

**Asset Store item page:** <https://www.assetstore.unity3d.com/#!/content/64348>

**Video demo:** <https://www.youtube.com/watch?v=8hiWF10rg2s>

**Real-time demos:** [Windows EXE](#), [Mac APP](#), [Linux](#), [Android APK](#)

**Email us:** [moonflowercarnivore@gmail.com](mailto:moonflowercarnivore@gmail.com)

**Facebook group:** <https://www.facebook.com/MoonflowerCarnivore>



**Enchanted Edge** is a particle effect library created using Unity's default Shuriken particle system without post-processing image effect, non-built-in script or shader. This package provides visual effects of elemental attribute being attached on sword, hammer or arrow. This document details the suggested usage and methods of customization.

If you are not using Unity Editor 5.3.0 or onward, we strongly recommend to upgrade to the newest [stable](#) or [patched](#) version of Unity 5.3 due to all the important bug fixes and new essential features of particle system. Unity 5.3.0 to 5.3.3 in particular cannot batch draw calls of different particle prefabs which loads the same material, this greatly hampers optimization.

**Due to a security vulnerability, update your Unity editor to the [latest patch version](#).**

## *Change log*

- V1.00:
  - First release
- V1.01:
  - Fixed prefab hierarchy for more straightforward scaling result
  - Minor revision of Wind and Darkness tier 2
- V1.02:
  - Added “Sustain”, “Swing” and “On-hit collision” sub-folders for standalone sustain, swing and on-hit (collision) effects similar to “On-hit” (non-collision) effect sub-folder.
  - New element: (Elven) Wood
  - Added “Character sustain” effects.
  - Greatly tightened speed cap of all “Swing” effects to prevent swing particles from flying off uncontrollably.
- V1.03:
  - New weapon type: Arrow
  - New element: Earth
  - New “Enchanted Lightning swing cheap” variants
- V1.04 (August 2017):
  - Support for Unity 5.6.

## *Basic usage of Shuriken particle system*

All effects are contained in the “Prefabs” folder. The root “Prefabs” contains the complete (sustain-swing-on-hit) version of the effects. “On-hit only” effects are in the sub-folder. To view any of these effects, drag any prefab from the Project window to Hierarchy window and the effect will play itself when it is selected in the Hierarchy window. Before you press the “Play” button, the displayed effect is under “Simulation” (or “Preview”) mode.

You can check the little “Particle Effect” module at the bottom-right corner in the “Scene” window to manipulate the simulation, however you change its values will not affect values of the actual effect. Simulation mode is less accurate in terms of particle emission and lifetime than in the “Play” mode, this will become more obvious when the effect has complicated Sub Emitter setup like this asset. You can modify the values of the particle effect loaded in the Hierarchy and press “Apply” to overwrite the one in the Project folder when you are satisfied, otherwise you should rename the effect prefab and drag it back to the Project folder for version backup.

Normally only the selected prefab(s) will simulate, Shuriken particle system will check if the selected prefab’s top parent and its other child prefabs also contain the particle system component. If so the whole effect tree will be simulated altogether without needing to re-select all effect prefabs.

During the Play mode, you can still drag the effect prefab from Project window to Hierarchy window and all effects will play simultaneously. If you have changed any values of the particle effect during Play mode, it will not be applied after you stop playing it unless you drag the modified effect back to the Project window.

This package includes a default\_scenes for testing purpose. You drag any effect prefab (except “on-hit non-collision”) from the project folder to the “Enchanted effect container” prefab (at the bottom of “z\_weapon\_container” prefab) of hierarchy window because it defines the proper central position (the base of blade) of the testing sword to emit particle effects. Under no circumstances should you change the position or rotation of the root effect prefab because this will cause confusion during implementation of the particle effect. If you really need to change the initial position or rotation of the particle effect, do that to the “effect container” instead so the root effect prefab is unaffected.

The default\_scene uses sword, you can delete the “z\_sword” prefab in hierarchy and drag “z\_hammer” under “weapon rotate” prefab for testing hammer effects. Again you drag the Enchanted hammer effect under “Enchanted effect container” because it define the proper base position of the hammer head.

### *Always on top glow*

All large glow and ray particle prefabs are put under the “TransparentFX” instead of “Default” layer. Refer to the default scene setup, you place an extra “FX camera” under the “main camera” prefab hierarchy so the former always follows the latter. FX camera is only used to render objects of TransparentFX and this layer is removed from the culling mask of the main camera. Because the FX camera is given higher “[depth](#) (draw order)” than the main camera, objects of the TransparentFX layer will always appear before those rendered by the main camera.

This setup helps prevent uncontrolled 3D rotation of the main camera:

- Empty prefab (for general camera position and y rotation)
  - Empty prefab (for x rotation only)
    - Main Camera prefab (-1 depth, excludes “TransparentFX” layer; for z position [distance from origin] and z rotation)
      - FX Camera prefab (0 depth, renders “TransparentFX” layer only)

This practice prevents the glows from intersecting with other objects in the “lower layer”, the ground and the target particularly. The drawback is that anything rendered by the FX camera will cover absolutely everything of lower camera depth regardless of distance. You can set lower clipping plane of the FX camera or simply choose not to set up an FX camera if you do not care of the intersection. The sorting fudge and sorting layer of all particle prefabs are given appropriate values so they should appear in the correct particle order.

Another drawback of this method is that even if the main camera and the FX camera render the billboard particle prefabs with the same material, their draw call cannot be batched. Nonetheless this is the cheaper way to get a glowing effect than adding a light which is something you need to choose between when developing for mobile devices.

### *Fire: Additive or Alpha Blended?*

Fire effects are provided in 2 more variants: Additive and Alpha Blended. This is because the shade of flame varies dramatically and is affected by numerous factors (fuel type, oxygen level, combusting material). For realistic approach, you should look at a real picture of fire on a scenery you want to reference. If the flame is more yellowish or brighter, go for Additive; if the flame is more reddish-orangey or dimmer, use Alpha Blended. If performance (e.g. on mobile platform) concerns you much more than realism, opt for Additive and remove “particle sustain birth birth glow” prefab.

### *On-hit Collision*

The on-hit effects in the complete Enchanted effect prefabs use the default Unity collision function, so it will trigger the on-hit effect when the “particle on-hit master” prefab hits any collider in the “Ignore Raycast” layer to avoid collision with other unrelated colliders. If you have assigned a different on-hit collision layer, you need to change the collision layer in the “collides with” of “collision” module in the “Enchanted on-hit collision” prefab.

Everyone knows that Unity’s collision detection is a nightmare. “Mesh collider” in particular is totally unreliable. That is why we instead use “sphere collider” in our demo scenes, also it is much cheaper, but it still misses occasionally. If you have trouble to get the collision of on-hit effect works in your project or you want to avoid collision detection as much as possible on mobile platform, use the individual (non-collision) on-hit effects in the “On-hit standalone” folder under “Prefabs” folder, time and position the on-hit effect on the target manually.

### *On-hit non-collision*

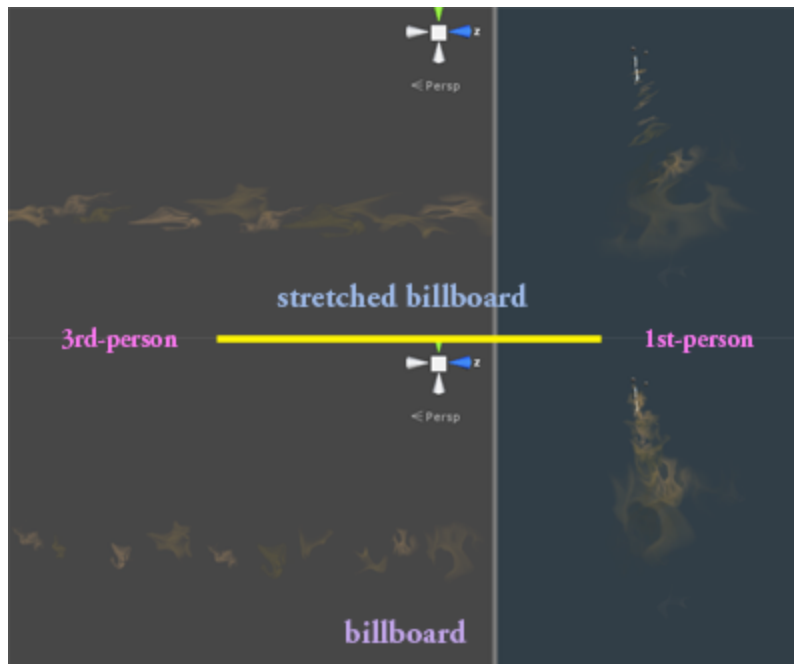
If you have decided to use on-hit non-collision effect, remove the on-hit collision effect from the Enchanted preset in case you are using it. Set Up an empty child prefab (on-hit effect container) under the target prefab and adjust the container prefab’s y axis position if needed. For the scale in our demo scene, we recommend  $y=1.3$  which is about the height of the character’s chest. Finally you write your own script to load and activate the desired on-hit non-collision effect.

## *Point Light*

Each Enchanted effect preset contains a point light because it just looks more awesome. For mobile game, you would want to remove it straight away. Even though newer mobile models can handle it without drop of framerate, an extra light still drains battery and warms up the device quickly.

## *Arrow swing in first-person perspective*

The swing effect of arrow mostly use “stretched billboard” just as the other weapon types, the problem is that it may only look nice in third-person perspective, but not so in first-person (e.g. for VR) because the path of the arrow is usually flying away from the attacker and causing the stretched billboard to look flattened. To fix that, go to the “renderer” module of the swing effect prefab and change the “render mode” to just “billboard” if the default setting is “stretched billboard”.



## *Customizations*

Even under same scale, each weapon has different measurements. Even for sword or hammer, you would likely have to adjust the position or emission shape of the effect to fit into the weapon. To make things less painful, we recommend to create an empty child prefab under the weapon's prefab hierarchy (c.f. "z\_sword" and "z\_hammer" prefabs) and name it "Enchanted effect container". Adjust the z position of the container so it locates at the visible base of the blade or hammer head. You only load the Enchanted effect prefabs into the container prefab.

For example, mage staff would use similar Enchanted effect as hammer, so you define the base position of "Enchanted effect container" similar to z\_hammer and then load the hammer Enchanted effect to the container and your work is done.

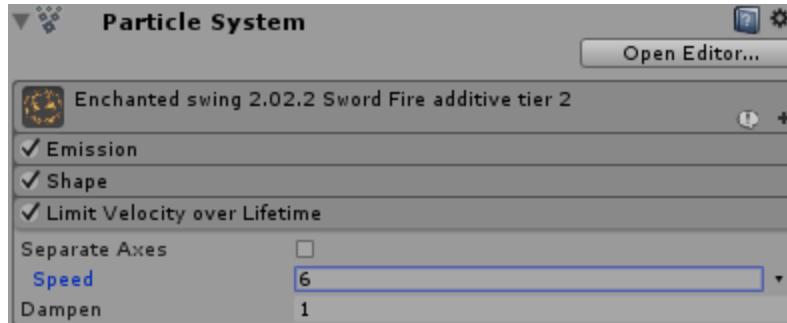
### *Length of particle emitter shape:*

In the primary particle "master" prefabs if it uses "cone" emission shape and emit particles from "volume" or "volume shell", adjust the "length" in the "shape" module. For effects like "Lightning" and "Holy" with tall oval glow particles, adjust "start size" for tuning the width of the glow; for the length of glow, tune "length scale" in the "renderer" module.

If the blade is curved such as [shamshir](#) or [katana](#), you may, depending on how curved the blade is, set up 2 to 3 more Enchanted effect container prefabs alongside the blade evenly. Load the same Enchanted sword effect to all container prefabs. Rotate x axis of the container prefabs and shorten length of the cone emission shape until they roughly follow the contour of the blade.

### *Swing effect goes crazy and needs fixing:*

If your weapon (or the character who carries it) moves or rotates too fast, swing particles may stretch and fly like crazy. To fix the effect, tune the "[limit velocity over lifetime](#)" module of the swing effect prefab to cap the maximum speed (smaller speed value) of the swing particles. You can also reduce values of "[inherit velocity](#)" module but this may require you to edit "[size by speed](#)" module as well if most swing particles are below the default visualization threshold.



### ***Increasing particle emission rate to no avail:***

Increase the “max particles” in the [main module](#) as well. We set it just right for performance on mobile platform. If your target platform is desktop or console and high particle count and alpha overdraw do not concern you at all, 10,000 should do just fine, but we still highly recommend to set a reasonable cap instead of a virtually unrestrained value.

### ***Scaling:***

Although Unity allows scaling in the [prefab's transform component](#), it is far from perfect because it does not scale the distance emission value (heavily used in “swing” prefabs), limit velocity over lifetime module or any “scale” value of stretched billboard render mode. If you multiply the transform scale value by 2, the distance emission value should be divided by 2 and speed value of limit velocity over lifetime and all stretched billboard scale values should be multiplied by 2. We have submitted a bug and a [suggestion ticket](#) to Unity and we do not know when they will deliver a fix, so do not scale the prefab transform component at all (except for the spiral mesh which will be explained in the next section) and instead save a separate effect prefab and scale every particle parameter manually, including “start size”, emission shape and velocities. However, this issue has been officially changed from “suggestion” to “bug” which means it will get fixed sooner, so all particle prefabs are given the “hierarchy” value of “scaling mode” which means their scale is affected by all of its parent prefabs if scaled. If this gives you undesirable result, change scaling mode of all particle prefabs from “hierarchy” to “local” or change the weapon prefab structure from this:

- Root weapon prefab containing weapon's mesh filter and renderer
  - Melee Weapon Trail prefab
  - Enchanted effect container prefab

Into this:

- Root weapon prefab (empty)
  - Weapon's mesh filter and renderer prefab



- Melee Weapon Trail prefab
  - Enchanted effect container prefab

And retain the scale value of the root and container prefab to 1 unless you want to scale both the weapon model and Enchanted effect with full knowledge of the consequence.

If you have serious trouble of getting the Enchanted effect positioned properly on your weapon due to different measurements and/or scaling than our demonstration weapons, you can send us your weapon model (or a rough replica of same scale without texture for the sake of secrecy) and we will help you to customize the effect and tell you exactly which values are modified and how much should be multiplied.

***Scale of the spiral mesh in Wind effects:***

It is used in the “particle sustain spiral” prefab of any Wind sustain effect, it loads the “ray\_add” material. Technically you can scale its z axis in the transform component, but you cannot control its size over lifetime of separate axes.

In Unity 5.4 and onward, you can enable “3D start size” of the spiral prefab and change the z start size for its height, or x/y start sizes for its width and length, so you do not need to modify its transform component. You can also enable “separate axes” in “size over lifetime” module and adjust the z axis independently to give the mesh more dynamic.

No matter 5.3 or 5.4 you are using, if you want the spiral direction of the mesh in the opposite way, you need to change the x scale in transform component to -1 and make sure the simulation space is “local” instead of “world”, otherwise negative scale is ignored due to a Shuriken bug ([issue 791796](#)).

***Adding [Melee Weapon Trail \(MWT\)](#) effect:***

MWT is a free asset available on the Asset Store by another publisher. Compared to the built-in [Trail Renderer](#), MWT is a better option for weapon attack effect because it is more flexible, more performant and (optionally) interpolates trail vertices to smoothen the trail mesh if the trail base moves or turns too quickly. We have added the MWT container, base and tip prefabs under our demonstration weapons. However, because of the [End User License Agreement](#) of Unity Asset Store, we cannot include any asset, even if it is free, by another publisher in our own asset, instead we include a screenshot of the MWT setting in the package root folder for your reference. If it is for desktop/console platform, we think you would be interested to combine the swoosh trail effect of MWT with the swing particle effect of Enchanted Edge. For mobile platform, you would opt one between the two.

MWT is not without its limitations (it is free after all):

- You cannot make the trail effect exactly look like [Kratos' Chaos Blades](#) because the trail source of MWT must be a straight line drawn from the base to the tip prefabs, unlike the curved chains attached to the Chaos Blades. You can line up the tip and base of different MWTs side by side to emulate a curve, though.
- If the base and tip of the trail, as well as their parent prefabs only rotate but none of their transform-position changes arithmetically, the trail will not be generated. Before you pay to buy other similar trail asset, ask that asset provider explicitly if this issue with MWT presents in their asset as well.

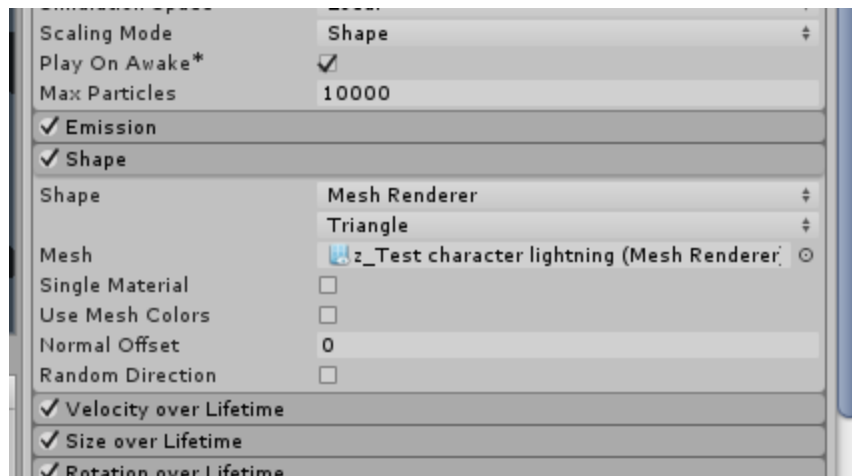
### *Extra: Mounting Enchanted effect on non-weapon model*

[Not really a selling point of this asset](#), but there you go~

In the “z\_extra particle mesh renderer on non-weapon object” scene, you can see the character models are mounted with Enchanted effects, these are slightly modified Enchanted effects from the “sustain” effects of the regular Enchanted weapon effect prefabs.

If this is going to be used on skinned mesh, you need to change the particle prefab which uses “Mesh Renderer” emitter shape to “Skinned Mesh Renderer”.

In practice, you need to program all particle prefab with the “(Skinned) Mesh Renderer” emitter shape to load the target mesh renderer **from the scene** (not project folder). For precision, the whole Enchanted particle prefab should be parented to the target mesh hierarchy (unlike weapon effects, you do not need a specific weapon container prefab for correct position), otherwise some particle may not follow the mounted mesh correctly when the latter moves.



## *FAQ*

### *Why so little mesh particles used in this asset?*

Out of concern of performance. Mesh particles prevent draw call batching whatsoever, unlike billboard particles. It is still fine to use mesh particle with restraint on mobile platform.

### *I want XXX element but it is not in your package.*

You can email us about your suggestion. You can provide us reference images or image links which you really like.

If you like this asset, please rate it or write a review in our Asset Store [item page](#). However, if you need technical support, you should [email us](#) directly and preferably state the target platform (mobile or desktop/console) of your project. You may also be interested in other assets created by us, so check out our [publisher page](#) on Asset Store. Some assets are even offered for free, so don't delay.



**Moonflower Carnivore**

**2015-2016**

[moonflowercarnivore@gmail.com](mailto:moonflowercarnivore@gmail.com)