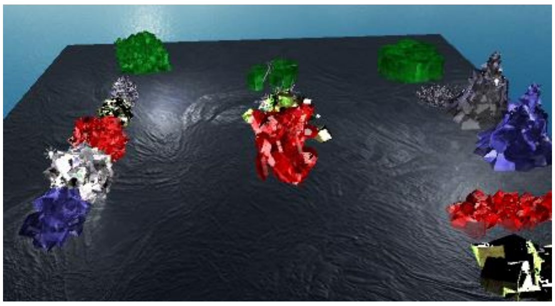
Crystal Generator

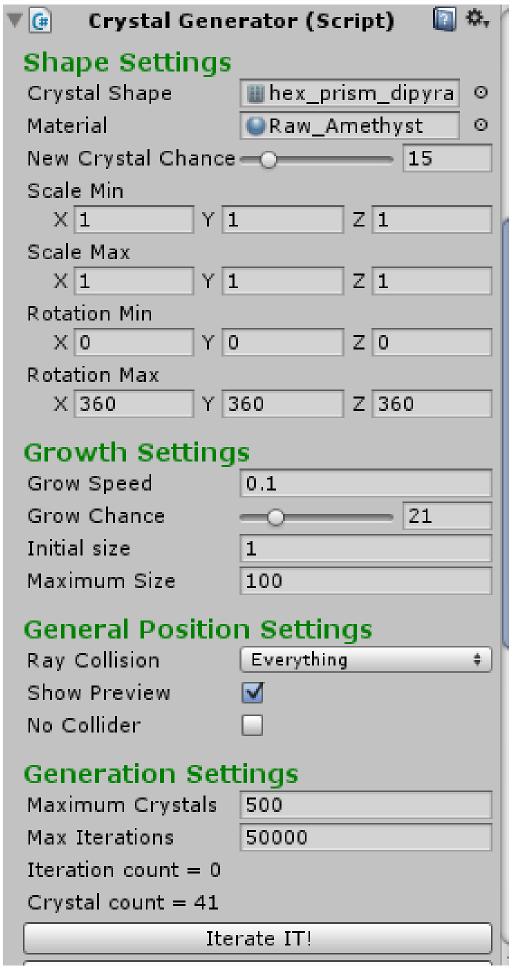
Introduction:



I am fascinated from crystals and minerals found in Nature. However as a game programmer, I thought that it must be able to simulate crystal growth in nature. So I decided to create a tool which is capable of doing this. It is possible to create all kind of natural crystals by describing the crystal shape.

How to use:

It is very easy to use. Just drag one of the prefabs included in the prefabs folder into the scene. Make sure that you have at least one solid surface in your scene as some growing algorithms require solid surfaces to grow crystals on. The values are already preset and you only have to click on one of the iterate buttons. You can also change those values described below to get your desired behavior.



Crystal Generator (Script)

Shape Settings

Crystal Shape: hex_prism_dipyra
Material: Raw_Amethyst
New Crystal Chance: 15
Scale Min: X 1 Y 1 Z 1
Scale Max: X 1 Y 1 Z 1
Rotation Min: X 0 Y 0 Z 0
Rotation Max: X 360 Y 360 Z 360

Growth Settings

Grow Speed: 0.1
Grow Chance: 21
Initial size: 1
Maximum Size: 100

General Position Settings

Ray Collision: Everything
Show Preview: ☑
No Collider: □

Generation Settings

Maximum Crystals: 500
Max Iterations: 50000
Iteration count = 0
Crystal count = 41

There are many options how crystals could grow in real life. These options are positioning algorithms which are attached to the game object. An example would be the crystal growing process of salt crystals where a small stitch is inside a solution where they grow from the stitch. Further information is in the Chapter “Positioning”

If the crystal generator has no positioning algorithm attached, crystals will be created at local position (0,0,0) which will result in a sphere after many iterations.

Parameters:

Shape Settings:

- **Crystal Shape:** Describes the basic structure of a crystal. As every natural occurring crystal has a preset shape. This asset contains a set of isometric and non-isometric shapes. However you can also use your own model as preset. It is recommended to keep the vertex count of the preset small.
- **Material:** Material of the crystal.
- **New Crystal Chance:** The probability a new single crystal is generated when iterating.
- **New Crystal Scale Min/Max:** Minimum and Maximum starting scale vector. New crystal has a random value between those two. Values of Zero are forbidden as it screws up the generation process.
- **Random Rotation Min/Max:** Orientation of a new crystal.

Grow Settings:

- **Grow Speed:** The speed how fast a crystal will grow.
- **Grow Chance:** The chance in percent a crystal will grow. Low probability will result in many smaller crystals.
- **Initial Size:** The initial size of a new crystal. (Value 0 is forbidden)
- **Maximum Iterations:** The maximum amount of iterations allowed.

General Positioning Settings:

- **Ray Collision:** Some algorithms use the environment to create crystals. This setting describes what collider can be hit. Keep in mind that crystals also grow on previous generated crystals.
- **Show Preview:** Displays a preview simulation. Lines show where crystals will grow.
- **No Collider:** If set, no mesh collider will be created. Set it if the crystals are used as special effect.

Generation Settings:

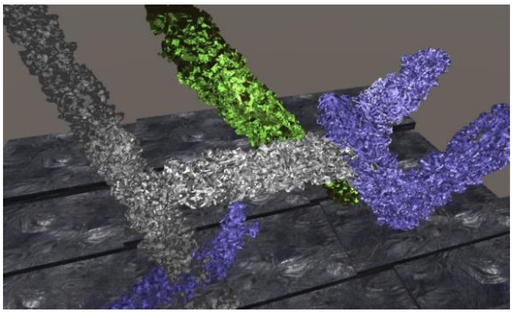
- **Maximum Crystals:** Maximum amount of crystals the generator is allowed to create
- **Maximum Iterations:** Maximum amount of grow cycles allowed
- **No Collider:** If set, no mesh collider will be created. Set it if the crystals are used as special effect.

Buttons:

- **Iterate It /10/100:** Does grow cycles.
- **Reset:** Removes the current crystal.

Extreme Crystal Generator

It is possible to create crystalline structures with an extreme amount of crystals. This became possible as I found out how to deal with large amounts of meshes. In order to do this, either drag in one of those Sample Prefabs in the Extreme folder which is in the Prefabs folder, or simply add the ExtremeCrystalGenerator script and a positioning algorithm script to an object.

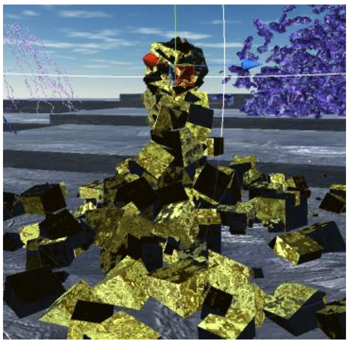


This special Generator is capable of creating an extreme amount of crystals. For easier handling, it has an Iterate It * 1000 and 10000 Button. For optimization reasons, this generator does only spawn crystals in their initial state and growth is a random stage between Sizefactor_Min/Max. Small iteration counts can be used for real time growing. Also the Maximum Crystal count is removed as it is not needed. The maximum limit is what your computer can handle where the limiting factor is your memory and graphics card.

Positioning

The old Positioning system was inside the Crystal Generator which was not flexible enough. Therefore the positioning is now a separate component. The old positioning is REMOVED now! If the generator has no CrystalPositioning script attached, the result will be a crystal sphere like in the right image. The only setting which sticks to the generator is the Collision Layer as it is needed by most positioning algorithms. The scene “NewPositioning” shows every positioning algorithm currently implemented.

Spherical:

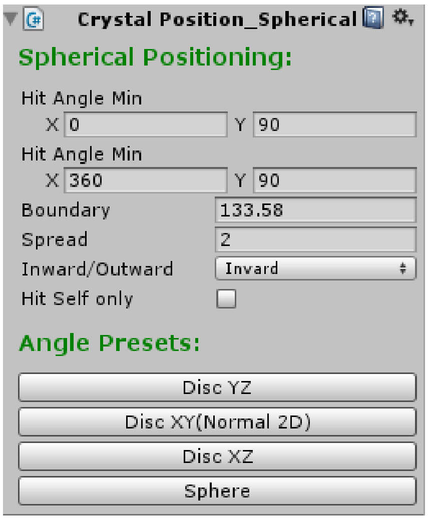


The spherical system represents the original positioning which was included directly into the crystal generator. Crystals will grow from the inside or outside and has the option to collide only with itself or the world. The position is determined by ray casts origin from the center or outer boundary. The hit point of such ray is the position of the next crystal. If it doesn’t hit a collider, the current iteration is suspended and no crystal will be created.

If crystals can collide with its previous crystals, the result will grow towards the center like in the left image where the crystal grew from the ground towards the center.

Parameters:

- **Hit Angle Min/Max:** Possible direction of the ray casts. When draw seeds flag is set, green lines will show the possible directions.
- **Boundary:** Crystals can only grow inside the boundary. Ray casts will not hit anything outside the boundary
- **Spread:** Random spherical offset which is added to the point determined by the ray cast.
- **Inward/Outward:** Describes the Direction. When set to inward, ray casts will be cast from the boundary towards the center and opposite when set to outward.
- **Hit Self only:** Ray casts will ignore the surrounding world and will only hit itself. The initial crystal will be in the center to have something to grow on. Is only useful when direction is set to inward. The result will be a crystal sphere when set to outward



Crystal Position_Spherical

Spherical Positioning:

Hit Angle Min X 0 Y 90

Hit Angle Min X 360 Y 90

Boundary 133.58

Spread 2

Inward/Outward Invard

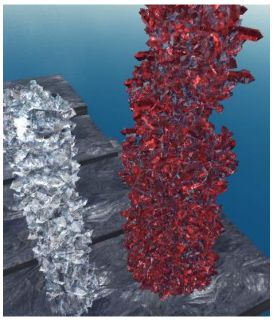
Hit Self only □

Angle Presets:

Angle Presets:

- Setting the Hit Angle correct is not easy sometimes so these buttons will change it to get the according result. Common are the disc shapes for each plane direction (for 2D Games)

Beam:

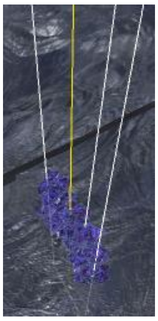


As the name suggests, this system acts like a beam laser where crystals will grow on any collider it hits. The direction of the beam is described by the transform component and is the local x axis. After many iterations, the resulting crystal will end up becoming a pillar if it can collide with itself like in the image.

This algorithm has only three settings which are:

- Range: Length of the Beam
- Boundary: Radius of the Beam
- Spread: Random offset added to the hit point

Box

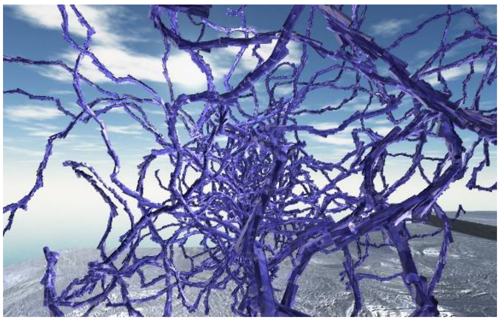


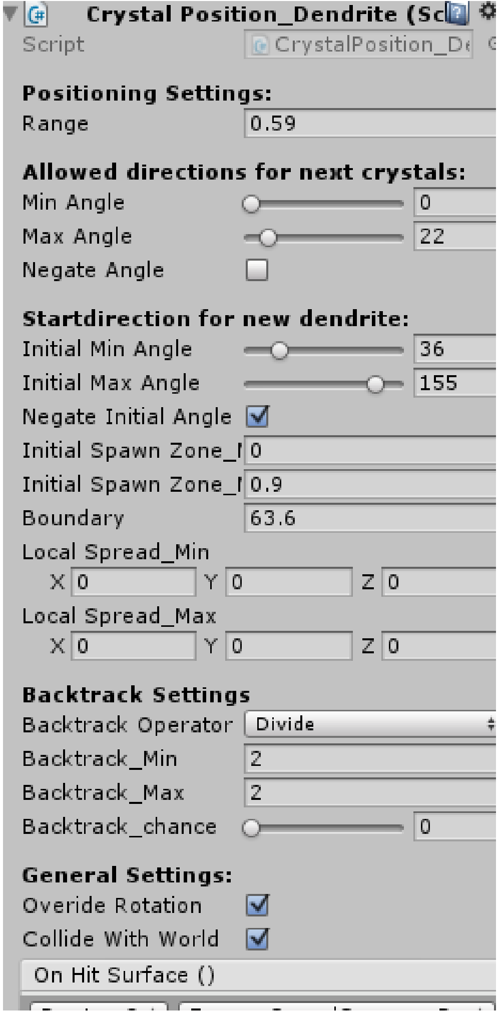
This concept is similar to the Beam script. It also acts like a beam laser but the range, width and height is described by the Box Size.

The x and y axis of the Box Size setting describes the length and width of the beam. The z value is the height value which acts like the Range value. The direction is also described by the transform. If x or y is 0, the result will be a wall like crystal.

Dendrite (UPDATE 3.0):

This is a new positioning system which creates dendrite like structures and is included in the recent update. It creates structures outgoing from the center and grows like roots in all direction. This algorithm also ended up becoming an unusual lightning generator by accident during development. It is also the most complex positioning system in this asset at the moment so you probably have to play around with it.





Crystal Position_Dendrite (Sc

Script CrystalPosition_De

Positioning Settings:

Range 0.59

Allowed directions for next crystals:

Min Angle 0

Max Angle 22

Negate Angle □

Startdirection for new dendrite:

Initial Min Angle 36

Initial Max Angle 155

Negate Initial Angle ☑

Initial Spawn Zone_I 0

Initial Spawn Zone_I 0.9

Boundary 63.6

Local Spread_Min

X 0 Y 0 Z 0

Local Spread_Max

X 0 Y 0 Z 0

Backtrack Settings

Backtrack Operator Divide

Backtrack_Min 2

Backtrack_Max 2

Backtrack_chance 0

General Settings:

Overide Rotation ☑

Collide With World ☑

On Hit Surface ()

Parameters:

Positioning Settings:

Range: Space between crystals.

Angle Settings: This describes the allowed angle between the next crystal point and the previous point. The white lines indicate the possible directions. For every new crystal, the angle must be between **Min Angle** and **Max Angle**. If **Negate Angle** is set, the opposite is the case so it is not allowed to be between these angles.

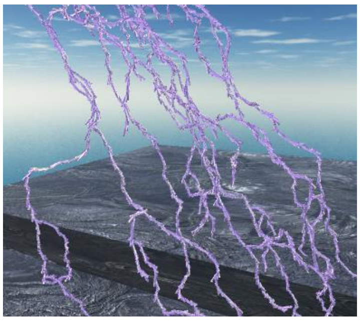
Initial Angle Settings: Describe the start direction of a new dendrite. It follows the same rules as the Angle Settings

Initial Spawn Zone Min/Max: Set the possible start position of a new dendrite. The position will be a point with a distance between Min and Max. A new dendrite will form at the beginning or if the old one exceeds the **boundary.**

Local Spread: These values force dendrites into a certain direction. The direction is local and the result will end up becoming similar to lightning strikes if the y value is high.

Backtrack Settings: For every crystal generated, there is a chance to go back to previous points and start a new dendrite from this point. The direction is also described by the initial angle settings. Two operators are possible which are Subtract and Divide. The influence value x is a random value between Backtrack Min/Max.

- Subtract just goes back x iterations and continues from there.
- Divide goes back to position current dendrite count/x which means if x is set to 2, it will go back half the way and continues from there on. Note if x is 0, x will become 1 to prevent divide by zero issues.

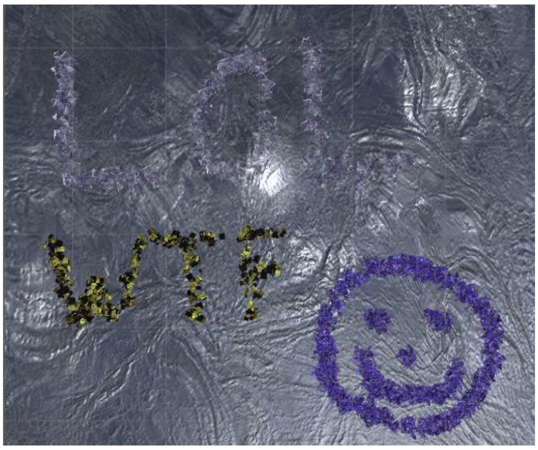


General Settings: Override rotation influences the orientation of the crystals. For lightning effects, set this to true and change the rotation setting of the crystal generator to very small values. Larger values will make the lightning more zig/zag like. Sample scene “New Positioning” shows such lightning which is even generated in real time when you hit play.

When collide with world is set, a new dendrite will be generated from the center if the old on did collide with world geometry. Collision Layer of the generator is used. Make sure it does not collide with the layer of the generator itself or you end up with a sphere ☺

If a dendrite collides, the unity event “On Hit Surface” is called.

Painting Crystal Generator:



Update 2.0 contains an amazing new crystal generator which allows 3D drawings in real time. To do this, select the game object which has the Crystal Generator Paint script attached to it and hold CTRL. When you do this, you see this blue crystal shape and a grey circle. While this Handle is visible, you can paint crystals by holding the left mouse button.

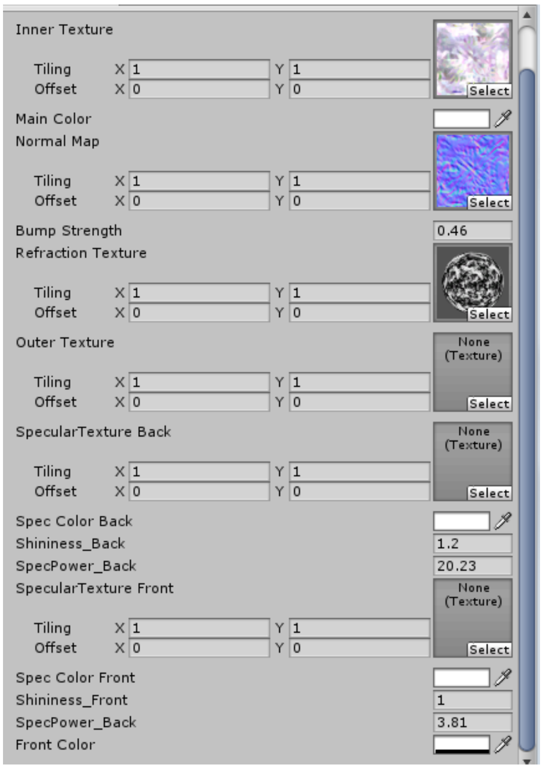
There is also a mode for 2D Painting, to use this mode, check the 2D Paint mode flag and go into the 2D view. You can paint crystals on the screen and their Z value is the Z position of the game object. These painted structures also have collisions and the collision mask can be adjusted in the inspector. If it allows collision with the same layer the painted crystal has, new crystals will be drawn on top of the previous drawn crystals. Now if you paint on one spot, the result will be a pillar with grows towards the camera. This can be very fast and can be very frightening as it looks like a worm moving very fast towards you

Parameters:

➢ Collision:	Set the layer mask
➢ Spread:	Draw radius
➢ ZSpread:	Variation in the vertical direction (y in 3D and z in 2D)
➢ PaintSpeed:	Amount of crystals generated per click/drag
➢ 2D Painting:	Activate 2D painting

Crystal Shader:

This asset also includes three shader which attempts to simulate crystals. As transparency especially those of crystals is very difficult, it is not possible to simulate physical correct crystals in real time yet. A main part of their beauty comes from their inclusions and errors in the crystal structure, created during the growing process. All three shader have several render passes. In the first pass, the interior of the crystal is rendered with its own lighting settings. Afterwards the outside is rendered using standard phong lighting.



Inner Texture
Tiling X 1 Y 1
Offset X 0 Y 0
Main Color
Normal Map
Tiling X 1 Y 1
Offset X 0 Y 0
Bump Strength 0.46
Refraction Texture
Tiling X 1 Y 1
Offset X 0 Y 0
Outer Texture None (Texture)
Tiling X 1 Y 1
Offset X 0 Y 0
SpecularTexture Back None (Texture)
Tiling X 1 Y 1
Offset X 0 Y 0
Spec Color Back
Shininess_Back 1.2
SpecPower_Back 20.23
SpecularTexture Front None (Texture)
Tiling X 1 Y 1
Offset X 0 Y 0
Spec Color Front
Shininess_Front 1
SpecPower_Back 3.81
Front Color

All 3 Types have similar settings which are:

- **Inner Texture:** Main texture of the interior
- **Main Color:** Color multiplier of the main texture
- **Normal Map:** Simple normal map for roughness
- **Bump Strength:** Normal map power
- **Refraction Texture:** Cubemap for the refractions
- **Outer Texture:** Texture of the surface.
- **Specular Texture back:** Specular map of the Interior
- **Spec Color back:** Color multiplier of the spec map.
- **Shininess back:** Shininess of the interior.
- **Spec power back:** Additional multiplier.
- **Specular Texture Front:** Specular map of the surface
- **Spec Color Front**: Color multiplier of the surface
- **Shininess Front:** Shininess of the surface
- **Specpower front:** Additional multiplier
- **Front Color:** Multiplier of the outer texture.

Note: This asset contains 6 preset materials. Keep in mind that with such amount of parameters, getting a good looking crystal material is difficult and has to be tweaked.

However some crystals are complete opaque in real life like pyrite. In this case, the settings for the interior can be ignored. It took a lot of time to develop this shader until I was satisfacted with results. Although it is still far away from being perfect. Future updates may include new shaders as half transparency is still a hazzle in real time computer graphics.

Shader1: Raw_Crystal_Surf

Shader implemented as Surface Shader using the StandardSpecular version, has Transparency and **needs an reflection Probe to work!**

Shader2: Raw_Crystal_Surf_Cutout

It is similar as Shader 1 but uses Cutout instead of transparency.

Shader 3: Raw_Crystal(UPDATE 3.0)

This shader got a whole overhaul and the lighting now works without edge artifacts. Also has more options. The settings are now split up into 3 sections which are Main Properties, Inner Side and Front Side.

Main Properties: Contains standard settings like Main Texture and Bumpmap which are common in other shaders like the standard one. The Refraction Texture is used for the inner and outer reflections.

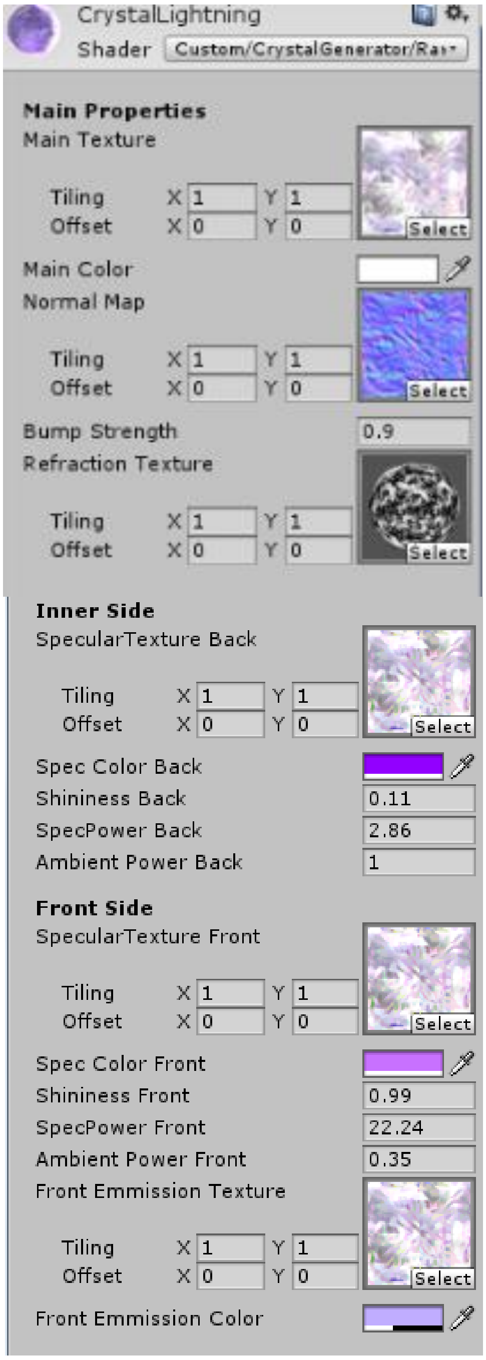
Inner Side: Describes properties which changes tha appearance of the Inner Side of the crystal. It contains specular settings which includes Texture, Color and Shininess. Spec Power multiplies the specular result and can even be negative.

Ambient Power is multiplies influence of ambient light and directional light. Can also be negative

Front Side: These are settings for the front side of the crystal. The parameters are similar to those from the inner side.

Additional parameters are the Front Emission Texture and Front Emission Color. These values give the crystal a slight glow. The alpha value of the emission color sets the final alpha value of the front side.

This means if alpha is zero, only the inner side will be visible.



CrystalLightning

Shader Custom/CrystalGenerator/Ra

Main Properties

Main Texture

Tiling X 1 Y 1

Offset X 0 Y 0

Main Color

Normal Map

Tiling X 1 Y 1

Offset X 0 Y 0

Bump Strength 0.9

Refraction Texture

Tiling X 1 Y 1

Offset X 0 Y 0

Inner Side

SpecularTexture Back

Tiling X 1 Y 1

Offset X 0 Y 0

Spec Color Back

Shininess Back 0.11

SpecPower Back 2.86

Ambient Power Back 1

Front Side

SpecularTexture Front

Tiling X 1 Y 1

Offset X 0 Y 0

Spec Color Front

Shininess Front 0.99

SpecPower Front 22.24

Ambient Power Front 0.35

Front Emmission Texture

Tiling X 1 Y 1

Offset X 0 Y 0

Front Emmission Color

Real Time growing (Update 3.0)

Currently there is one script included which is the Crystal Controller. It is currently very experimental and may be changed in the future. The "Dendrite Lightning" game object in the "New Positioning" scene has this script attached. It simply grows crystals during play mode.

It has Growspeed Min/Max as setting which describes the amount of grow cycles each fixed update frame. It is limited between 0 and 20.

If "Use Fuel" option is set, crystals will only grow if the fuel value is greater than 0. Fuel is consumed for each grow cycle.

Wait Frame for Mesh Update: Updating the Large Mesh Combiner every frame is slow so it is recommended to just do it ever 6th frame to increase the performance.

Extending:

The first major Update has revamped everything to make it easier to extend the crystal generator. For extending, simple derive your custom generator from the crystal generator class.

The most important functions are:

- `CrystalGrowth()`   Function called for each growth cycle
- `CreateCrystal()`   Function called when a new crystal should be created
- `UpdateFinalMesh()`   Function to update final Mesh.

The positioning also can be extended

The base class "CrystalPosition" contains 2 virtual functions which are:

- `Preview(CrystalGenerator Generator)`: Preview visible when DrawSeeds is true in the generator
- `CalculatePosition(CrystalGenerator Generator): Calculates the position`

It is not forbidden to let some fancy algorithm do the painting when using the painting crystal generator. To do this, set the PaintPosition and call CrystalGrowth afterwards. Call UpdateFinalMesh to update the mesh.

Exporting Meshes:

It is possible to export every mesh now as the LargeMeshCombiner allows exportation of meshes. However large meshes consist of several meshes so you have to export every piece to get the full object.

Further information is in the Large Mesh Combiner documentation!

Possible Errors:

No Crystal is generated:

- New crystal chance is 0.
- Bounds is 0 or negative.
- Bounds is so small that it doesn’t hit any collider in the world.
- Maximum Amount reached
- All parameters of New Crystal Scale Min/Max are zero. It is possible to create two dimensional crystals by setting one coordinate to 0. It would work but it wont look fancy.
- New Crystal Size is 0. If you want small crystals set it to 0.01 at least.

Bad Looking mesh:

- Mesh looks horrible: YES this can happen if your system has so many crystals that the vertex count of the final mesh exceeds the maximum vertex count suported by unity. If you need extreme amounts of crystals, use the new Extreme Crystal Generator

If the start scales are 0 the error `"Creating a Crystal with Scale 0 is not supported." is generated. To solve this, simply don’t set New Crystal Size and CrystalScale Min/Max to zero.`

Dangers when letting them grow in the Update method

Be careful when using this asset for crystallization effects. Set the maximum iteration count lower and reduce the maximum amount of crystals possible. For example set Max iterations to 180 and call the Grow Crystal function in the fixed update to get a 3 second crystallization effect. The growth of many crystals may slow down your frame rate.

Last Notes:

If you have any questions, suggestions, bug reporting don’t hesitate to contact me. If you are going to sell a game which uses this asset, inform me because I may buy your game and play it ☺

Contact Information:

E-Mail: mhartl.mmt-b2013@fh-salzburg.ac.at

Homepage: http://lostinpixels.org/