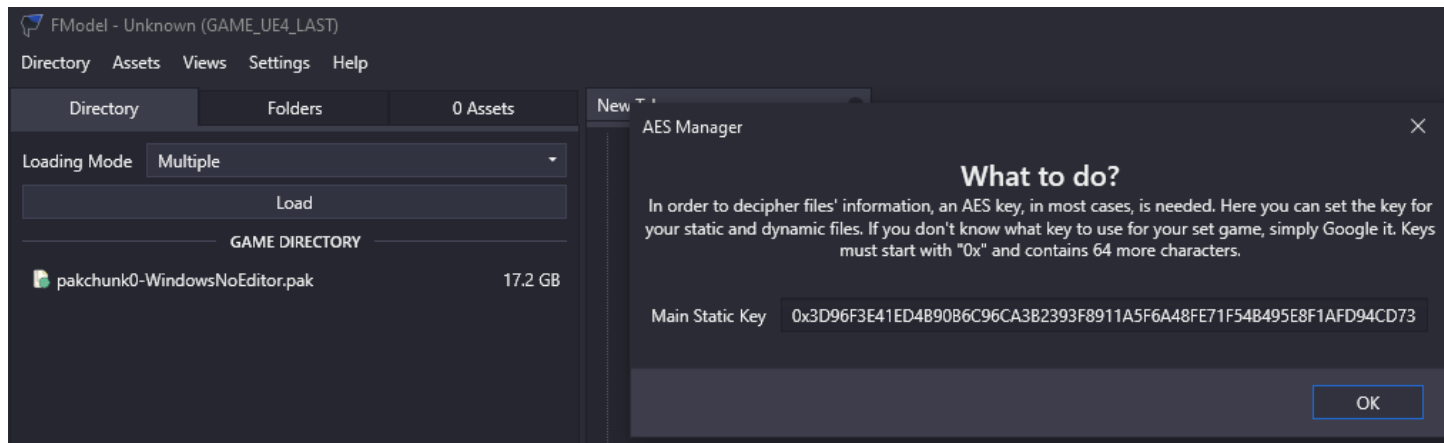
FModel Setup

This section I used Strive as an example.

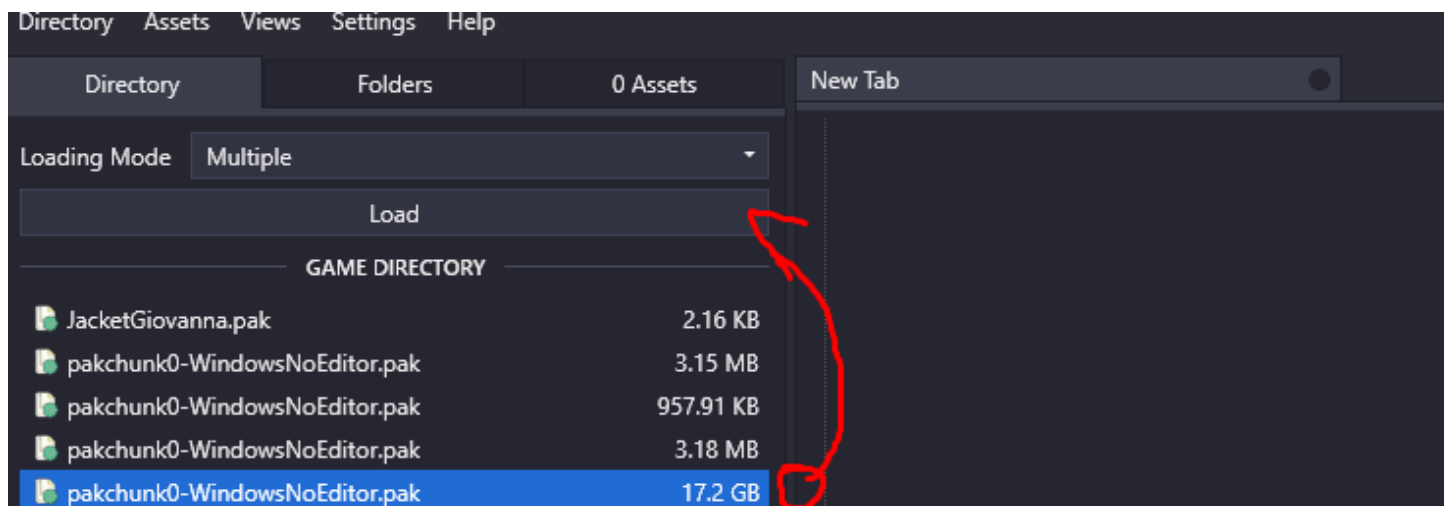
- Download FModel from here: <https://github.com/iAmAsval/FModel/releases/tag/4.0.0.2>
- Once installed it should ask for the directory of your game, set this to wherever the **Paks** folder is.



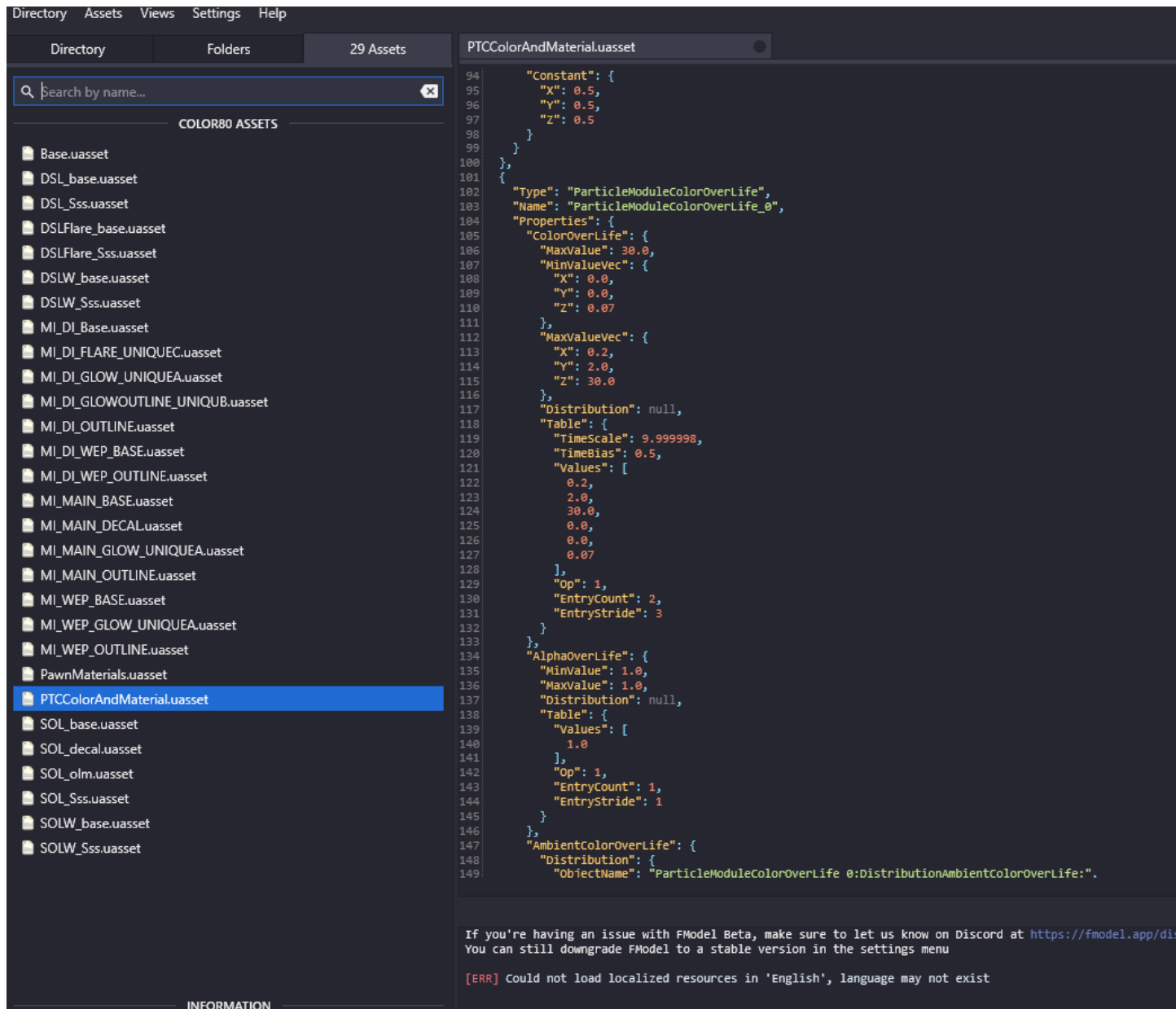
- Once it's setup, select Directory, then AES. Input the AES key.



- select the main game pak (probably the biggest one there), and hit the **Load** button.



- From there you can navigate to whichever file you want to see the data of.



Unreal Package Extractor

This section I used FZ as an example.

- <https://gildor.org/downloads>
 - On this site is Unreal Package Extractor, download it and put it in a folder
 - Create a .bat in that folder, and paste this into it in notepad:
 - For %a in (*.uasset) do extract.exe -game=ue4.17 "%a"
 - For %a in (*.umap) do extract.exe -game=ue4.17 "%a"

Other tools

Unreal Package Extractor	Size 101.65 KB	Updated 30 September, 2020 - 19:03	152967 downloads
Unreal Package Decompressor	Size 99.21 KB	Updated 30 September, 2020 - 19:03	89655 downloads
Batch exporter for UE Viewer (pre-UE4)	Size 2.18 KB	Updated 1 November, 2012 - 13:00	29665 downloads
Skeletal Mesh Editor	Size 2.42 MB	Updated 26 March, 2012 - 10:49	43094 downloads
EpicGames ActorX for 3ds Max/Maya 2012-2016	Size 4.76 MB	Updated 25 December, 2020 - 11:55	48863 downloads

- from there drop the .uasset and .uexp of a file into the folder, and run the .bat you made

Name	Date modified	Type	Size
cg_SteppeFar03.uasset	3/9/2021 8:59 PM	UASSET File	5 KB
cg_SteppeFar03.uexp	3/9/2021 9:30 PM	UEXP File	169 KB
extract.exe	9/30/2020 11:00 AM	Application	188 KB
ExtractMap.bat	1/26/2021 5:43 PM	Windows Batch File	1 KB
ExtractUasset.bat	1/26/2021 5:43 PM	Windows Batch File	1 KB

- It should create a new folder named the same thing as the .uasset. Inside is `NameTable.txt` which is the important one.

Name	Date modified	Type	Size
cg_SteppeFar03.MaterialInstanceConstant	5/26/2021 9:02 PM	MATERIALINSTAN...	169 KB
ExportTable.txt	5/26/2021 9:02 PM	TXT File	1 KB
ImportTable.txt	5/26/2021 9:02 PM	TXT File	1 KB
NameTable.txt	5/26/2021 9:02 PM	TXT File	4 KB

- In the name table, we can search for the name of a value we find in Ftool, for example here's `Fresnel_Color`

```

{
  "ParameterName": "Color_Lerp_Param",
  "ParameterValue": {
    "R": 0.0,
    "G": 0.664659,
    "B": 0.91,
    "A": 0.65,
    "Hex": "A600A9E8"
  },
  "ExpressionGUID": "94748D3A-4D797EA4-FA6B48B9-973E"
},
{
  "ParameterName": "Color_Multiply_Param",
  "ParameterValue": {
    "R": 1.0,
    "G": 1.0,
    "B": 1.0,
    "A": 0.0,
    "Hex": "FFFFFF"
  },
  "ExpressionGUID": "C2A74578-4806731-427296A3-AEC3"
},
{
  "ParameterName": "Fresnel_color",
  "ParameterValue": {
    "R": 0.046001,
    "G": 0.11,
    "B": 0.079315,
    "A": 0.0,
    "Hex": "0C1C14"
  },
  "ExpressionGUID": "D234DAC2-4AE66A70-A77818B9-1E27"
}

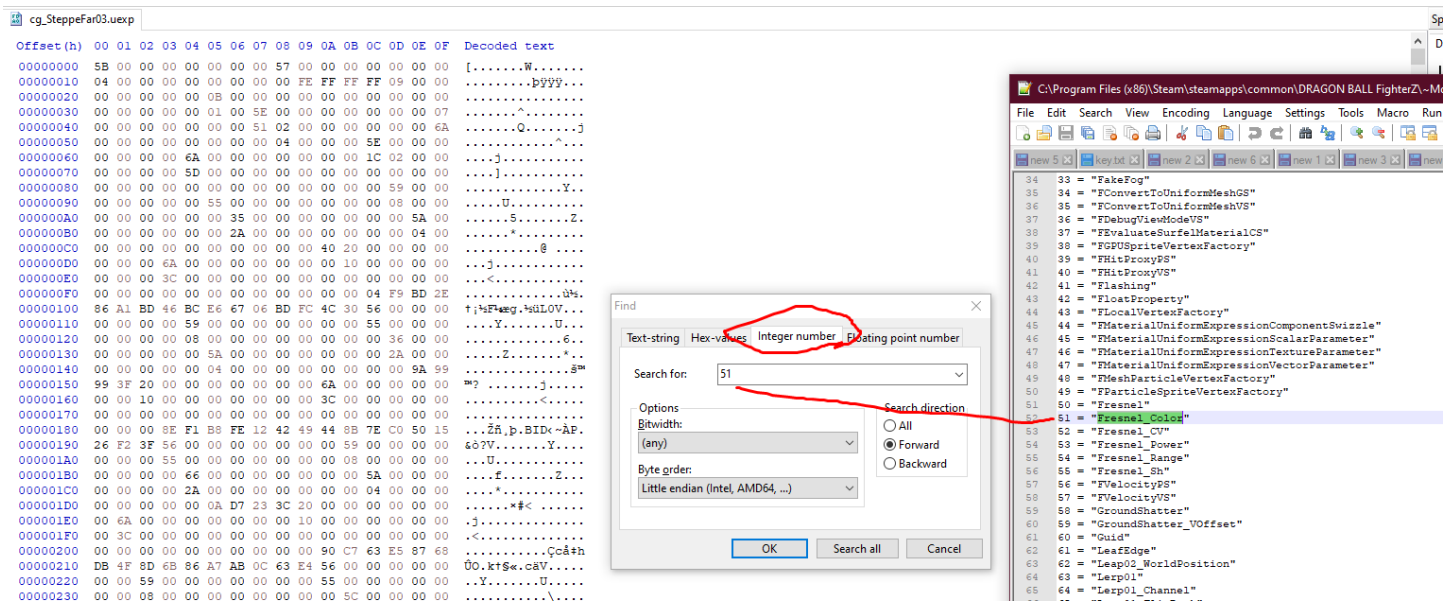
```

```

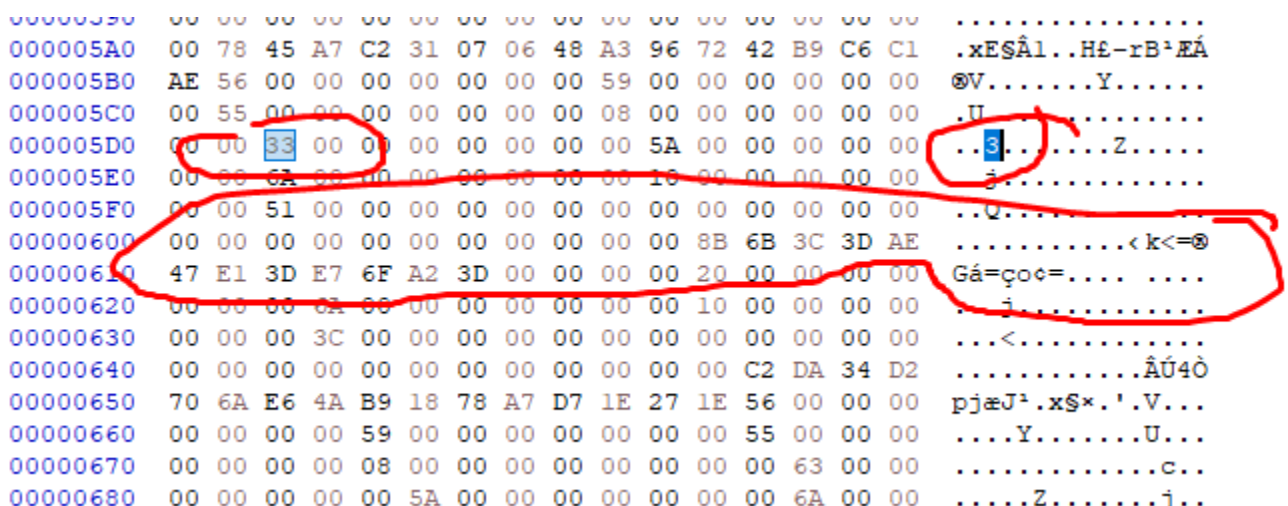
34 33 = "FakeFog"
35 34 = "FConvertToUniformMeshGS"
36 35 = "FConvertToUniformMeshVS"
37 36 = "FDebugViewModeVS"
38 37 = "FEvaluateSurfelMaterialCS"
39 38 = "FGPUSpriteVertexFactory"
40 39 = "FHitProxyPS"
41 40 = "FHitProxyVS"
42 41 = "Flashing"
43 42 = "FloatProperty"
44 43 = "FLocalVertexFactory"
45 44 = "FMaterialUniformExpressionComponentSwizzle"
46 45 = "FMaterialUniformExpressionScalarParameter"
47 46 = "FMaterialUniformExpressionTextureParameter"
48 47 = "FMaterialUniformExpressionVectorParameter"
49 48 = "FMeshParticleVertexFactory"
50 49 = "FParticleSpriteVertexFactory"
51 50 = "Fresnel"
52 51 = "Fresnel_Color"
53 52 = "Fresnel_CV"
54 53 = "Fresnel_Power"
55 54 = "Fresnel_Range"
56 55 = "Fresnel_Sh"
57 56 = "FVelocityPS"
58 57 = "FVelocityVS"
59 58 = "GroundShatter"

```

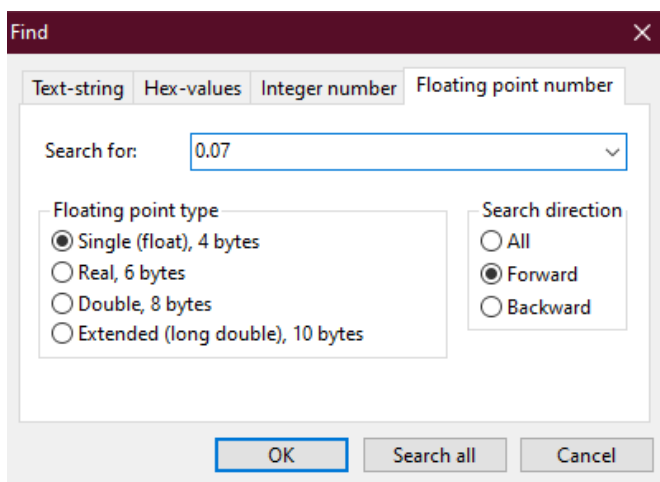
- Search for it's numeric value in Integer Number Search in the .uexp in your favorite hex editor (I used HxD for this example).



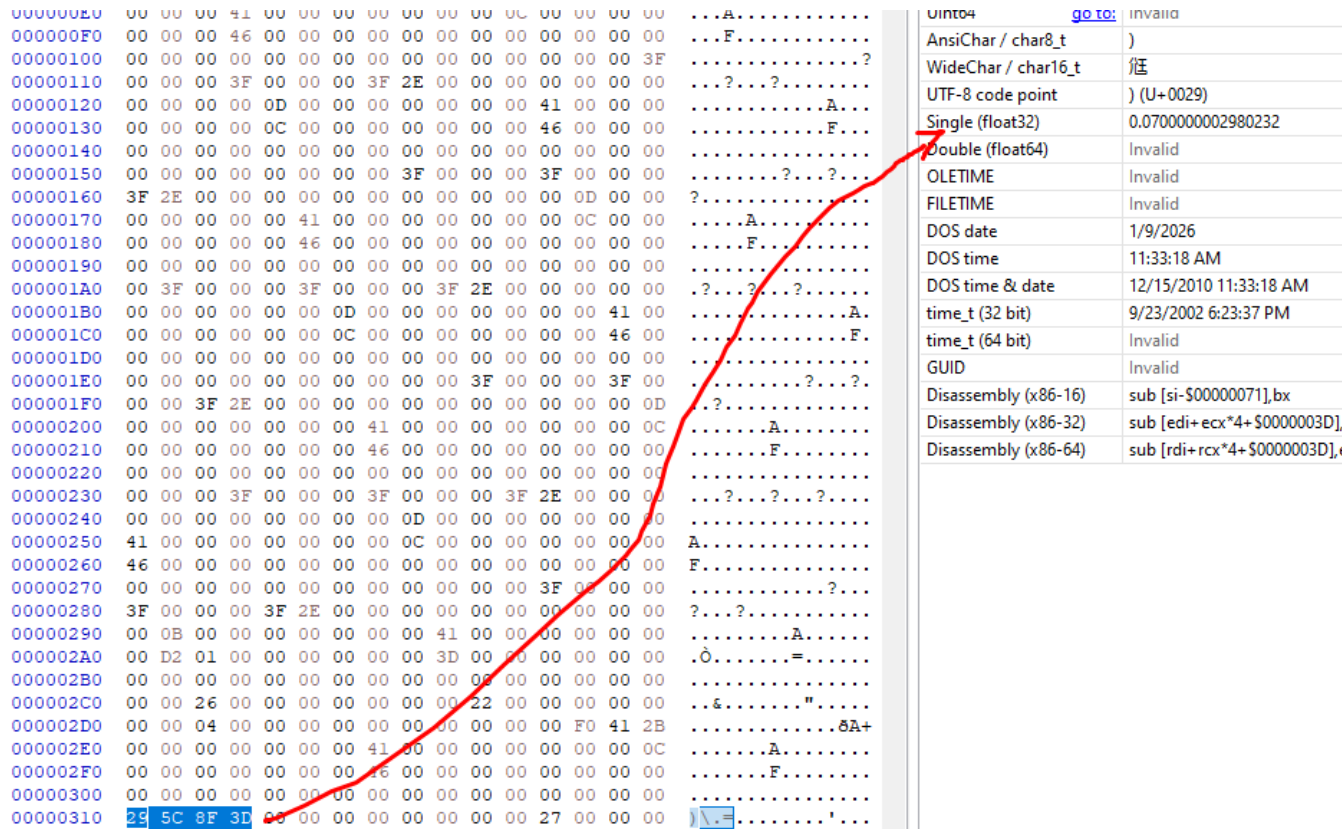
- It should send you to where that value is in the .uexp, and after some more information should be the actual data you want to modify (from here search for the actual float or integer values basically.)



- Don't forget to take advantage of HxD's search window, since you can search for individual float values along with actual integer values that you'd find in Fmodel and the Nametable respectively.



- You can also use the right sidebar to edit the values you selected as other kinds, like floats.



The screenshot shows a memory editor with a list of memory addresses and their contents. A red line highlights a specific memory location, and a red arrow points to the 'Double (float64)' option in the right sidebar.

Address	Content	Comment
000000E0	00 00 00 41 00 00 00 00 00 00 00 00 00 00 00 00	...
000000F0	00 00 00 46 00 00 00 00 00 00 00 00 00 00 00 00	...
00000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3F	...
00000110	00 00 00 3F 00 00 00 3F 2E 00 00 00 00 00 00 00	...
00000120	00 00 00 00 0D 00 00 00 00 00 00 00 41 00 00 00	...
00000130	00 00 00 00 0C 00 00 00 00 00 00 00 46 00 00 00	...
00000140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
00000150	00 00 00 00 00 00 00 00 3F 00 00 00 3F 00 00 00	...
00000160	3F 2E 00 00 00 00 00 00 00 00 00 00 00 0D 00 00	...
00000170	00 00 00 00 00 41 00 00 00 00 00 00 00 0C 00 00	...
00000180	00 00 00 00 00 46 00 00 00 00 00 00 00 00 00 00	...
00000190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
000001A0	00 3F 00 00 00 3F 00 00 00 3F 2E 00 00 00 00 00	...
000001B0	00 00 00 00 00 00 0D 00 00 00 00 00 00 41 00 00	...
000001C0	00 00 00 00 00 00 0C 00 00 00 00 00 00 46 00 00	...
000001D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
000001E0	00 00 00 00 00 00 00 00 3F 00 00 00 3F 00 00 00	...
000001F0	00 00 3F 2E 00 00 00 00 00 00 00 00 00 00 0D 00	...
00000200	00 00 00 00 00 00 41 00 00 00 00 00 00 00 0C 00	...
00000210	00 00 00 00 00 00 46 00 00 00 00 00 00 00 00 00	...
00000220	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
00000230	00 00 00 3F 00 00 00 3F 00 00 00 3F 2E 00 00 00	...
00000240	00 00 00 00 00 00 00 00 0D 00 00 00 00 00 00 00	...
00000250	41 00 00 00 00 00 00 00 0C 00 00 00 00 00 00 00	...
00000260	46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
00000270	00 00 00 00 00 00 00 00 00 00 00 00 3F 00 00 00	...
00000280	3F 00 00 00 3F 2E 00 00 00 00 00 00 00 00 00 00	...
00000290	00 0B 00 00 00 00 00 00 41 00 00 00 00 00 00 00	...
000002A0	00 D2 01 00 00 00 00 00 3D 00 00 00 00 00 00 00	...
000002B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
000002C0	00 00 26 00 00 00 00 00 00 00 22 00 00 00 00 00	...
000002D0	00 00 04 00 00 00 00 00 00 00 00 00 00 F0 41 2B	...
000002E0	00 00 00 00 00 00 00 41 00 00 00 00 00 00 0C 00	...
000002F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
00000300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	...
00000310	28 5C 8F 3D 00 00 00 00 00 00 00 00 27 00 00 00	...

The right sidebar shows a list of data types and their corresponding values:

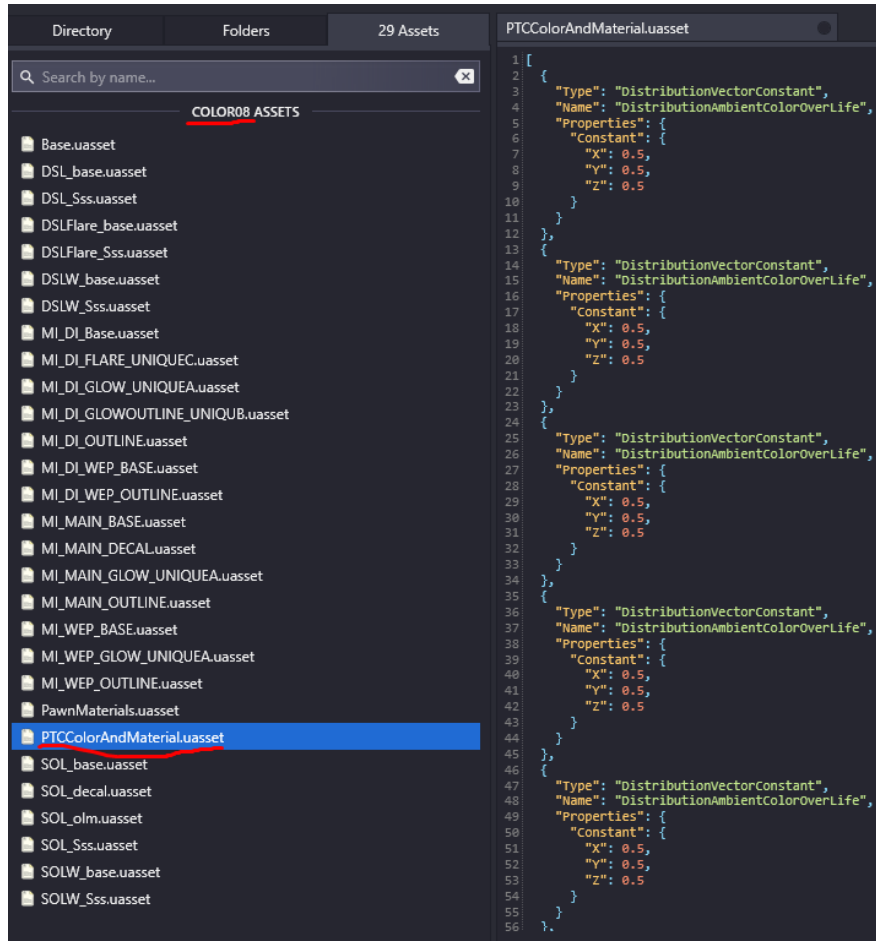
Type	Value
UInt04	Invalid
AnsiChar / char8_t	)
WideChar / char16_t	経
UTF-8 code point	) (U+0029)
Single (float32)	0.0700000002980232
Double (float64)	Invalid
OLETIME	Invalid
FILETIME	Invalid
DOS date	1/9/2026
DOS time	11:33:18 AM
DOS time & date	12/15/2010 11:33:18 AM
time_t (32 bit)	9/23/2002 6:23:37 PM
time_t (64 bit)	Invalid
GUID	Invalid
Disassembly (x86-16)	sub [si-\$00000071],bx
Disassembly (x86-32)	sub [edi+ecx*4+\$0000003D],
Disassembly (x86-64)	sub [rdi+rcx*4+\$0000003D],

- After modifying the uexp you should be able to pack up and test in game.

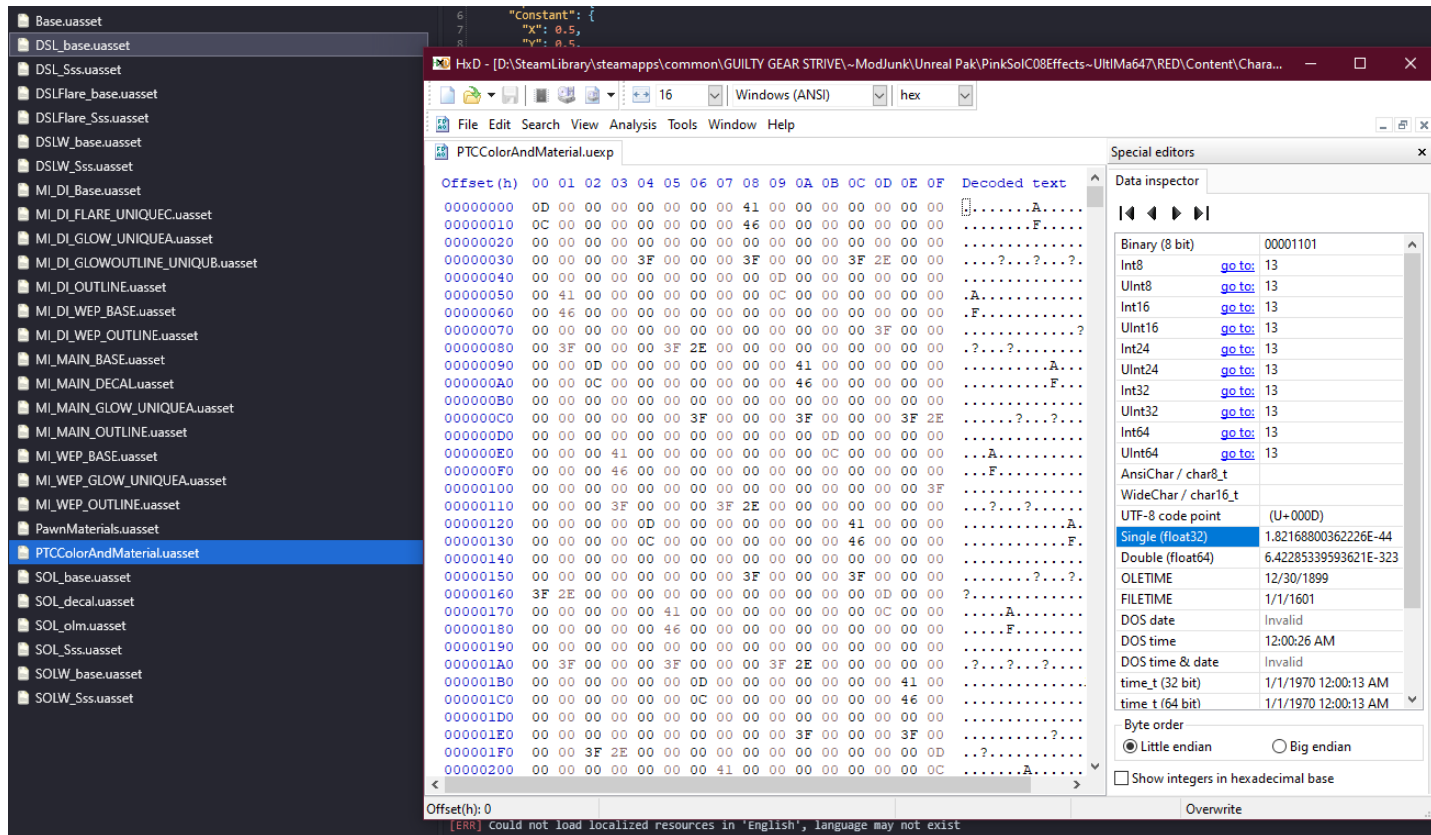
GG Strive Per-Color Effects

NOTE: Although almost everyone has the file for it, Sol and Ky by far have the most in-depth files for this due to their EX colors.

- Find the **PTCCColorAndMaterial** file in the character's color folder that you want to edit in FModel.
 - For this example I'm going to be making Sol's flames pink for his Color08.



- In Umodel, find the same file and save it instead of exporting it. Open the .uexp of the file in a hex editor (again I'll be using HxD).



- In FModel, scroll down to the first **ParticleModuleColorOverLife** section. We only need to worry about this section for every particle.

```

101 {
102   "Type": "ParticleModuleColorOverLife",
103   "Name": "ParticleModuleColorOverLife_0",
104   "Properties": {
105     "ColorOverLife": {
106       "MaxValue": 30.0,
107       "MinValueVec": {
108         "X": 0.07,
109         "Y": 0.0,
110         "Z": 0.0
111       },
112       "MaxValueVec": {
113         "X": 30.0,
114         "Y": 2.0,
115         "Z": 0.2
116       },
117       "Distribution": null,
118       "Table": {
119         "TimeScale": 9.999998,
120         "TimeBias": 0.5,
121         "Values": [
122           30.0,
123           2.0,
124           0.2,
125           0.07,
126           0.0,
127           0.0
128         ],
129         "Op": 1,
130         "EntryCount": 2,
131         "EntryStride": 3

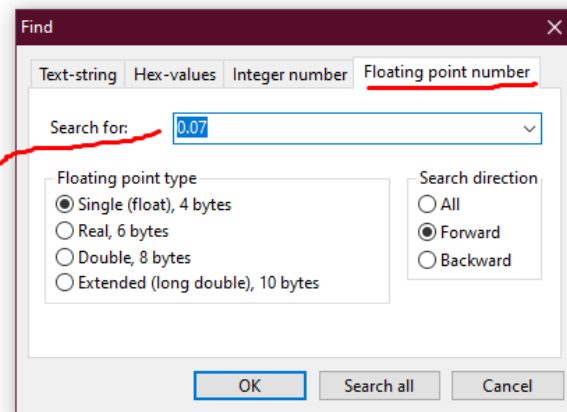
```

- Since we're looking at float values here, we will need to search for **0.07** as a float value in HxD. It should bring us to the first **MinValueVec** value, which is split up into it's RGB components.

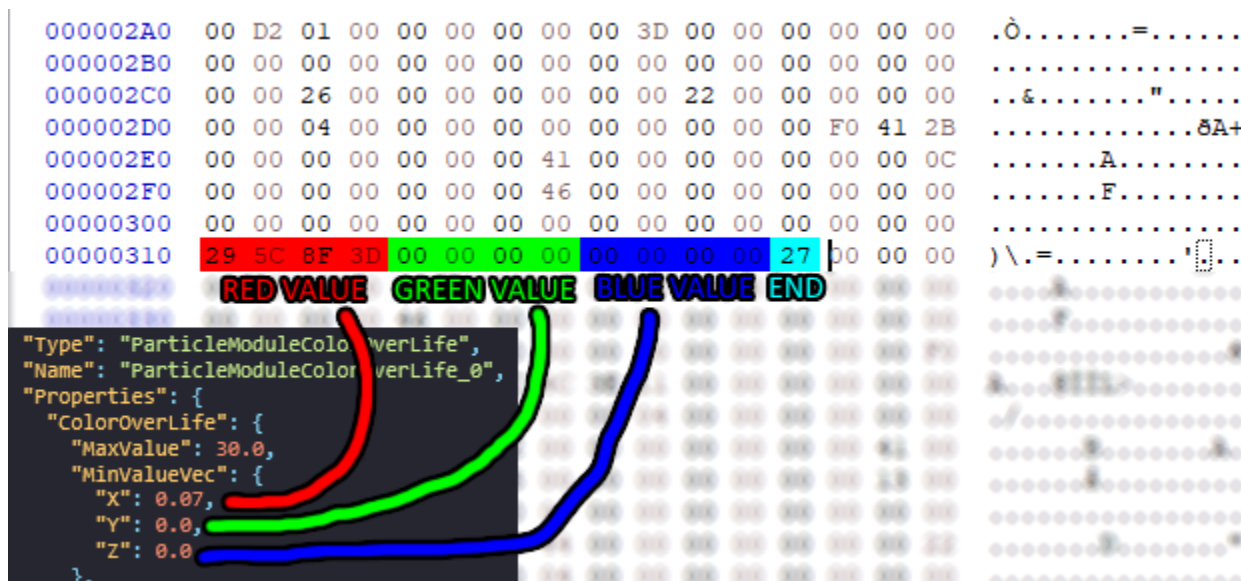
```

000001D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000001E0 00 00 00 00 00 00 00 00 00 00 3F 00 00 00 3F 00 .....?..?.
000001F0 00 00 3F 2E 00 00 00 00 00 00 00 00 00 00 00 0D ..?..?..
00000200 00 00 00 00 00 00 00 00 41 00 00 00 00 00 00 0C .....A.....
00000210 00 00 00 00 00 00 00 00 46 00 00 00 00 00 00 00 .....F.....
00000220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....F.....
00000230 00 00 00 3F 00 00 00 00 3F 00 00 00 3F 2E 00 00 00 .....?..?..?..
00000240 00 00 00 00 00 00 00 00 0D 00 00 00 00 00 00 00 .....
00000250 41 00 00 00 00 00 00 00 0C 00 00 00 00 00 00 00 A.....
00000260 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F.....
00000270 00 00 00 00 00 00 00 00 00 00 00 00 3F 00 00 00 ?..?..?..?..
00000280 3F 00 00 00 3F 2E 00 00 00 00 00 00 00 00 00 00 ?..?..?..?..
00000290 00 0B 00 00 00 00 00 00 00 00 41 00 00 00 00 00 .....A.....
000002A0 00 D2 01 00 00 00 00 00 00 00 3D 00 00 00 00 00 .....ò.....=.....
000002B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000002C0 00 00 26 00 00 00 00 00 00 00 22 00 00 00 00 00 .....&.....".....
000002D0 00 00 04 00 00 00 00 00 00 00 00 00 F0 41 2B .....8A+
000002E0 00 00 00 00 00 00 00 00 41 00 00 00 00 00 00 0C .....A.....
000002F0 00 00 00 00 00 00 00 00 46 00 00 00 00 00 00 00 .....F.....
00000300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000310 29 5C 8F 3D 00 00 00 00 00 00 00 00 27 00 00 00 ) \ . = ..... ' 0
00000320 00 00 00 00 41 00 00 00 00 00 00 00 0C 00 00 00 .....A.....
00000330 00 00 00 00 46 00 00 00 00 00 00 00 00 00 00 00 .....F.....

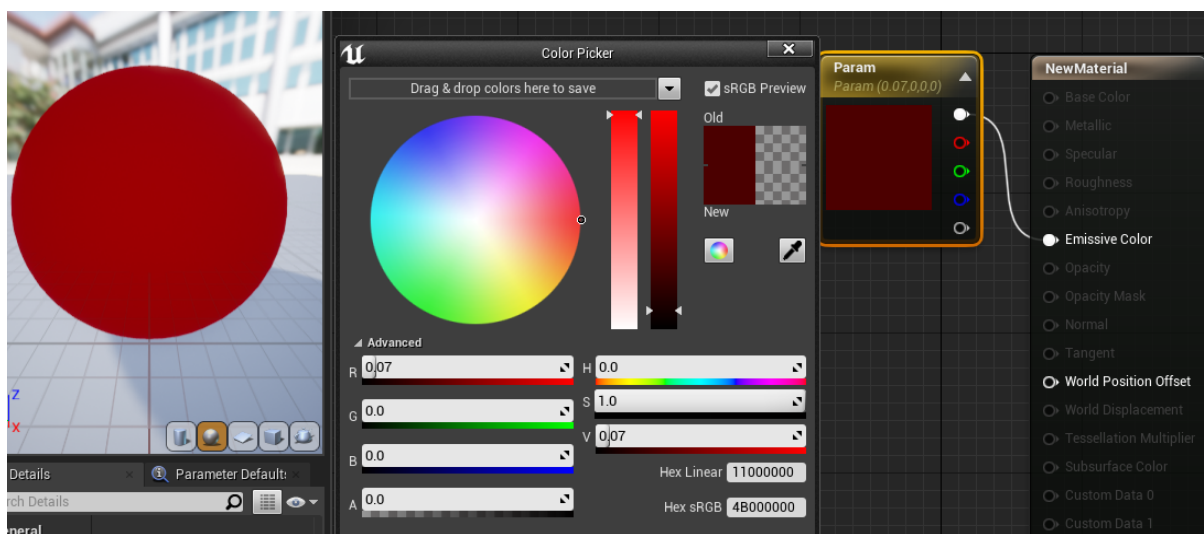
```



- We know this is the float value we want, because after the **0.07** byte is two **0.0** float values, which matches up with FModel. Keep this in mind for any other particle color you edit, since they'll always have their color values right next to each other.



- [OPTIONAL] If you want a better idea of what your color is going to look like, you can create a temporary unlit material in UE4 that you can input colors to. It'll also be nice to find the colors you want to actually put in game.



- It's also nice to figure out how to set your Min and Max colors, as the actual effects kinda blend the two together.



- Once you're ready to edit a float value, you can use the bar on the right to edit it easier. For example I'm changing the blue value here to 0.055 (HxD rounds it oddly but it'll stick to whatever you set it to).

00000280	3F 00 00 00 3F 2E 00 00 00 00 00 00 00 00 00 00	?...?.....	WideChar / char16_t	越
00000290	00 0B 00 00 00 00 00 00 00 41 00 00 00 00 00 00	A.....	UTF-8 code point	Unexpected continuation byte
000002A0	00 D2 01 00 00 00 00 00 00 3D 00 00 00 00 00 00	.Ö.....=.....	Single (float32)	0.0549999997019768
000002B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		Double (float64)	Invalid
000002C0	00 00 26 00 00 00 00 00 00 00 00 00 00 00 00 00	...&.....".....	OLETIME	Invalid
000002D0	00 00 04 00 00 00 00 00 00 00 00 00 00 F0 41 2B	8A+	FILETIME	Invalid
000002E0	00 00 00 00 00 00 00 00 41 00 00 00 00 00 00 00	A.....	DOS date	Invalid
000002F0	00 00 00 00 00 00 00 00 46 00 00 00 00 00 00 00	F.....	DOS time	Invalid
00000300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		DOS time & date	Invalid
00000310	29 5C 8F 3D 00 00 00 00 AE 47 61 3D 27 00 00 00	)\.=.....@Ga='...	time_t (32 bit)	8/19/2002 7:31:58 PM
00000320	00 00 00 00 41 00 00 00 00 00 00 00 00 00 00 00	A.....	time_t (64 bit)	Invalid
00000330	00 00 00 00 46 00 00 00 00 00 00 00 00 00 00 00	F.....		
00000340	00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0	8		
00000350	41 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00			

- After that move onto the **MaxValueVec** and do the same thing (it starts with 30.0 so I searched for that first). For this example I changed it from 30.0, 2.0, 0.2 to 30.0, 12.8, 23.0.
 - I highlighted the end bytes here (27 and 11), but make sure not to edit those since they're important for the .uexp.

000002D0	00 00 04 00 00 00 00 00 00 00 00 00 00 00 F0 41 2B	8A+
000002E0	00 00 00 00 00 00 00 41 00 00 00 00 00 00 00 00 0C	A.....
000002F0	00 00 00 00 00 46 00 00 00 00 00 00 00 00 00 00	F.....
00000300	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00000310	29 5C 8F 3D 00 00 00 00 AE 47 61 3D 27 00 00 00	)\.=.....@Ga='...
00000320	00 00 00 00 41 00 00 00 00 00 00 00 00 00 00 00	A.....
00000330	00 00 00 00 46 00 00 00 00 00 00 00 00 00 00 00	F.....
00000340	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0	8
00000350	41 CD CC 4C 41 00 00 B8 41 11 00 00 00 00 00 00	AfLA...A.[].....
00000360	00 2F 00 00 00 00 00 00 00 00 04 00 00 00 00 00	./.....
00000370	00 00 00 00 00 00 00 42 00 00 00 00 00 00 00 41 00	B.....A.
00000380	00 00 00 00 00 00 00 E5 00 00 00 00 00 00 00 13 00	â.....
00000390	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
000003A0	00 00 00 00 00 00 00 44 00 00 00 00 00 00 00 22	n....."

- The last thing we need to change for a particle is the Values array. All we need to do is copy-paste our values over it (this time it starts with Max instead of Min, so keep that in mind).

```

Distribution: null,
"Table": {
  "TimeScale": 9.999998,
  "TimeBias": 0.5,
  "Values": [
    30.0,
    2.0,
    0.2,
    0.07,
    0.0,
    0.0
  ],
  "Op": 1,
  "EntryCount": 2,
  "EntryStride": 3
}

```

- The last thing to keep in mind is the `MaxValue` and `MinValue` variables. **ONLY CHANGE THESE IF YOU HAVE EXCEEDED THE CURRENT VALUE IN ANY SLOT.**
 - For example, if one of your values is now 64 rather than 55, be sure to change the `MaxValue` to 64. Same should be done for `MinValue` if it does exist and you changed it, simply make it the new lowest value.

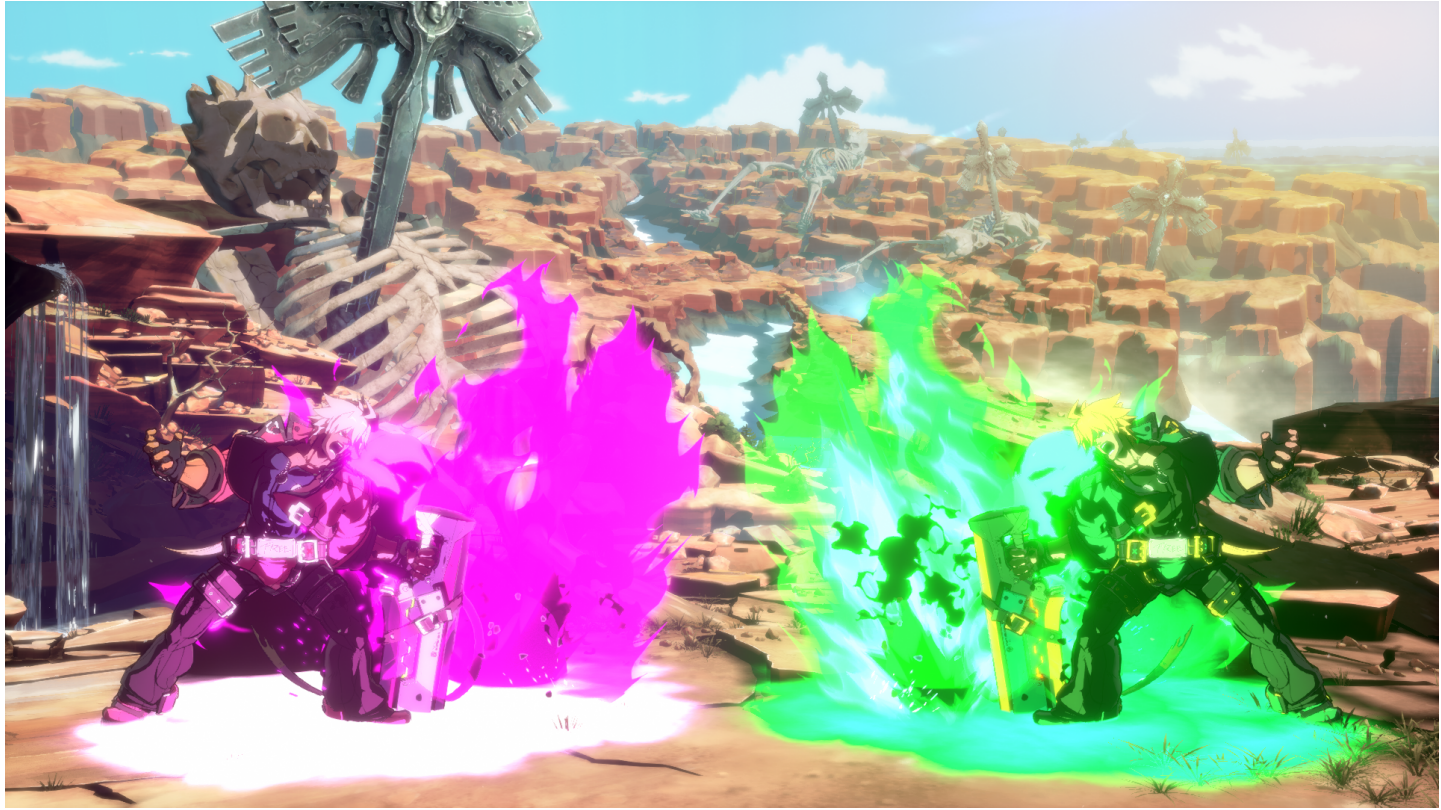
```

"Type": "ParticleModuleColorOverLife",
"Name": "ParticleModuleColorOverLife_1",
"Properties": {
  "ColorOverLife": {
    "MaxValue": 55.0,
    "MinValueVec": {
      "X": 0.07,
      "Y": 0.0,
      "Z": 0.0
    },
    "MaxValueVec": {
      "X": 55.0,
      "Y": 2.0,
      "Z": 0.2
    }
  },
  "Distribution": null,
  "Table": {
    "TimeScale": 19.999996,
    "TimeBias": 0.65,
    "Values": [
      55.0,
      2.0,
      0.2,
      0.07,
      0.0,
      0.0
    ]
  }
}

```

- From there it should just be rinsing and repeating until you've done everyone in the file. Sol and Ky obviously have the most in theirs so they'll take the longest out of everyone, but it's largely a game of paying attention to where you are and not getting lost.

Final Product:



- If you want the pink Sol effects used in the example you can grab them here (ill probably fix them later lmao):

https://drive.google.com/file/d/1bdwynZSWE8tGjHsiVYyYj_PPSE9ZiqIX/view?usp=sharing