

# Particle Tool

by Atlas and ImBee

Manual

## **What is Particle Tool By Atlas and ImBee and what is it used for?**

**Particle Tool By Atlas and ImBee** is a package of two programs used to convert particle files from **Reality Pump** games in order to enable easier editing and running of the same particle in different games from this studio. This package of programs is a “bridge” between different particle formats. These programs support the following games/titles:

- Earth 2150: Escape from the Blue Planet,
- Earth 2150: The Moon Project,
- Earth 2150: Lost Souls,
- World War III: Black Gold,
- Heli Heroes,
- Frontline Attack: War over Europe/World War II: Panzer Claws II,
- Polanie II/KnightShift/Once Upon a Knight,
- Polanie III/KnightShift II Curse of Souls
- Earth 2160,
- 3D ParticleGen Visual FX (steam),
- Two Worlds,

**The package consists of two programs:**

- Particle2MyAod.exe
- MyAod2Particle.exe

### Particle2MyAod:

**Particle2MyAod.exe** is a program used to export data from particles in **.msh** and **.prt** formats from the titles listed above to **.myaod** format, which is very similar to **.aod**. Simply enter the name of the particle file along with the format.

### File with ParticleEmitter:

If ParticleEmitter is detected, the program will ask you to specify the final format. You can choose **y/n**. If you choose **n**, the program will extract the particle file to **.myaod** without specifying a specific format, i.e., the program will simply extract the particle. If you enter **y**, you will be able to force the format to which **.myaod** will later be compiled (e.g., if you want to run particles from **KnightShift** in **Earth 2160**, enter **y**, and then force the **e2160** format. The **MyAod2Particle** program will then know how to compile **.myaod** to the **Earth 2160** format). The program will then extract the particle file.

The exported particles folder contains a **.myaod** file and an **\_extra\_data.cpp** file. The **\_extra\_data** file contains the entire beginning of the particles file in the form of specified variable values. You can modify this, but you don't have to. The most important file is **.myaod**.

The program also works in **argc&argv** mode, so you can call it from **cmd** or **powershell** and enter the argument next to the program name, which will export the particles without forcing the format:

**Particle2MyAod.exe <input\_file\_name>**

You can also enforce the correct format in the following way:

**Particle2MyAod.exe <input\_file\_name> -force <ks/tw/pg/e2160>**

or

**Particle2MyAod.exe <input\_file\_name> --force <ks/tw/pg/e2160>**

### File with Dynamic particle:

Unfortunately, Dynamic particles are handled by editing the **DynamicParticle.cfg** configuration file. Depending on the title from which the particles originate, this configuration file must be modified – preferably immediately. The following options are set by default, which mean:

*dynamic\_particle\_input\_format = ks;*

**(The input Dynamic particle will come from KnightShift.)**

*force\_specific\_export\_format = false;*

**(Should a specific format be enforced?)**

*forced\_export\_format = ks;*

**(The final output format for Dynamic particles will be the format used in the game KnightShift.)**

To handle a specific format from another game where there are dynamic particles, simply enter one of these formats: **e2150**, **ww3\_or\_hh**, **ww2\_or\_fa\_or\_pc2**, **ks**

in this place after the sign =

***dynamic\_particle\_input\_format = <enter the format here>;***

Aby konwertować do innej gry należy odblokować przełącznik ***force\_specific\_export\_format*** na ***true***.

The final format to which we are converting should be entered in ***forced\_export\_format*** after the = sign.

***forced\_export\_format*** = <enter the format here>;

The compiler should already know how to compile a directory with exported particles.

I created my own special compiler, **MyAod2Particle.exe**, to compile the **.myaod** format.

To automatically extract all files without forcing a format, you can use **\_EXPORT\_ALL.ps1** or **\_EXPORT\_ALL.bat**.

### **MyAod2Particle:**

**MyAod2Particle.exe** is a program used to compile **.myaod** particle files into a predefined format from a given **Reality Pump** game.

The program is launched by double-clicking and entering the name of the input directory. If the **extra\_data** file in the directory does not exist, the program will create a classic particles header by generating the appropriate information, but if the **extra\_data** file exists, it will recompile this file into a particles header.

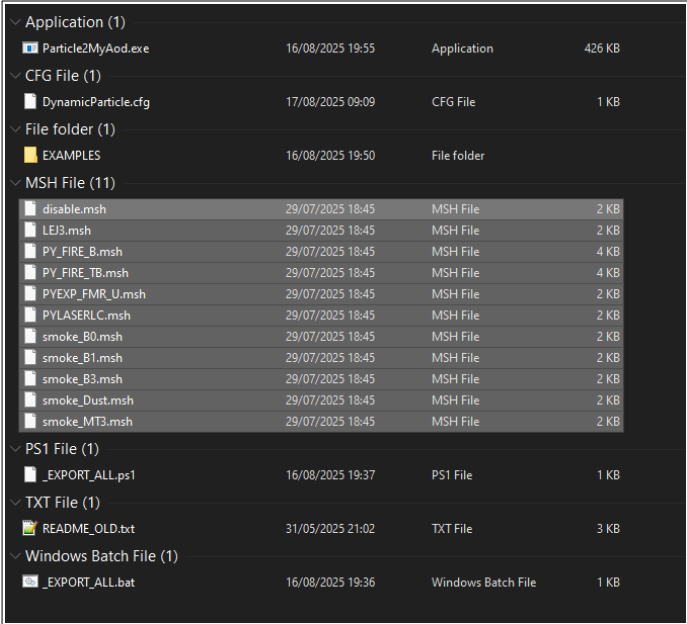
The program also works in **argc&argv** mode, so you can call it from cmd or powershell and enter the argument next to the program name, i.e.

**MyAod2Particle.exe** <input directory name>

To automatically compile/import all folders, you can use **\_IMPORT\_ALL.ps1** or **\_IMPORT\_ALL.bat**.

### **Typical and example scenarios for using ParticleTool:**

a) I want to export Dynamic particles from Earth 2150:



Application (1)			
Particle2MyAod.exe	16/08/2025 19:55	Application	426 KB
CFG File (1)			
DynamicParticle.cfg	17/08/2025 09:09	CFG File	1 KB
File folder (1)			
EXAMPLES	16/08/2025 19:50	File folder	
MSH File (11)			
disable.msh	29/07/2025 18:45	MSH File	2 KB
LEJ3.msh	29/07/2025 18:45	MSH File	2 KB
PV_FIRE_B.msh	29/07/2025 18:45	MSH File	4 KB
PV_FIRE_TB.msh	29/07/2025 18:45	MSH File	4 KB
PVEXP_FMR_U.msh	29/07/2025 18:45	MSH File	2 KB
PYLASERLC.msh	29/07/2025 18:45	MSH File	2 KB
smoke_B0.msh	29/07/2025 18:45	MSH File	2 KB
smoke_B1.msh	29/07/2025 18:45	MSH File	2 KB
smoke_B3.msh	29/07/2025 18:45	MSH File	2 KB
smoke_Dust.msh	29/07/2025 18:45	MSH File	2 KB
smoke_MT3.msh	29/07/2025 18:45	MSH File	2 KB
PS1 File (1)			
_EXPORT_ALL.ps1	16/08/2025 19:37	PS1 File	1 KB
TXT File (1)			
README_OLD.txt	31/05/2025 21:02	TXT File	3 KB
Windows Batch File (1)			
_EXPORT_ALL.bat	16/08/2025 19:36	Windows Batch File	1 KB

Figure 1: For example, I have a pool of particles from E2150.

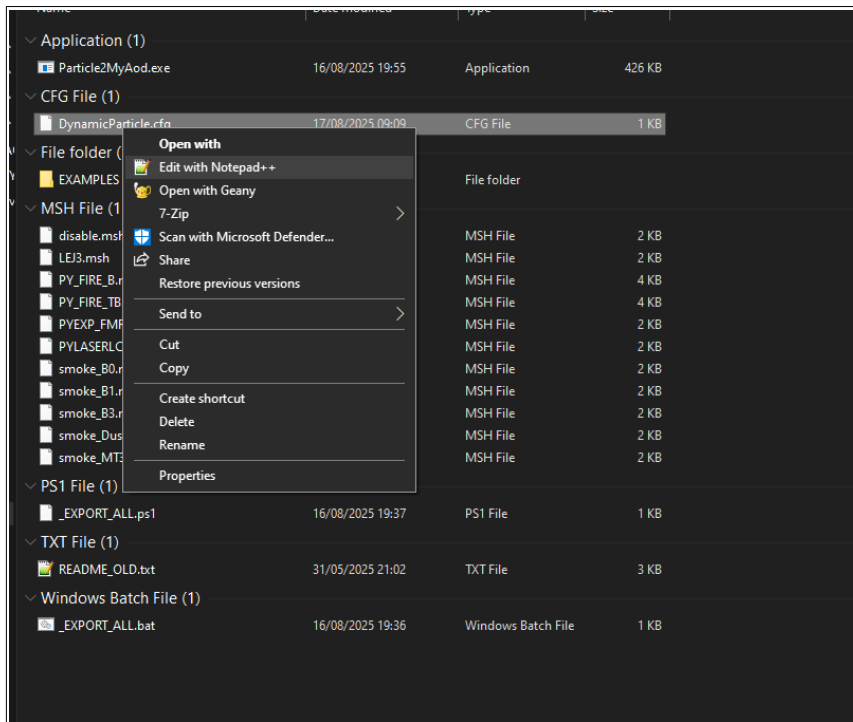


Figure 2: I am editing *DynamicParticle.cfg*.

```
// Dynamic Particle Config:
dynamic_particle_input_format = ks;
force_specific_export_format = false;
forced_export_format = ks;
// Available formats:
// e2150
// ww3_or_hh
// ww2_or_fa_or_pc2
// ks
```

Figure 3: I see options.

```
// Dynamic Particle Config:
dynamic_particle_input_format = e2150;
force_specific_export_format = false;
forced_export_format = e2150;
// Available formats:
// e2150
// ww3_or_hh
// ww2_or_fa_or_pc2
// ks
```

Figure 4: I change the format to the appropriate input format.

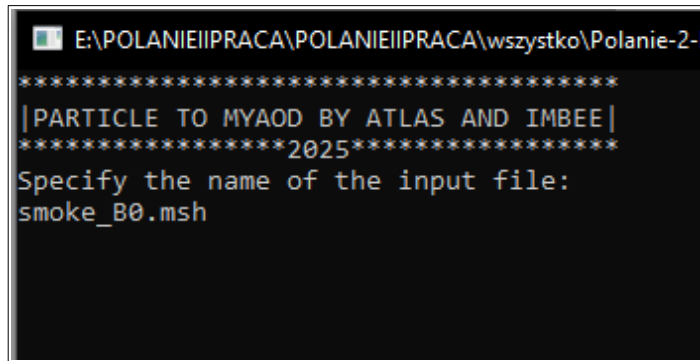


Figure 5: I open the program and enter the file name.  
Then I press enter.

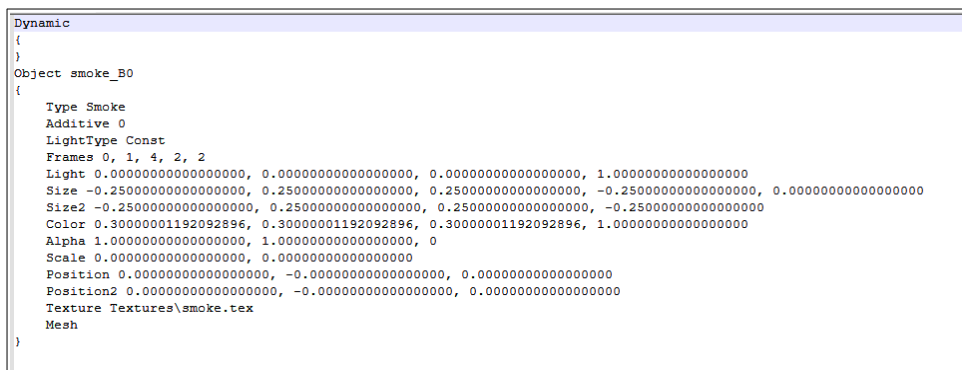


Figure 6: Done—I can edit Dynamic Particles data.

b) I want to convert Dynamic particles from Earth 2150 to World War III Black Gold:

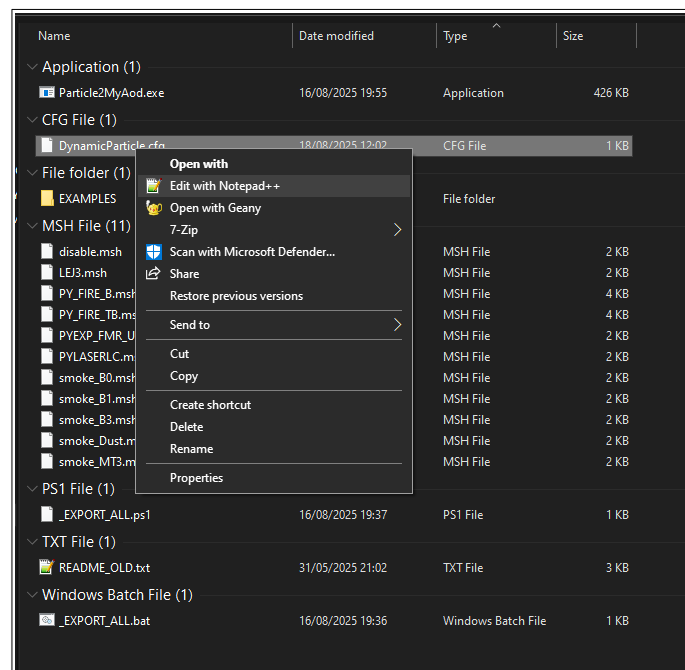


Figure 7: I am editing DynamicParticle.cfg.

```
// Dynamic Particle Config:
dynamic_particle_input_format = e2150;
force_specific_export_format = false;
forced_export_format = e2150;
// Available formats:
// e2150
// ww3_or_hh
// ww2_or_fa_or_pc2
// ks
```

Figure 8: I notice the options.

```
// Dynamic Particle Config:
dynamic_particle_input_format = e2150;
force_specific_export_format = true;
forced_export_format = ww3_or_hh;
// Available formats:
// e2150
// ww3_or_hh
// ww2_or_fa_or_pc2
// ks
```

Figure 9: I set the input format to e2150, unlock force\_specific\_export\_format to true, and set forced\_export\_format to ww3\_or\_hh.

```
E:\POLANIE\PRACA\POLANIE\PRACA\wszystko\Polanie-2-Da
*****
|PARTICLE TO MYAOD BY ATLAS AND IMBEE|
*****2025*****
Specify the name of the input file:
LEJ3.msh
```

Figure 10: I open the program, enter the name of the particle, and press enter.

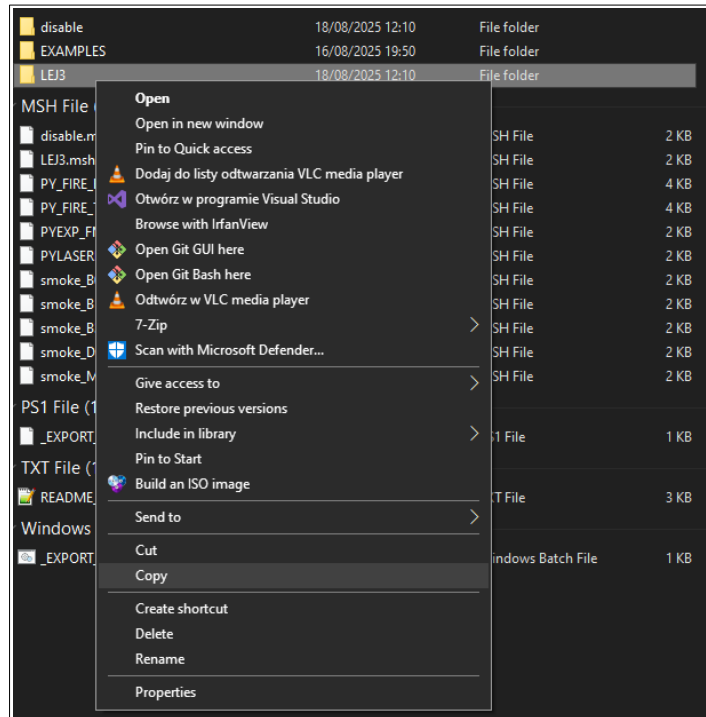


Figure 11: I copy the received directory with the exported particles to the directory from MyAod2Particle.

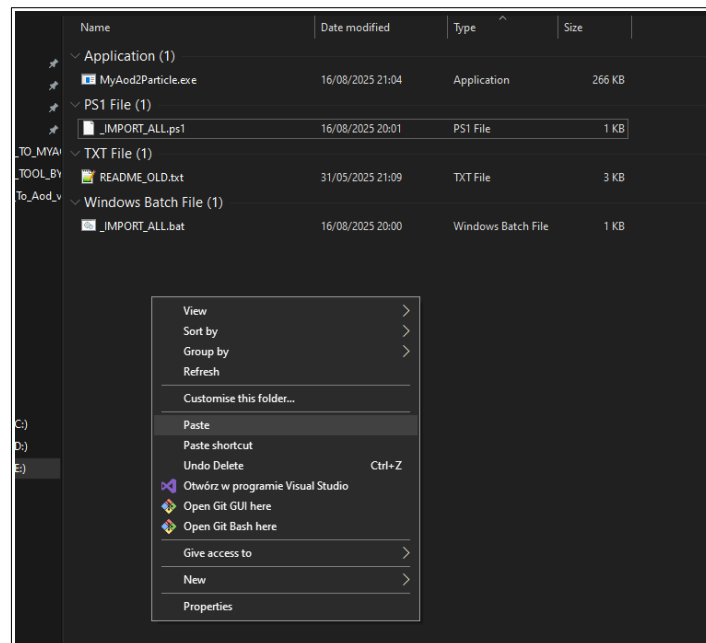
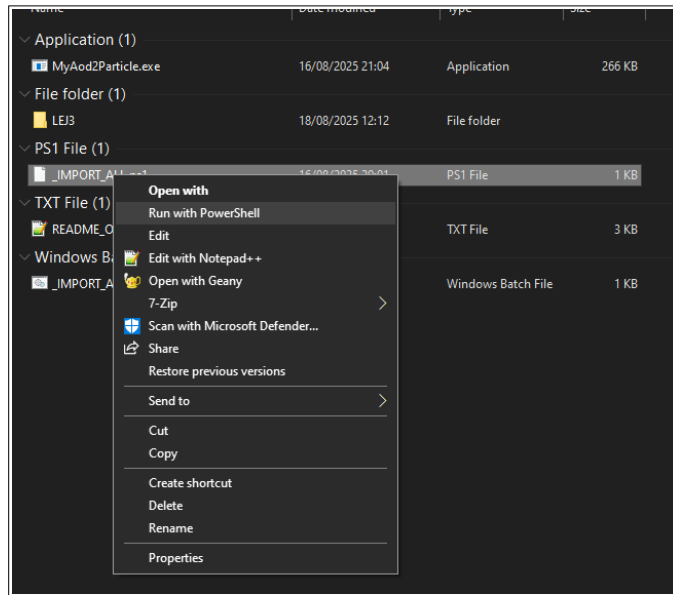
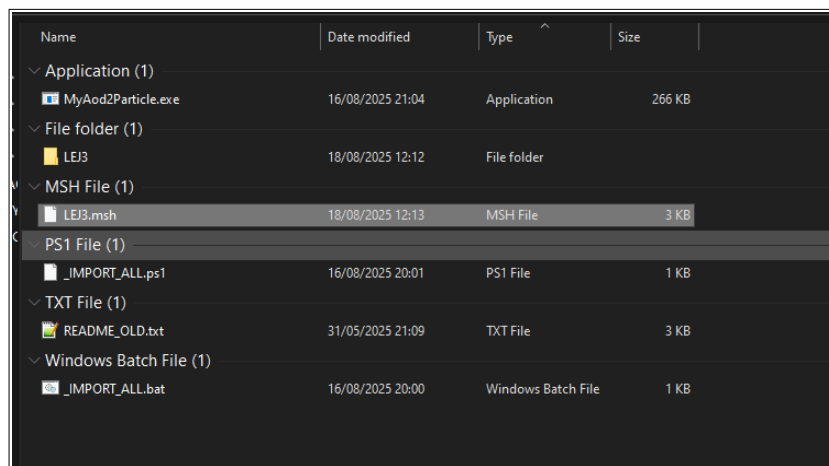


Figure 12: I'll paste it here.





*Figure 13: We can launch the program by double-clicking and entering the name of the directory to be compiled, or we can launch the `_IMPORT_ALL.ps1` script via PowerShell to compile all directories with particle data.*



*Figure 14: Done - Particles is ready to be added to the game.*

c) I want to convert ParticleEmitter from KnightShift to Earth 2160:

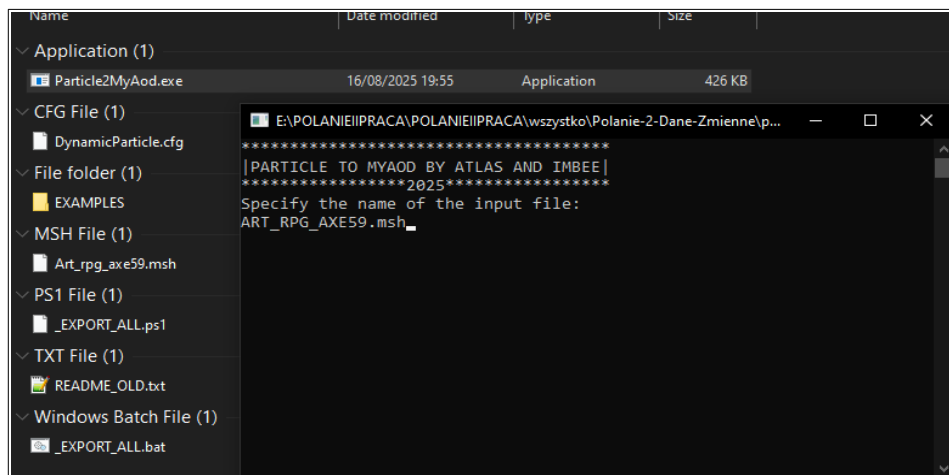


Figure 15: I double-click Particle2MyAod.exe and enter the name of the .msh file.

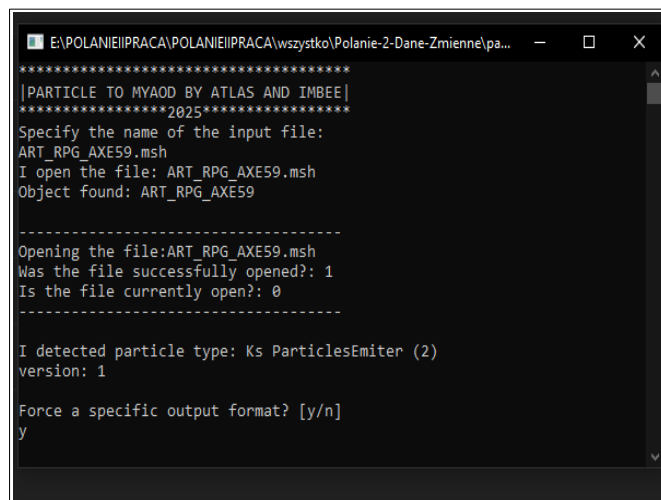


Figure 16: I am introducing y.

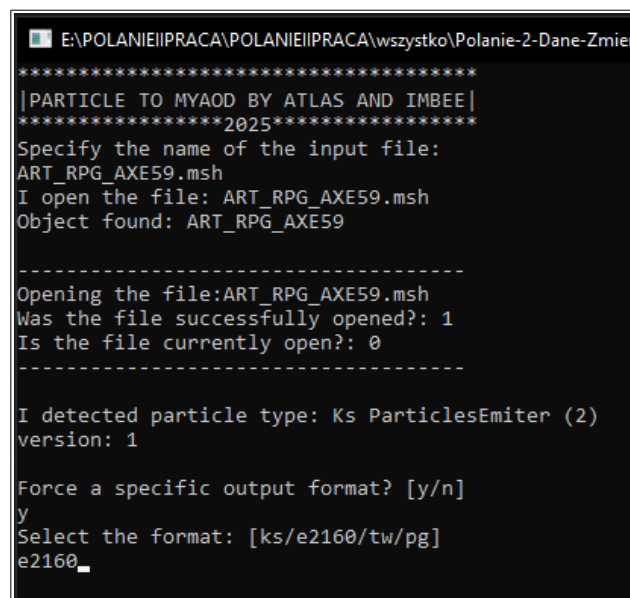


Figure 17: I am forcing the e2160 format.

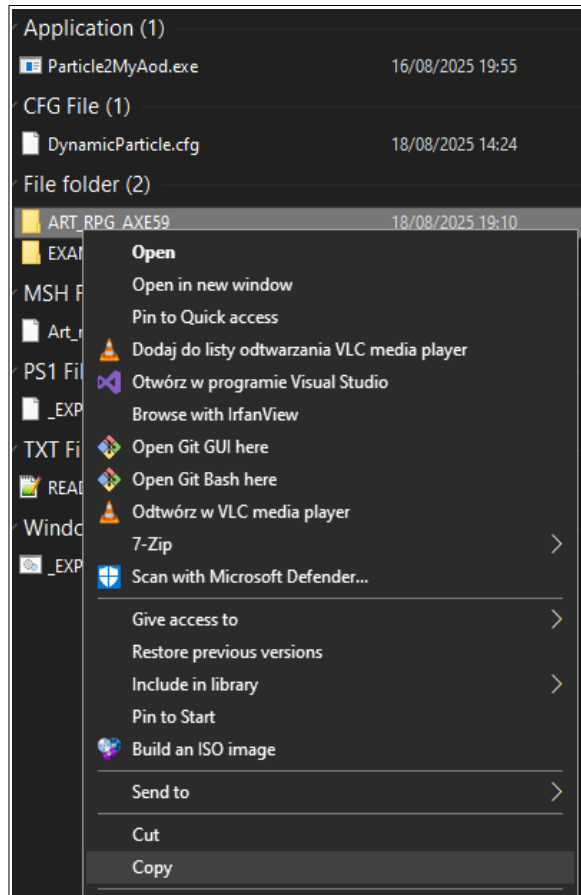


Figure 18: I copy the directory with particlesa data to the directory with MyAod2Particle.

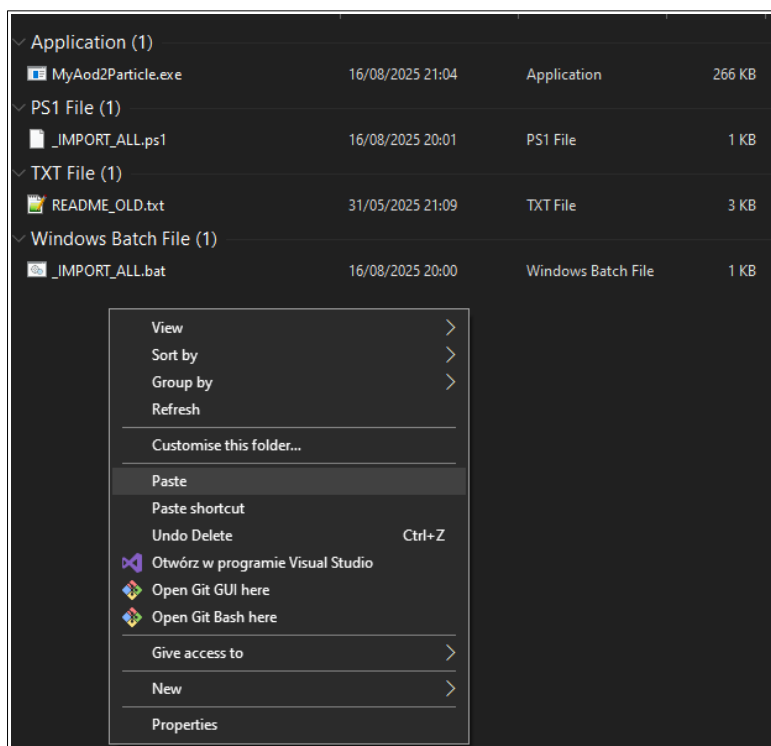


Figure 19: I'm pasting it.

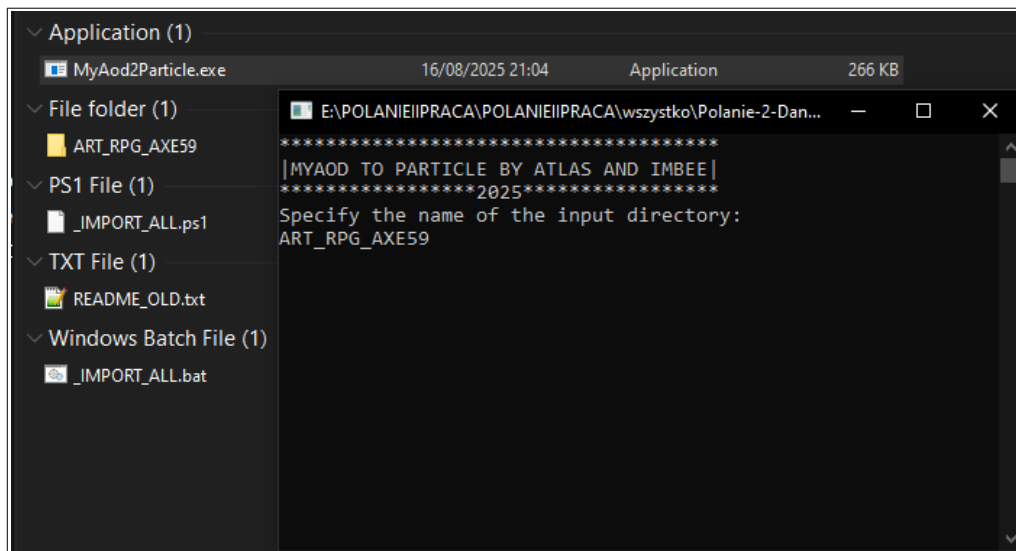


Figure 20: I double-click to open the program and enter the name of the directory containing the particles data.

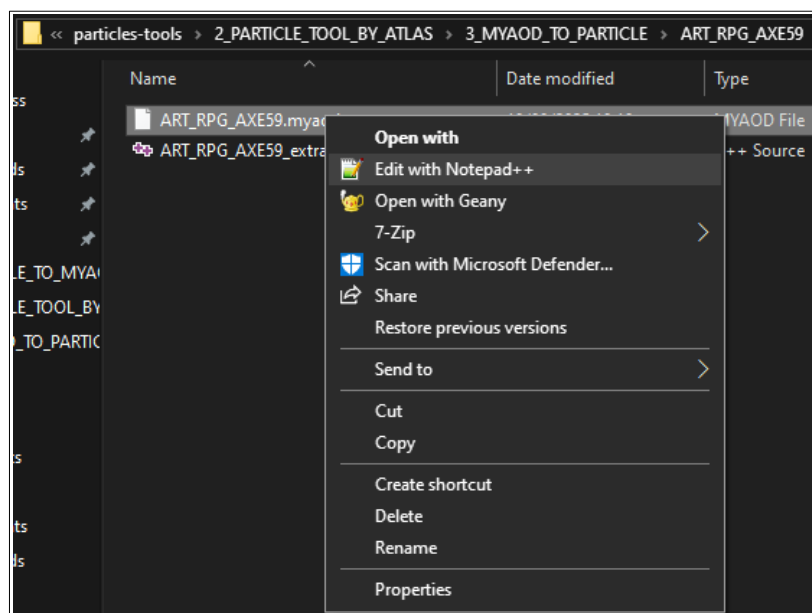


Figure 21: I go to the directory with the extracted particles data and check the contents of the **.myaod** file.

```

3418 {
3419     Emitter E4
3420     EmitterIndex 3
3421     Particle P4
3422     ParticleIndex 3
3423     annotation
3424     time 0.0000000000000000
3425     loopedEmission 2, 1, 0, 0.0000000000000000
3426     drawParticleEmitter 1
3427     simpleOneParticleSwitch 0
3428     2Dmask 0
3429     hardwareCursor 0
3430     stopInPartialPause 0
3431     finishMissile 0
3432     emitsGroupsSwitch 0
3433     emitsGroups
3434     onlyEmittedByOtherEmitterSwitch 0
3435 }
3436 gameRate 20
3437 endValue_0 0.0000000000000000
3438 endValue_1 0.0000000000000000

```

Figure 22: I check and remember the gamerate.

Name	Date modified	Type	Size
ParticleEdit_NO_SSE.exe	17/08/2025 09:55	Shortcut	3 KB
ParticleEdit_SSE.exe	17/08/2025 09:55	Shortcut	3 KB

Figure 23: I launch ParticleEdit.exe from E2160.

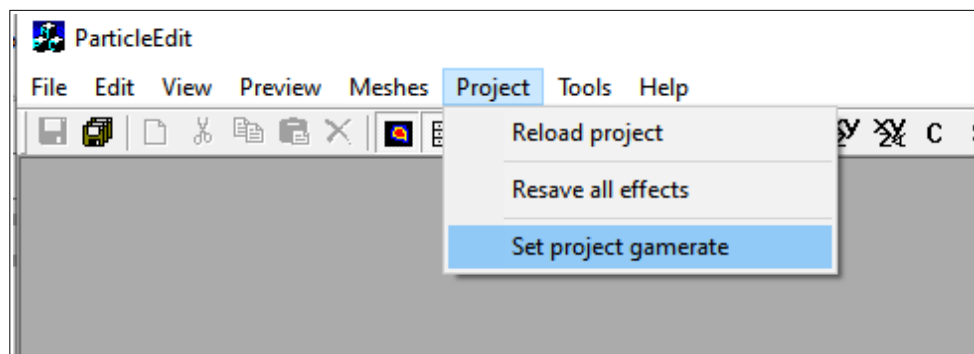


Figure 24: I locate and click on the **Set project gamerate** option.

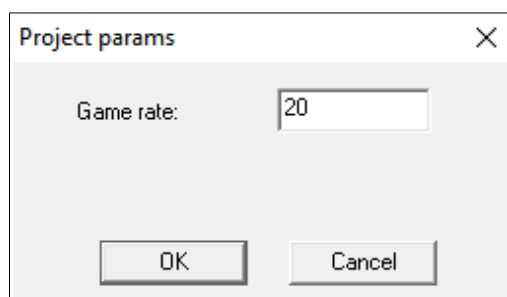


Figure 25: I set the value to the one we read from the file earlier.

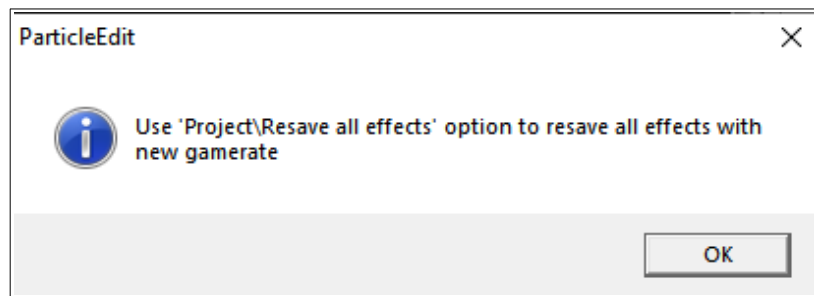


Figure 26: This message is displayed.

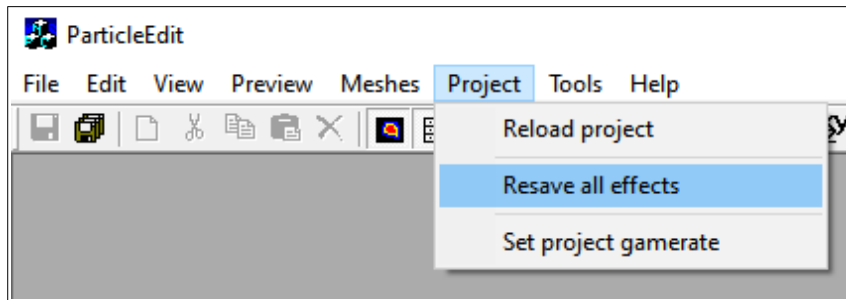


Figure 27: We locate and click **Resave all effects**. To ensure that the operation is performed correctly, please wait a moment.



Figure 28: Once the operation is complete, we shut down the program.

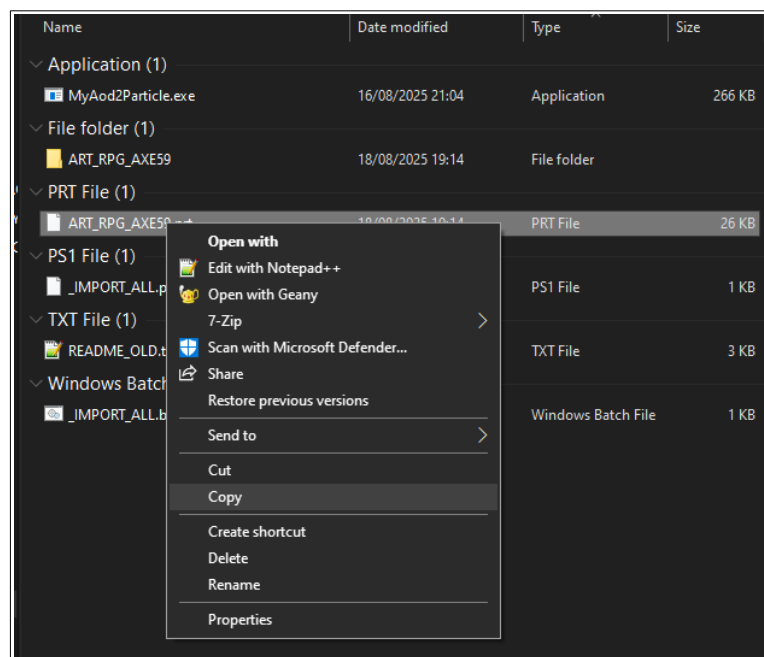


Figure 29: Copies the output .prt file.

Name	Date modified	Type	Size
Backups	08/07/2025 21:29	File folder	
Resources	10/06/2025 17:37	File folder	
1970.prt	18/08/2025 10:21	PRT File	19 KB
AMOEBE.prt	18/08/2025 10:27	PRT File	48 KB
ART_AMMO.prt	06/07/2025 12:21	PRT File	22 KB
ART_MINE.prt	07/07/2025 16:34	PRT File	18 KB
BLACK_HOLE_VORTEX.prt	18/08/2025 10:25	PRT File	20 KB
BM_7_2.prt	16/07/2025 22:19	PRT File	113 KB
Cursor01.prt	28/07/2025 14:35	PRT File	6 KB
e2160_prt1.prt	09/07/2025 20:30	PRT File	35 KB
e2160_prt2.prt	04/07/2025 12:27	PRT File	24 KB
Fire5.prt	18/08/2025 10:33	PRT File	34 KB
Geyser1.prt	07/07/2025 18:09	PRT File	28 KB
KOLUMNA.prt	19/07/2025 21:07	PRT File	43 KB
L_MI_EM_01_1.prt	21/06/2025 19:25	PRT File	12 KB
linked_e21.prt	09/07/2025 12:12	PRT File	5 KB
linked_e22.prt	08/07/2025 19:51	PRT File	5 KB
M8_TEMPLE_FX.prt	19/07/2025 21:32	PRT File	42 KB
particleslib.lprt	28/07/2025 13:29	LPRT File	1 KB
prt1.prt	04/07/2025 15:16	PRT File	32 KB
SaveEffects.log	01/08/2025 19:37	LOG File	29 KB
SG_OUT_OK.prt	07/07/2025 19:16	PRT File	81 KB
TEST.prt	05/06/2025 20:35	PRT File	5 KB
TEST.prt.bak	05/06/2025 20:33	BAK File	5 KB
TEST_or.prt	05/06/2025 20:46	PRT File	10 KB
TTEE.prt	08/06/2025 12:55	PRT File	8 KB
TTEE1.prt	08/06/2025 15:16	PRT File	8 KB
TTEE1.prt.bak	08/06/2025 15:14	BAK File	8 KB

Figure 30: I am locating the directory with .prt files for the ParticleEdit program from Earth 2160.

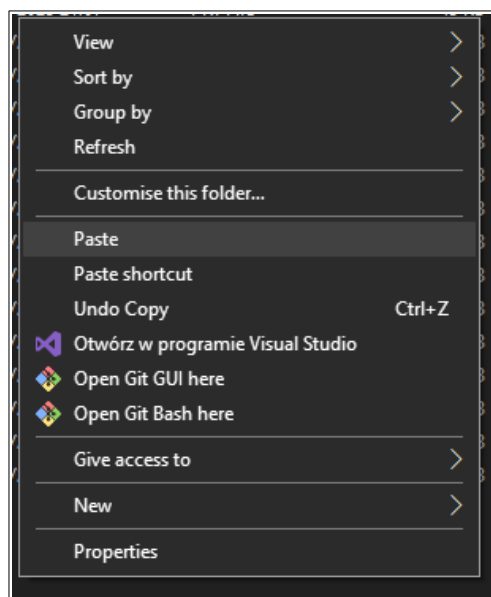


Figure 31: I'm pasting it.



	ParticleEdit_NO_SSE.exe	17/08/2025 09:55	Shortcut	3 KB
	ParticleEdit_SSE.exe	17/08/2025 09:55	Shortcut	3 KB

Figure 32: I am restarting ParticleEdit from Earth 2160.

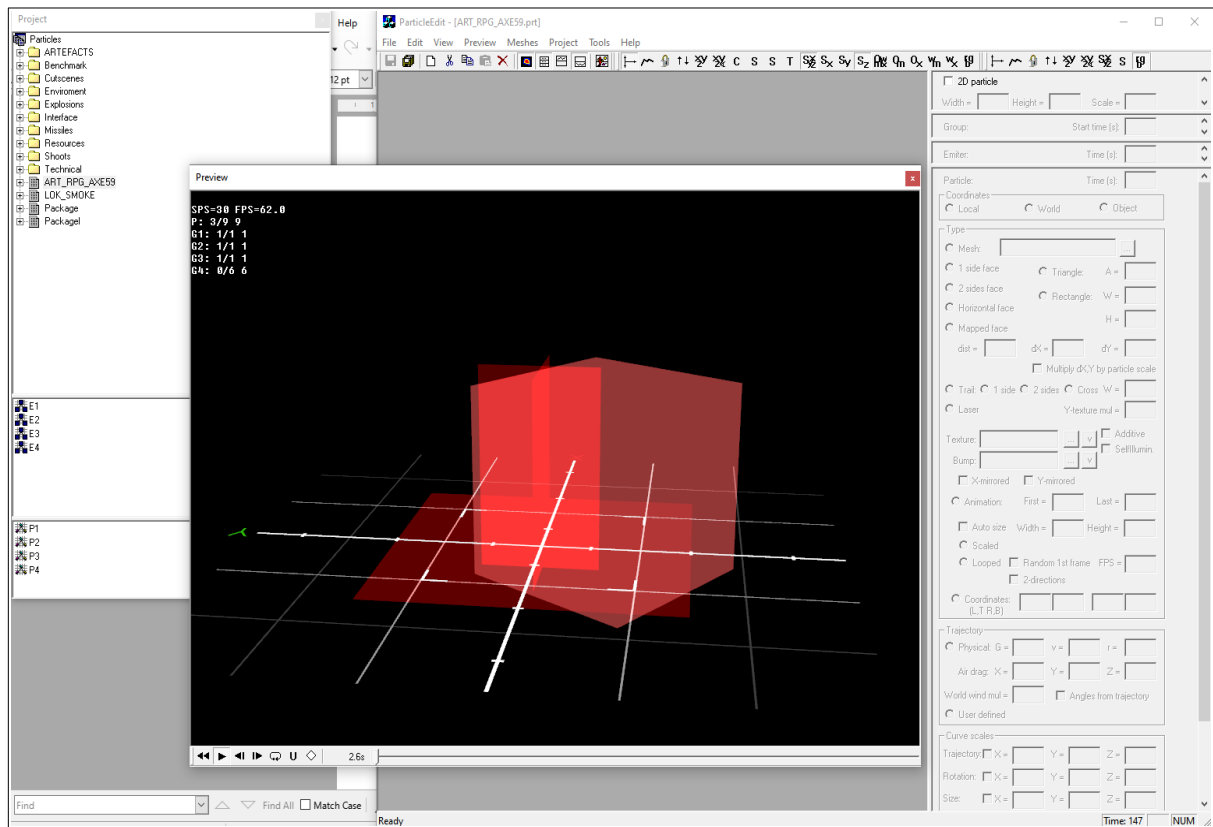


Figure 33: As we can see, the particle has started, but to get rid of these red textures, you need to convert .tex to .tga and put the textures in the Textures directory in the E2160 game directory.



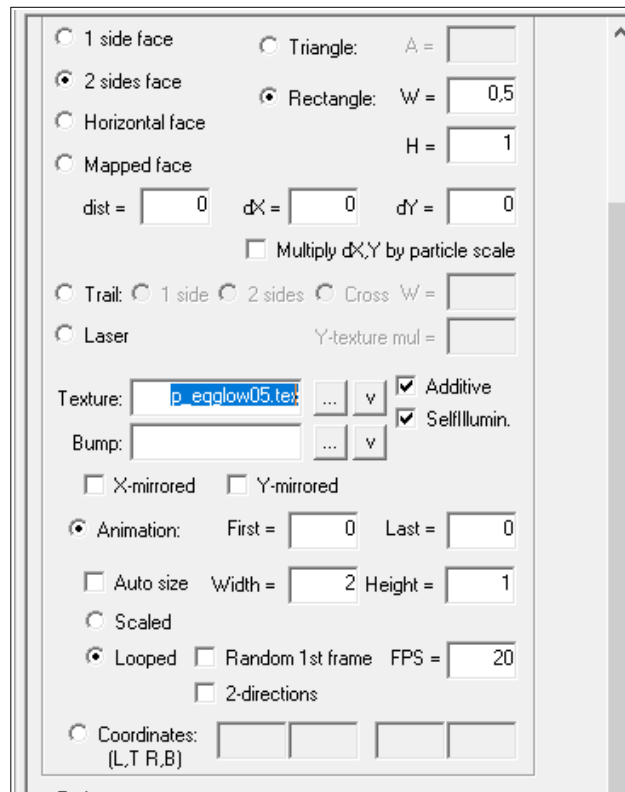


Figure 34: You can jump between ParticleEmitter objects and change the names of textures and meshes.

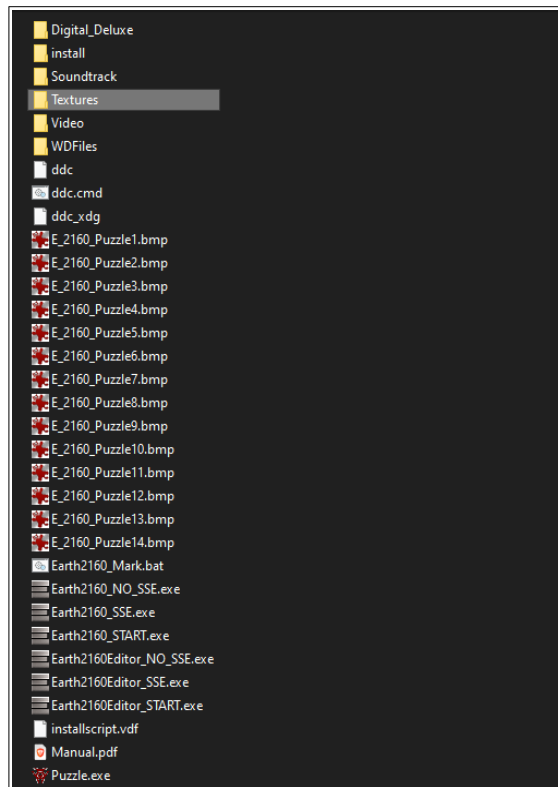


Figure 35: We create the Textures folder in the Earth 2160 game folder and place our textures in .tga format there. This way, you don't need to pack these files into WD.

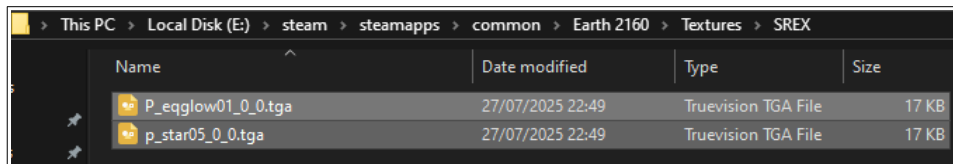


Figure 36: I added these textures myself.

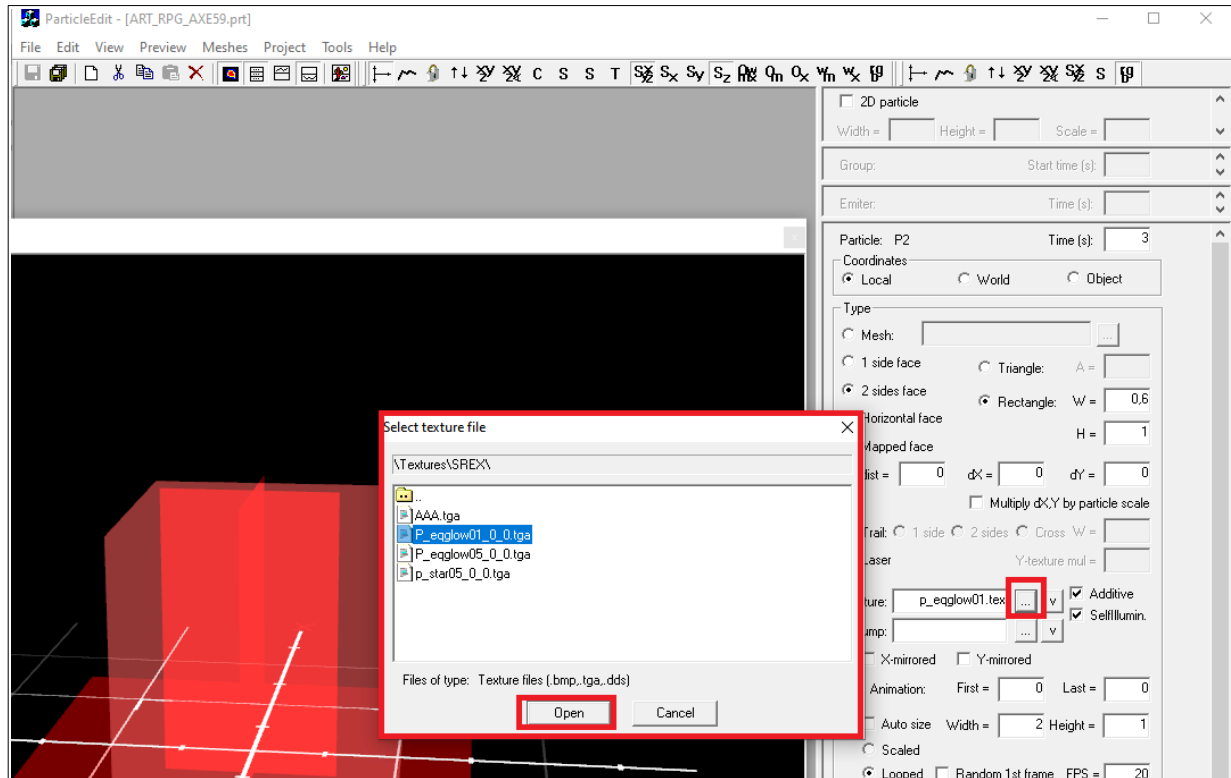


Figure 37: We select tga textures from the texture catalog. We do this for each .tex texture. At the same time, we can also change the mesh to, for example, Sphere.

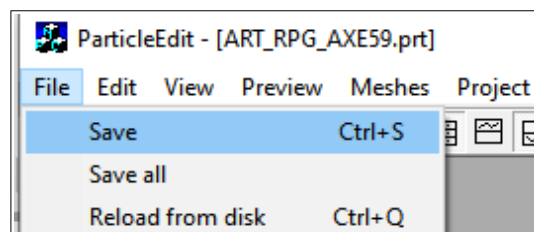


Figure 38: We save the particles.

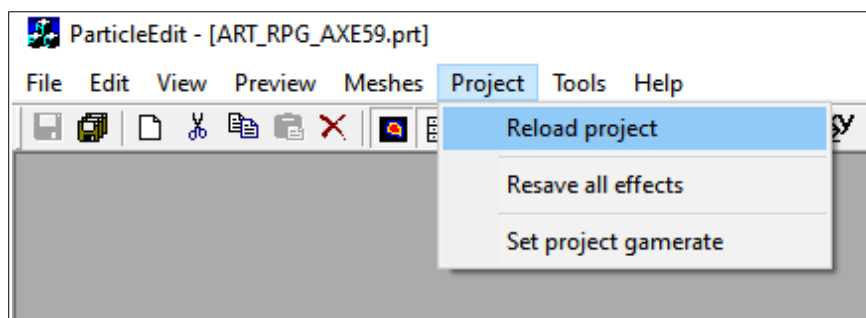


Figure 39: To refresh the project in order to see the changes, we can click on the **Reload project** option.

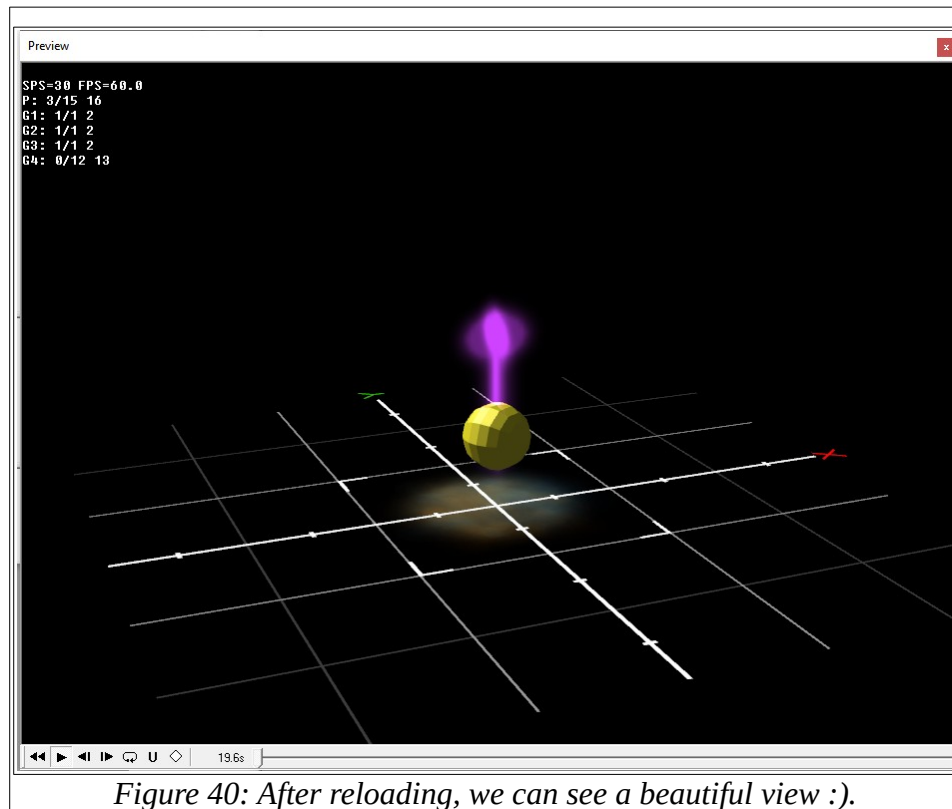


Figure 40: After reloading, we can see a beautiful view :).

d) I want to make changes to the ParticleEmitter from KnightShift.  
We export the file to E2160 format as in point c). We edit it as we see fit.

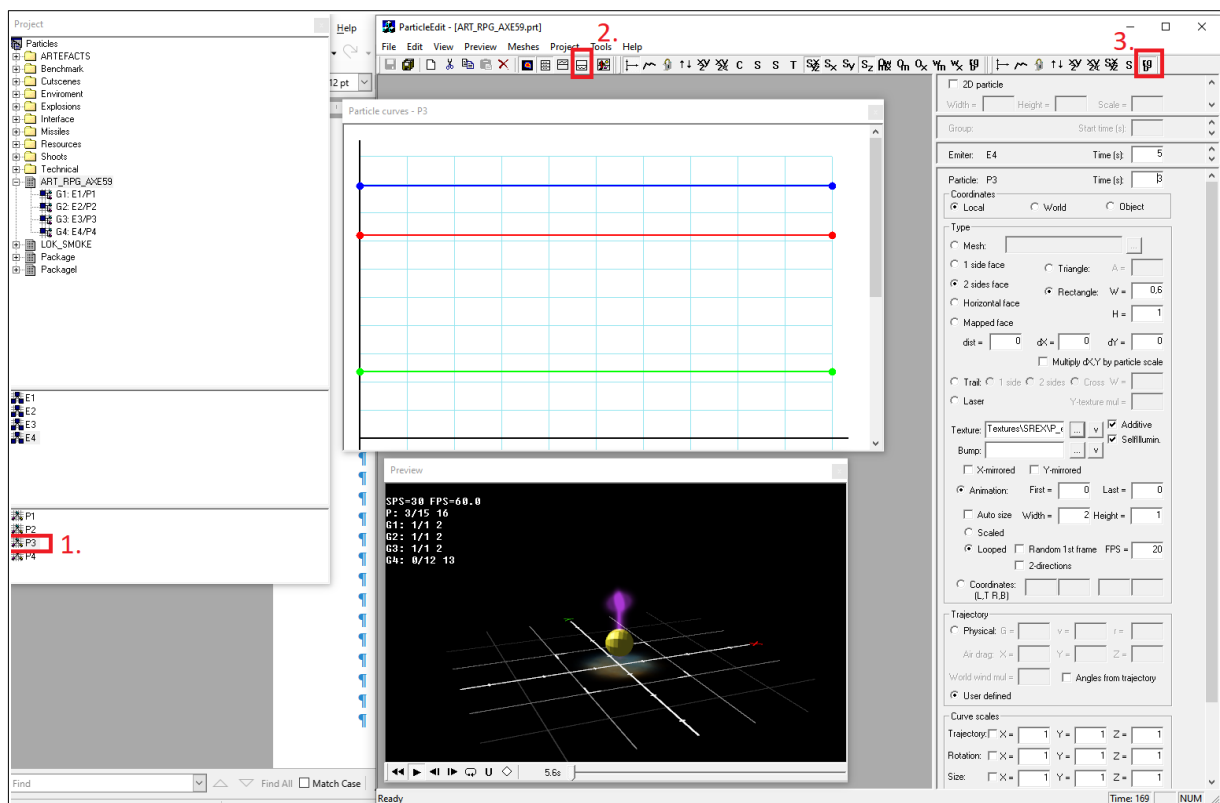


Figure 41: Now, for example, we identify what we want to change and make the changes. For example, I will change the color of this weapon border.

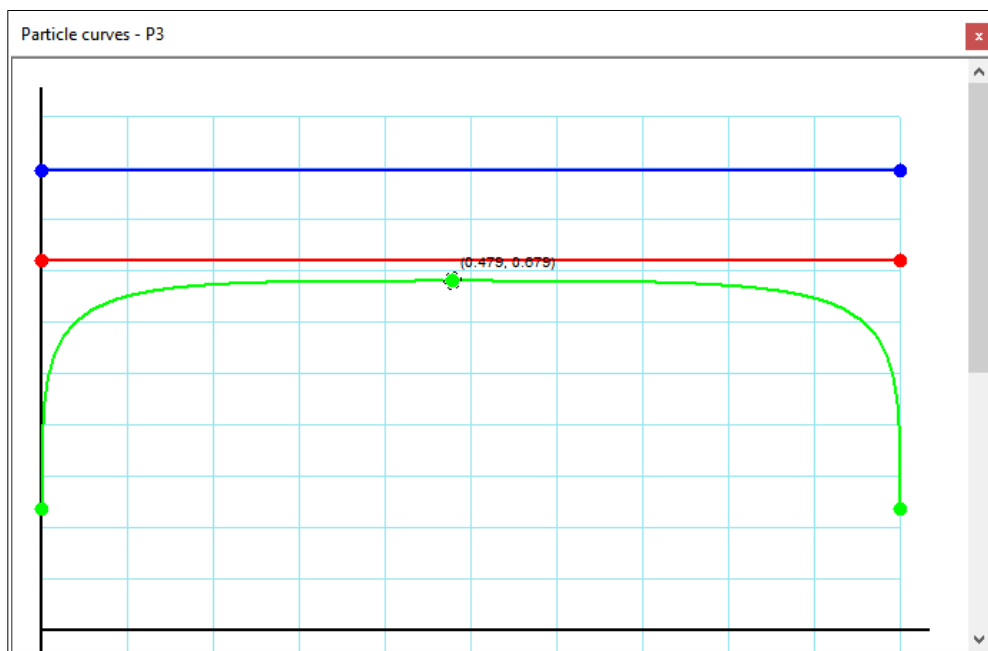


Figure 42: I manipulate points.

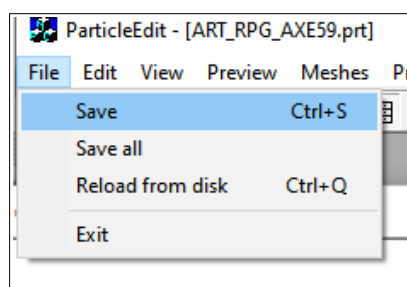


Figure 43: I am saving the file.

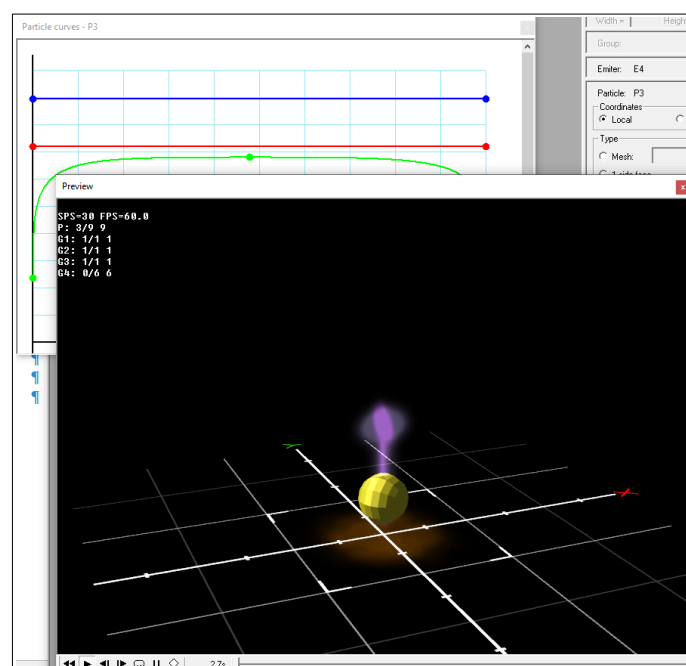


Figure 44: The border has changed to white.

If we want to put this file back into KnightShift, we need to restore the entries for the previous .tex textures and meshes. This can be done with ParticleEdit at this point or during the conversion back to .myaod format using a text editor such as Notepad++.

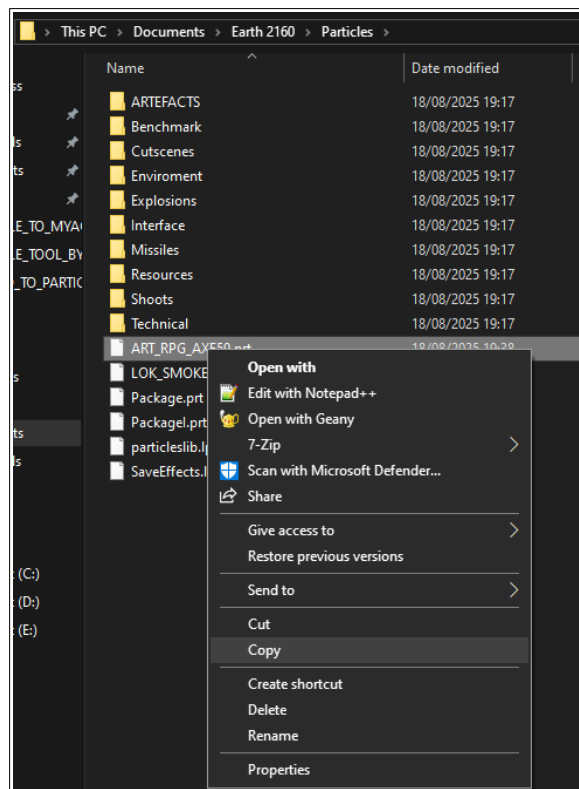


Figure 45: Copy the .prt file from the ParticleEdit particles directory.

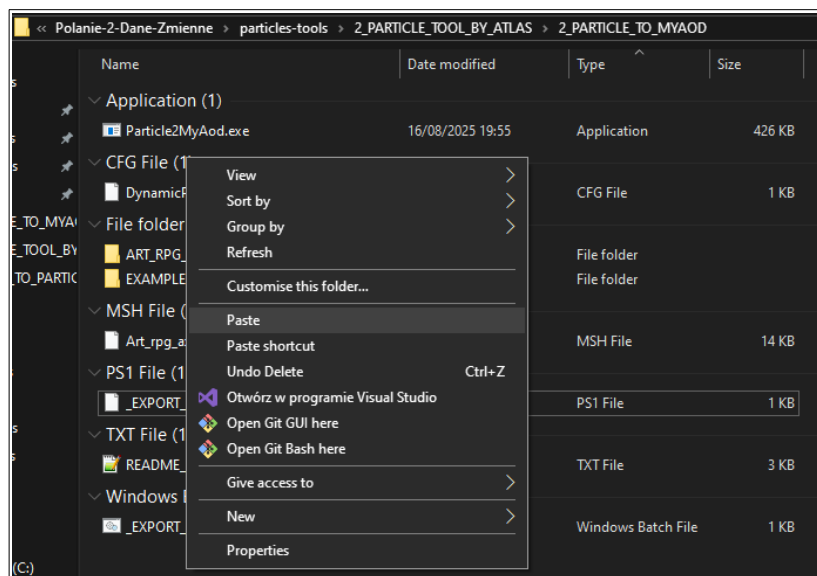


Figure 46: Paste into the Particle2MyAod directory.

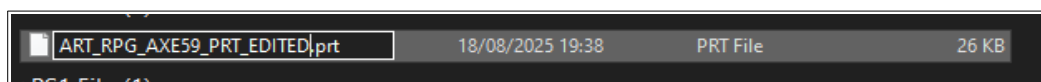


Figure 47: We are changing the file name as a precautionary measure so that the output directories do not get mixed up.

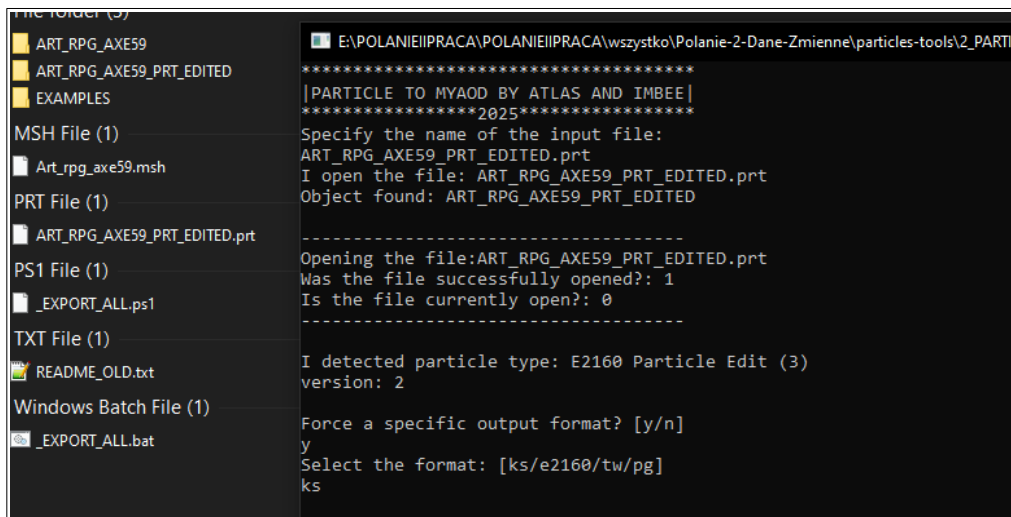


Figure 48: We export by enforcing the ks format.

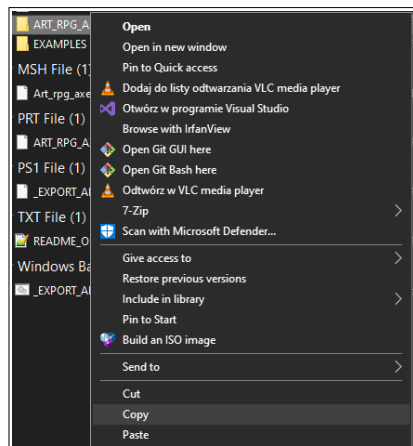


Figure 49: We copy the directory that was created.

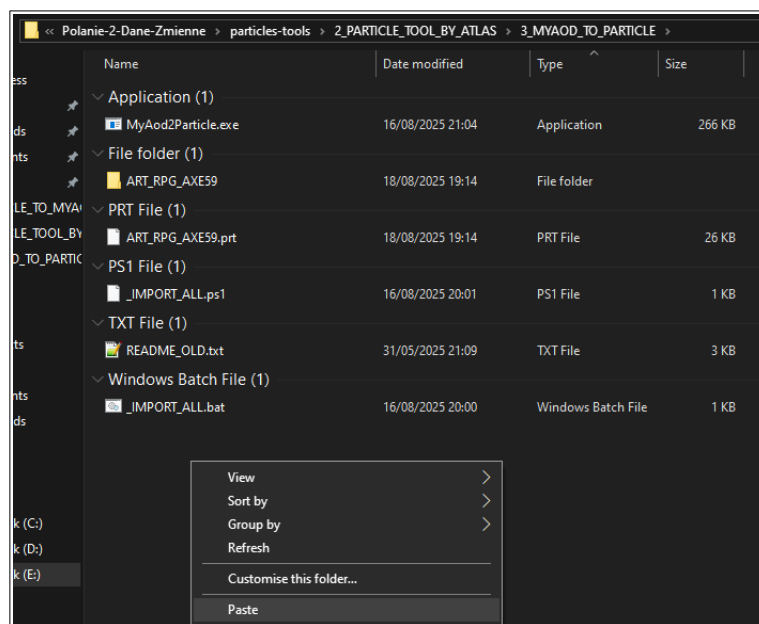


Figure 50: Paste it into the MyAod2Particle directory.

```

E:\POLANIEIIPRACA\POLANIEIIPRACA\wszystko\Polanie-2-D
*****
|MYAOD TO PARTICLE BY ATLAS AND IMBEE|
*****2025*****
Specify the name of the input directory:
ART_RPG_AXE59_PRT_EDITED_

```

Figure 51: We import.

MSH File (1)			
ART_RPG_AXE59_PRT_EDITED.msh	18/08/2025 19:53	MSH File	14 KB
PRT File (1)			

Figure 52: Done—you can upload the file to KnightShift.

e) I want to run a .prt file from 3D ParticleGen Visual FX (Steam) in ParticleGenie from Two Worlds:

Name	Date modified	Type	Size
Application (1)			
Particle2MyAod.exe	16/08/2025 10:55	Application	176 KB
CFG File (1)			
DynamicParticle.cfg			
File folder (2)			
1970			
EXAMPLES			
PRT File (1)			
1970.prt			
PS1 File (1)			
_EXPORT_ALL.ps1			
TXT File (1)			
README_OLD.txt			
Windows Batch File (1)			
_EXPORT_ALL.bat			

Figure 53: I launch the **Particle2MyAod.exe** program. I enter the **file name**, then I enter **y** to force a different format, and I enter **tw**.

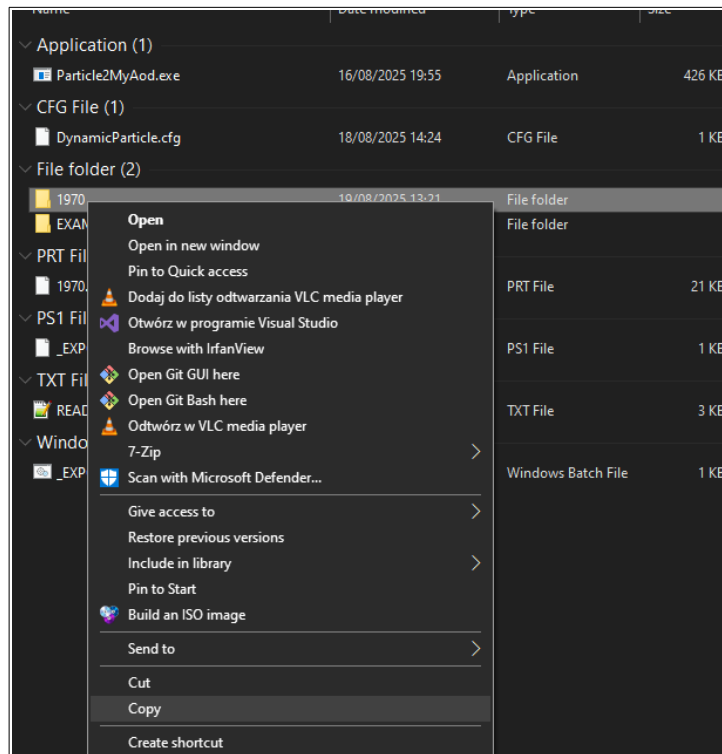


Figure 54: I copy the obtained directory.

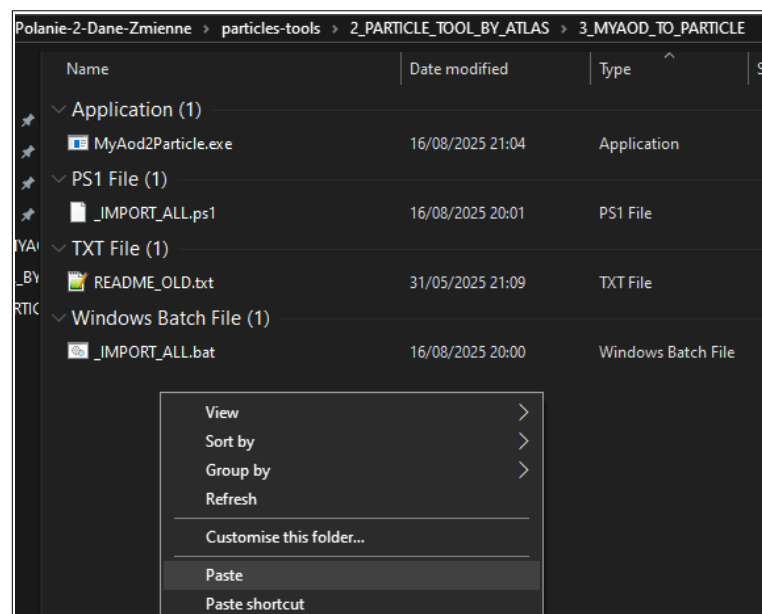


Figure 55: I paste it into the directory with **MyAod2Particle.exe**.



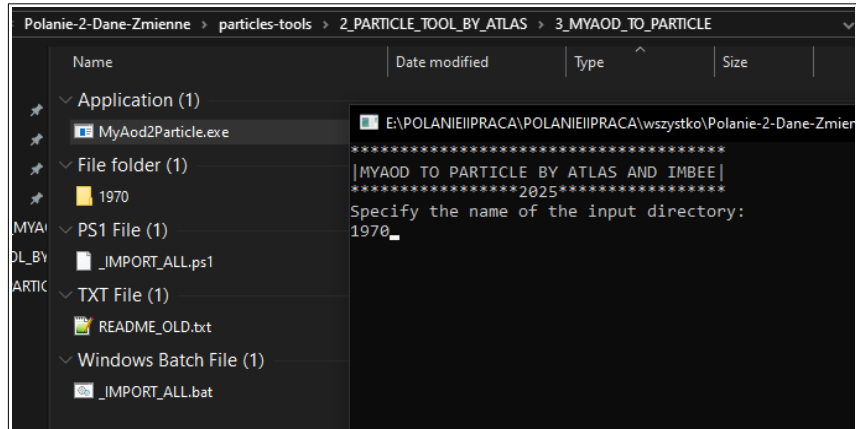


Figure 56: I turn on **MyAod2Particle** and enter the name of the pasted directory into it.

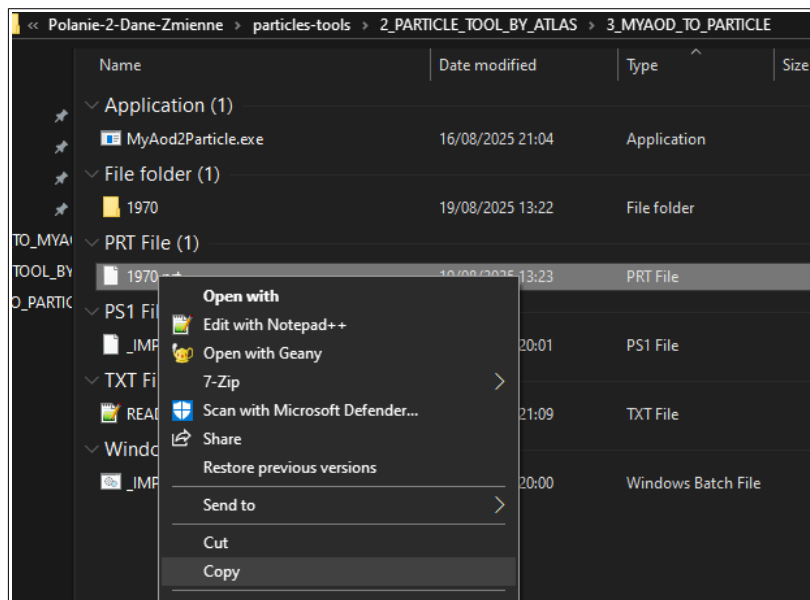


Figure 57: I copy the resulting **.prt** file.

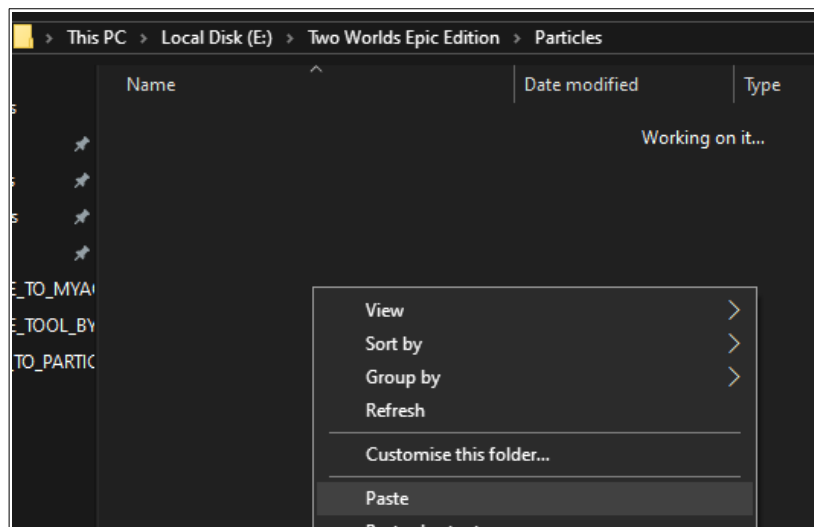


Figure 58: I paste it into the **Particles** directory in the **Two Worlds** game folder.

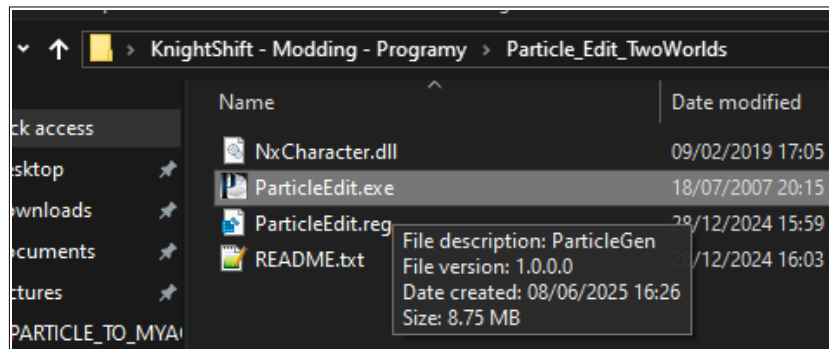


Figure 59: Uruchamiam **ParticleGena** z **Two Worlds** SDK.

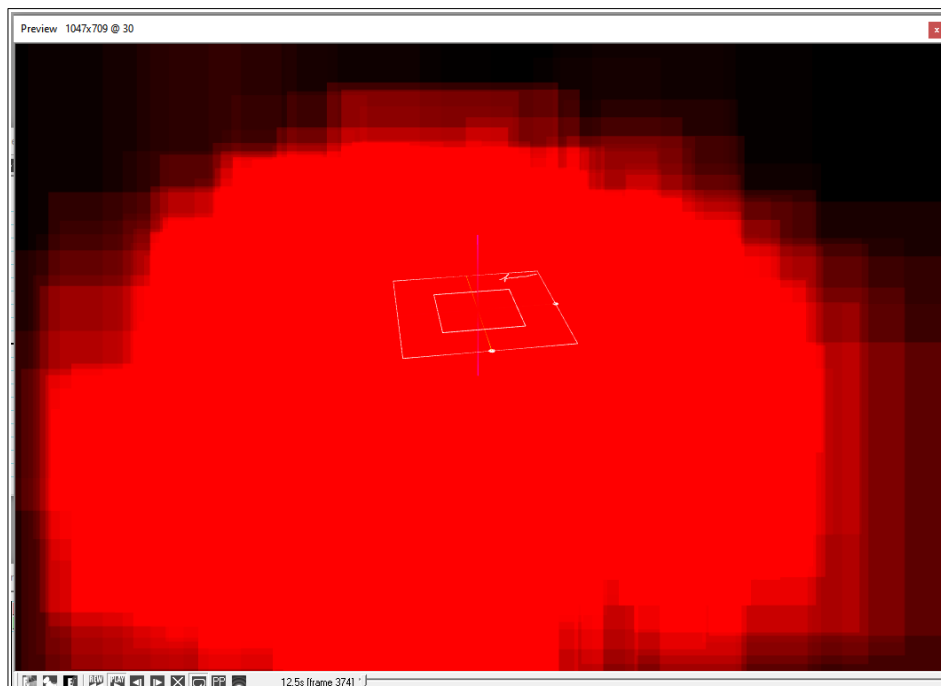
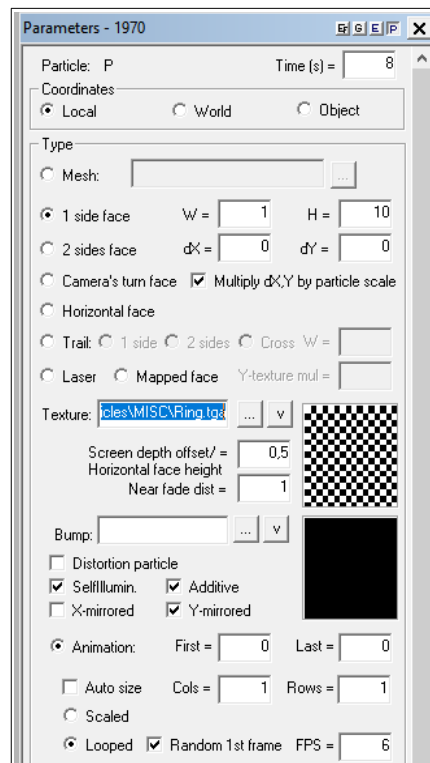
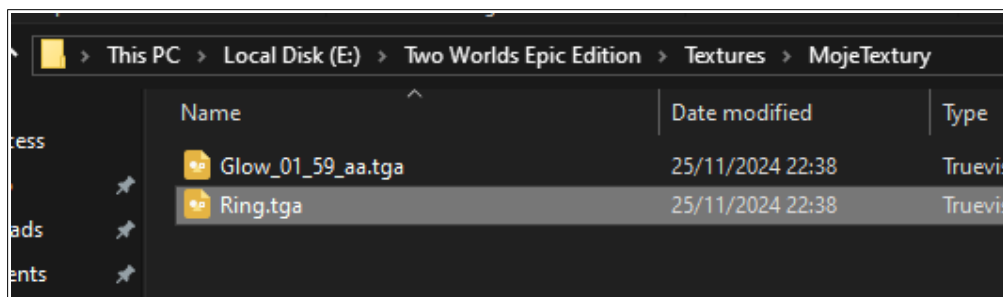


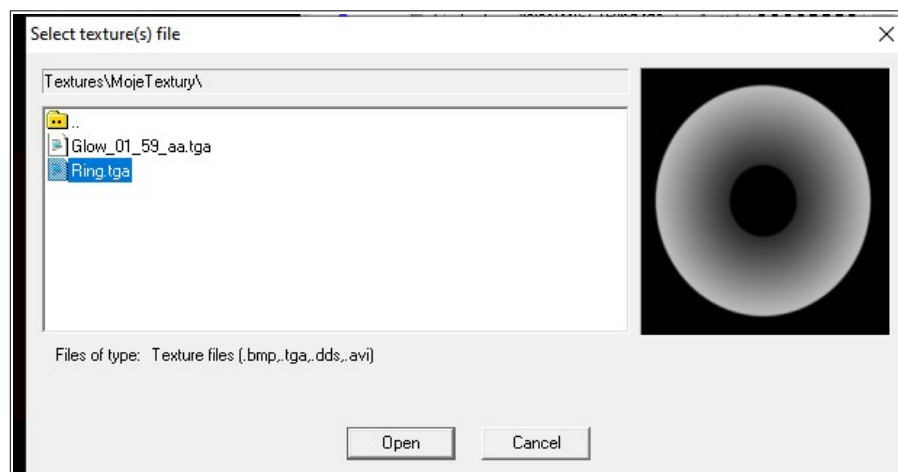
Figure 60: As we can see after launching particles in the program, the file works, but you still need to transfer the appropriate texture from 3D Particle Gena to Two Worlds.



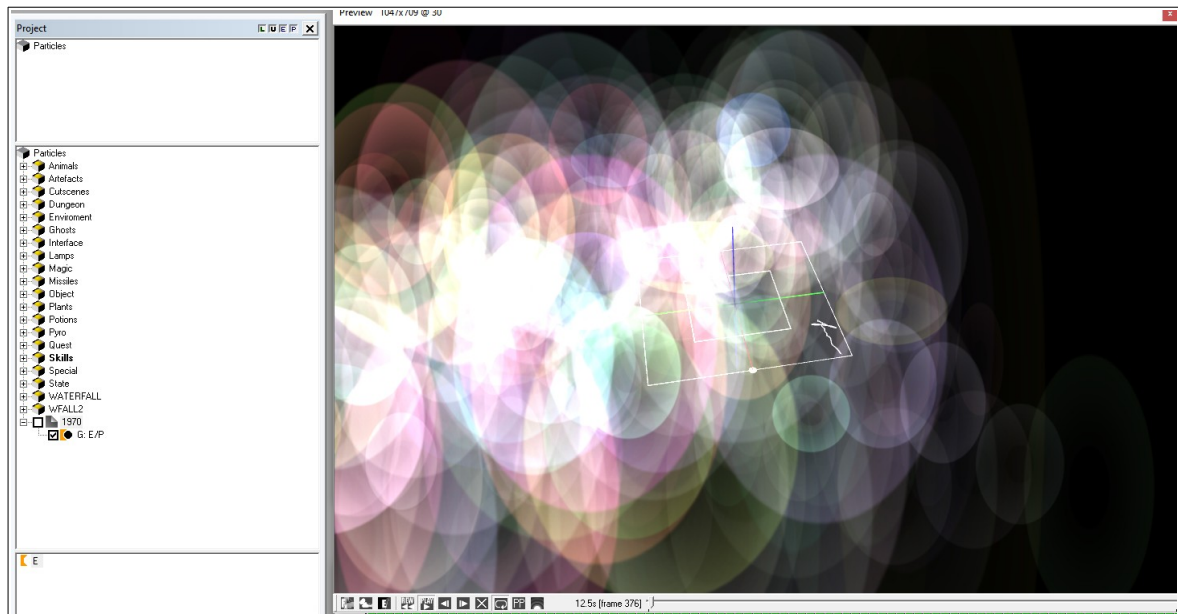
*Figure 61: We can locate the activated texture in the object parameters.*



*Figure 62: Place the required texture in the Textures directory in **Two Worlds** (I created an additional directory inside called MyTextures).*



*Figure 63: In the particle program, select the added texture.*



*Figure 64: After saving the particles and restarting the program, we can see the final effect.*