Mesh Tracer

Dog Eat Dog Games

Thank you for your purchase!

**Links**

Tutorial Videos on how to use this tool:

<https://www.youtube.com/playlist?list=PLc2O4sFLm5sQ94P6NUQdmGISNlnfa3Omt>

Support forum:

<http://www.dogeatdoggames.com/#!support/c1x9v>

(support link may change, visit [www.dogeatdoggames.com](http://www.dogeatdoggames.com) and navigate to support)

Checkout some of our other tools or games here:

[www.dogeatdoggames.com](http://www.dogeatdoggames.com)

[www.facebook.com/dogeatdoggames](http://www.facebook.com/dogeatdoggames)

[www.twitter.com/dogeatdoggames](http://www.twitter.com/dogeatdoggames)

**What’s Included**

* **Tons of premade Particle Effects and Trail Renderer prefabs to get you started.**
* **5 example scenes**
* **4 unique scripts** 
  + Vertex Particles
  + Vertex Tracer
  + Triangle Tracer
  + Edge Tracer

**Examples**

* Beach – relax at a particle beach!
* LineTracer – create awesome effects on top of 3D models.
* NightClub – get your dance on.
* Random – an overview of many different effects.
* Tree – an artistic tree scene.

**Scripts**

**This section describes the parameters used in each core script.**

**Vertex Particles**

* Wave animation – control particle animations.
  + AnimCurve – curve of the animation
    - If type == static, this controls the grow/shrink over time.
    - If type == wave, this controls the position of the wave over time.
  + AnimTime – time it takes for one cycle of the animation to complete.
  + Min/max size – min/max size of the emitted particles.
* Particle LifeTime – the lifetime of each emitted particle.
* Particle System Prefab – use particles from this prefab.
  + Note: In this script, we often have hundreds of verticies, so the script emits 1 particle at each vertex, instead of instantiating an entire particle system at each vertex.
* OnStart – PlayEffect () in the Start() method
* HideMesh – hide the original mesh when you play the effect
* Type
  + Static – put 1 particle at every vertex point. Make it grow/shrink with the waveAnimation.
  + Wave\_Right (and other directions) – grow/shrink particles in a wave across the mesh in local space.
* numTracers – for the wave animations, number of waves we can simultaneously move across the mesh.
* percentVerticies – what percent of the verticies to use (setting to a low value improves performance on larger meshes).
* randStartPos – for the wave animations, start the waves at random positions on the mesh. If this is false, the waves will be evenly spaced.

**Vertex Tracer**

* Trace animation – control tracer animations.
  + AnimCurve – curve of the animation
    - This controls the postion of the tracer over time.
  + AnimTime – time it takes for one cycle of the animation to complete.
* Particle System Prefab – use a tracer from this prefab.
  + Note: In this script, this must be something with a TrailRenderer attached to it. The script then creates multiple instances of the prefab and moves them across the mesh.
* OnStart – PlayEffect () in the Start() method
* HideMesh – hide the original mesh when you play the effect
* Type
  + All of these types work by instantiating 1 or more instances of the Particle System Prefab. It then moves them across the mesh in different patternes, using TrailRenderes to draw lines.
  + Natural – move through all verticies, use whatever order they we loaded into the machine with.
  + Random – move through each vertex, ordered randomly.
  + Right,Left,Up,etc – move through each vertex, ordered in local space using the direction specified.
* numTracers – for the tracer animations, number of tracers we can simultaneously move across the mesh.
* randStartPos – for the tracer animations, start the tracer at random positions on the mesh. If this is false, the waves will be evenly spaced.

**Triangle Tracer**

* Note
  + Attach TrailRendererHelper.cs to ParticleSystemPrefab’s used in this class if you want trails to disappear in between triangle traces.
* Line animation – control tracer animations.
  + AnimCurve – curve of the animation
    - This controls the postion of the tracer over time.
  + AnimTime – time it takes for one cycle of the animation to complete.
* Tracer – use a tracer from this prefab.
  + Note: In this script, this must be something with a TrailRenderer attached to it. The script then creates multiple instances of the prefab and moves them across the mesh.
* Type
  + All of these types work by instantiating 1 or more instances of the Particle System Prefab. It then moves them across the mesh in different patternes, using TrailRenderes to draw lines.
  + All – trace out every triangle. This is very expensive for high poly meshes.
  + Loop – trace out each triangle, 1 by 1, using their natural order.
  + Trace Random – trace out each triangle, in random order.
  + Trace Connected – trace out each triangle, moving across touching triangles.
  + Disco – trace out ‘numTracer’ number of triangles at once across the mesh. Get new triangles every animation loop.
* numTracers – for the tracer animations, number of tracers we can simultaneously move across the mesh.
* OnStart – PlayEffect () in the Start() method
* HideMesh – hide the original mesh when you play the effect
* randStartPos – for the tracer animations, start the tracer at random positions on the mesh. If this is false, the waves will be evenly spaced.
* Debug – display the triangles being traced in the Scene view.

**Edge Tracer**

* Line Tracers – list of lines used to trace out the mesh.
  + Name – name each line in the inspector.
  + Draw Point – where the edge is drawn from (Only valid of IntantDraw is disabled and if start/end times for each LineTracer have been set).
  + Start Time – the time after calling AnimatedDraw(), that this line will start drawing.
  + End Time – the time after calling AnimatedDraw(), that this line will end drawing.
  + Curve – each edge on your mesh is replaced with this curve.
  + Amplitude – height of the curve.
  + LineMat – material used in the LineRenderers that get created.
  + Start/End Color – start/end color of each line.
  + Start/End Width – start/end width of each line.
  + Line Segments – line segments in each line, reduce when possible to increase performance.
* Use Outline – only use the outline of the mesh, instead of the whole thing. Significantly improves runtime performance when enabled.
* On Start – start tracing the edges OnStart().
* InstantDraw – draw the edges instantly, instead of using start/end time.
* HideMesh – hide the mesh when you draw lines.
* OnFinish() – add events that get called when the edge has been traced.

**Trail Renderer Helper**

* This helper class handles the issue where trails persist though disable-move-enable logic.

**Sources**

* SpaceShip Model - <http://www.turbosquid.com/3d-models/free-max-model-space/588767>
* Stone Pine Tree Model- <http://www.turbosquid.com/3d-models/realistic-stone-pine-obj/857432>
* Triangle Renderer Helper - <http://forum.unity3d.com/threads/trailrenderer-reset.38927/>
* Unity Standard Assets Packages
  + Camera
  + Particle Effects
* Sky5X Unity Skyboxes - <https://www.assetstore.unity3d.com/en/#!/content/6332>

**License Info**

Subject to Unity’s EULA: <http://unity3d.com/legal/as_terms>