FRAGMENTUM

Fragmentum 2.3

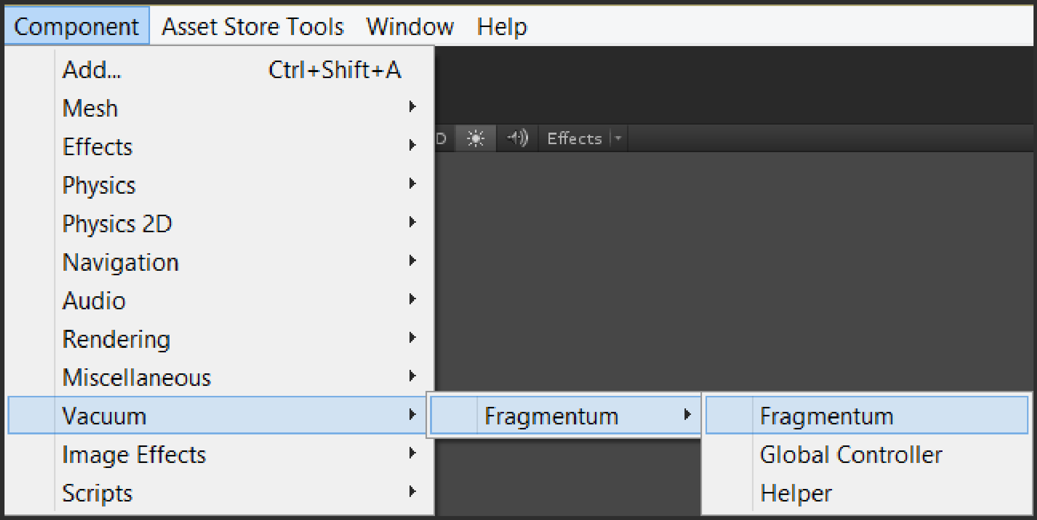
By Davit Naskidashvili

https://www.facebook.com/VacuumShaders

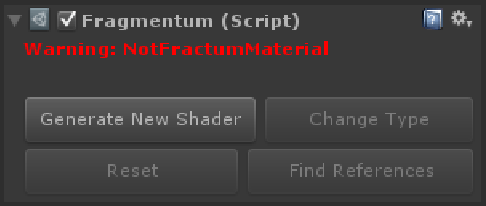
2014

Quick Start

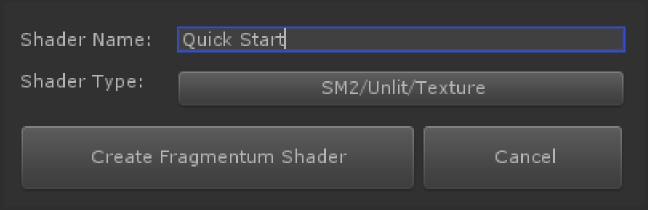
1. Assign Fragmentum script and fresh material to the renderer.



2. Generate new Fragmentum shader.

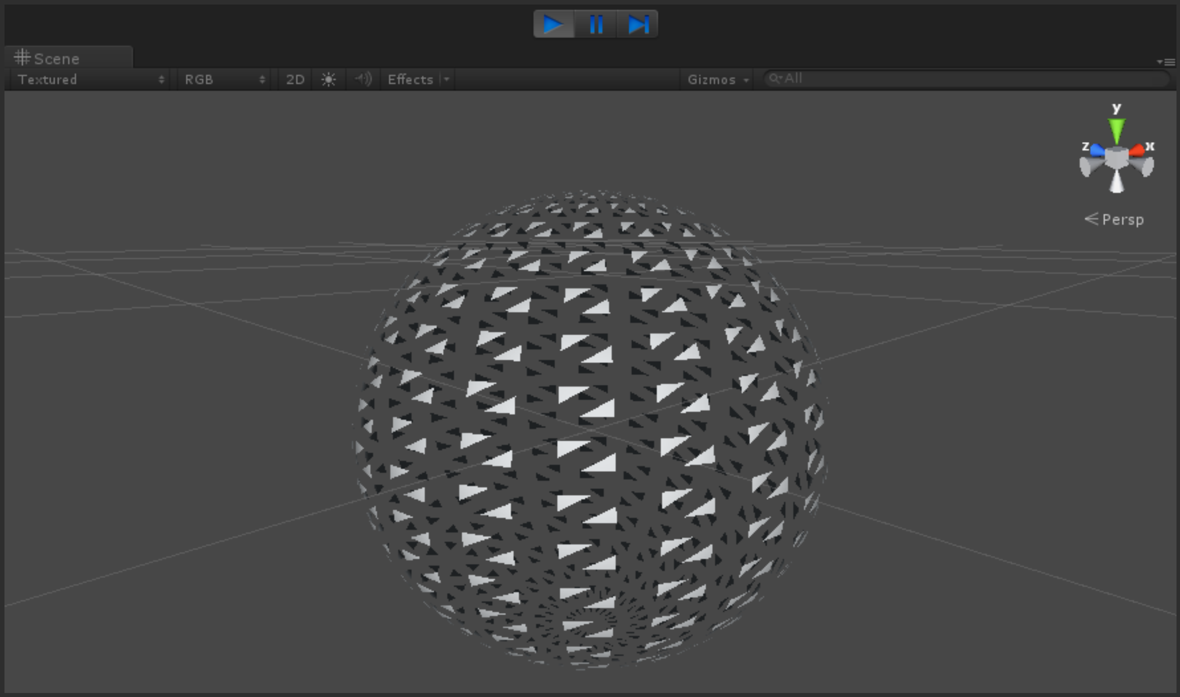

Fragmentum (Script)

Warning: NotFractumMaterial


Shader Name: Quick Start

Shader Type: SM2/Unlit/Texture

3. Push Play button.

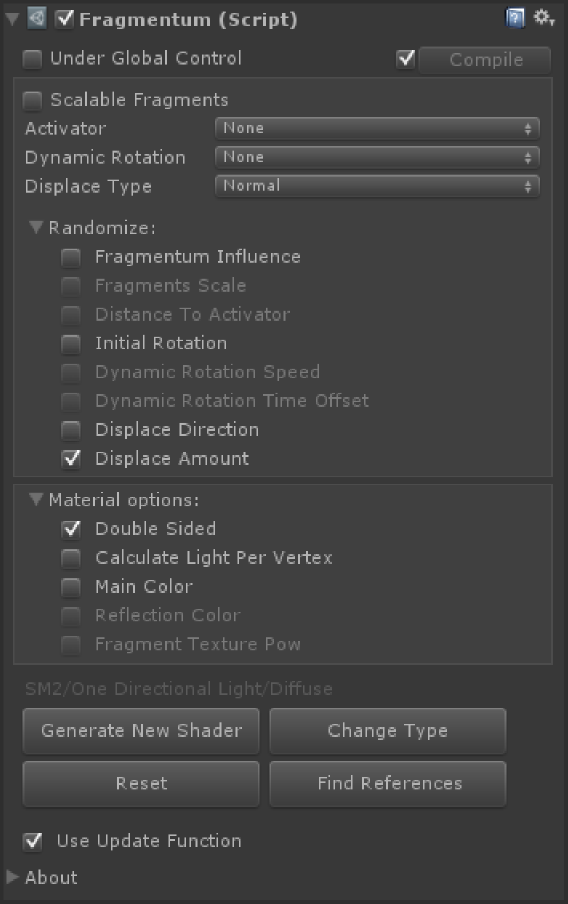


4. Or open example scene: 01. Simple Fragmentum

Note

- Shader fragments objects using only their polygons (triangles). No new triangles are generated.
- Before start, Fragmentum script prepares mesh for the shader - generates some additional data and thus rebuilds it.
- In Play mode Fragmentum script does nothing.
- Fragmentum shader is 100% vertex shader.
- Fragmentum does not work with meshes, with more than 21.000 vertices.
- Fragmentum should be the only material on the object.
- Shader is not dedicated for Skinned Meshes, but still can be used with some restriction. Works only with DX11.
- DX11 version of the shader requires SM5 and Compute Shaders support.
- Fragmentum does not require Unity Pro.
- Demo scenes cannot be used in any commercial or non-commercial projects.

Fragmentum script overview



Fragmentum (Script)

☐ Under Global Control ☑ Compile

☐ Scalable Fragments

Activator: None

Dynamic Rotation: None

Displace Type: Normal

Randomize:

- ☐ Fragmentum Influence
- ☐ Fragments Scale
- ☐ Distance To Activator
- ☐ Initial Rotation
- ☐ Dynamic Rotation Speed
- ☐ Dynamic Rotation Time Offset
- ☐ Displace Direction
- ☑ Displace Amount

Material options:

- ☑ Double Sided
- ☐ Calculate Light Per Vertex
- ☐ Main Color
- ☐ Reflection Color
- ☐ Fragment Texture Pow

SM2/One Directional Light/Diffuse

☑ Use Update Function

Fragmentum script is dedicated to minimize shader's calculations and allows choosing only those parameters that are required to achieve specific effect.

Fragmentum script is shader analyzer. It reflects shader data. Every time changing parameters inside script, changes shader too, as it recompiles shader file. If several materials use same Fragmentum shader, than changing this shader throughout any of Fragmentum script, will change effect on other materials too.

Fragmentum script displaces shader's state not objects.

Fragmentum effect depends on Fragmentum script attached to the object and Fragmentum shader, used by renderer's material. If one of them is missed than there is no fragmentum effect. Enabling parameters inside Fragmentum script activates appropriate parameters inside material editor.

Auto Compile on. Changing parameters inside script will automatically recompile shader file, to reflect changes.

Auto Compile off. If it is required to change several parameters, it is better to disable auto compile. Button becomes red to notify that shader requires recompilation.

All fragmentum effects of the object depend on **Fragmentum Influence (FI)** parameter (it is the first topmost parameter inside material editor). It changes from 0 to 1. When it is zero, there is no fragmentum on the object. All effects: displace, rotation, scale, dissolve and others are multiplied by this value. Value of 1 gives full fragmentum effect.

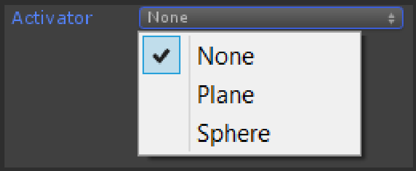
In the Play mode try reduce **Fragmentum Influence** slider from 1 to 0, to view how object is losing all its effects (“fragmentumness”).

It is better to set up all parameters while **Fragmentum Influence** is set to 1.

This parameter is explained later at the end.

Enables fragments scale changing with ***Fragments Scale*** parameter inside material editor. Scale like all other parameters depends on **FI**.
If **FI** is 0, than fragments scale is - 1, same as the original mesh.
If **FI** is 1, than fragments scale is equal to ***Fragments Scale*** value inside material editor.
View example scene: 02. Scale

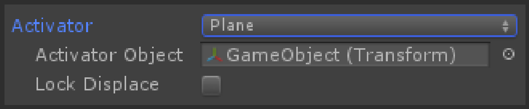
Activators



Activator defines the rule, by which triangles of the mesh are effected by Fragmentum.

3 types of activators are available.

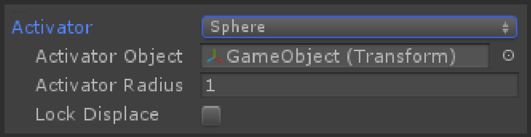
1. None – There is no rule. All triangles are affected by Fragmentum.
2. Plane – Plane’s position and its rotation (normal) defines which triangles will be affected by Fragmentum.
3. Sphere – Sphere position and its radius define which triangles will be affected by Fragmentum.



Activator: Plane

Activator Object: GameObject (Transform)

Lock Displace □



Activator: Sphere

Activator Object: GameObject (Transform)

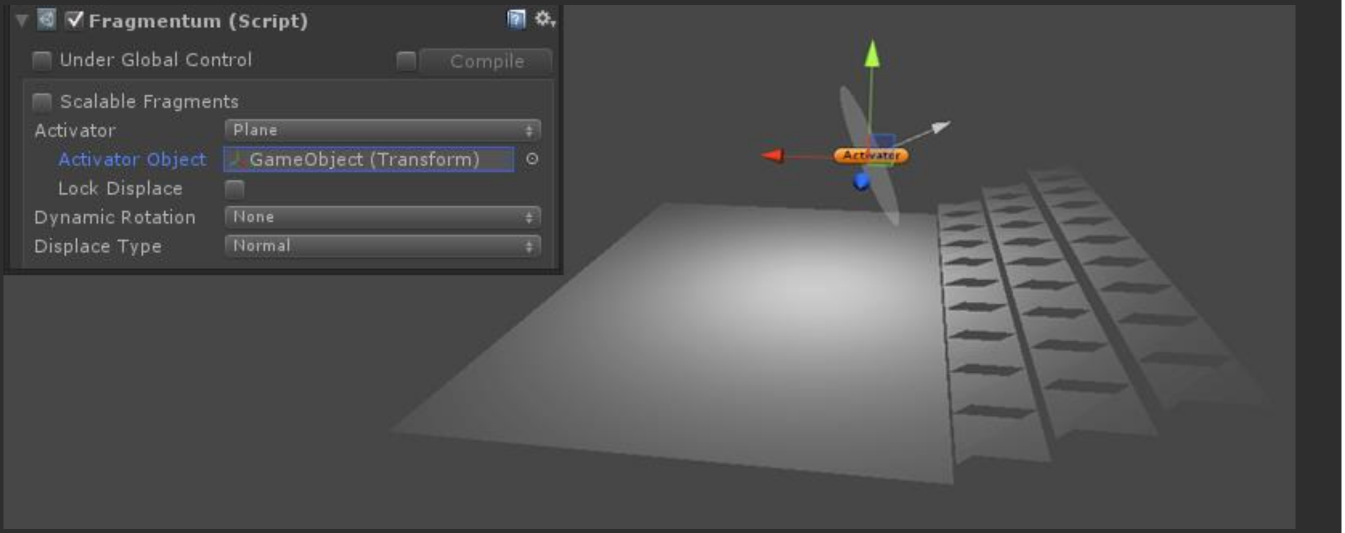
Activator Radius: 1

Lock Displace □

If activator is used, several parameters become enabled.

- Activator Object – For plane activator it is the object whose position and rotation (normal) “activates” fragmentum effects of the mesh. For sphere activator only objects position is used.
- Activator Radius – Used only by sphere activator. All triangles of the mesh within this radius are affected by fragmentum.
  If radius is negative Fragmentum effect is inverted: Triangles outside of the radius are fragmentumed, triangles inside radius do not.
- Lock Displace – More triangles are under influence of the activator, more there displace is. If these parameter is enabled, triangles will stop displace after reaching ***Displace Amount*** (parameter inside material editor).

See examples scenes about activators: 03. Plane activator and 04. Sphere activator



☑ Fragmentum (Script)

□ Under Global Control □ Compile

□ Scalable Fragments

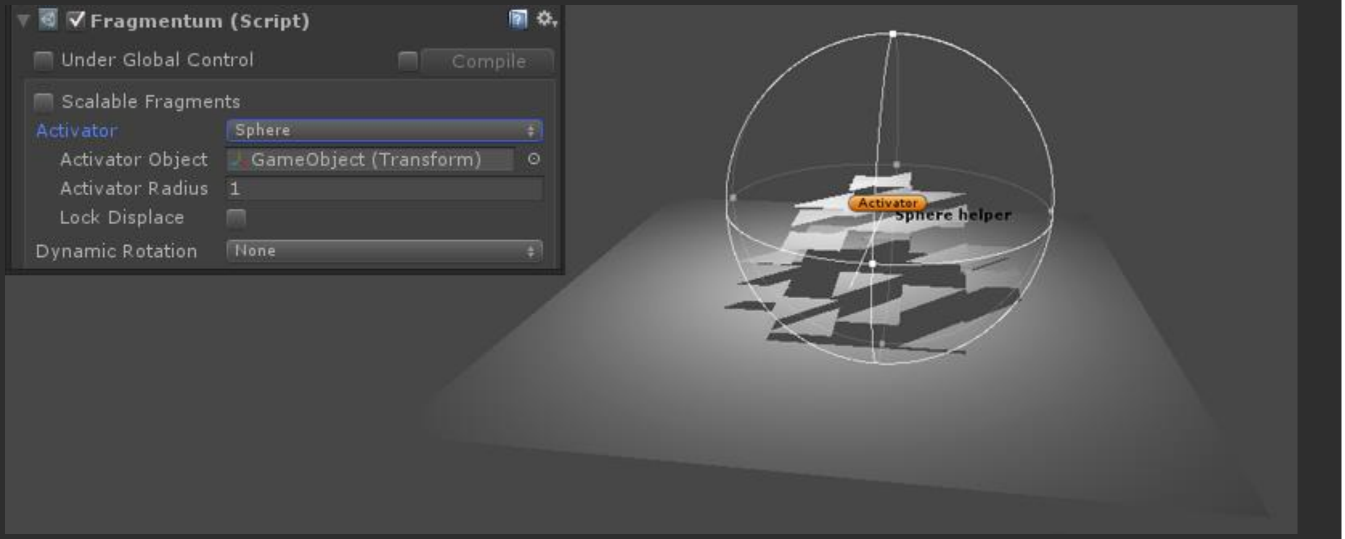
Activator: Plane

Activator Object: GameObject (Transform)

Lock Displace □

Dynamic Rotation: None

Displace Type: Normal



☑ Fragmentum (Script)

□ Under Global Control □ Compile

□ Scalable Fragments

Activator: Sphere

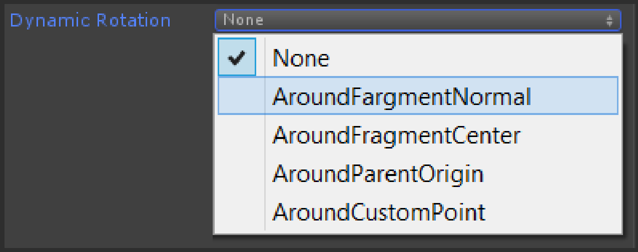
Activator Object: GameObject (Transform)

Activator Radius: 1

Lock Displace □

Dynamic Rotation: None

Dynamic Rotation



Fragmetnum has 4 different types of dynamic rotations.

If any one of them is used, ***Rotation Speed*** becomes enabled inside material editor.

- Around Fragment Normal – Triangle (fragment) rotates around its own normal and in the direction of normal axis.
- Around Fragment Center – Triangle (fragment) rotates around its center, but direction axis is random.
- Around Parent Origin – Triangle (fragment) rotates around parent’s origin.
- Around Custom Point – Fragments can rotate around custom point and along normal of that point.

If dynamic rotation is enabled, this parameter defines mathematical order whether rotation should be calculated after displace or before it.

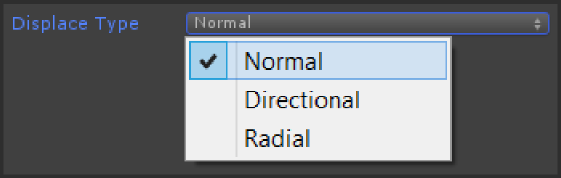
See example scenes for rotations: 07. Rotations (not included in Free version)

(*Teapot meshes with different rotation types*)

Note: Rotation is the most expensive effect.

d3d11 requires 14 instructions, d3d9 - 51 and opengl - 67.

Displace Types


Displace Type: Normal

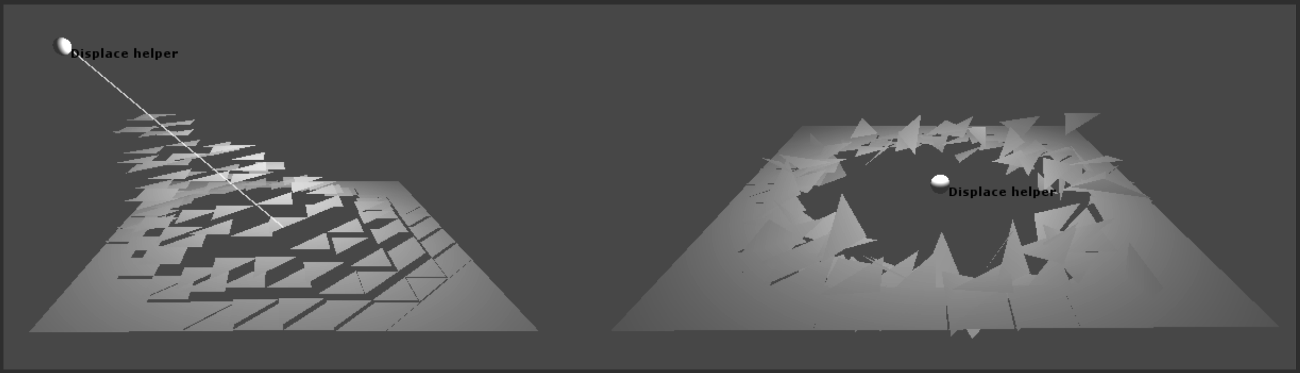
- ✓ Normal
- Directional
- Radial

Three types of displace are available in Fragmentum.

There determine displace direction type.

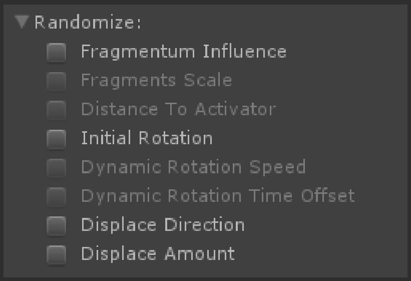
- Normal – Triangle’s (fragments) are displaced in the direction of their normal
- Directional – Triangle’s (fragments) displace direction vector is set from material editor, or custom object can be used to control displace vector.
- Radial – Triangle’s (fragments) move toward one point or away from it, it defends on ***Displace Amount*** value. Point’s position can be controlled from material editor or from Inspector editor.

See examples scenes: 05. Directional Displace



(*Directional and Radial displace types*)

Randomize

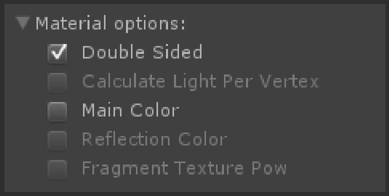

▼ Randomize:

- ☐ Fragmentum Influence
- ☐ Fragments Scale
- ☐ Distance To Activator
- ☐ Initial Rotation
- ☐ Dynamic Rotation Speed
- ☐ Dynamic Rotation Time Offset
- ☐ Displace Direction
- ☐ Displace Amount

Enable randomize of the corresponding parameter.

After enabling randomize can be controlled from the material editor.

Material options


Material options:

☑ Double Sided

□ Calculate Light Per Vertex

□ Main Color

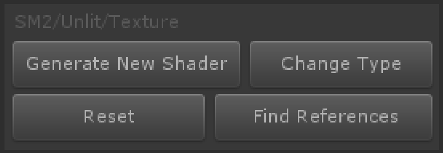
□ Reflection Color

□ Fragment Texture Pow

Allows reduce instruction count of the shader.

- Double Sided – Whether material should be double sided.
- Calculate Light Per Vertex – For sm2 and sm3 enables light calculation to be done per vertex, instead of standard per pixel calculation.
- Main Color – Enable to change material's color. If material is transparent A channel controls transparency.
- Reflection Color – Enable to change reflection color (RGB) and reflection intensity (A).
- Fragment Texture Pow – DX11 and sm3 shaders have ***Fragment Texture*** to control are of the **Fragmentum Influence** (**FI**). This parameter enables "Pow" function over texture.

Other options


SM2/Unlit/Texture

- Gray string displays current Fragmentum shader's type.
- Generate New Shader – Opens shader creator window. Generated material will be created inside Fragmentum subfolder "shaders/custom" and should stay there. After generating, new shader will be applied to the material.

- Change Type – Changes shader type without losing active parameters.
- Reset – Resets parameters of the shader.
- Find References – Find all objects inside scene editor which are using current Fragmentum shader.

☑ Use Update Function

Use Fragmentums's Update() function to update:

1. Plane activator's position and rotation
2. Sphere activator's position and radius
3. Point position for Directional and Radial Displace
4. Point's position and rotation for Rotation Around Custom Point

Fragmentum shader types

Fragmentum has Mobile, SM2, SM3 and DX11 shader types support.

Supported lighting types are:

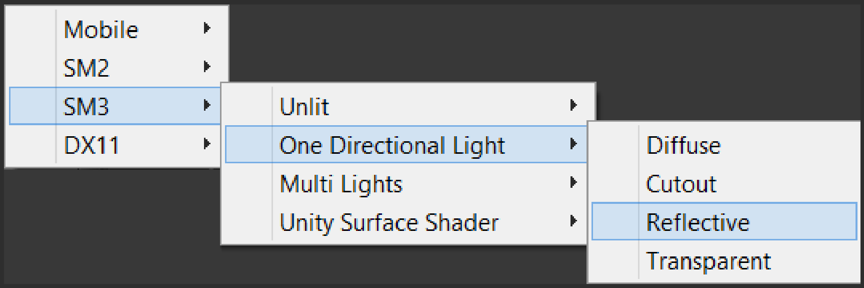
- Unlit
- One Direction Light
- Multiple Light (Directional, Point and Spot lights)
- Unity Surface shader. Same as above, but light is calculated through Unity built-in Surface shaders

All of them support light maps.

Per pixel effects are:

- Texture (unlit)
- Diffuse
- Reflection
- Transparent
- Cutout (dissolve). Only triangles (fragments) under **FI** get this effect, not the whole mesh.
- Bump Specular (DX11)
- Bump Specular Reflection (DX11)
- 3D Extrude (DX11)

Shader type can be changed by clicking Change Type button or while creating new Fragmentum shader.



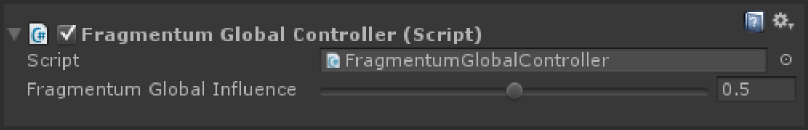
Global Controller

□ Under Global Control

If enabled than current shader's **Fragmentum Influence** parameter is under global control. It is used when scene contains multiple fragmentum objects, with deferent fragmentum shaders and it is necessary to manipulate their "fragmentumness" simultaneously.

Fragmentum Influence is the only one parameter that can be "global".

To control global **Fragmentum Influence** scene should contain only one reference of the script **Fragmentum Global Controller**.



☑ Fragmentum Global Controller (Script)

Script FragmentumGlobalController

Fragmentum Global Influence 0.5