If you are reading this documentation in PDF or TXT formats, unless you have trouble connecting to internet or accessing Google, preferably use the online version: https://docs.google.com/document/d/1_afCffe86f9AiCHI2Udp4xIS58F_oh_MVo1pYtveK5I

Unity Asset Store Link: https://www.assetstore.unity3d.com/#!/content/52746
Email us: moonflowercarnivore@gmail.com
Facebook group: https://www.facebook.com/MoonflowerCarnivore

Shuriken Salvo is an effect library created using Unity's default particle system, codenamed "Shuriken Particle System" to distinguish from the legacy particle system. This documentation details the recommended usage and methods of customization. To use this manual more efficiently, press Ctrl+F to search for keyword of your question. If nothing matches, email us for technical support.

!!! IMPORTANT !!!

Due to a security vulnerability, update your Unity editor to the latest patch version.

Table of content

Change log

- V1.00:
    - First release
- V1.01:
    - + “Grand” version of all Elemental and Physical Missile effects.
    - + “Recovery Cure”.
    - + “Recovery Elixir”.
    - Renamed “Revive” to “Recovery Revive” for better alphabetic sorting in folder.
- V1.02:
    - + "Elemental Water" series of effects.
    - Revised “roundblade” objects in all "Elemental Wind" effects for Unity 5.3.4p4 fix of 3D rotation.
    - Improved performance of “Elemental Wind_grand” (swarm).
    - Removed “Physical Missile (Unity 5.2)” and “Physical Missile (offset pivot) because both now merge into just "Physical Missile" effects with accurate positions of missiles and trails.
    - Fixed texture icecube.png so it will no longer ask for fixing to “normal map” upon import. If this still happens to you, select “ignore”, it should remain as “texture”.
    - This update is not available to Unity 5.3.3 fallback version (only updated to v1.01) due to instability and bugs. The latest update is submitted using Unity 5.3.4p5.
- V1.03 (June 2017):
    - Update for Unity 5.6.1.
    - Fixed mesh “arch” and all particle emitters (feather, arch) using this mesh for the intended rotation result which currently rotate uncontrollably random.
    - Changed scaling mode of all particle objects to “hierarchy” due to the soon-to-be fix of Shuriken scaling bug by Unity.
    - Some particle emitters use Lights or Noise modules.
- V1.04 (September 2017):
    - New “homing” scripts and effect variants.

Basic usage of Shuriken Particle System

All effects are contained in the sub-folders of “Prefabs” folder and divided by “homing”, “swarm”, “single projectile”, “desktop” and “mobile” variants. 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 to begin run-time, the displayed effect is under “Preview/Edit” mode.

You can check the Particle Effect panel at the bottom-right corner in the “Scene” window to manipulate the preview, however you change its values (e.g. playback speed) will not affect the actual effect. Edit mode is less accurate in terms of particle emission and lifetime than during run-time, this will become more obvious when the effect has complicated Sub Emitter setup. 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 parent object and drag it back to the Project folder for version backup.

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

During the run-time, 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 run-time, it will not be applied after you stop playing it unless you drag the modified effect back to the Project window.

This package includes scenes for testing purpose. The “default scene swarm” is for the “swarm” effects. You drag a swarm effect prefab from project folder to the “effect_container” object of hierarchy window because it defines the proper central position (the chest) of the dummy to be hit by the projectiles and trigger the collision effect. Under no circumstances should you change the position or rotation of the prefab parent because this will cause confusion during instantiation 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 effect parent is unaffected. OK, by knowing the consequence you still want to change the initial position of the effect object, remember that changing the position of the (slave) sub emitters will have no effect at all. The change of position only works on particle object when it is not any sub emitter to another emitter. So only transposing the “particle master” object will show the difference.

It also includes a “default scene single projectile” for effects in the “single projectiles” folder. You drag the combined effect (without “collision” or “projectile” suffix) to the “effect_container_radius” object and play the scene to examine the “single” effect.

Particle Homing scripts

There are 2 scripts for handling dynamic homing/guidance behavior of each particle emitted by Shuriken Particle System, namly “particleHoming.cs” and “particleHomingMultiTarget.cs”, both in the “Scripts” folder. The former can only select 1 target at a time while the latter handles multiple targets as much as you need.

Attach either homing script on any object with Particle System component in “World” simulation space, then assign a target object. For multi target, enter the number of targets then assign the targets to the elements. Programmatically, create a new target array and apply it to the target property of the script. If no object is assigned to the “target” property, the script assumes the attached object by default to avoid null-reference exception. All particles emitted from this particle system will be guided to the target position. Set the “Dying Range” (radius from target position) so the particle is killed once it is in range of the target, otherwise you need to add a collider component (preferably sphere collider or capsule collider for human character) on the target and enable Collision module of particle system with “world” mode, 0.1 bounce and 1 life loss, modify layers of “collide with” if you want the particles collide with collider of specific layer only. Lastly the collision effect requires sub emitter spawned on collision.

“Speed” dictates how fast the guided particle would be corrected to the direction towards the target. Zero speed means the projectiles will not be guided at all. High homing speed will result in sharp turn of each projectile path.
“Max speed” dictates how much speed of the guidance is limited at for preventing the projectiles from getting flung too far away from the target after the projectile misses it.
“Homing Delay” is the duration since the emission of each particle before being guided towards any target, this is allows all projectiles to spread apart before moving towards the target and avoids all projectiles to only focus on a single target when multiple targets are assigned.

particleHomingMultiTarget provides two target selection modes: Random and Closest, both pick target from the target array. For Closest mode particularly, if you set a very low homing speed, there is chance the guided particle would miss its initial target. If the particle leaves the target too far away, it may get closer to another, subsequently it will be guided to this new target instead. If you want each projectile to only focus on one single target and never change target, use either Random mode or the single homing variant and instantiate extra prefab with different target.

If the target does not move (such as in turn-based RPG battle) or the effect simply requires no homing behavior during simulation of particle system, use the “swarm” effect prefab for the sake of performance. Use the single homing version if the effect only targets one unit to bypass the comparison of distances between the projectile and the targets.

Ending or replaying an effect

Before Unity 5.5, stopping simulation of the effect parent while its particles are still being animated will kill them instantly. To end the looping effect smoothly, you can deactivate the Emission module of the “particle master” because all other particle emitters (slave) of any “swarm” effect are dependent on the emission of this “master” object. (Doing so in edit mode will kill all particles as well.) The reason that the master emitter is not placed in the real top parent (which contains an empty particle system component for triggering preview) is because there is situation that the transform-rotation of the master object should not be succeeded by the slave objects.

From Unity 5.5 and onward, ParticleSystem.Stop() stops simulation without killing the remaining particles by default which is simpler than accessing Emission module via script.

Always on top glow

All large glow and ray particle objects 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” object 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” than the main camera, objects of the TransparentFX layer will always appear before those rendered by the main camera.

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 objects are given appropriate values so they should appear in the correct order.

Scaling

You only modify the transform scale value of the prefab parent and the rest will scale automatically, because all child particle objects are given the “hierarchy” scaling mode to multiply with the parent scale.

The scale of our prefabs follow the general rule of thumb that 1 unit length in the Unity scene equals to 1 (metric) meter. The test character in the demo scene is about 1.7 meter tall, chest (preferred targeted position) is 1.2 meter from soles.

Customizations

Lifetime of projectiles (homing and swarm only): Change the “start lifetime” value of “particle master” object. And then change the “duration” of all “particle slave birth (...)” objects with the exact same value as particle master’s start lifetime. This is because the gradient of “start color” and emission rate graph of all “particle slave birth” depend on the duration value until Unity 2017.1. If the 2 values do not match, the trail will not appear or fade in/out at the right timing.

Launching distance and range (swarm only) is dictated by radius of circle in Shape module. Changing shape radius also require modify either start lifetime or start speed in Main module. Because changing lifetime of particle master also requires changing of its “birth” sub emitters which can be quite troublesome considering the amount of birth sub emitter linked, changing speed is more preferable. You can refer to the *locateAttacker.cs* script, the formula of the 3 parameters for each projectile precisely locate the attack from target is:

```
main.startSpeedMultiplier = shape.radius / main.startLifetimeMultiplier
```

Launching angle (swarm only): Most swarm effects are given certain tilt to the “particle master” or “master rotation” object of “Shuriken”. Usually you only need to adjust the Z rotation of transform. We do not rotate the parent object because there are certain cases that we do not want the child objects to be rotated such as the collider plane of “Meteorite” and “Missile” which is supposed to be constantly horizontal. If their collision plane are rotated, their collision on the supposed ground will be rotated as well which just looks silly.

Range of spiral motion (swarm only): In the “particle master”, check the Velocity Over Lifetime module, the values are given the “ranges between curves”. You do not need to modify the curves but their curve multiplier values. X dictates the linear velocity and Y/Z dictates the range of spiral. Whenever you have done entering any value in the graph, press enter to confirm otherwise the value may not be registered.

One tip when editing the multiplier value of multiple axes in the particle curve graph is that you can select multiple curves of different axes or just press ALT+A to select all visible curves in the curve graph and then type the multiplier value so the new value will be applied to all selected curves. This trick only works when all selected axes share the same multiplier value before selection (which is mostly true for Y/Z velocity over lifetime curves). When the selected curves have different multiplier values, the multiplier value will disappear, prevents you from modifying it.

Range of spiral motion (homing only): In the “particle master”, adjust the strength or frequency of the Noise module, enable the module if it is not activated.

Launching direction (homing only): Change the emitter Shape module. If you only want the projectiles to be launched in 1 direction, you may use “cone” or “edge” shapes and then adjust transform rotation accordingly.

Length of trail: Modify the “start lifetime” of the trail object. If it is for modern desktop or console platforms, we affectionately recommend increasing the smoke trail of meteorite and missile to longer for more dramatic effect. But you may need to increase “max particles” in Main module accordingly, most particle system are given appropriate cap to prevent excessive particles from being emitted accidentally which can easily lag the game, so always consider a reasonable cap instead of a virtually unlimited figure. You can estimate the new cap after increasing trail emission rate and trail lifetime by first increasing max particles to something like 100,000, then look at the “particle count” in the Particle Effect panel during preview and add slightly more (e.g. +200 for smoke trail) to that particle count for the final max particles cap.

Density of trail: Modify the emission rate of the trail object. The default values are just right and you should not worry if the trail disperses when the projectile travels too fast, because the trail emission is calculated over “distance” instead of “time”, that means more particles are emitted to fill in the gaps when the emitter travels faster. “Distance” is not used when the emitter is stationary which is why this option is usually overlooked by artists. There are 2 situations where you do need to adjust the emission rate:

- If you have changed the trail start size, adjust the emission rate conversely.
- The flame trail object of the fire effect applies “inherit velocity”. This option is applied to give additional force for the flame particles to move along the initial direction of the leading emitter. If you set the leading emitter travels too fast, you can increase the emission rate of the flame trail to reduce the gap. But the interpolation of Shuriken particle system for evening the gaps between the initial positions of particles does not take inherit velocity into account, that means if you want the flame trail to join absolutely coherently at high speed, you need to disable the Inherit Velocity module besides increasing flame particles emission rate.

Collision with scene objects: All swarm effects except Shuriken (swarm) are defaulted to collide with the plane collider of particle system itself because it is cheaper and guarantees no projectile particles sneak through the collider. If you want the projectiles collide with other colliders, in the Collision module of “particle master”, you change the “Plane” to “World” mode, setup the colliders in a specific layer which you intend to be interacting with the particle effect by checking or unchecking the layers in the “collides with” parameter. If you want other particle emitter (e.g. the sub smoke trails in the Missile swarm effect) collides too, you add/enable the Collision module of “particle slave collision birth master”.

Collision in “World” mode is unpredictable, also not all colliders are equally reliable. Sphere collider is second to plane mode collision because it only needs to calculate the distance of the particle position from the target collider center. Mesh collider is the most unreliable which in most cases should be avoided, usually replaced with capsule collider for human character. You adjust the position and radius of the sphere or capsule collider to roughly match the size of your visible mesh model.

Accuracy of projectile (homing only):

- Reduce “start speed” in particle system Main module. Because the default emitter shape is “sphere”, start speed will launch the particles in random directions. If the particle is initially facing away the target, the projectile will have harder time to align itself against the target.
- Increase “speed” in particleHoming script: This value determines the level of track correction of the projectile towards the target.
- Reduce “max speed” in particleHoming script: This may sound counter-intuitive. The truth is if the projectile is moving too fast, it will be flung farther and faster after missing the target. Max speed is to undo this undesirable gravitational force, allowing the projectile to re-align easier, otherwise it will keep orbiting the target until death.
- Increase distance between emitter and target: If the initial direction of the projectile does not face towards the target and the two parties are too close, the close distance leaves little room for the projectile to correct its path.

Angle of shadow of smoke: This is applicable to any particle system using the “volumetric_smoke_alpha”, “volumetric_smoke_sheet_alpha” or “volumetric_smoke_sheet_anim_alpha”. You adjust the “start rotation” of the smoke particle objects. Zero degree means the lighter side is facing upward and shadow side downward. For some randomness of the shadow angle, change the “constant” mode of start rotation to “random between constants” and give the initial angle ±10 degree range. e.g. if the initial angle is 0 degree, change it to -10 and +10 degrees; 40 and 60 if the initial is 50. “Volumetric_smoke_sheet_alpha_dir_light” which is used in Lightning, Meteorite and Missile effects changes the shadow angle automatically to the primary directional light in the scene via shader. If your scene has no directional light, you should change the smoke material in these effects to other smoke materials not dependent on directional light.

Brighter flame (all fire effects): If you do not use bloom post-processing effect, change the material of flame emitter from “flame_alpha” to “flame_add” which uses additive shader. Because additive is for emissive texture, be careful that it will make the flame texture loses its definition on brighter background. If you use bloom and have enabled HDR mode of your main camera, increase the “intensity” value in the material which uses any particle shader with the suffix “Intensify”. Gamma and linear color spaces differ significantly here. Do not be alarmed if you need to crank up intensity beyond 3 to get the desirable blooming under linear color space.

Mesh particles alignment: Only from Unity 2017.1 onward you can align mesh particles to their direction vectors. For our meteor mesh, you enable “3D start rotation”, set start rotation: “x=0, y=90, z=0”, change “simulation space” to “world” in Main module, change “render mode” to “mesh”, load the meteor model and change “alignment mode” to “velocity” in Renderer module.

Light receiving particle

The custom shader “Particle Alpha Blended Amb Dir Lit” used in the particle material uses an additional gradient map for cheap shadow effect. The gradient map automatically rotates itself according to the rotation of the main directional light. The whole material also takes the color, intensity of the directional light and ambient color into account. Their effective power can be tuned in the material setting. When using any material with this shader on any other particle system, you must enable custom vertex streams in Renderer module, add “UV2”, “Tangent” and “custom1.xyz”.

If your texture is a flipbook with multiple frames which requires enabling Texture Sheet Animation module, remove “UV1” from the list “enabled UV channels”, otherwise the gradient map will not be applied on the main texture correctly.

Optionally enable the Custom Data module to fiddle with the additional intensity color on the whole texture which is useful in HDR bloom. It requires “color” mode, then you can change the data type from constant to gradient. If desired, you can increase the “current brightness” in the custom color picker which will grant access to HDR color mode. In desktop effects like Missile grand, we give the smoke particle system gradients from dark orange to grey then black which changes the smoke particles’ emissive color at its early age to emulate heat.
Both “emissive color” in the material properties and “custom1” color ignore alpha value (which they multiply before being added to the texture color). The resultant emissive power is determined by the grey scale of these 2 color inputs (the darker the weaker).

All mobile variants replace material with this shader by another version which uses default Mobile/Particles/Alpha Blended shader and texture with pre-painted shadow.

Light emitting particle

Some particle system object uses Lights module to emit light which can be projected on other light-receiving objects rendered by the same camera. This requires Lights module has an object with any qualified light component. For consistency sake, this light object is disabled which does not affect the result of the light emitting particle system. All mobile variants have this module and referenced light objects removed.

Performance and optimization

Unity 5.3.0 no longer batches drawcalls of particle systems which use the exact same material **until 5.3.4p2**, instead the engine uses multi-threading to enhance the processing speed which could mean more power is consumed. On mobile platforms this can be more sensitive as the device might heat up and lose battery power quickly. Nonetheless particle count and overdraw (2 or more objects overlap before the camera) are the biggest culprits.

Even if you are using the version which cannot batch draw calls, we still recommend creating particle texture sheet/atlas by grouping frequently used textures of the same shader into 1 big material and then enable “Texture Sheet Animation” in the particle system to select the required texture in order to reduce the amount of texture read-write. So literally you would have at least 2 particle texture sheets, one additive and one alpha blended. You can include the less used texture into the sheet if it still has room for it. Because different games use different set of particle textures, the particle texture sheet sadly has to be done on your side for maximum optimization of the specific game.

Draw call batching of particle system also requires some strict rules. If it fails to batch, a exclaimation mark in a speech bubble will appear at the top-right corner of particle system Main module, it will pop up a message box to brief the reasons. You can further learn the details in Frame Debugger (can be opened under the “Window” menu).

Also you can produce another set of “power saving” particle effects by reducing the emission rate and/or lifetime of all particle effects. Exactly how much should be reduced really depends on how you value aesthetics. There is hardly a right answer. Some might just remove the seemingly minor Sub Emitters (but be careful if it is a master emitter to another sub emitter, in such case, only uncheck its Renderer module unless its slave emitter is not needed either) instead of reducing other emitters’ emission rate or lifetime. The laziest and safest approach would be reducing all lifetime values by half, but remember that before Unity 2017.1, the “duration” of any trail Sub Emitter should agree with the lifetime of the spawning (master) emitter, otherwise the color cycle of the trail will not be properly played.

Known issues

- Particle Homing script usually should run during edit/preview mode, but there is chance it would fail to initialize. Should that happen, play the scene (preferably do not attempt the pause-play trick), immediately stop it to return to edit mode and the script should simulate again.
- Particle system usually with emission rate over distance would clump the emitted particles together if the framerate is uncapped, for instance without v-sync. Cap the framerate with tarrgetFrameRate or just enable v-sync again should solve it.
- Generally a particle sends “death” or “collision” messages for spawning sub emitter as intended unless you have enabled Trails module without checking “die with particles”, the death message is only sent after the trail is dead as well. If the particle is already dead upon collision (1 life loss) but the trail is still alive, it will continuously send collision message. The result is either sub emitter spawned much later than expected (on death) or excessive amount of sub emitter being spawned (on collision). For homing effect which uses trails module for particle master object, the issue is dealt by either enabling “die with particles” (healing effects) or giving the trails lower lifetime ratio.
- You could replace the Trails module by sub emitter on birth so the small glow dots stack to form a trail like the Darkness desktop effects. This method is not recommended for mobile and for VR requires careful performance test because trail by stacking particles can hinder performance when 2 conditions are met: transparency overdraw (including cut-out/cutoff) is too high and the portion of high transparency overdraw occupies a large portion of the screen (usually when camera zooms in to the effect). You can have slightly high transparency overdraw as long as those portions do not fill more than one third of the screen.
- Unity 2017.1 has removed “interior collisions” from particle system Collision module (world mode), also collision message is sent whenever the particle is within or exits the collider. In the worst case this could trigger the “Invalid WorldAABB” error messages.

Terms of Use

Disclaimer: This section only summerizes a portion of Unity’s Asset Store Terms of Service and End User License Agreement (EULA). It does not override the agreement detailed on Unity’s website if there is contradiction between the two.

This asset published by Moonflower Carnivore (the licensor) is intended to be only distributed publicly in your (the end user’s) electronic game and interactive media, in the form of an executable which cannot be modified in any editor again, including but not limited to Unity editor.

You are free to use and modify this asset if you obtain this asset legally via Unity Asset Store firsthand and distribute your executable containing this asset legally.

You are not allowed to publicly reproduce, distribute, sublicense, rent, lease or lend this asset, either untouched or retouched, or any resource included in this asset for commercial or non-commercial purposes.

If you obtain this asset secondhand, not via Unity Asset Store firsthand, you are not authorized to use this asset, unless you obtain this asset legally via Unity Asset Store.

Per EULA, sharing this asset to other computers is only allowed:
(...) *provided that these computers are either all (i) physically located at a single physical location ("Site") belonging to END-USER, or (ii) laptops belonging to END-USER which have been made available by END-USER to its employees that are employed at the same Site provided all such computers have appropriately licensed Unity software installed.*

If you really want to include any single content of our assets in your own asset to be sold commercially, you must email us to negotiate a separate contract for permission.

If you like this asset, please rate it or leave your comment in our Asset Store item page. However, if you need technical support, you should email us directly. 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