

Xffect

The Xffect component is the base of each effect. It manages all the effect layers which are all its children.

properties

- **name**: the effect layer's name you are going to add.
- **life**: the whole effect's life time(if value < 0, life time is infinite). when the time expires, all the layers stopped and become non-active.
- **Add Layer**: when pressed this button, it will add a child game object, and automatically add a **Effect Layer** component to this game object.

How it Works?

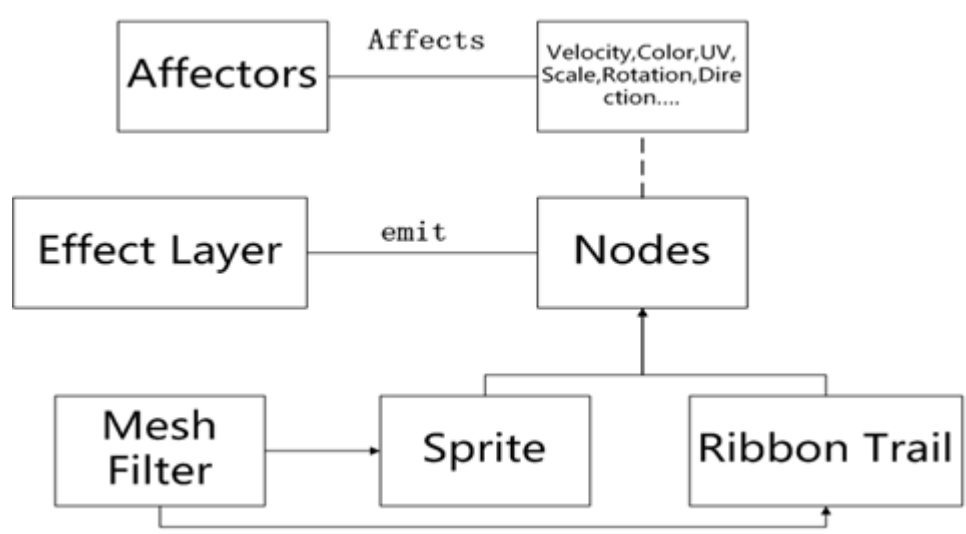
when the component start, it will check all the effect layer's material, then create an equal number of **Mesh Objects**. all the render types in this effect will use these meshes, this will greatly improve the performance.

In each update loop, it is responsible for updating all its children—the **Effect Layer**

EffectLayer

The **EffectLayer** is the core of "Xffet Editor", and it's completely independent, has nothing to do with the build-in particle system. currently it has two render types, a simple emitter and flexibly way to change the scales, rotations, colors,uv frames and velocity.

How it works?



briefly, in each update, **Effect Layer** will emit corresponding nodes, and each affector will change these nodes' attribute(velocity, color,scale...), then the node set this attribute to its render type: sprite or ribbon

trail. after that, the render type will update the mesh's verts. at last, the Mesh Filter will render it with one draw call.

render type:

- sprite: 4 verts and six indices quad.
- ribbon trail: just like the build-in trail render, but has more flexibility. if you want to know more, please check the OGRE's ribbon trail.

client transform:

the client of this **Effect Layer**, it is used by **sync to client**.

sync to client

if you not set **sync to client**, only the **emit position** will relative to the client position. if you set it, all the nodes' position is in the local space of client transform.

material

the material of this Effect Layer, basically, use the build-in particle shaders is enough. for advancing use, please check the "Xffect/heat_distortion" shader.

start time

specify the start time(in seconds) of this **Effect Layer**. When the **Xffect** starts, it will check each Effect Layer's **start time**, if the elapse time > **start time**, the corresponding **Effect Layer** will be updated.

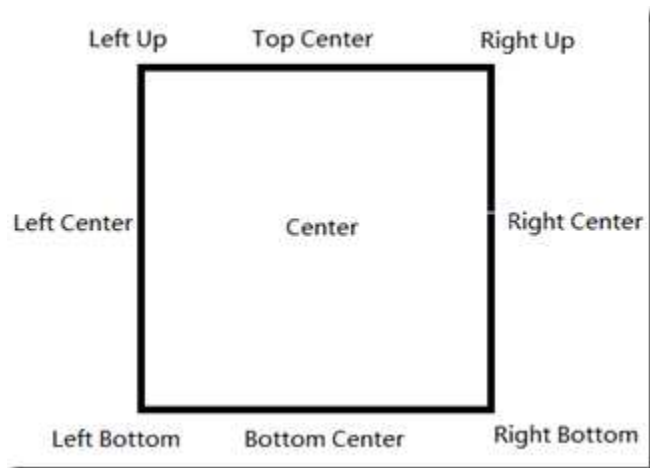
Sprite Configuration

if you choose **render type to Sprite, this configuration will appear.**

sprite type:

- Billboard: the sprite will always face the camera
- Billboard Self: by default it is on the xz plane, its direction will along the **Direction Configuration** that you set, and it will face the camera like a ribbon trail, but with only two elements. as it's like a trail, you need to set the uv stretch direction: vertical or horizontal.
- XZ Plane: just like the Billboard Self, but will not face the camera.

original point:



this is important to the **Scacle Configuration**, for example, suppose we change the y scale to 2, if you set original point to Center, the sprite will stretch both the top and bottom. if you set original point to Bottom Center, the sprite will only stretch the top part.

width, height:

the original width and height of the sprite quad.

RibbonTrail Configuration

if you choose **render type to Ribbon Trail, this configuration will appear.**

width:

the width of the trail.

max elements:

the max quads count in a trail is (max elements – 1), so the more elements you set, the more smooth the trail will be. and this value should > 2, or you should just use **Sprite** with **Billboard Self**.

trail length:

if the current trail length exceed the **trail length**, the trail head will stretch and tail will shrink.

uv stretch type:

- Up to Bottom: trail head is the texture's top and trail tail is the texture's bottom.
- Left to Right: trail head is the texture's left and trail tail is the texture's right.

Rotate Configuration

this configuration is for **Sprite only**

random original rotation:

- original rotation min: the minimum of the random rotation
- original rotation max: the maximum of the random rotation

original rotation:

the sprite's original clockwise rotation.

rotate change type:

- None: the rotation will not change during the node's life time.
- Simple: the rotation will change by "**delta angle per second**".
- Curve: the rotation will change by "**rotation curve**".

Scale Configuration

this configuration is for **Sprite only**

random original scale:

if check this option, each node when emitted will be set to the random size that you set.

original x, y scale:

if not toggle **random original scale**, each node's size will be the x, y scale you set.

scale change type:

- None: the scale will not change during the node's life time.
- Simple: the scale will change by "**delta scaleX,Y per second**". if you set scaleX,Y to negative, the scale will shrink.
- Curve: the scale will change by "**scaleX,Y curve**".

Color Configuration

color change type:

- None: the node's color will not change during its life time.
- 2Gradual: the node's color will gradual between two colors.
- 4Gradual: the node's color will gradual between four colors.

gradual time:

the time of the color gradual from start to end. if value < 0, the gradual time is the node's own life time.

gradual type:

- Clamp: when the color reached the end, it will not change.
- Loop: when the color reached the end, it will restart.
- Reverse: when the color reached the end, it will gradual from end to start.

UV Configuration

No UV Animation:

- original lower left uv: specify the lower left uv coord.
- original uv dimensions: the dimension of the uv coord. but you should be noted that the default uv direction is from lower left to upper right, if you want an other direction, eg, form upper left to lower right, just set the dimensions y to negtive.

Build UV Animation:

- x tile: the cols of the texture.
- y tile: the rows of the texture.
- time: the animation's play time, if value < 0, the play time is the node's own life time.
- loop: the play count of the animation, if value < 0, it will loop forever.

Emitter Configuration

emitter type:

- Point: each node will be emitted from "**emit position**".
- Box: each node will be emitted at a random position in the box. the box is specified by **box center** and **box size**.
- Sphere Surface: each node will be emitted at a random position on the sphere surface.and each node's direction is set to from sphere center to the node's position.the sphere is specified by **sphere center** and **radius**.
- Circle: each node will be emitted around the circle center. the Circle is specified by **circle center** and **circle direction**.
- Line: each node will be emitted on the line. the line's direction is based on the **client transform's** forward.and the line's length is specified by **line center**, **line left length** and **line right length**.

max nodes:

the maximun nodes the emitter can emit.

is node life loop:

if check this option, each node's life is infinite.

node life min, max:

each node's life time(in seconds) will be a random range from min to max.

emit by distance:

if check this option, a node will be emitted when the client transform's position exceed **diff distance**. this is useful for missiles.

chance to emit per loop:

from 0 percent to 100 percent, this will give the emission some randomness.

emit duration:

each emission loop's duration, when the duration expired, the **emit loop count** will -1.

emit rate:

for example, suppose emit rate is set to 100 and emit duration is set to 1, then when the loop expired, there should have 100 nodes be emitted.

emit loop count:

when loop count becomes zero, the emit will stop. and if the value < 0, the loop is infinite.

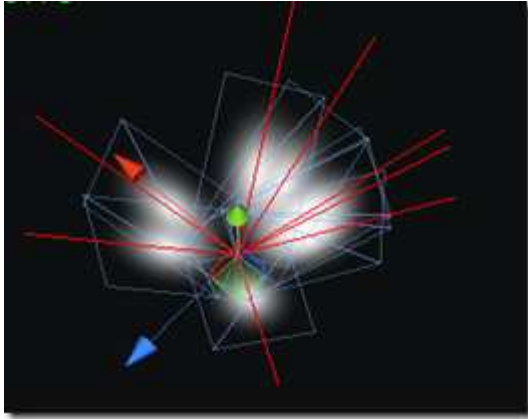
delay after each loop:

after each loop, the emission will stop a while that you set(in seconds).

Direction Configuration

here is a tricky part. the direction is not only the "**Billboard Self, XZ Plane**" sprite's along direction, but also the original velocity direction. you may ask how about the **Billboard** sprite and **RibbonTrail**? the answer is they don't have the along direction. but could have the original velocity direction.

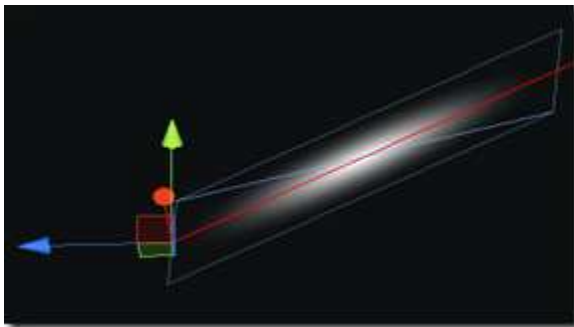
what is "along direction" ?



the red line is the along direction. remind again, this only influence the Billboard Self and XZ Plane Sprite.

is random direction:

if check this option, each node's direction will around the **original axis**, and the angle is **angle around axis**, for example, the below image, **original axis** is (0,1,0) , **angle around axis** is 60, and the sprite **Original Point** is **Bottom Center**.



original direction:

specify the exactly original direction you want.

original velocity.

if this value > 0, then the node will move along the original direction by default.

dir along velocity?

if set this option, the along direction will be set to the velocity dir in each update loop.

Affector Configuration

just choose the affector you want, and press "**Add Affector**" to add and press "**Delete Affector**" to delete.

LinearForceAffector:

like gravity, this affector will give the node a directional acceleration.

JetAffector:

this affector will give each node a random acceleration range from **jet min** to **jet max**, could be negative.
the acceleration direction is the node's current velocity dir.

VortexAffector:

each node will move around the emit point like they are in a vortex. the affector is usually used with
emitter type: Circle.

AttractionAffector

each node will be attracted by the **attraction position** like gravitational force.