

Modding Tutorials

About the Author

Greetings Martian,

I'm Sylvain, you know me under my nickname "Silva". I started modding on Surviving Mars a few months after the release and I created many famous mods for 2 or more years.

In November 2020, I was contacted by Paradox to create the first Content Creator Pack (CCP1) before I joined Abstraction Games to work on the Below and Beyond DLC.

Today I'll share with you parts of my knowledge to help you to mod Surviving Mars. This basic Art Pipeline documentation covers how to create and import a new Asset used in your Mods. You can find plenty of information about the Texturing, Animating, Metadata and how to set up Blender with the Export Tools.

Art Pipeline

You can follow this roadmap to create your Asset. In this guide I will give Art specifications on how to set up your Asset for the Game, presuming you have enough knowledge in 3D modeling and Texturing.

1. Concept Art – Design
2. Modeling
3. Animating
4. Texturing
5. Metadata (Lighting, Collision Box, Footprint, Path, Door...)
6. Set your Asset with Blender 2.79b
7. Export/Import into the Game
8. Create the Icons, LUA functions and more

Sections

- [Tools – Documentation](#)
- [Scene – Metadata](#)
- [Texturing – Animation – Lighting](#)
- [Blender Settings – Exporting](#)
- [Integrate your Asset In-game](#)
- [Extra : Pipe Connection](#)
- [Extra : Transparency Material](#)
- [Extra : Hook Up SoundFX or ParticlesFX](#)
- [Extra : Terrain Hole](#)
- [Extra : Create a Building Skin](#)

Tools – Documentation

Applications

Surviving Mars (*latest version*)

 The Mod Tool is integrated In-game.

Blender 2.79b + Plugin HGE Exporter (*this plugin works only with the 2.79b*)

 Plug-in location : [Game_Install_Folder]\Surviving Mars\ModTools\HGBlenderExporter.zip

3D Software (Blender 2.9+, Maya, 3DsMax...)

Texturing Software (Substance, Photoshop...)

3rd Party Tools

ChoGGi's Expanded Cheat Menu Mod: a useful tool to debug and test your mod.

 Available on Steam Workshop & Paradox Mod <https://steamcommunity.com/sharedfiles/filedetails/?id=141115781> 

Documentation & Samples

To complete this Tutorial, read the **Official Documentation**, it contains a lot of useful information.

 Official documentation is here [Game_Install_Folder]\Surviving Mars\ModTools\Docs

Reference Code <https://github.com/surviving-mars/SurvivingMars>

Samples [Game_Install_Folder]\Surviving Mars\ModTools\Samples

Tutorial : How to create an Animal (Art Pipeline) by Silva <https://steamcommunity.com/sharedfiles/filedetails/?id=1759820629>

Download Tutorial Source files

Download the example mod files called "FoodFactory" from our git repository

Contains an asset folder and mod folder that can be used as reference alongside this guide.

Assets → Blender & Textures files

FoodFactory/Mod → Mod files (must be placed in the Mod folder)

Scene – Metadata

What is a Metadata ?

There are 3 types of metadata : Spot, Path and Surface

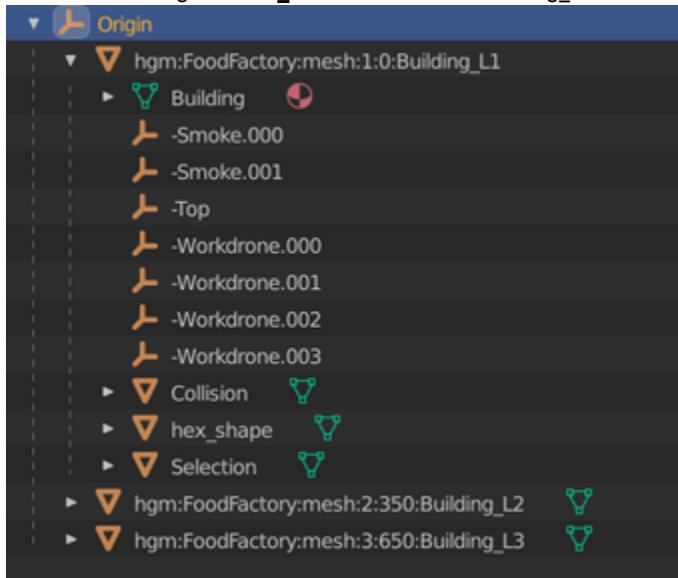
- Spot : Used to set the position for the Lights, Logo, FX, Attaches, Indicators, Entrances and many other things
- Path : To define the Path to enter/exist/visit a building for the Drones, Colonists...
- Surface : To define Collision Box, Footprint
 - **Collision** is a simplified building shape
 - **hex_shape** is a Plane (not a volume) object
 - **Selection** is a Plane (not a volume) object

After modeling your 3D model in your favorite 3D Software, you must import it into **Blender 2.79b** (*FBX format is recommended*)

 Check the Official Documentation to get more information.

Scene hierarchy

- **Origin**
 - hgm:ENTITY_NAME:mesh:1:0:Building_Lod1
 - All essential Metadata
 - -Top
 - ... other Spots
 - hex_shape
 - Collision (if your asset has a Collision Box)
 - Selection
 - ... other Surfaces
 - ... other Paths
 - hgm:ENTITY_NAME:mesh:2:350:Building_Lod2
 - hgm:ENTITY_NAME:mesh:3:650:Building_Lod3



 **Note :** Blender automatically adds a number after the name “.001 .002 ...” because the names are unique in Blender. This is not a problem.

Naming convention to mark your Entity

```
hgm:<entity_name>:<mesh_name>:<lod>:<distance>:s=<state_name>:<blender_name>
```

entity_name

The name of the entity needs to be identifiable as it will appear in the game editor (Latin letters, digits, underscore permitted).

mesh_name

Static objects need to have one mesh per state. By convention, the “normal” mesh (e.g. for the default *idle* state) is called *mesh*.

lod

LOD (level of detail) variant of the mesh, 1-based; 1 is the most detailed mesh

distance

Distance from the camera (in meters) where the engine starts to use this LOD; must be 0 for LOD 1

state_name

The name of the state in which this state will be used. Optional; defaults to *idle*.

blender_name

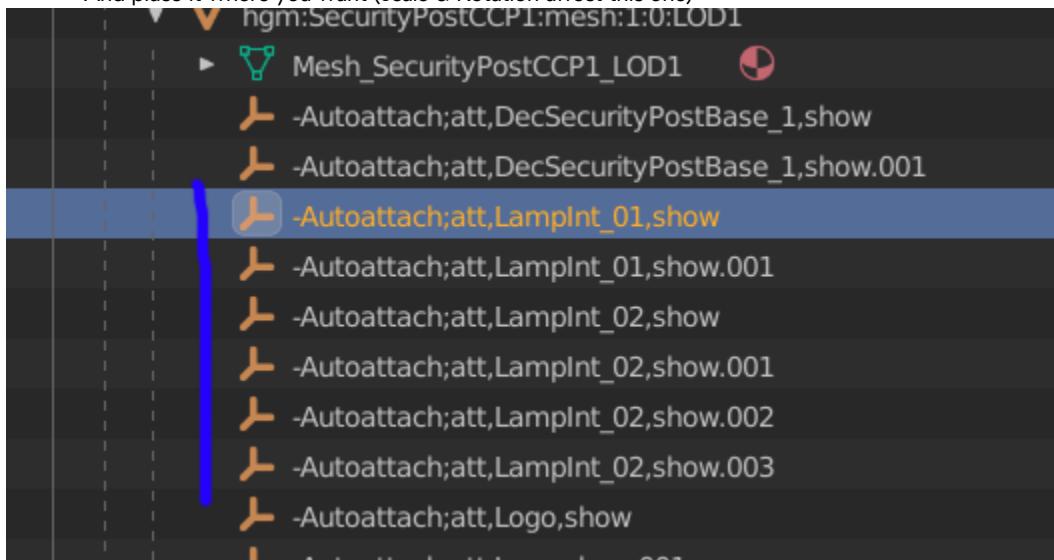
Additional string you can use to distinguish between several objects which need to be separate meshes in Blender, but will be merged by the exporter into one in-game mesh.

AutoAttach

The Attach method is an easy way to add an existing Asset (useful to add foliage and other decorations). This metadata is an Empty object with a specific naming convention.

```
-AutoAttach;att,[entity_name],show
```

- [entity_name] : Replaced by the Entity name
- And place it where you want (Scale & Rotation affect this one)



Some useful Decoration names :

- Lamps:
- LampInt_01 to LampInt_05
- LampWallInner_01 to _05

- LampWallOuter_01 to _05
- Bush & Tree
- Bush_01 to _05
- Tree_01 to _05
- and all other Assets...

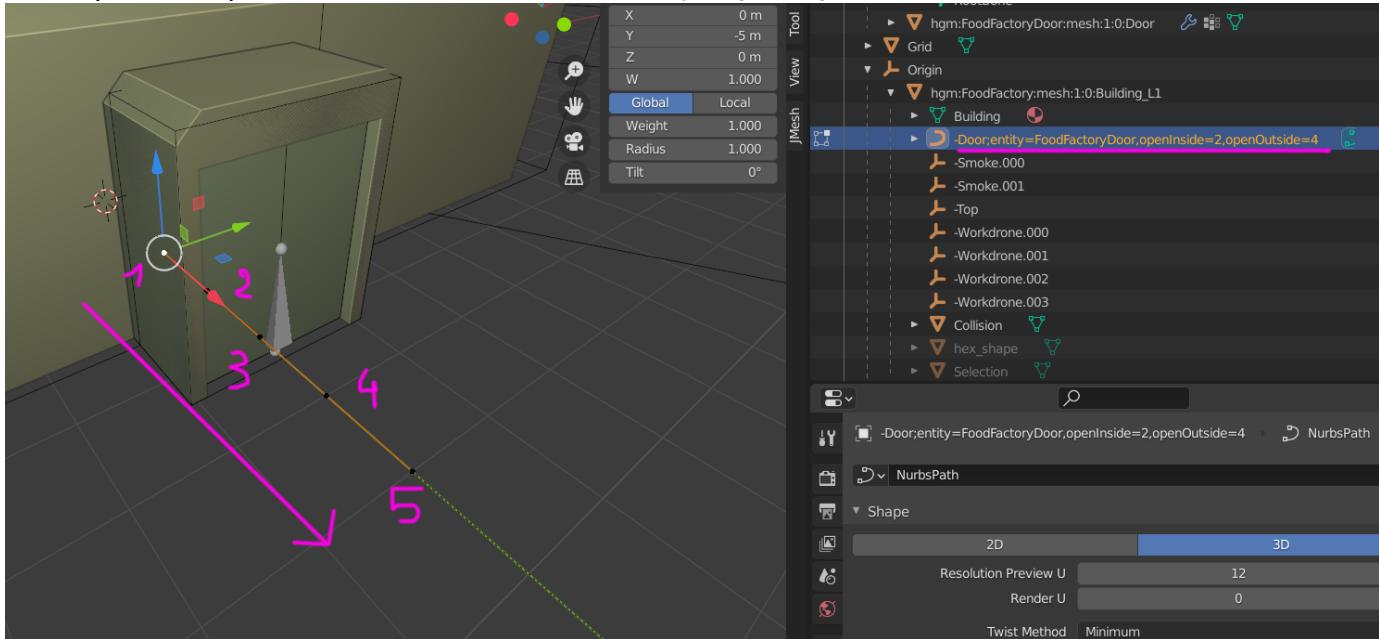
How to add an Entrance path ?

In Blender you can use the Path object, in other 3D Softwares it is a Line or Spline.

Follow the following naming convention to add a door.

-Door;entity=[DOOR_ENTITY],openInside=2,openOutside=4

- Replace [DOOR_ENTITY] with your door entity (the example uses *FoodFactoryDoor*)
- **openInside & openOutside** define the Vertex number when opening/closing the Door



- **The first Vertex** must **Always** start inside the Building and the last vertex needs to end outside. This is the Direction of the Path. The Vertex order is important for the Doors and Paths.
- If your Building has an entrance but no Door. Add a Path with the name -Door only.
- To create a Path to walk around the Building, name it -Path instead of -Door

Texturing – Animation – Lighting

UV Mapping

Your Asset **must have 2 UV channels !**

- **UVChannel_1**
- **UVChannel_2 for the Dust, Frozen, and Emissive effects**

Both UV Mappings can have the same UV Wrapping or a different/separate one.

The Textures

Surviving Mars uses the PBR textures. There are **5 textures**

- **Base Color (1) Required**
- **Normal (1) Required**
- **Roughness Metallic (1) Required**
- Colorization (1)
- Emissive (2)
- Dust (2)

* (UV Channel number)

Specifications :

- **File format : TGA (supports Opacity)**
- **Resolution : 4*4 to 2048*2048 (only use square ratios)**

Substance Painter

You can use your own settings if you want but I will share my own settings.

Shader

In “**Shader Settings**” select “**pbr-metal-rough**”

Channel Settings



Export Settings

User0 & User1 are the custom user channels to define the Dust & Colorization channels.

Output maps

	Create:	Gray	RGB	R+G+B	RGB+A	R+G+B+A	
\$textureSet_BC	\$	RGB	A				png ✓ X 8 bits ✓ X
\$textureSet_RM	\$	R	G	B			png ✓ X 8 bits ✓ X
\$textureSet_NM	\$	RGB					png ✓ X 8 bits ✓ X
\$textureSet_EM	\$	Gr					png ✓ X 8 bits ✓ X
\$textureSet_Dust	\$	Gr					png ✓ X 8 bits ✓ X
\$textureSet_Colorization	\$	RGB					png ✓ X 8 bits ✓ X

Ambient occlusion

- Base color
- Anisotropy angle
- Anisotropy level
- Blending mask
- Coat color
- Coat normal
- Coat opacity
- Coat roughness
- Coat specular level
- Diffuse
- Displacement
- Emissive
- Glossiness
- Height
- Ior
- Metallic
- Normal
- Opacity
- Reflection
- Roughness
- Scattering
- Scattering color
- Sheen color
- Sheen opacity
- Sheen roughness
- Specular
- Specular edge color
- Specular level
- Translucency
- Transmissive
- User0
- User1
- User10
- User11

Normal

- World space normal
- ID
- Ambient occlusion
- Curvature
- Position
- Thickness

Converted maps

- 1/ior
- Mixed AO
- Diffuse
- f0
- Glossiness
- Glossiness²
- Normal DirectX
- Normal OpenGL
- Reflection
- Specular
- Unity4 Diffuse
- Unity4 Gloss
- 2D View

Drag and drop maps on output's channels to setup your export.
6 output maps will be exported.

Colorization

The Colorization map is used by the Color Scheme feature to switch the color for all Buildings in-game. You can compare it with a **RGB Mask System**, this feature is optional, you can skip it if you want. It's not necessary to use all Masks, you can work with only the **Red**, or **Red + Green**, or **Red + Green + Blue**. But impossible to use Green or Blue alone or Green + Blue

In Substance

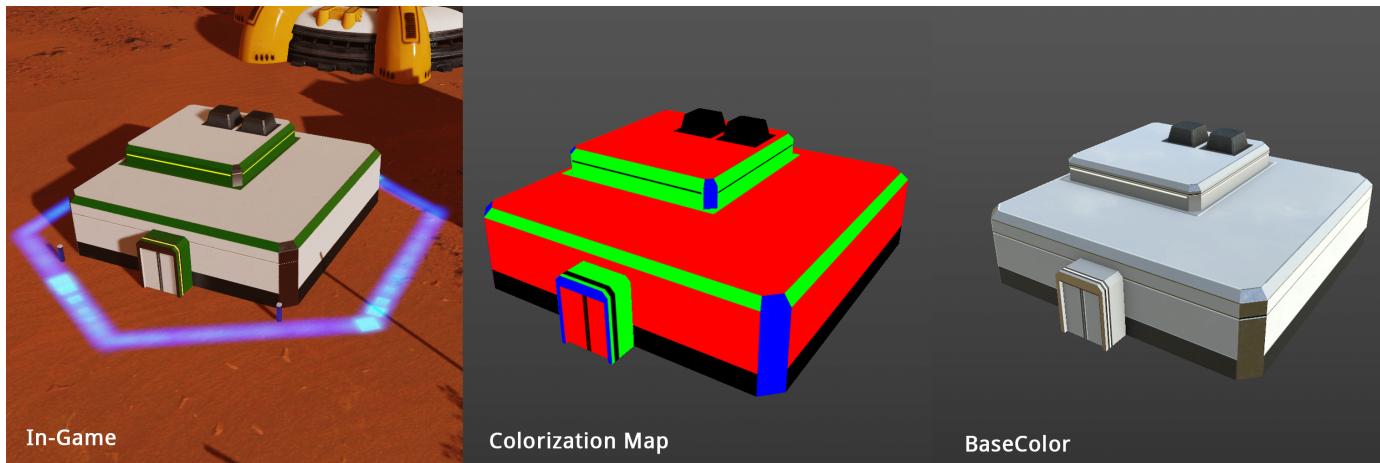
Red = R255 G0 B0

Green = R0 G255 B0

Blue = R0 G0 B255

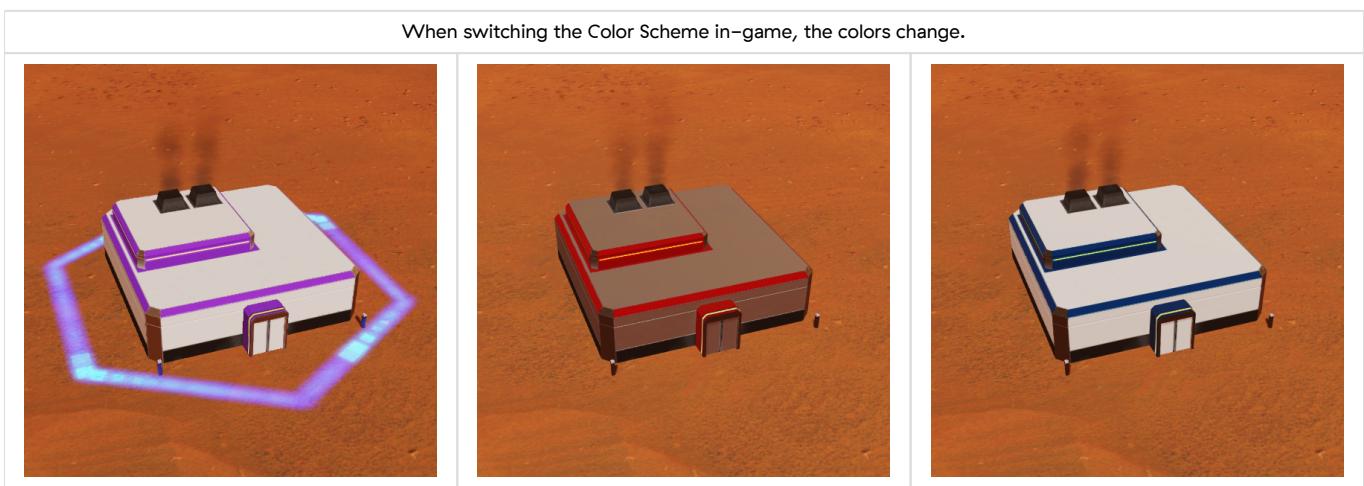
Black = R0 G0 B0

⚠️ All parts in **Black** are not affected by the Colorization. This is useful for the details, decals...



Palette color 1	<input type="text" value="outside_base"/>
Palette color 2	<input type="text" value="inside_accent_food"/>
Palette color 3	<input type="text" value="outside_metal"/>
Palette color 4	<input type="text" value="none"/>

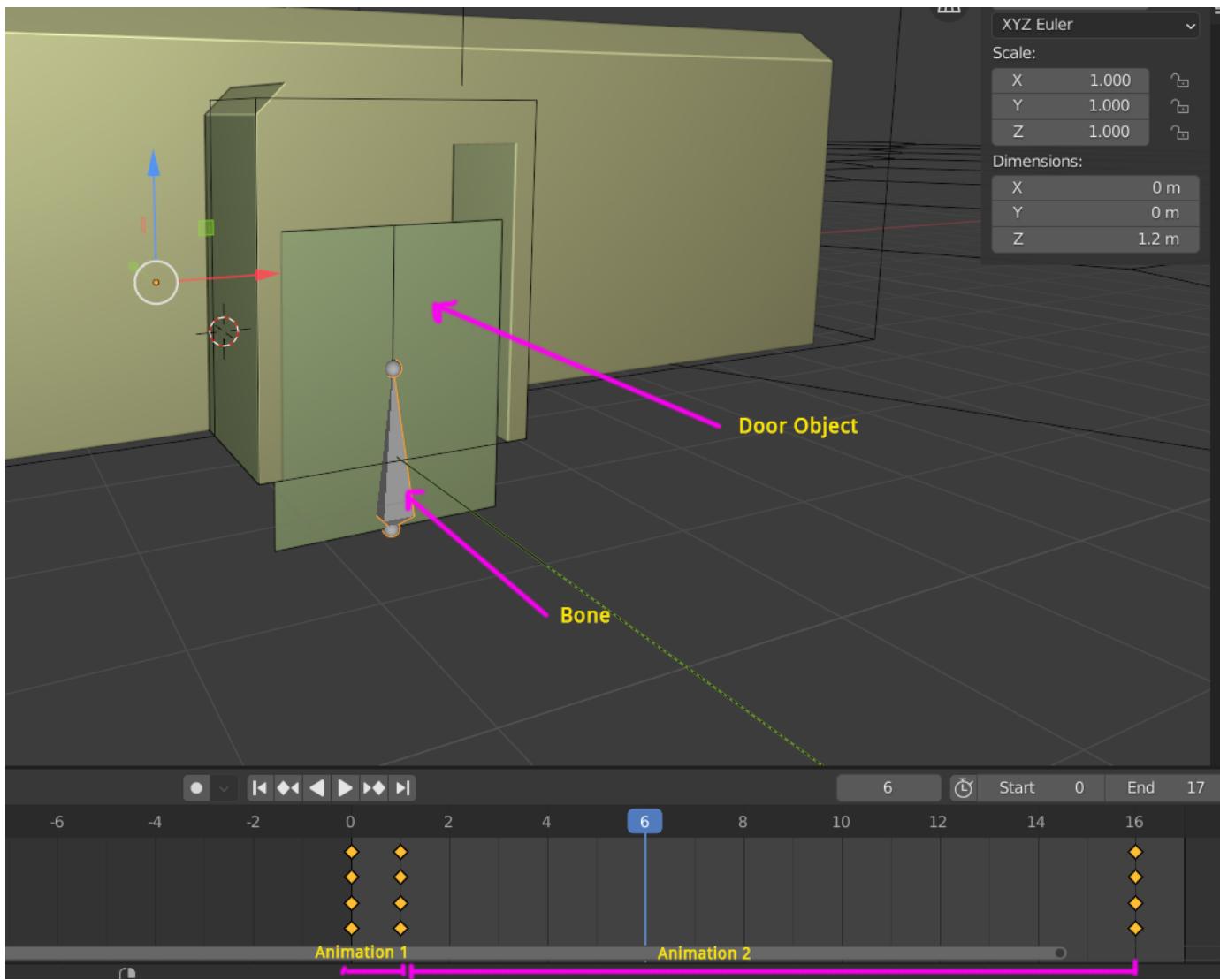
Look at the **BaseColor**, there is no color, only white, grey and dark grey. The Colorization is set in the Building Template.



Animation

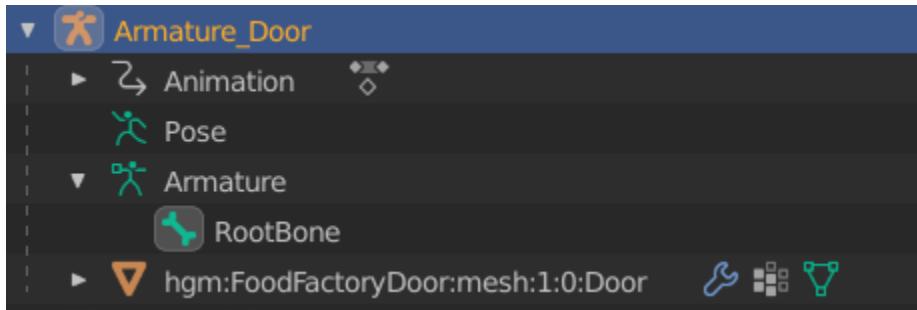
You can create your Animations with your favorite 3D Software but make sure **Blender 2.79b** accepts your animations properly. Surviving Mars uses the Bone System, so you must use the Bones to animate your model; don't animate an object without Bones, this is doesn't work.

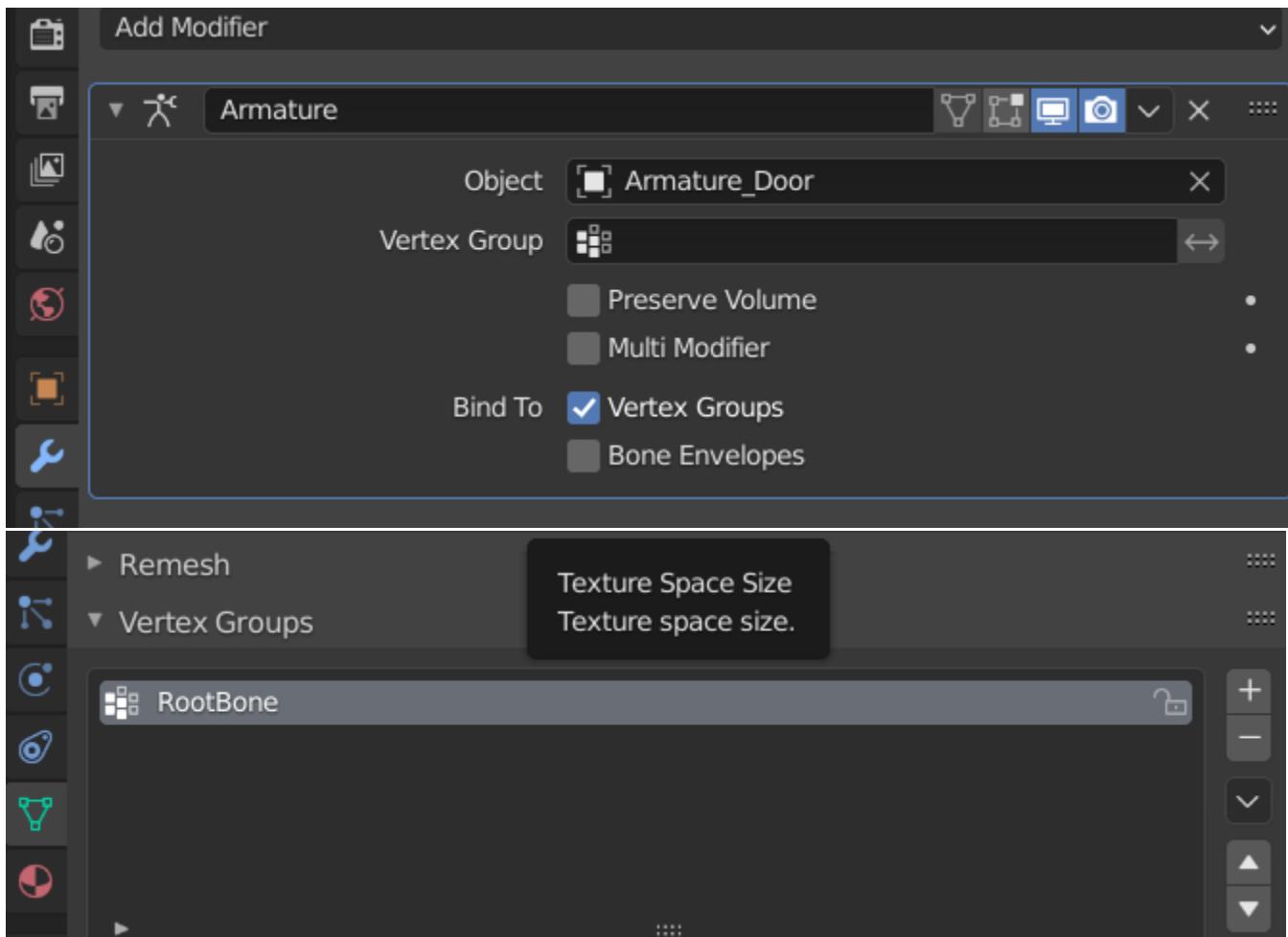
i More details can be found in the **Blender Settings & Exporting** section of this guide.



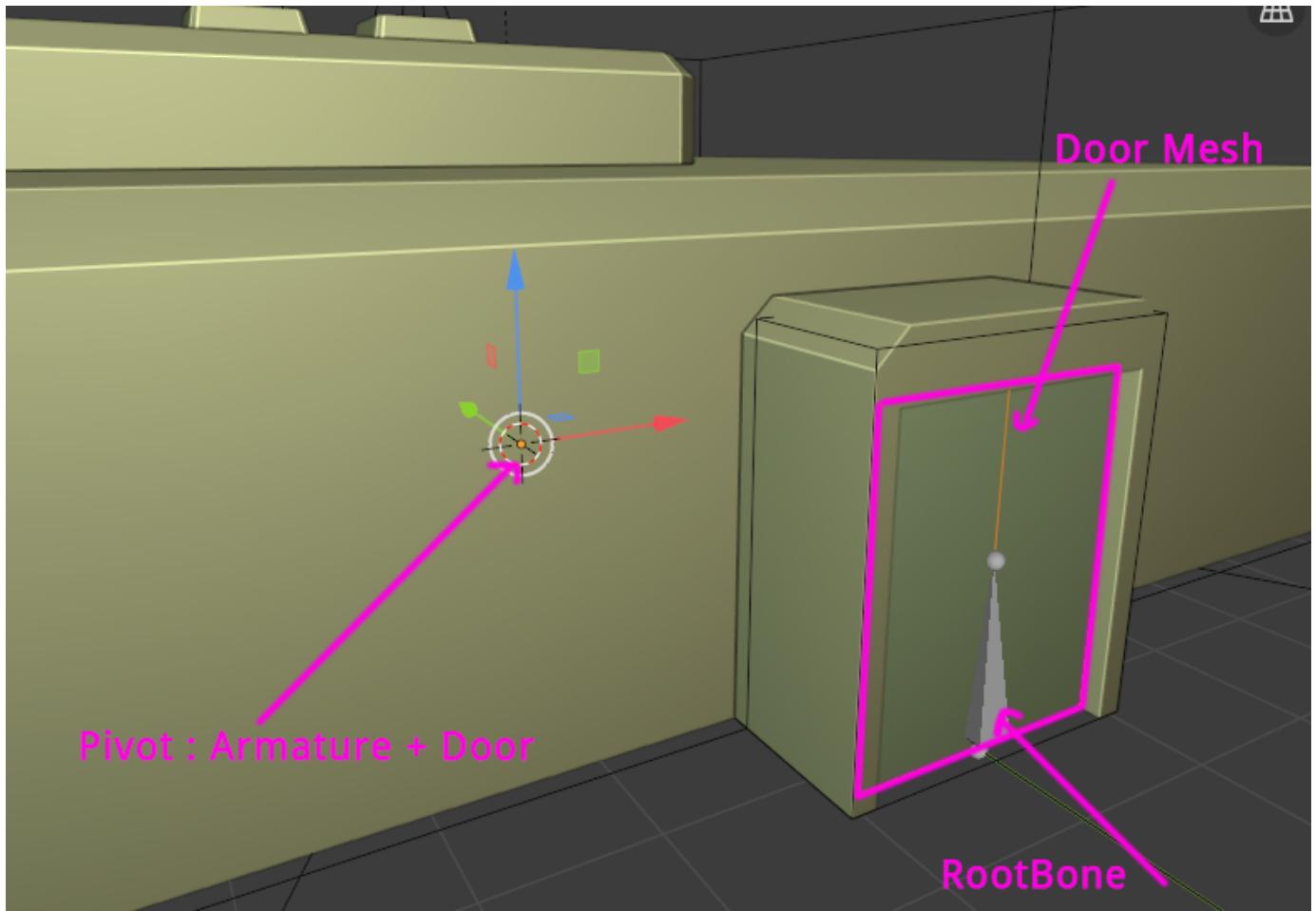
In Blender 2.79b

This step is complex; many errors can be made here. The animated Mesh must be a child of the Armature. This Armature has only 1 Bone, I named this “RootBone” (you can set more Bones if you have a complex animation). You can find more information on Internet about Animations made with Blender.





Caution The Pivot point of your Animated Mesh must be on the Pivot point of the Armature

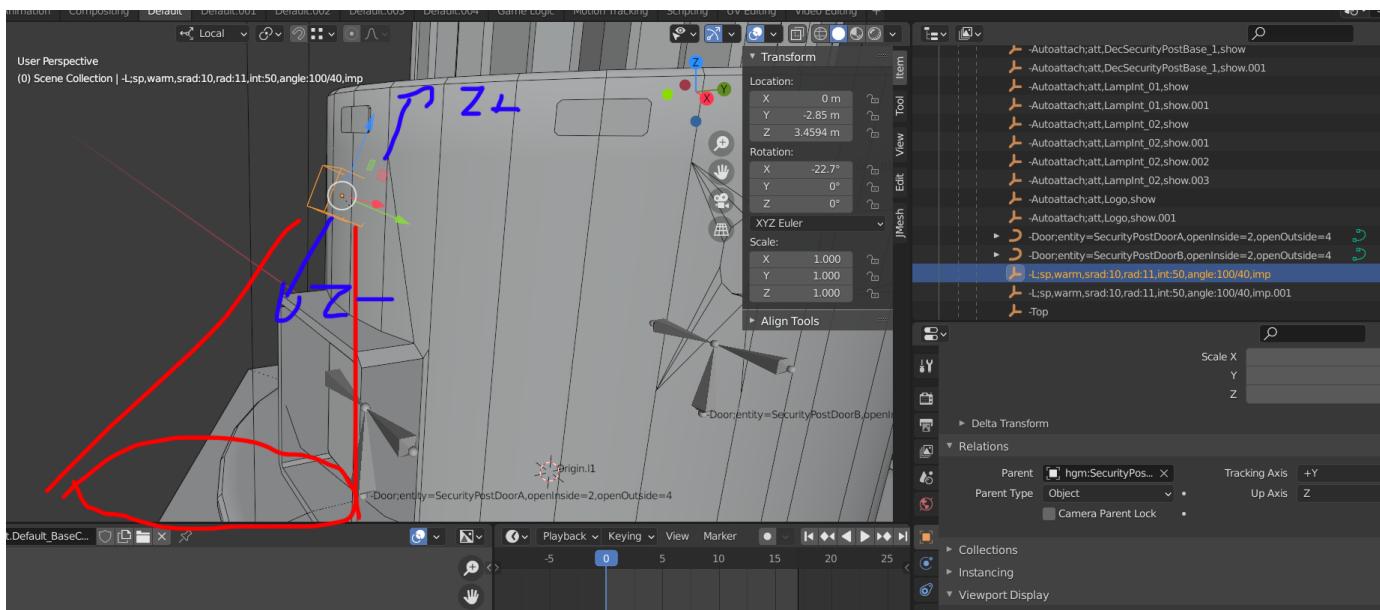


Lighting

Don't use the Light Objects from your 3D Software, this is another Metadata with some specific values. So, a simple Empty or Dummy Object is required.

Spot Light

- Z- Direction of the light
- `-L:sp,[color name],srad:[in cm],rad:[in m],int:[from 0-200],angle:[angle of the outer cone]/[angle of the inner cone],imp`
- The *color* parameter can have one of three values: *warm*, *neutral* or *cool*.
- *rad* – the attenuation radius of the light.
- *int* – the intensity of the light.
- *srad* – Optional; Defines the physical dimensions of the light source (its radius).
- *imp* – Optional; If marked as important, the light will always be turned on at night. Otherwise the game might ignore it for optimization purposes.
- *sim* – Optional; All lights marked with *sim* will light up simultaneously, instead of one by one in random order (think of apartment building rooms vs. vehicle headlights).



Point Light

- To create a point light (spherical), the spot name must follow this pattern (without square brackets):

-L:[color name],srad:[in cm]:rad:[in m],int:[from 0-200],imp,sm

Blender Settings – Exporting

 Remember: Use **Blender 2.79b** to set up your Asset!

Material settings

If there is a Material (*from your 3D software*) on your Mesh remove it and add new one with Blender.

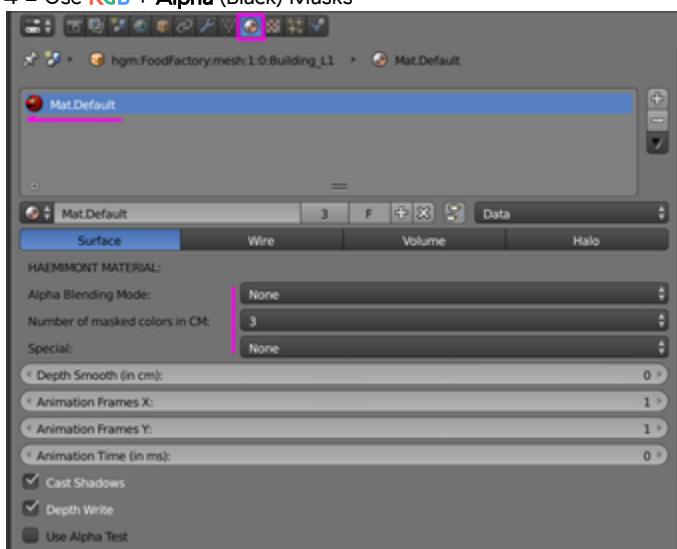
Number of Masked colors in CM is an important property to set the Colorization Map.

1 = Use **Red** Mask only **OR No Colorization map**

2 = Use **Red + Green** Masks

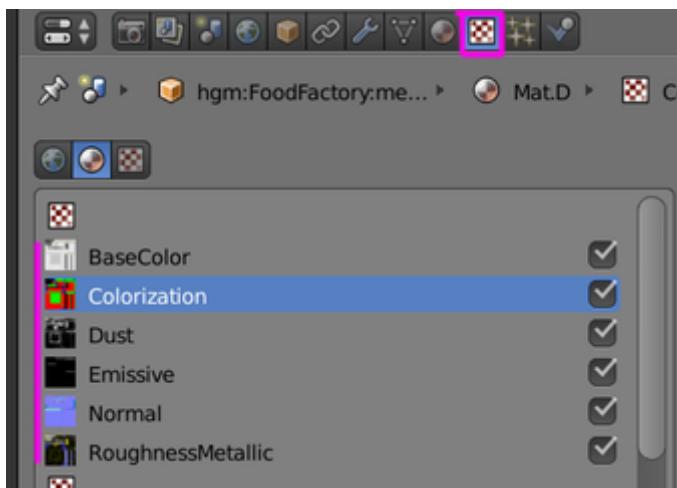
3 = Use **Red + Green + Blue** Masks

4 = Use **RGB + Alpha (Black)** Masks

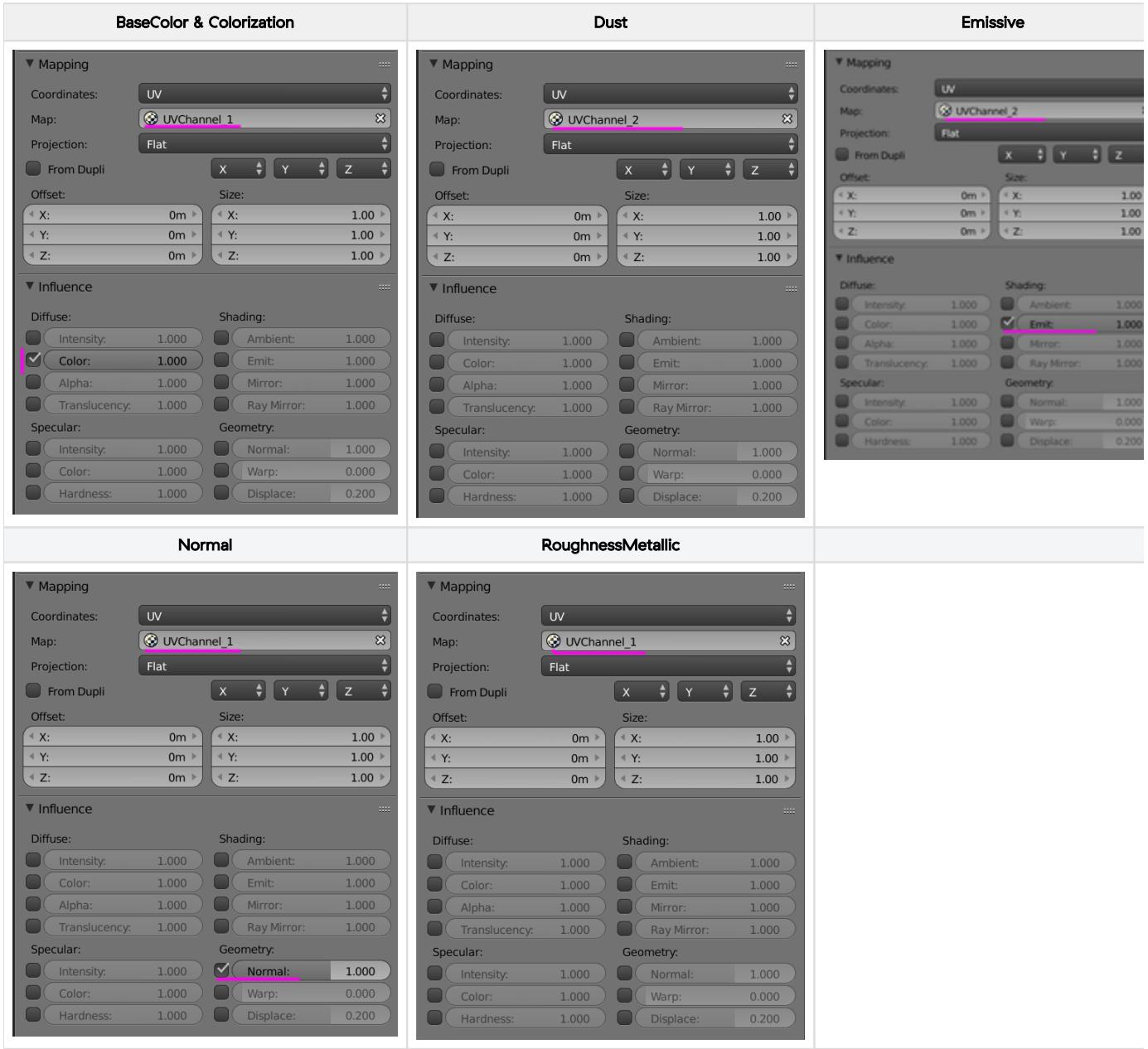


 This part is added by the Plug-in. If it is missing, make sure you've installed it properly.

Texture Names



Texture settings



⚠️ Obviously don't forget to add the Material for each Mesh (LODs, Door...) you can use the same material if the other meshes share the same UV Map. Otherwise create a new Material.

Set the Animations

If your Asset doesn't have any animations you can skip this part.

In the **HGE Tools Tab**

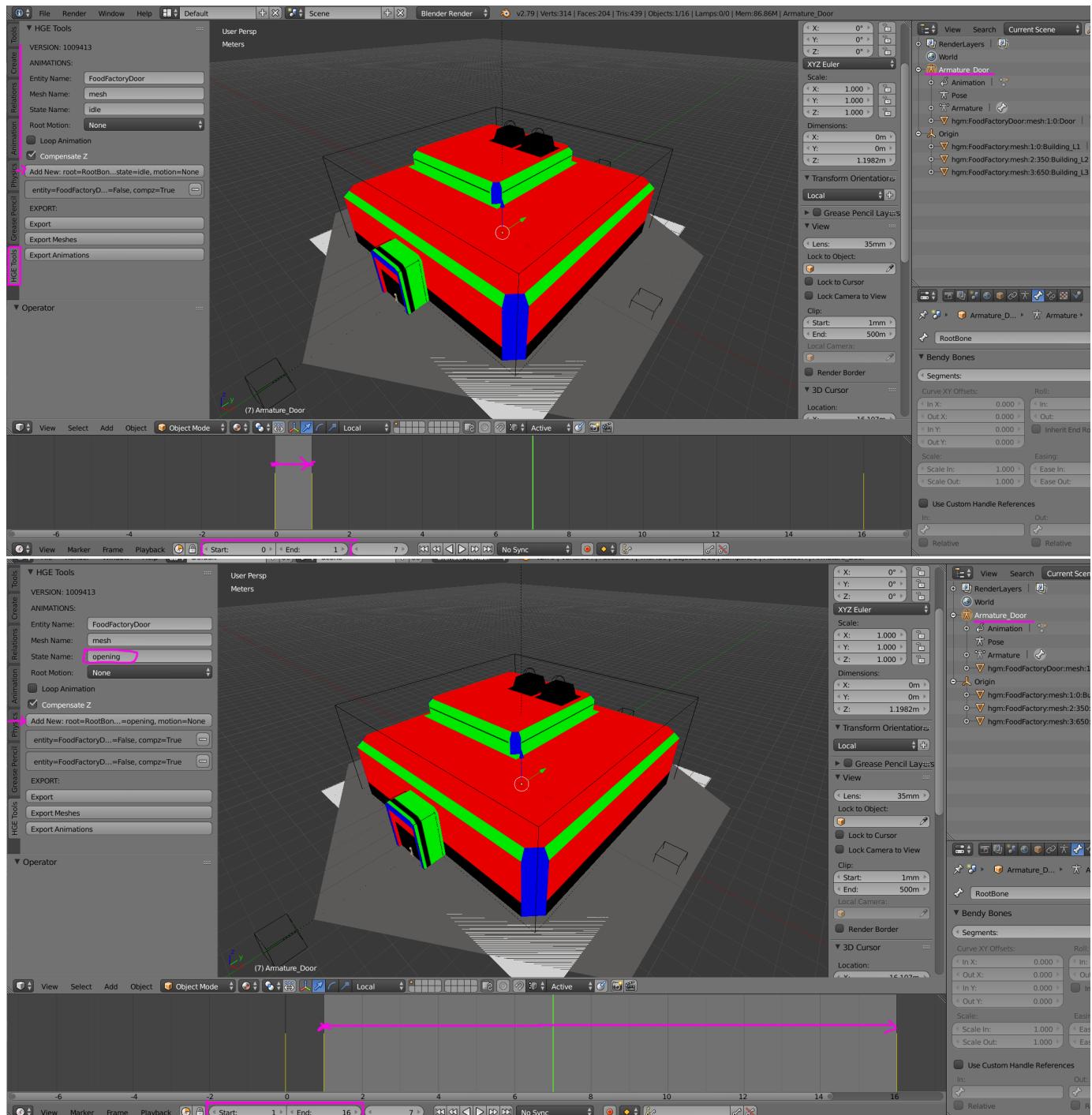
1. Select your Armature
2. Set your Entity, Mesh Name
3. Set the name of the first animation : by default all Assets have the “idle” animation
4. Set the Frame Range, here our idle animation start at 0 to finish at 1
5. Click on **“Add New ...”** to add the animation state
6. Repeat the Steps N°3 to N°5 (changing the State Name & Frame Range)

Info There are some Default states in-game.

Door default states : idle, opening

Building default states : idle, start, working, end

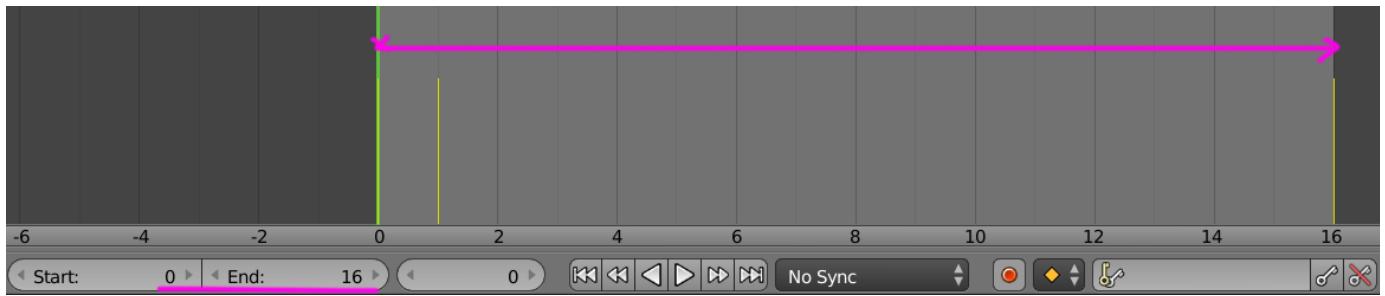
You can add your custom states but you must code their triggers.



Important After setting each Animation, you must set the Frame Range before exporting.

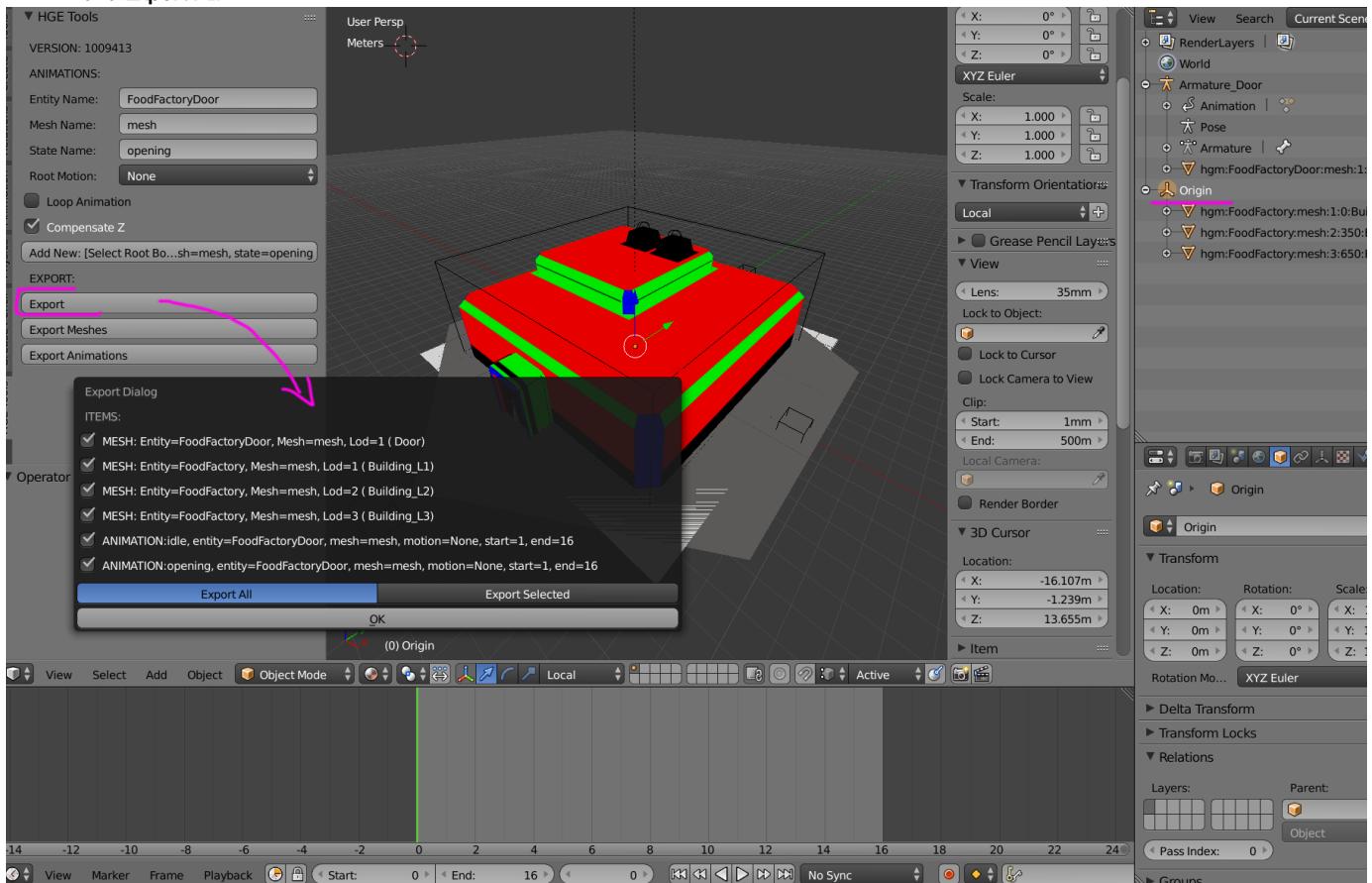
Start must be set on the **first** frame of the **first** Animation.

End must be set on the **last** frame of the **latest** Animation.

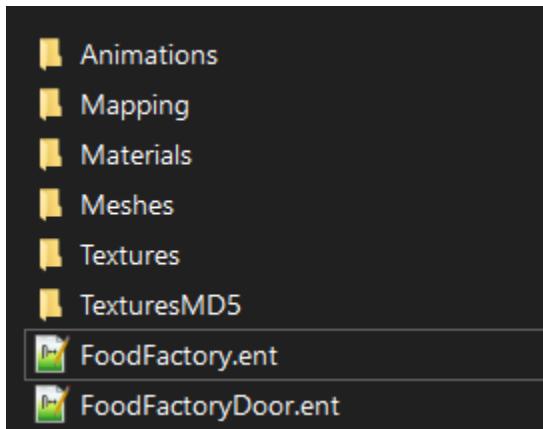


Exporting

1. Select the **Origin**
2. Click on **Export** button
3. and **Export All**



You can find the Exported Files in this folder : AppData\Roaming\Surviving Mars\ExportedEntities

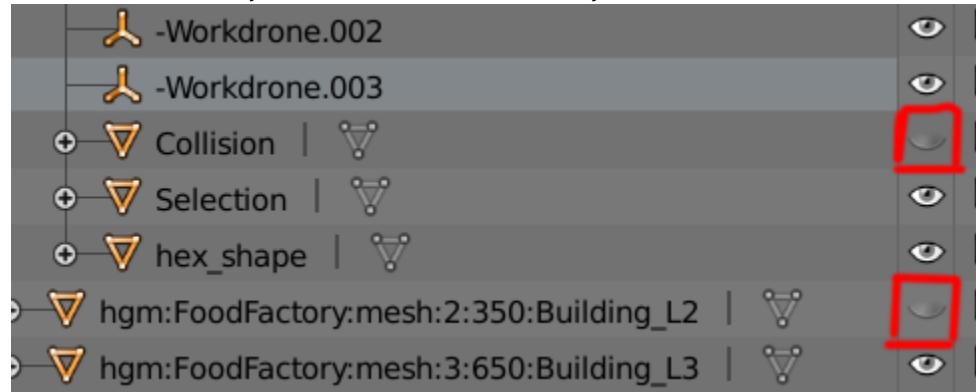


The next step will be Importing these files into your Mod.

Frequent Exporting Errors

If you get an error during the exporting process...

- Check if each object is Visible. Because a hidden object creates an error.



- Check all Texture paths, a missing file texture or a wrong path causes errors.
- Check the Armature and Bone settings, this is a complex part so many errors can be made.

Integrate your Asset In-game

The screenshot shows the Mod Editor application window. The left panel, titled 'Mod Items', displays a loaded mod named 'Food Factory (loaded)' with ID 'wybLV6m' and version '0.00-003'. It lists entities: 'BuildingTemplate FoodFactory', 'Code Script', 'Entity FoodFactory', and 'Entity FoodFactoryDoor'. The right panel, titled 'Item Properties', is set to the 'FoodFactory' entity. It has three tabs: 'Mod', 'Type', and 'General'. The 'Mod' tab contains fields for Comment, Group (set to 'Life-Support'), Id (set to 'FoodFactory'), Copy from group (set to 'Default'), and Copy from. The 'Type' tab contains the Class field set to 'FoodFactory'. The 'General' tab contains fields for Display Name (set to 'Food Factory'), Build Menu Category (set to 'Life-Support'), Build Menu Icon, Build Menu Pos (set to '1'), Entity (set to 'FoodFactory'), Dome Comfort (set to '0'), Dome Morale (set to '0'), Show Range for All (unchecked), Show Range (unchecked), Infopanel (set to 'ipBuilding'), Suspend on Dust Storm (unchecked), Color Modifier (a color swatch), and four Palette color fields (set to 'outside_base', 'inside_accent_food', 'outside_metal', and 'none').

Open the **Mod Tools** and create your new **Mod**.

Import the Entities

First of all, you must import each Entity. In this example, there are two entities **FoodFactory** and **FoodFactoryDoor**.

1. Click on "**New Mod**" in the top menu to add a new item and select **Entity**

2. Select the item in the left column and Click on ... to open the file browser and find your .ent files in **AppData\Roaming\Surviving Mars\ExportedEntities** and **Import**
3. Repeat each step for each Entity

Now your entities are imported into your Mod and the Game.

Add a Script

The LUA script is essential to define your new Building and its Attachments. You can also customize and design your own functions here.

1. Click on "**New Mod**" in the top menu to add a new item and select **Script**
2. Select the item in the left column and Click on **Open** (*You can edit with Notepad*)
3. And write your code here

To test your Building quickly you must add the base.

```
DefineClass.FoodFactoryDoor = {
    __parents = {"Door"}, -- This parent class add functions to open
    /close the door
    entity = "FoodFactoryDoor",
}

DefineClass.FoodFactory = {
    __parents = {"Building"}, -- This parent class contains various
    basic functions for the buildings

    fx_actor_class = false, -- Reset the FX Class
    -- some properties here
}

function FoodFactory:GameInit()
    self.fx_actor_class = "MetalsExtractor" -- Here we can re-use the
    same FX & Sounds from the Metals Extractor
end

-- write your functions here
```

✖ **self.fx_actor_class = "MetalsExtractor"** The Metals Extractor has some -Smoke spots to show a smoke FX. Your Asset must have one or more -Smoke spots too (same name).

Add a BuildingTemplate

Finally add the **BuildingTemplate** to set each property (*production consumption cost, name, upgrades...*)

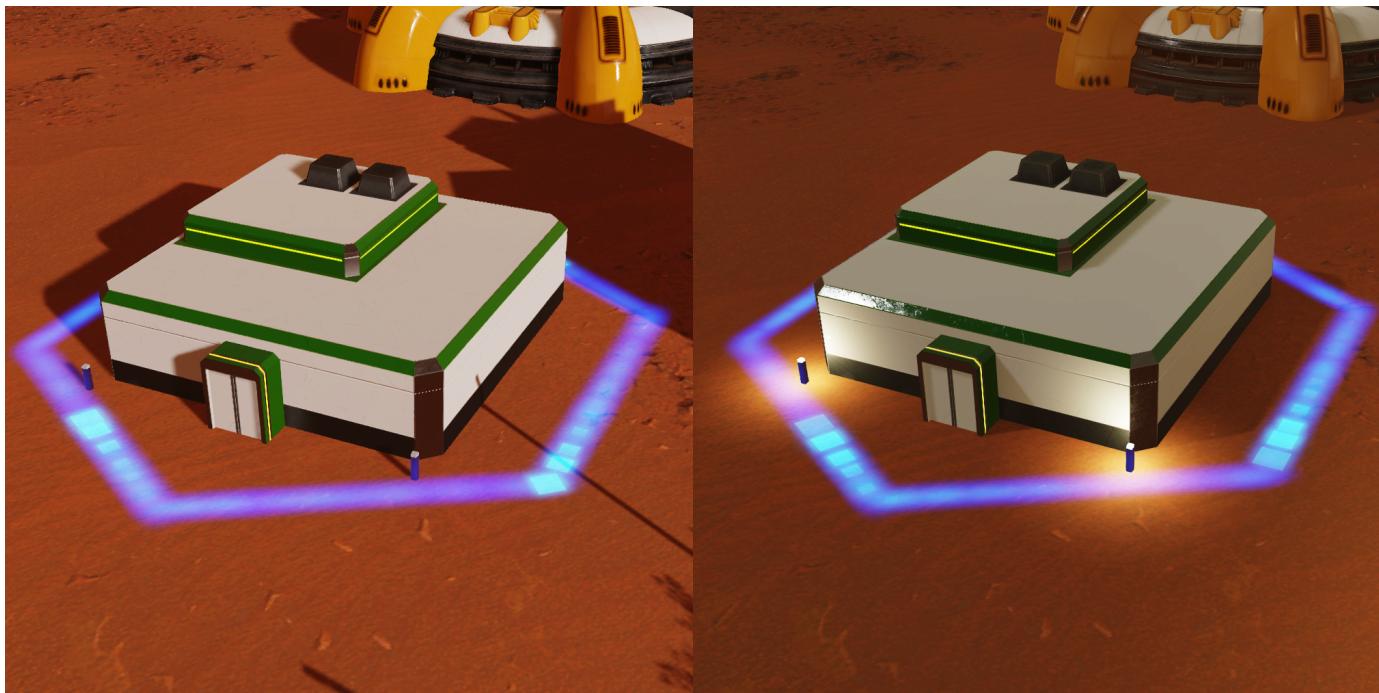
1. Click on "**New Mod**" in the top menu to add a new item and select **BuildingTemplate**
2. Select the item in the left column and Set it

Most important properties

- **ID** – Must be Unique
- **Class** – Set an existing class or your own class you defined before
 - If your custom class is not available in the list, select your Script and click on the "Test" button in the tool bar to reload the LUA file.
- **Build Menu Category** – Your asset becomes available to build

- **Entity** – Set your Entity (3D Model) here.
- **Palette Color 1 to 4** – for the Colorization Map

ⓘ **Test your Mod :** **Save** it before and **Reload** the Game. **Check** in the Mod Manager if your mod is enabled properly.



Extra : Pipe Connection

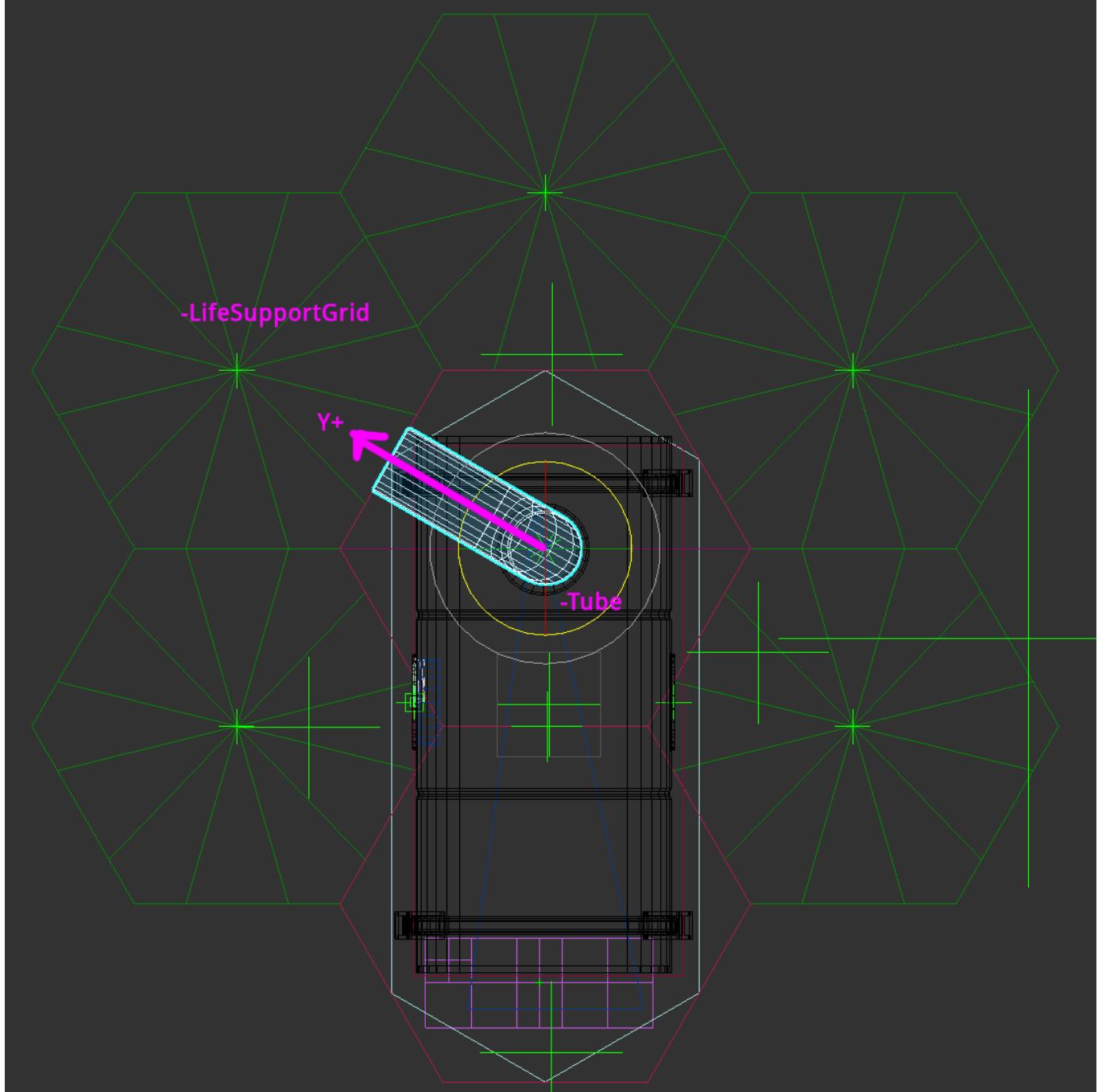
To add a connection pipe on your Asset, you must add 2 new metadata. It's not necessary to create a new tube model; we can re-use the same one from the game (Recommended)

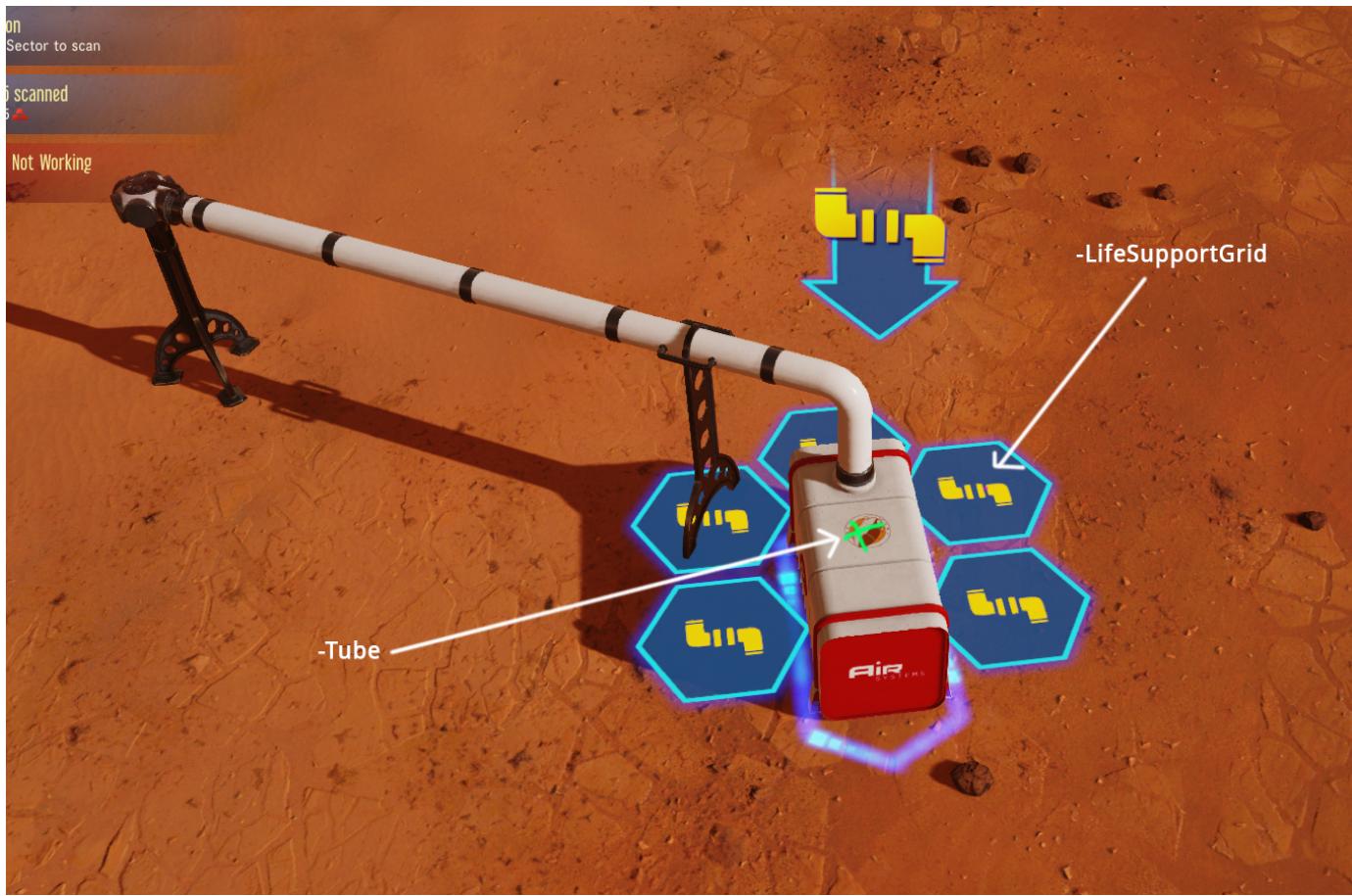
-LifeSupportGrid is placed out of the footprint, and must be **in the middle of the hex at Z 0**. This metadata creates a Pipe connection on the ground.

-Tube is placed in the footprint and must be **in the middle of the hex at Z 0** near a **-LifeSupportGrid**. **Y+** is the direction of the Pipe. This metadata creates a Pipe.

 Each **-LifeSupportGrid** must have a **-Tube** in the right direction to create the connection.

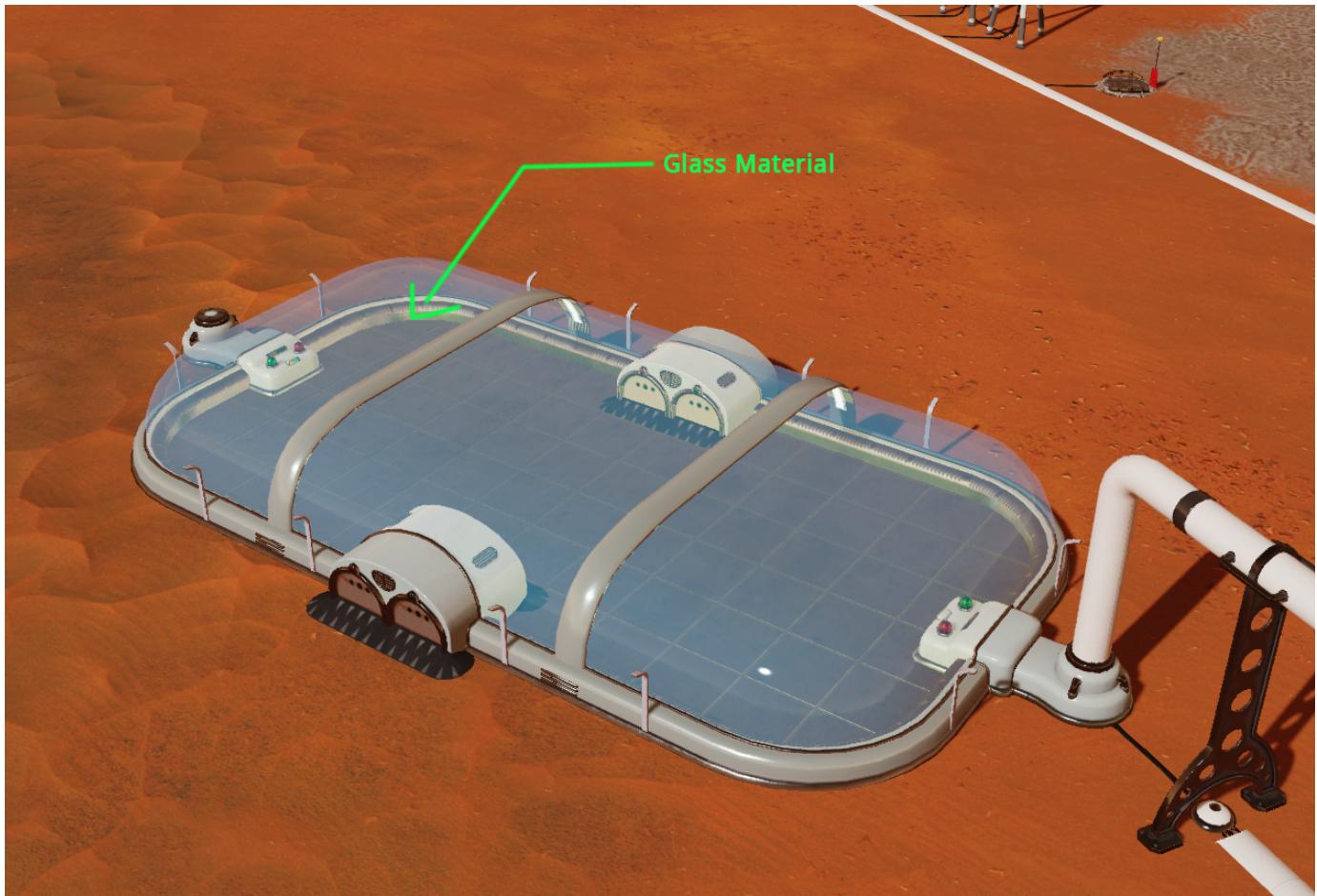
Example : Air Tank has 6 **-LifeSupportGrid** and 6 **-Tube**





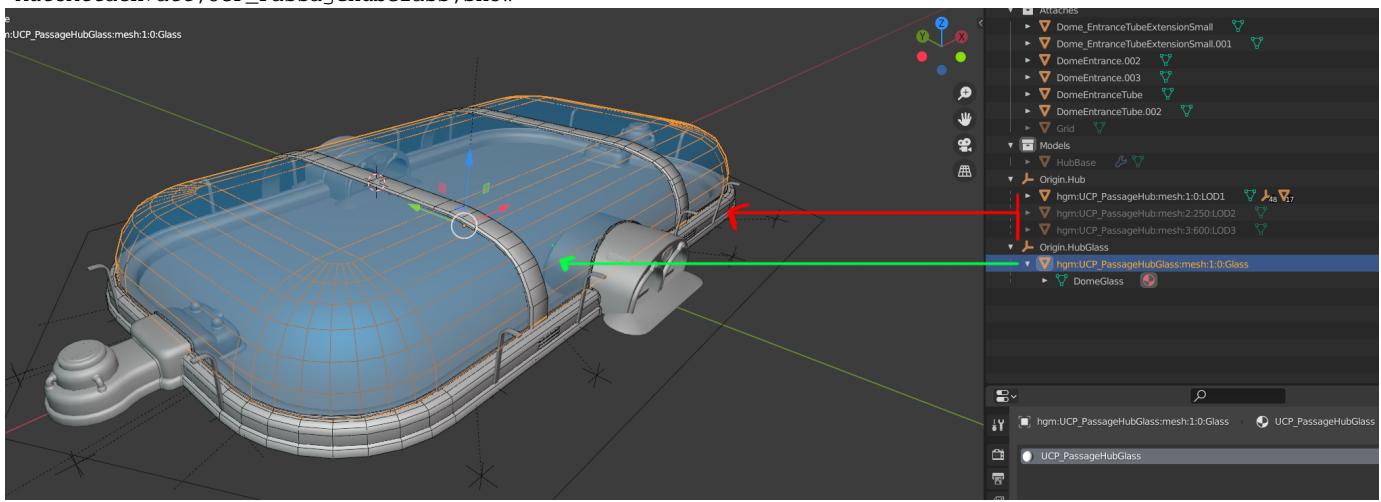
Extra : Transparency Material

How to create a Window or Dome glass Material ?



The Glass must be an separated mesh (Entity), that is not included in your main mesh because they use different Material settings. We will use an AutoAttach metadata to add your Glass into the Building Entity.

-AutoAttach;att ,UCP_PassageHubGlass ,show



HAEMIMONT MATERIAL:

- Alpha Blending Mode: Glass
- Number of ...lors in CM: 1
- Special: None
- Depth Smooth (in cm): 0
- Animation Frames X: 1
- Animation Frames Y: 1
- Animation Time (in ms): 0
- Cast Shadows
- Depth Write
- Use Alpha Test
- ▶ Preview
- ▶ Diffuse
- ▶ Specular
- ▶ Shading
- ▼ Transparency

Mask	Z Transparency	Raytrace
Alpha: 0.350	Fresnel: 0.400	
Specular: 1.000	Blend: 1.250	

BaseColor

- Dust
- Emissive
- Normal
- RoughnessMetallic

BaseColor

Type: Image or Movie

Source: Single Image

Can't Load Image

Color Space: sRGB

View as Render

Use Alpha

Alpha: Straight

Fields

Upper First Lower First

▶ Image Sampling

▼ Image Mapping

Extension: Repeat

Repeat: Mirror

- Select **Glass** or **Blend** in the **Alpha Blending Mode** property.
- Check the **Transparency** section.

- **BaseColor** map must have a **Opacity** in the TGA file.
- **Normal** is required
- **RoughnessMetallic** is required
- **Dust**, **Emissive** are set on **Channel_UV2** and **Optional**
- **Colorization** works too if needed.

Extra : Hook Up SoundFX or ParticlesFX

You can add your custom **Triggers** called **ActionFX** in Surviving Mars. An **Action** is a state triggered by your Asset, for example : Turning On a building enabling the “**Working**” state. So you can hook up some SFX and VFX. Each Action is composed by “**Moments**” : **start** Working , **end** Working, and more ...you can play a specific FX when starting or/and finishing an **Action**.

First of all, In your **Script** you must set the `fx_actor_class` property inside the **GameInit** method by the **Classname**.

```
DefineClass.FoodFactory = {
    __parents = {"Building"},

    fx_actor_class = false, -- Reset the FX Class
}

function FoodFactory:GameInit()
    self.fx_actor_class = "FoodFactory" -- String value
end
```

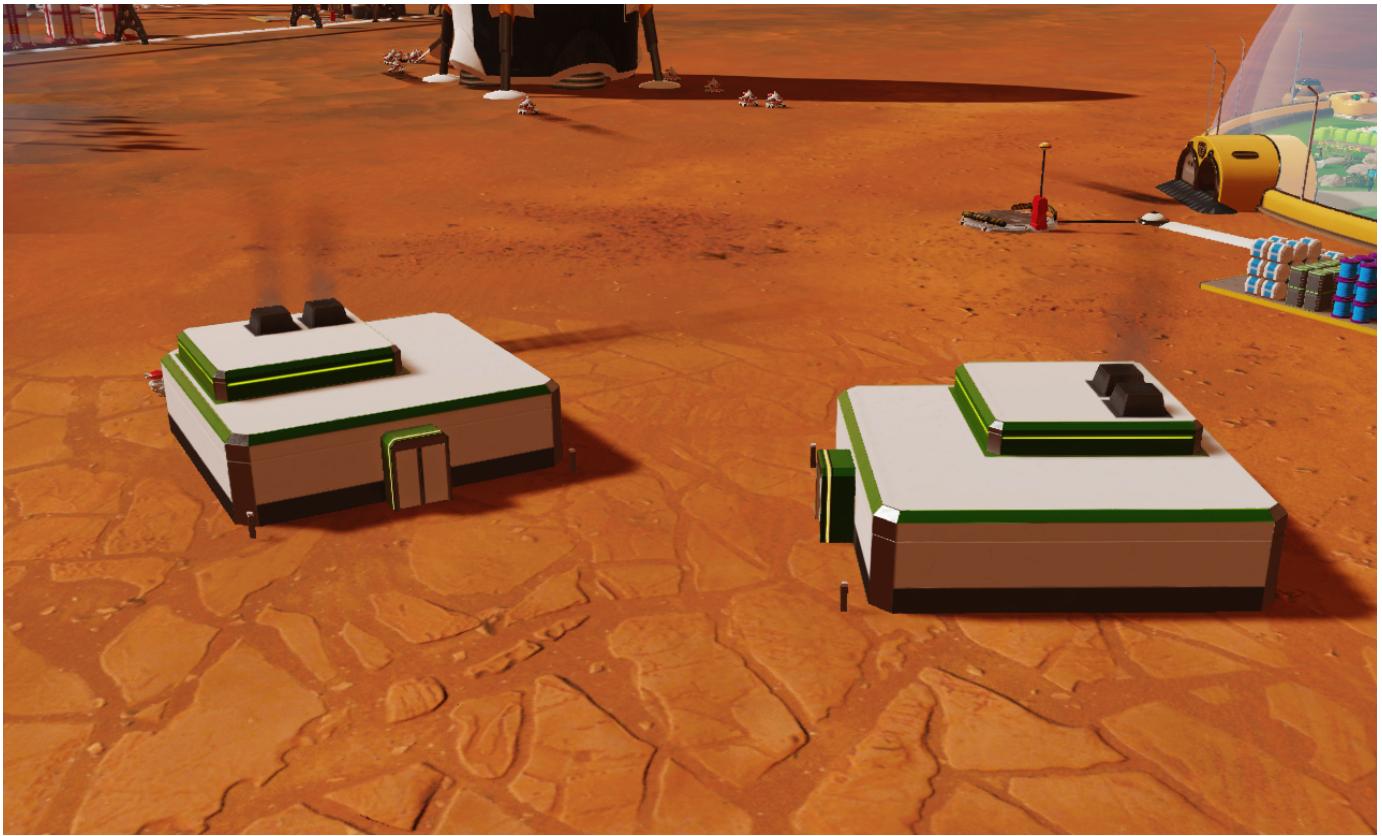
Now you can add your first **ActionFX** with the **Mod Tools** : *New Mod > ActionFXParticles* . Add an **ActionFXSound** to hook up a Sound FX.

Match	
Action	x Working
Moment	any
Actor	x FoodFactory
Target	any
Environment	any
Map	any
FxId	
Detail level category	<unspecified>
Chance	100
Disabled	<input type="checkbox"/>
Solo	<input type="checkbox"/>

Lifetime	
Delay	0
Time	0
EndRules	x (1 objects)
- Action " & Moment 'end' EndAction EndMoment x end	
Behavior	
BehaviorMoment	

Particles	
Particles	x ConcretePlant_Steam
Flags	
AlwaysVisible	<input type="checkbox"/>
Scale	100
ScaleMember	
FollowTick	100

Placement	
Source	Actor
SourceProp	
Spot	x Smoke
SpotsPercent	x 100
Attach	<input checked="" type="checkbox"/>
SingleAttach	<input type="checkbox"/>
Offset	0, 0, 0
OffsetDir	SourceAxisX
Orientation	
PresetOrientationAngle	0
OrientationAxis	x Y



List of ActionFXSounds

There are many Actions already programmed, you can also add your own Actions (states) through code. There are some essential Actions where you can hook up a specific sound or a particles system.

This is the ActionFXSound settings from the Metals Extractor building.

Action	Moment	Source	Sound ID	Note
Breakdown	start	Actor	<i>Object MetalExtractor Fail</i>	Malfunction, Shutdown sound
Destroyed	start	Actor	<i>Object MetalExtractor Demolition</i>	Destroyed, Demolished sound
SelectObj	start	Actor	<i>Object MetalExtractor Select</i>	Sound when selecting the building.
SelectObj		UI	UI SelectBuilding	Default UI Sound when Selecting or Spawning a building
Spawn	start	UI	UI BuildingComplete	
Working	start	Actor	<i>Object MetalExtractor LoopStart</i>	Start working loop
Working	start	Actor	<i>Object MetalExtractor Loop</i>	Working loop sound (<i>this sound must be repeated</i>)
Working	end	Actor	<i>Object MetalExtractor LoopStop</i>	End working loop

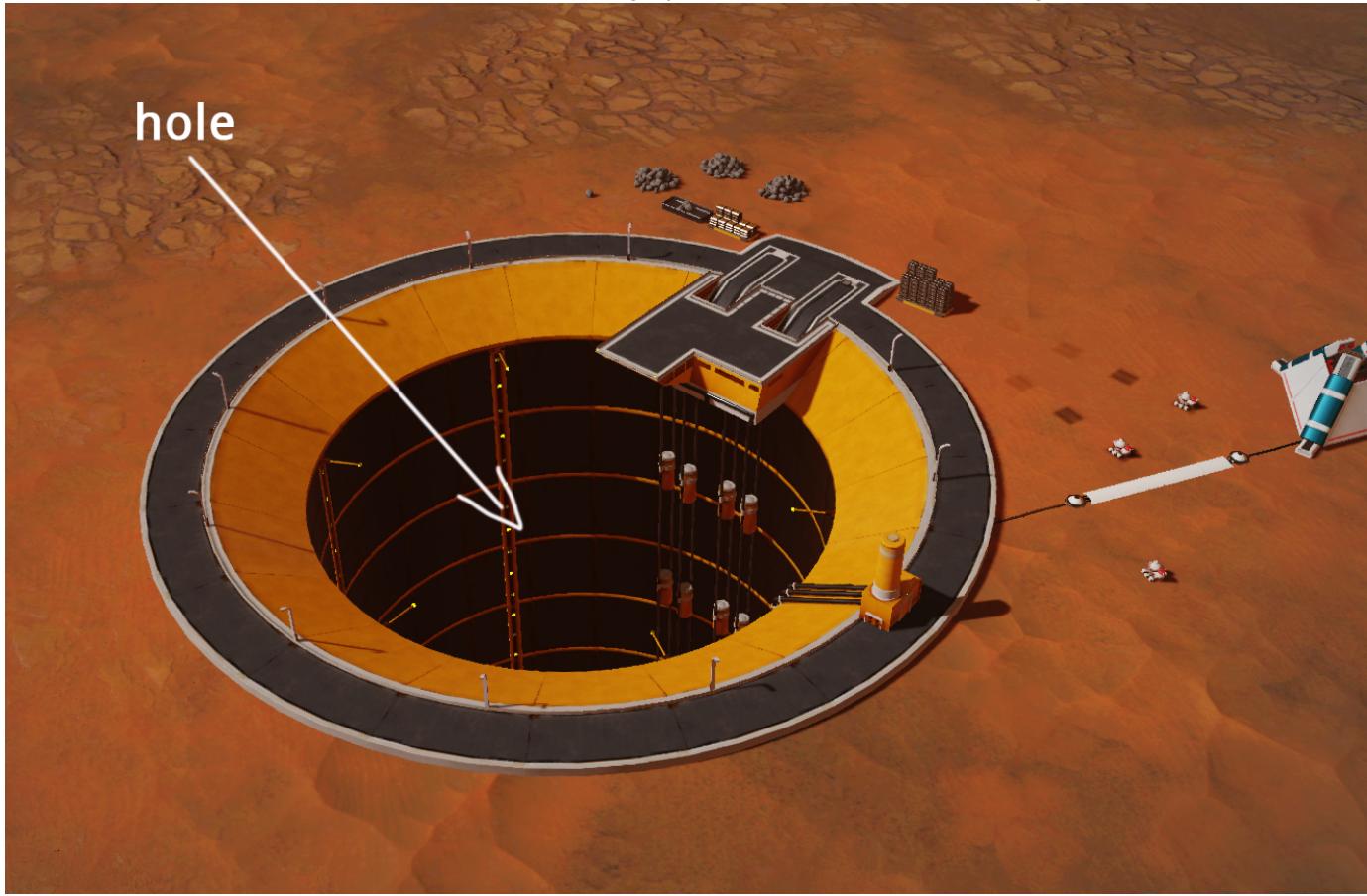
 You can find the existing Sound & Particles IDs in the code reference files (**ActionFXSound.lua** & **ActionFXParticles.lua**) in this folder : **Data/FXPreset/**

```
9040 PlaceObj('ActionFXParticles', {  
9041     Action = "Working",  
9042     Actor = "ConcretePlant",  
9043     Attach = true,  
9044     DetailLevel = 40,  
9045     EndRules = {  
9046         PlaceObj('ActionFXEndRule', {  
9047             'EndMoment', "end",  
9048         }),  
9049     },  
9050     Moment = "start",  
9051     Particles = "ConcretePlantCylinder_Steam",  
9052     Source = "Target",  
9053     Spot = "Cylinder",  
9054     Target = "ConcretePlantCylinder",  
9055     handle = 1084202036,  
9056 })
```

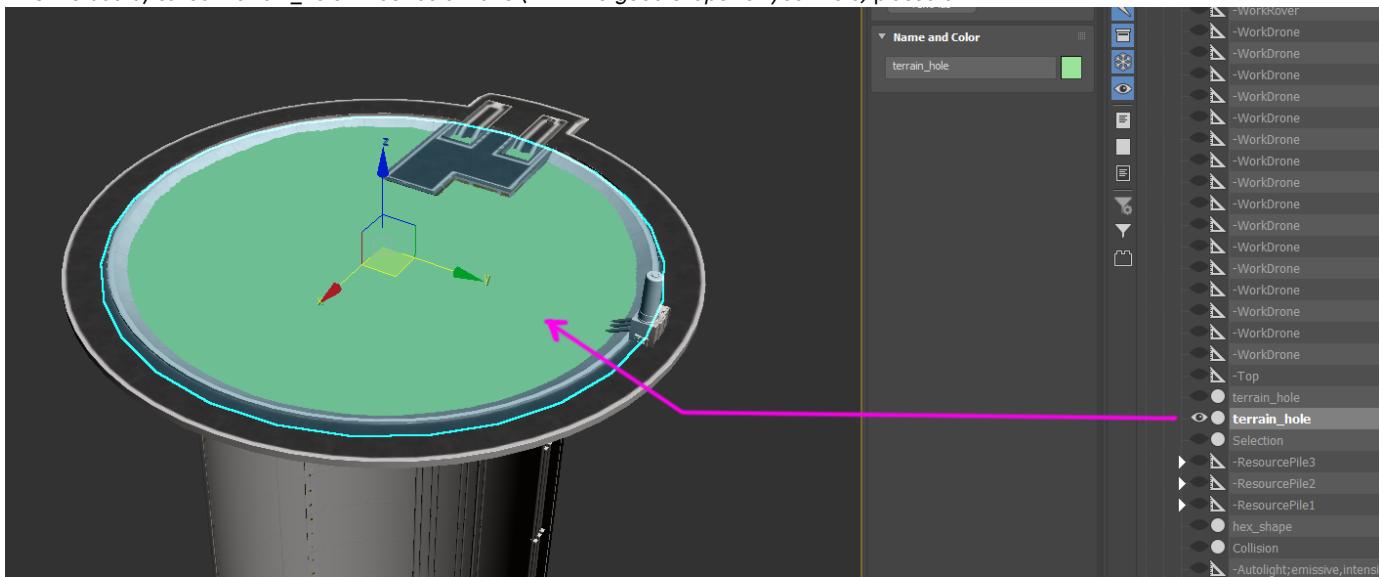
```
848 PlaceObj('ActionFXSound', {  
849     Action = "Breakdown",  
850     Actor = "ProjectMorpheus",  
851     Moment = "start",  
852     Sound = "Object ProjectMorpheus Fail",  
853 })
```

Extra : Terrain Hole

How to add a hole in the Terrain; this feature could be interesting if you want to create a Mineable Building.



This metadata, called “terrain_hole” must be a Plane (with the good shape for your hole) placed at Z0.

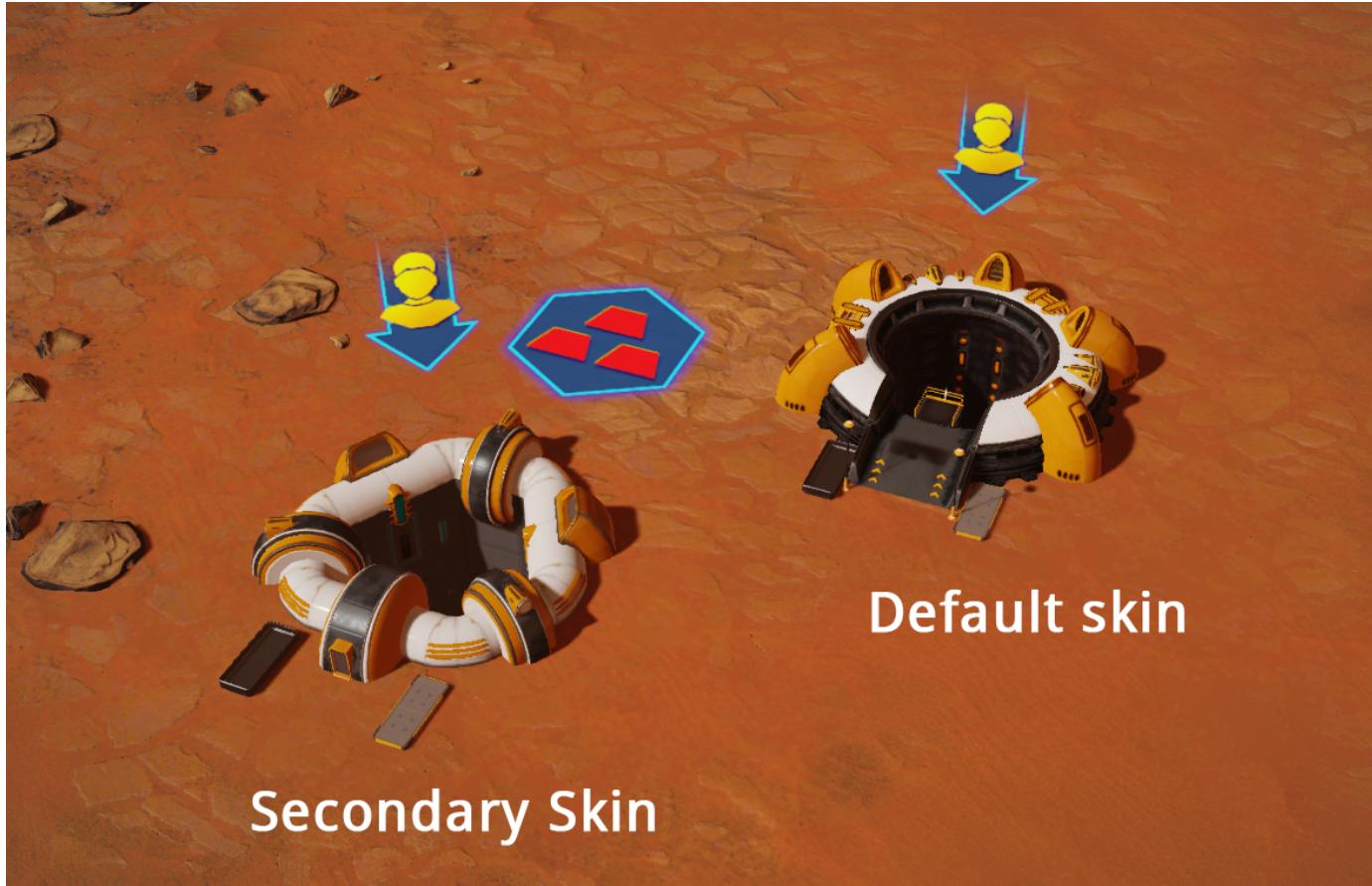


That's all !

Extra : Create a Building Skin

Use the same process to create a new Skin, but there are various constraints.

Surviving Mars allows 6 secondary Skins per Building, they are set through the Building Template.



Constraints

Some metadata from the Default Skin must be preserved for all secondary skins.

- Pivot Point
- Footprint
- All Specific Spots used by the script
- All Spots used by the VFX
- The pipe connections must have the same positions
- Animations can be modified but it's not possible to add new states

All other things can be changed or removed.

Set the Skin

If you create a new skin for an existing Building from the Game or from another Mod you must duplicate the **Building Template** and give the same **ID** to overwrite the original template.

In the Building Template find the “**Alternative Entities**” section, choose a free slot and set the **Entity** and the **Palette Colors** (if your skin uses the colorization), ignore the **DLC** property.

Alternative Entities

Alternative Entity 2	x MetalExtractorCP3	<input type="button" value="▼"/>
Alternative Entity 2 DLC	x contentpack3	<input type="button" value="▼"/>
Alternative Entity 2 Palette Color 1	x mining_accent_1	<input type="button" value="▼"/>
Alternative Entity 2 Palette Color 2	x mining_base	<input type="button" value="▼"/>
Alternative Entity 2 Palette Color 3	x outside_metal	<input type="button" value="▼"/>
Alternative Entity 2 Palette Color 4	x outside_dark	<input type="button" value="▼"/>
Alternative Entity 3		<input type="button" value="▼"/>
Alternative Entity 3 DLC		<input type="button" value="▼"/>
Alternative Entity 3 Palette Color 1	none	<input type="button" value="▼"/>
Alternative Entity 3 Palette Color 2	none	<input type="button" value="▼"/>
Alternative Entity 3 Palette Color 3	none	<input type="button" value="▼"/>
Alternative Entity 3 Palette Color 4	none	<input type="button" value="▼"/>