MAX PROUDE

HAYATE 3
PARTICLE TURBULENCE

HAYATE 3 Manual

Document Version 1.0

HAYATE Version 3.0

HAYATE has been designed to be a powerful tool while fitting in to the Unity3D workflow and being performant and easy to use. It's simple and easy to use Interface makes HAYATE the best and easiest tool for Visual Effect Artists.

This release comes with a couple of new features and for the first time includes source code.


HAYATE
PARTICLE TURBULENCE
3

Overview

HAYATE 3 Features

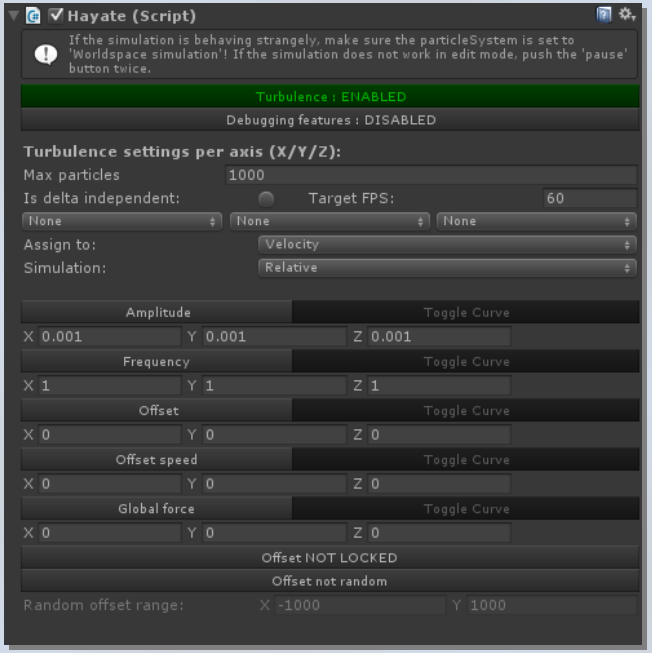
Contact: support@maxproude.com maxproude.com

Introduction to HAYATE 3 Features

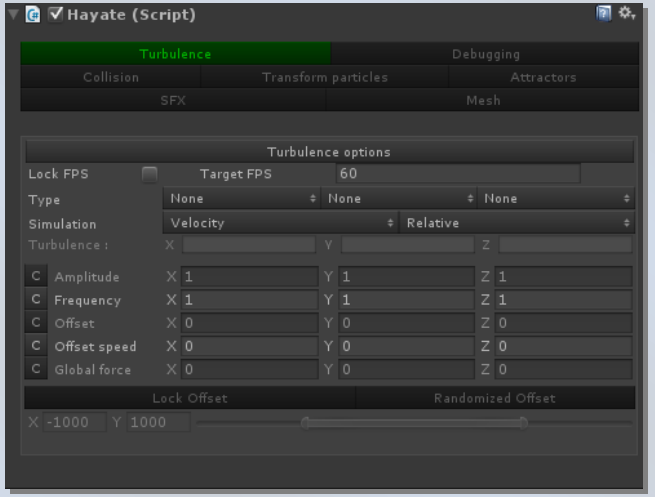
The Inspector

HAYATE 3 comes with a new custom inspector which uses the latest Editor API. Since this API is not always compatible with later Unity versions a second custom inspector will be automatically activated if your Unity version is earlier than Unity 4.5 . To test if your Unity version supports the new Inspector simply remove the #if UNITY_4_5 definitions in the HayateEditor.cs class. It is recommended to use the automatic features.

Compatible Inspector



Animated Inspector



Turbulence

Turbulence Settings

HAYATE Turbulence is a CPU based simulation that guarantees functionality on all Unity supported platforms. To achieve the desired effect HAYATE gives you a bunch of different turbulence types and simulation alternatives that allows you to be truly creative and create unseen AAA effects.

Turbulence Types (Sorted by performance)

- Sine
- Cosine
- AnimationCurve *NEW
- Perlin
- Texture
- Audio

These methods create a vector field by reading the position of two axes, calculating the Turbulence and assigning it to the third axis (x = T(y , z)). Once set up it does not require any additional input and working on the effect can begin.

To manipulate the Turbulence a couple of simple values can be used: Amplitude, Frequency, Offset, OffsetSpeed and GlobalForce. All Values can be set and animated per Axis.

The Amplitude represents the strength of the turbulence. The higher this value is, the faster the particles will move.

The Frequency manipulates the Turbulence by stretching or squeezing it. Higher Values stretch the Turbulence field while smaller Values squeeze it together.

Offset moves the Center of the Turbulence by +/- X per Axis.

OffsetSpeed adds movement to the Offset and adds +/- X per Axis per Frame.

GlobalForce is used to add Gravitation or Wind. Since Shuriken's Forces and Velocities are ignored if using Position Simulation, this is a possibility to add a Force to particle movement again.

Tip!

Use Lock Offset to keep the turbulence centered even when using Worldspace simulations.

Tip!

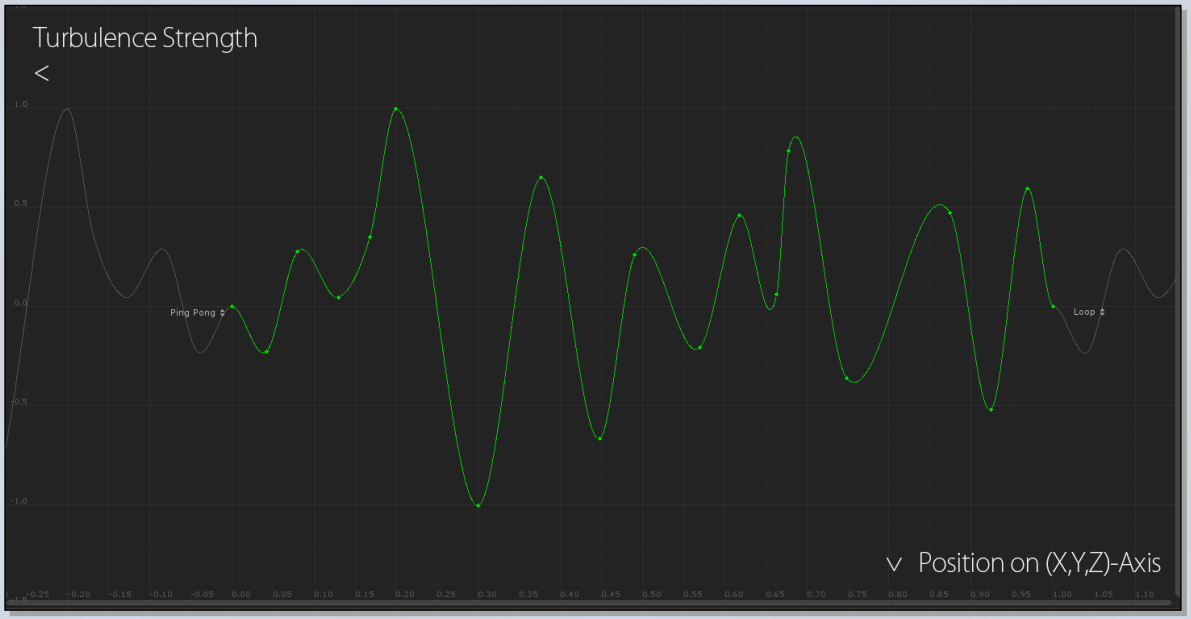
Use Randomize Offset (and set the range) to make every instance of your Effect unique.

AnimationCurve

HAYATE AnimationCurves Turbulence brings true freedom to Unity. Choose this Turbulence method to create your own Turbulence directly in Unity. Each AnimationCurve represents the Turbulence per axis. When editing the Animation curve the X-Axis represent the position of the particle on its X-Axis and the Y-Axis represents the strength and direction of the Turbulence.

Recommended workflow:

Click on the AnimationCurve field in the inspector to open up the Curve editor. Create two keyframes that are roughly the size of your level (X-Axis) and set them to 0 on the Y-Axis. You can then start to edit your custom Turbulence. Make sure that your curve stays between -1 and 1 on the Y-axis and is distributed evenly.



Tip!

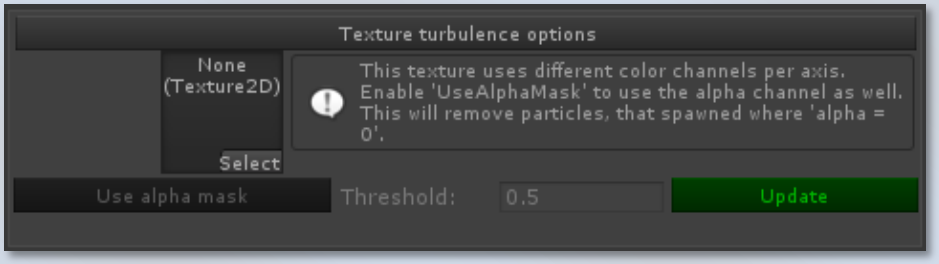
To save some time set the key frames at the end to Loop or PingPong to make the turbulence infinite on each axis.

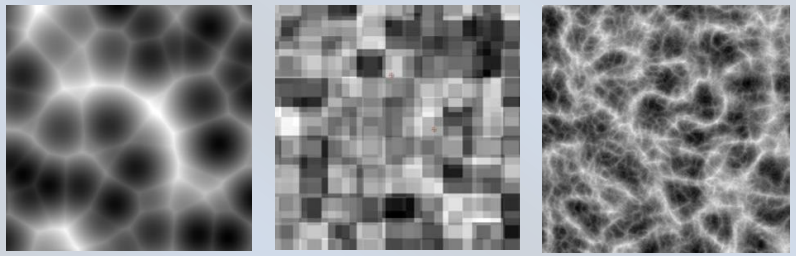
Texture Turbulence

This turbulence can be used to add additional Noise and Turbulence types into HAYATE. If you wish to use Voronoi Noise for Example you can do so using this Feature.

The texture is processed and the values are used to create a Vectorfield. The texture's channels represent the turbulence strength per axis (RGB = XYZ) The alpha channel is used to cancel out particles at the beginning. This features needs to be enabled (= Use Alpha Mask). Particles will be removed X seconds after spawning. X is defined in Threshold. Hit Update to create the Turbulence.

The Width and Height of the Texture used will represent the dimensions of the Turbulence (1 Pixel = 1 World Unit). By Changing the Frequency it is possible to adjust this value.





Other Noise types

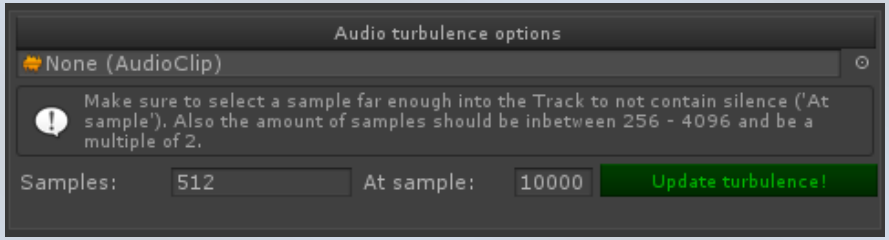
Tip!

The Texture needs to be read/write enabled to work. HAYATE will throw a warning otherwise.

Audio Turbulence

This feature is used to save memory (by reusing sound or music in your game) and give you an additional playground.

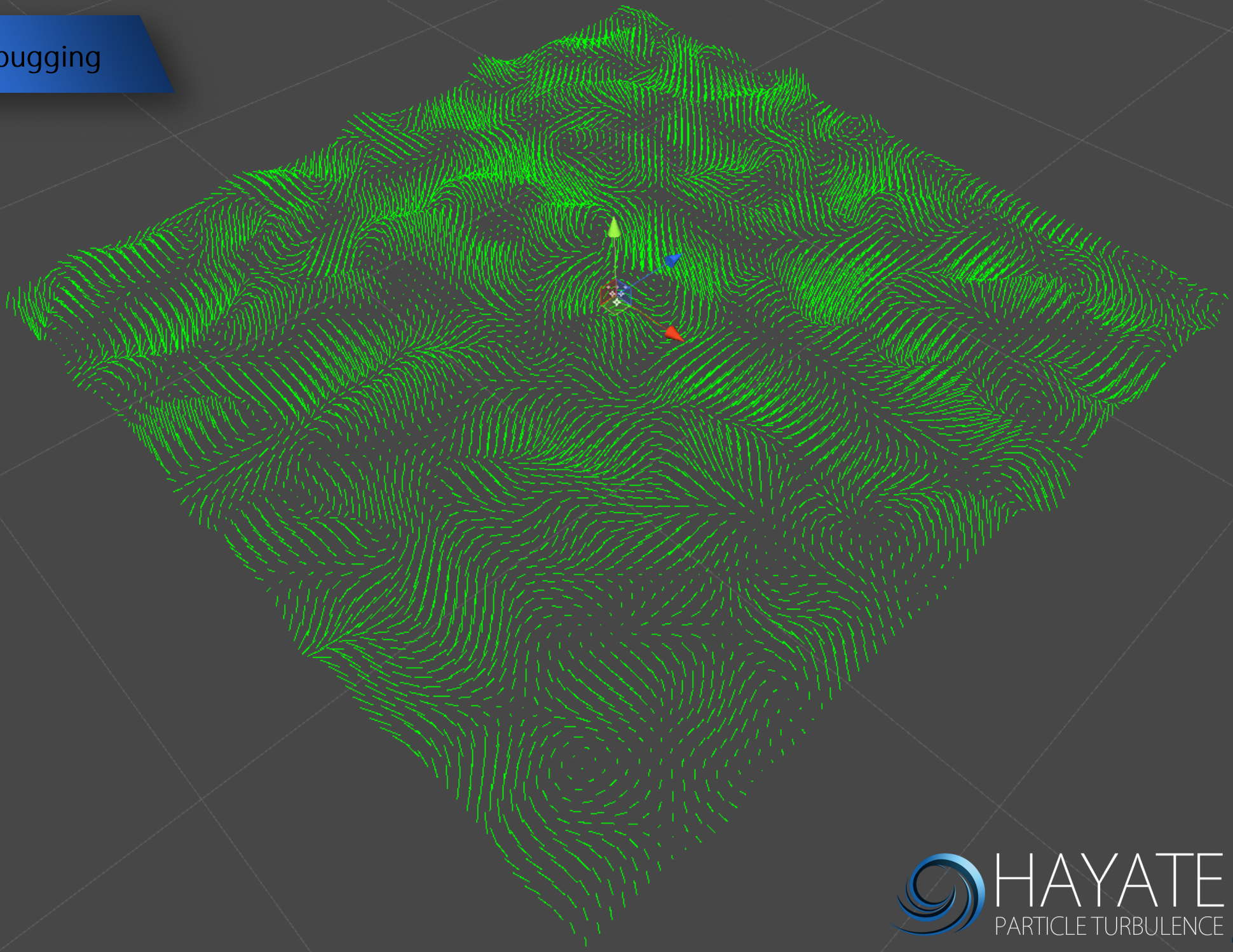
To set this up simply drag and drop an AudioClip into the AudioClip field and set the size of the Sample buffer (= Samples). If you hit Update Turbulence, HAYATE will create Turbulence beginning at sample X (= At sample) which is Y Samples long (= Sample buffer).



Tip!

The AudioClip also needs to be read/write enabled.

Debugging



HAYATE
PARTICLE TURBULENCE

Debug Options

Visualizing the Turbulence Field is sometimes essential to understand what the Turbulence is doing. That's why HAYATE comes with a tool that helps you with that.

Field Size sets the bounds of the Visualization and Step size the amount of drawn samples per world unit. Be careful not to use too many steps since it might make the Editor very slow and might even freeze! To increase visibility you can change the Vector Field's color and ray length.

Tip!

To See the Vector Field at least one axis needs to have som type of Turbulence enabled.

Transform Particles

HAYATE 3
PARTICLE TURBULENCE

Transform Particles settings

This feature automatically Instantiates and Destroys Transforms according to corresponding particles.

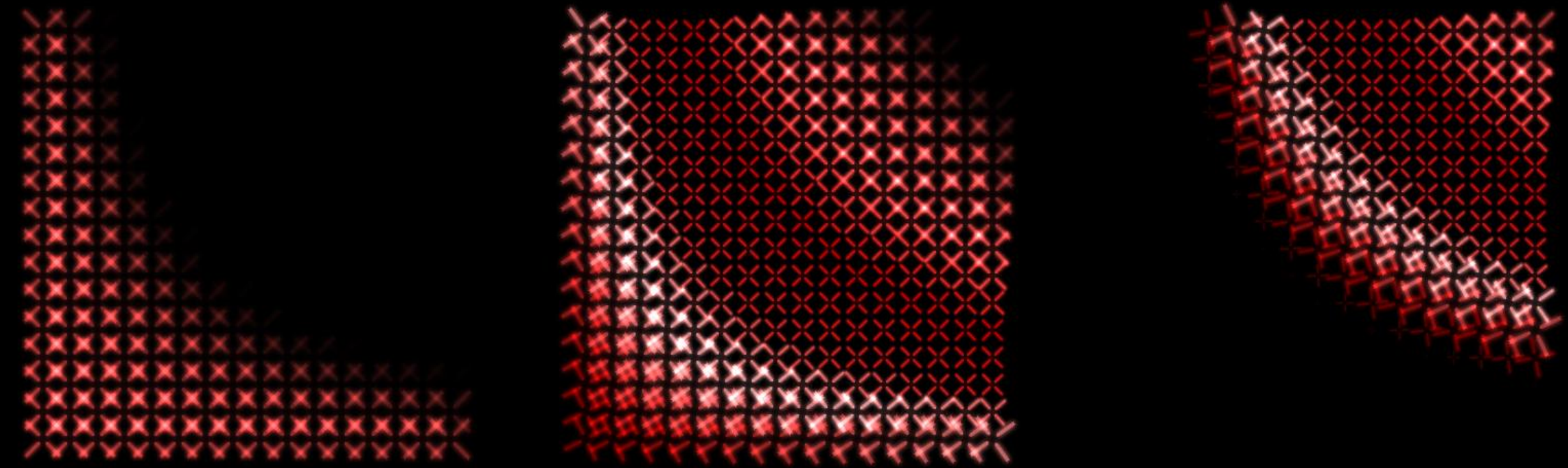
Assign a Prefab to the Transform slot in the Transform Particle Tab to enable this feature. You can now define how the Transforms behave.

They can be destroyed immediately or after a delay on particle death and rotated towards flight direction.

Tip!

When using Trails as Transform Particles make sure to set a delay (Trail lifetime) to prevent Trails from popping away when being destroyed.

Special Effects



HAYATE 3
PARTICLE TURBULENCE

Special Effect Settings

This Tab contains a mix of miscellaneous features and an UI Effect.

Misc features

Always on top – Set the render queue to foreground. Useful when using NGUI.

TimeScale independent – Effect continues even though time scale has changed

UI Effect

This effect has been designed to work in conjunction with UI but it can also be used normally. It puts Particles side by side to and alters their lifetimes to create a swipe effect. To make this particle fire OnStart(), enable the Create On Start button. Additional features are then revealed:

Dimensions, Particle Size, UIScale and Particle Order.

To make this effect work, you will have to set the dimensions (world units) and Particle Size of the Effect.

Example:

Effect Size 10 x 10 x 0.1, Particle Size 1 will create 10 x 10 x 1 = 100 Particles

10 x 10 x 0.1, Particle Size 2 will create 5 x 5 x 1 Particles.

UI Scale will give you the possibility to scale the Effect even though the dimensions are much higher. This is especially useful if working with a GUI system that relies on special scaling.

Example

10 x 10 x 0.1, Particle Size 1, UIScale 0.5, will create 100 Particles with dimensions 5 x 5 x 1

To use this feature from code simply use the following method:

```
hayateReference.CreateParticles( Vector3 _position, Float _width, Float _height, Float _depth, Float _particleSize, Float _uiScale, HayateEnums.buildOrder _buildOrder);
```

Particle Order

The direction of the swipe effect can be chosen here.

Top Left -> Bottom Right, Top -> Bottom, Top Right -> Bottom Left, Left -> Right, Bottom Right -> Top Left, Bottom -> Top, Bottom Left -> Top Right, Right -> Left

Tip!

Use the Emit button or press “E” in PlayMode to preview your settings.

Mesh Targets

HAYATE 3
PARTICLE TURBULENCE

Mesh Targets Settings

To make Particles follow or move to a Mesh's Vertices, you can use this feature. Both normal Meshes and Skinned Meshes are supported. The amount of vertices equals the amount of Targets, Particles can move to, but the amount of Particles can exceed the amount of Vertices. In that case multiple Particles will use the same Vertices.

In case you are using a normal Mesh specific subdivision features can be used to add more vertices to your Mesh. Note that those ParticleSystems can't be instantiated if HAYATE is used for subdivision because Unity can't save the Vertex List as a Prefab.

Enable Mesh Targets in the options to see additional settings:

- GameObject Field (needs MeshFilter or SkinnedMeshRenderer)
- Animate Button toggle Speed between Static Value and AnimationCurve
- Effect Type
- Speed Float Field or AnimationCurve

If using normal Meshes Subdivision options are enabled:

- Division Type
- Get Mesh Info / Subdivide / Reset Buttons

Recommended Subdivision workflow

To add more vertices to your Mesh using HAYATE's built in Subdivision Feature, you first have to know how big the smallest and biggest Triangle in your Mesh are, since HAYATE takes this into account to evenly distribute Vertices across your Mesh. Hit Get Mesh Info and check the Console window. It should look something like this:

```
Vertices: 525

Triangles: 2280

Smallest trangle: 0.0009600155 Largest triangle: 0.006831606

Mesh target ready to work with 525 particles.
```

Set the Threshold to a Value smaller than "Largest Triangle" to make all Triangles bigger than that smaller while adding vertices. Set the Threshold to a Value smaller than Smallest Triangle and all Triangles will be broken down. Note that even newly created Triangles will be Subdivided as well.

The maximum amount of Vertices per Mesh is limited to 65.000 Vertices by Unity. When subdividing you might have to stay well below this number since the Limit is set to Vertices on the GPU. Due to hard and soft edges the amount of Vertices may vary.

Tip!

Set Shurikens SimulationSpace to World in order to get the correct WorldSpace Vertex positions.

Collision Options

This is a small Feature that allows you to emit Particles if the collider attached to the ParticleSystem collides with something. The amount of particles to emit can be specified in the Emit Particles Int Field.

Additional Information

What to do if your Settings are not working as expected / at all?

If your Turbulence Settings behave strangely, don't worry! It's most likely a small Problem that can be resolved in seconds. Here are the most common problems you might experience when setting up HAYATE.

P: When using Turbulence Particles move in very quickly and change their direction frequently.

A: *This might be a Problem of your Settings in the Frequency Fields. Check if the Values are very small. If so Increase them. Values between 0.1 – 10 are the most usual settings*

P: Hayate's and Unity's Axes are not in sync and Particles don't fly to the correct Mesh Target positions

A: *Some Features of Hayate require Shuriken to be set to World Space Simulation. This setting should also be activated when you don't want the Particles to follow the Emitter..*

P: When Particles reach Mesh Target positions and become slower, they are still twitching around if Relative Simulation is used.

A: *To make some effects work with Shuriken and save Memory, Unity Particle Values are used. The Velocity of the Particles is reset every frame and used for other calculations. To prevent jittery movement from happening simply enable Velocity Oover Lifetime in the Shuriken settings and reduce the velocity towards the end of the Particle Lifetime.*

P: Attractors have no effect.

A: *To make Attractors work, Particles must be inside the Attractors Attenuation field and the Strength must be bigger than 0 to attract Particles. Here again Wolrd Space Simulation must be used in order to work correctly.*

P: Attractors don't create a Vortex like effect.

A: *In Order for Particles to get sucked in to an Attractor, they must become slower. You can regulate that by limiting the Velocity over lifetime (closing in to 0 Speed towards the end of a Particles Lifetime).*

P: Attractors have no effect.

A: *To make Attractors work, Particles must be inside the Attractors Attenuation field and the Strength must be bigger than 0 to attract Particles. Here again World Space Simulation must be used in order to work correctly.*

P: Particle Collision doesn't work anymore.

A: *If you are using Simulating the Particles Position instead of Velocity, Hayate will move them through colliders even though they collided with something. If you need Particle Collision you will have to use Velocity Simulation instead.*

P: Weird Logs when using Animation Curve Turbulence.

A: *This is a Bug by Unity. It has been reported and might be fixed in the later Versions than 4.01. These Warnings are harmless and only occur sporadically. It's safe to ignore them if you don't notice any Performance drops.*

…

P: I have a Problem that is not listed here!

A: *If you have a Problem with HAYATE, suggestions and or Feature request please don't hesitate to contact support@maxproude.com*

Request will be answered as soon as possible.

If you have any suggestions, ideas or want to have a special tutorial, please don't hesitate to contact me!

support@maxproude.com

More updates will follow soon!

MAX PROUDE