Reviser: Akin C. Start:27.05.2025

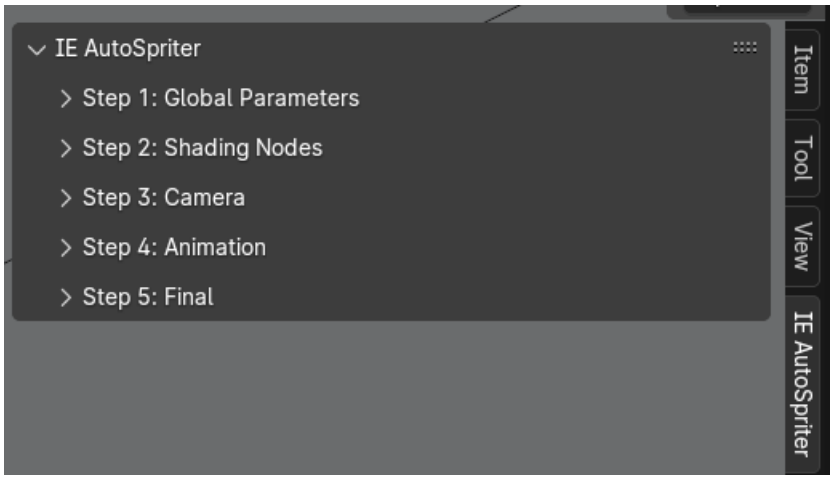
Index

Note: The content of this manual may change and therefore may not be applicable to every version of IE AutoSpriter. I'm also not a professional Blender user or Blender add-on developer. I learn by doing.

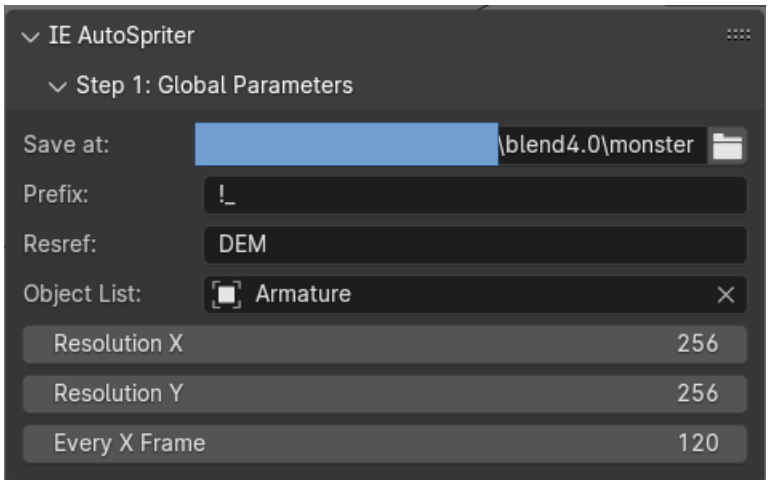
IE Autospriter Interface

This Blender add-on automates the process of rendering sprites specifically for Infinity Engine animations. The workflow is divided into logical steps to guide the user, though the sequence of the first four steps is flexible. Step 5 initiates the rendering process.

Closed form



Step 1: Global Parameters

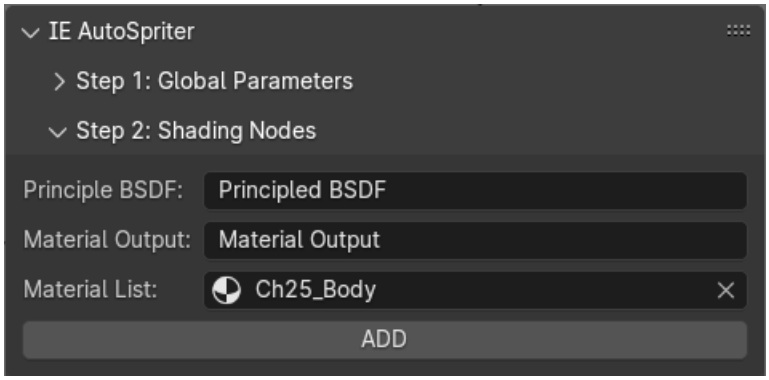


This step defines general settings that apply across the entire sprite rendering process.

- **Sprites Location:** Defines the directory where the generated sprite files will be saved.

- **Modder Prefix:** A user-definable prefix that will be added to the beginning of the sprite file names.
- **ResRef (Resource Reference):** This is the freely definable part of the sprite file name. Other parts of the file names are fixed and automatically generated by the add-on.
- **Resolution X:** This refers to the image width resolution
- **Resolution Y:** This refers to the image height resolution
- **Every X frame:** Saves every xth frame as a sprite. This can be useful, for example, when there are too many frames to save as sprites.

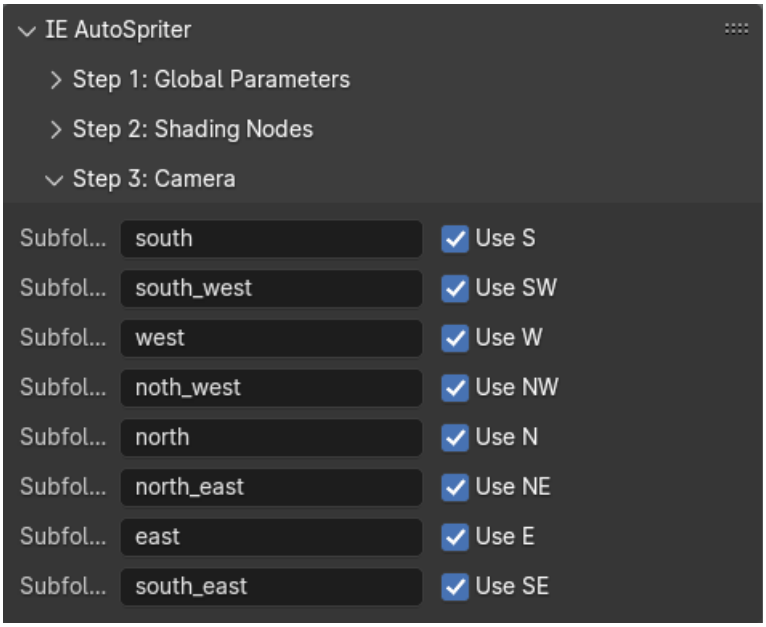
Step 2: Shading Nodes



This optional step allows for the addition of specific shaders to the material shading setup. This is crucial for Infinity Engine sprites, which require indexed color palettes.

Shader Node String Inputs: These input fields are used to specify the names of existing shader nodes within the material. The new, automatically added shader nodes will be placed and connected between these specified nodes.

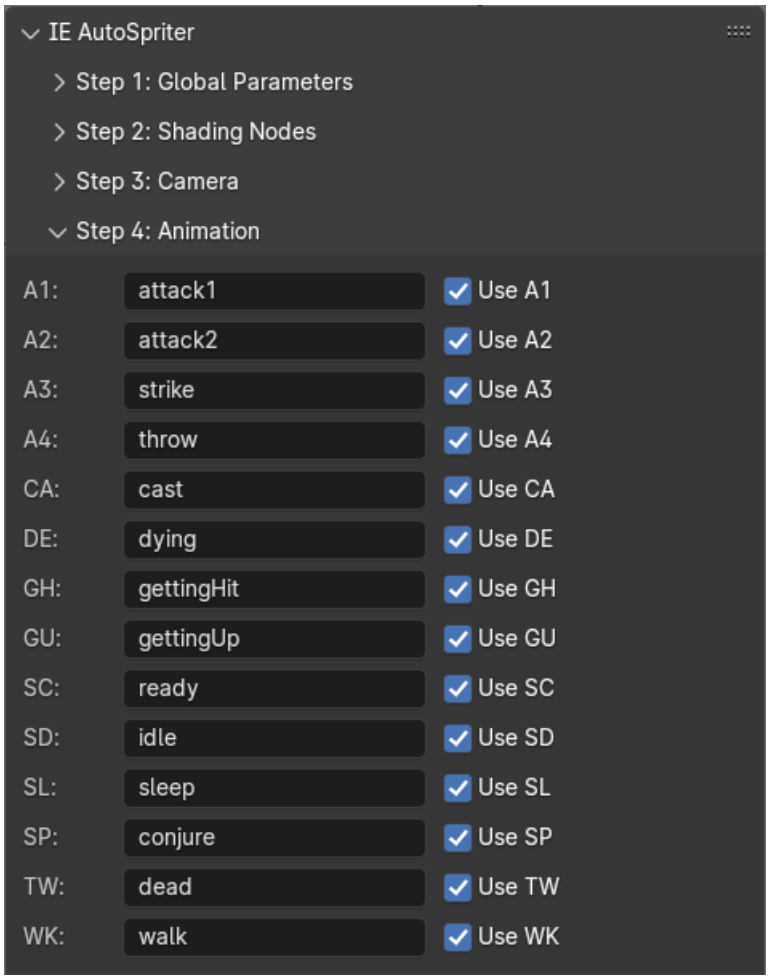
Step 3: Camera



This step manages the output folders and defines which camera orientations will be rendered.

- **Orientation Folders:** These checkboxes determine whether sprites for a specific orientation will be rendered and saved into their corresponding designated folders. If a checkbox is not activated, sprites for that particular orientation will be ignored and not rendered.

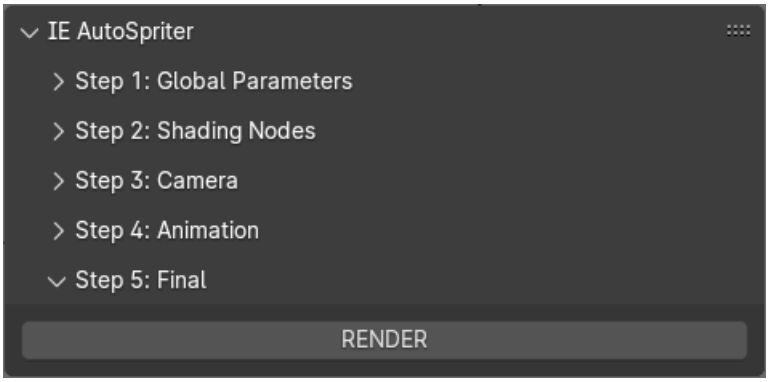
Step 4: Animation



This step defines which animations (Blender Actions) should be rendered and how they are named in the output.

- **Animation(Blender Action):** This refers to the type of animation, corresponding to an "Action" in Blender (e.g., "A1" for Attack Animation 1).
- **Checkbox:** If activated, the specified animation type (e.g., "A1") will be appended to the sprite file name.
- **String Input Field:** The name entered in this input field must precisely match the name of the corresponding Action in Blender.

Step 5: Final

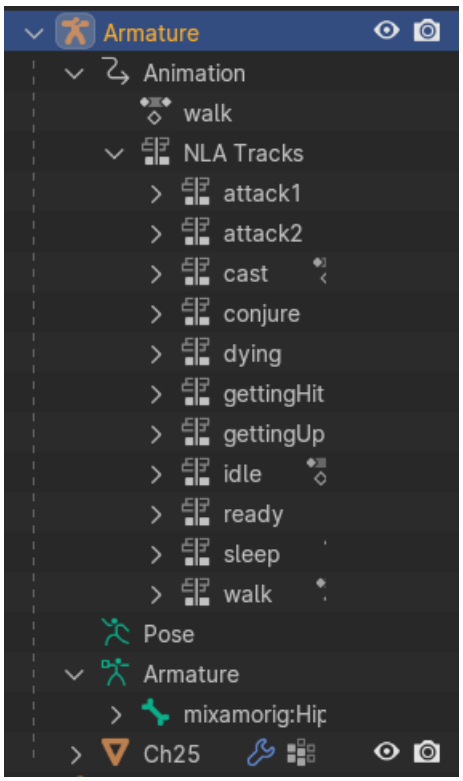


This is the concluding step that triggers the rendering process.

Render: Upon activation, this step initiates the rendering and saving of all sprites according to the settings configured in the preceding steps.

Requirements(See blend files for ~~demo and~~ saves)

It is necessary that the animation is listed in the **Armature** within the **Animation** object. This means that under **Animation** the actions are collected in the **NLA Tracks** (see example in the image below).

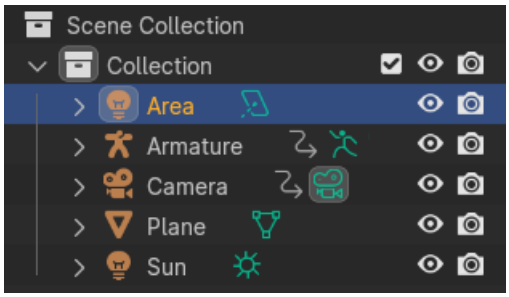


The reason for this is that **IE AutoSpriter uses this list to sequentially specify actions (type of**

animation) to render.

In addition, the names of the animation folders in **step 4** must exactly match the names found in the animation object (see **NLA Tracks**).

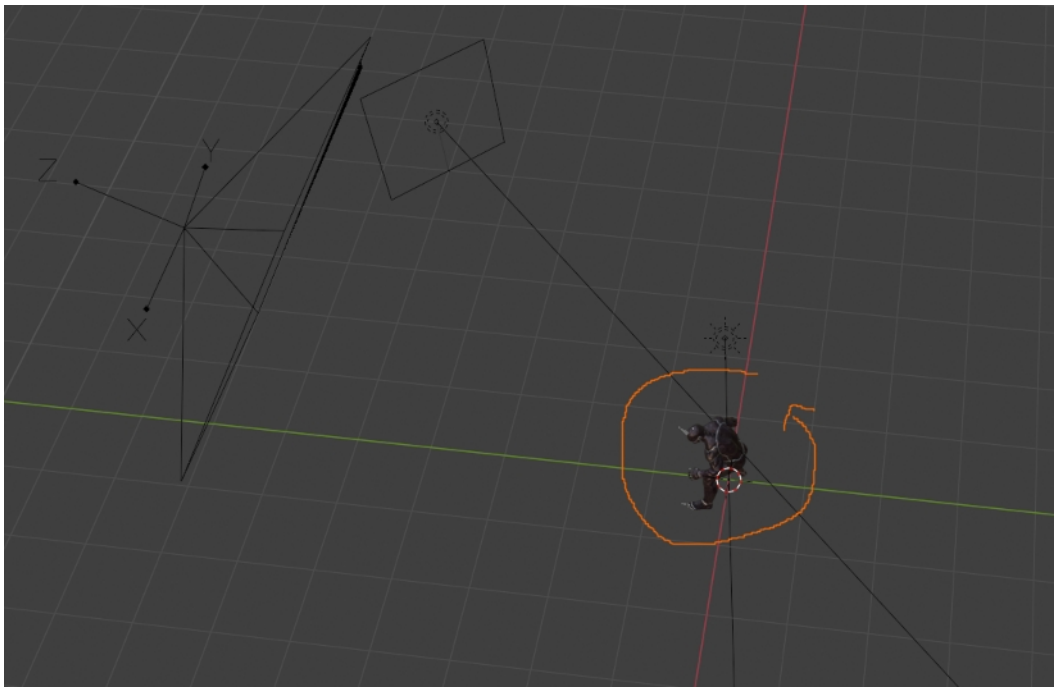
The blend file needs the following setup:



- **Camera**: This is needed to render the creature model.
- **Armature**: This has already been discussed.
- **Plane**: This is needed to create a shadow for the creature model.
- **Area**: This is a light source that should be adjusted as desired.
- **Sun**: This is needed to create a shadow on the **Plane**.

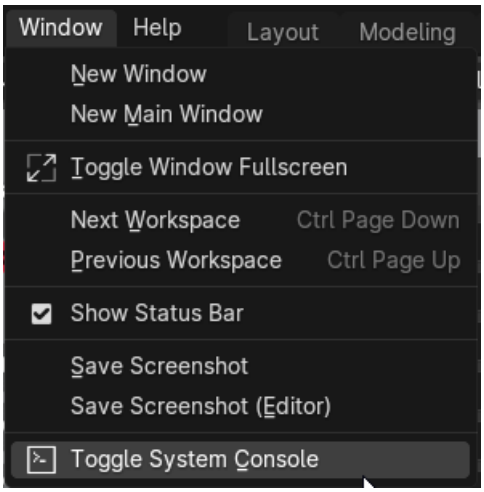
Helpful Insights

The different angles of the creature model are not created by moving the camera, but by rotating the model while the camera remains in a fixed position.



The objects in the collection have a number of specific important parameters that are not covered here, as the details can be found in the save files (these may be covered in later versions of IE AutoSpriter).

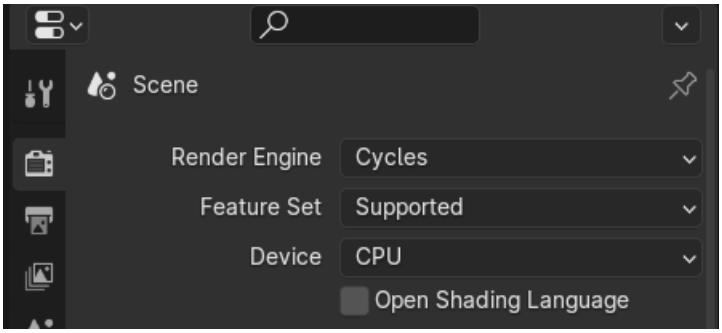
To view the rendering progress and elapsed time, go to "**Toggle System Console**" (see image below). This should be done **BEFORE** rendering, as rendering makes it impossible to interact with Blender.



Save file 4.0

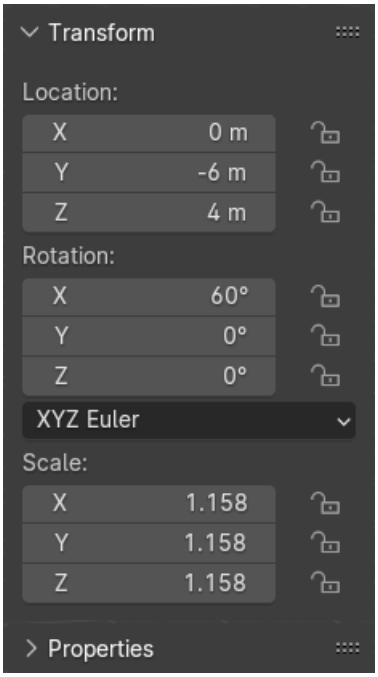
Found in „**IE-AutoSpriter-\save files\!_Template_Blender_4.0.blend**“

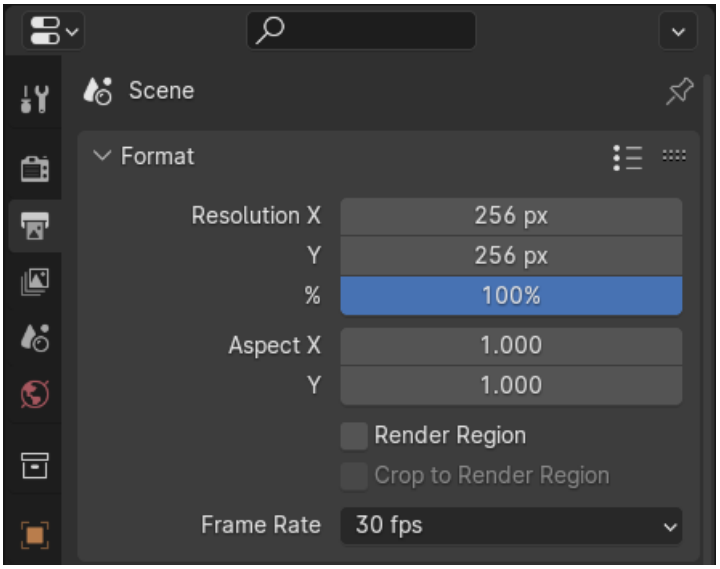
Render Engine



The rendering engine is set to Cycles (instead of EEVEE). This allows the "**Plane**" object to be rendered transparently without the "**Sun**" object losing its ability to cast shadows.

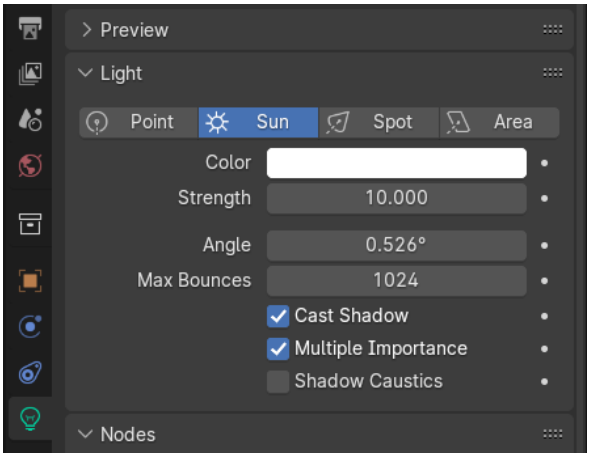
Camera settings

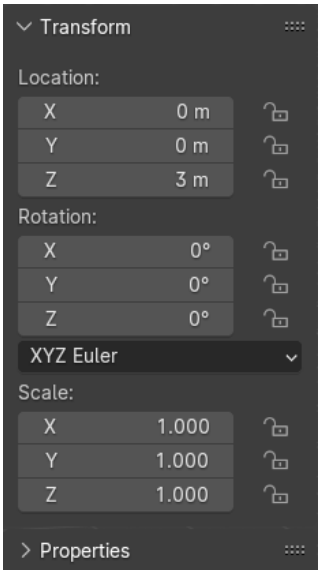




The camera settings can be adjusted to suit your personal preferences if needed. For example, the camera angle is set to 60°, although I've heard arguments for 45°. **After seeing someone else compare an image** of a custom character animation and an in-game animation at 45° and 60°, **I decided to go with 60**°. The rendering resolution is set to 256px to 256px(px stands for pixels) and frame rat to 30fps(frames per second).

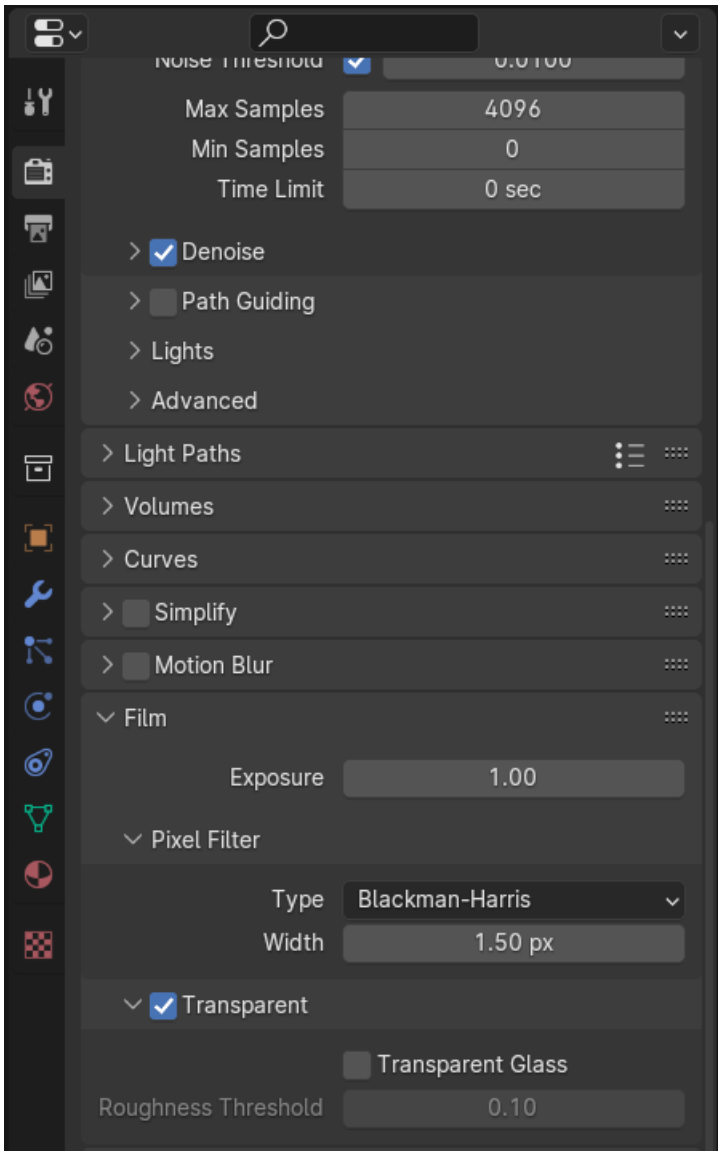
Sun





These settings are optional but worked reasonably well (lumens 10,000 and Z-distance 3m), but will likely need to be adjusted depending on the model that needs to be rendered.

Plane



The object „Plane“ is used to capture the shadow cast by sunlight on an object. To prevent the plane from being included in rendering without losing the shadow, the "Transparent" switch is enabled under „**Rend → Film is**“ checked(see screenshot above).