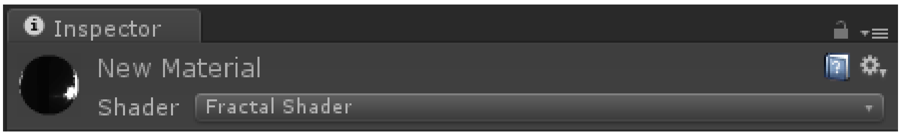
Fractal Shader

Fractal Shader consists of a shader and associated editor functionality for creating real-time fractal effects and associating them with a material so they can be rendered directly onto models, as 2D sprites or as scene backgrounds. The fractal shader supports the popular Mandelbrot and Julia fractals, with all associated variables fully customisable via the shader's properties, which are presented in a user-friendly way in the inspector via the custom material editor. The material editor also displays a preview image in the inspector that can be dragged and zoomed with the mouse.

An example scene is included, which shows off some of the functionality of the shader (what the various settings can achieve) and some example "pretty pictures" that can be generated. The following document will guide you through the process of creating a fractal, and introduce you to all of the available functionality. After that, the only limit is your own imagination!

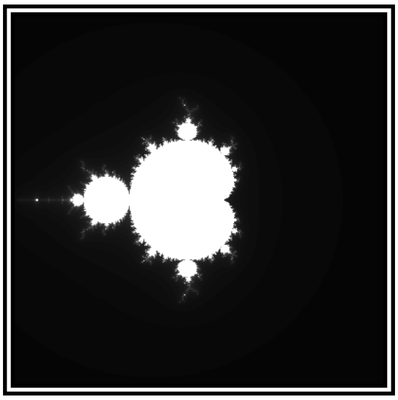
Creating a Material

In order to draw a fractal, you must draw it onto something. It doesn't matter what – you can create a fractal on a 3D model or 2D sprite – just bear in mind that the fractal itself is drawn within the UV co-ordinates 0..1 in both directions. For the following examples, let's use a quad, which helpfully has UV co-ordinates from 0 to 1 in both directions by default. So create a quad and add it to the scene. You'll also need a material, so create one of those and set its shader to "Fractal Shader":



Shader Fractal Shader

When you associate the material with the quad, the default fractal will be drawn onto it. The default fractal is a Mandelbrot fractal with 100 iterations and basic monochrome colouring, which looks like this:



Setting Fractal Parameters

This is far from the most interesting fractal that can be created, but is a good starting point to begin experimenting with the fractal's parameters. The following is a list of these parameters and how they affect the fractal image:

- **Fractal Type (Mandelbrot or Julia)** – Specifies which formula is used to calculate the fractal. The purpose of this document is not to explain how fractals are calculated – there are many resources on the web for this (or you could look at the shader source code, which is included in the package). For now it is sufficient to say that these options generate two different types of fractals.
- **Iterations** – The maximum number of iterations for each pixel of the fractal. More iterations mean more detail in the fractal image, but also more processing time. So this is a trade-off between what looks good, and what doesn't slow things down too much.
- **Convergence Threshold** – Technically this specifies the value at which each pixel in the fractal algorithm is deemed to have "converged" and therefore has reached its iteration limit. This is another detail/speed trade-off, but most of the time you will be better off leaving it fairly low (about 2-4). Convergence makes little to no difference for some colouring effects, so in this case set it as low as possible.
- **Smoothing (on or off)** – Whether to apply a smoothing algorithm to the fractal. When this flag is unset, "bands" of colour are sometimes visible and when it is set, colouring appears as more of a smooth gradient. Smoothing adds a little bit of processing power, so turn it off if it doesn't make a difference for your chosen fractal, which is the case with some colouring algorithms.
- **Multibrot (2 to 5)** – Changes the algorithm to produce a fractal with more "sides". This is best illustrated on the default Mandelbrot fractal, so just experiment to see what looks best.
- **Centre** – The X and Y co-ordinates of the centre point of the fractal. This can be changed manually or by dragging the fractal preview image (see below).
- **Scale** – The amount the fractal is "zoomed in". This can be changed manually or by scrolling the mouse wheel on the fractal preview image (see below).
- **Constant (Julia fractals only)** – The constant to apply to each pixel operation for Julia fractals. This defaults to *-0.4+i0.6* which is a popular Julia fractal. Minor changes to these numbers can cause a large change to the fractal itself.
- **Background Colour** – An optional colour that can be merged with the fractal, for example to fade it to black.
- **Background Amount** – The amount that the chosen background colour is merged with the fractal. A value of 0 (default) ignores the background colour completely, and a value of 1 displays the background as a solid colour.
- **Red/Green/Blue** – These settings specify how the fractal should be coloured, which is achieved by calculating the red, green and blue components of each pixel separately. When a pixel converges, it does so after a number of iterations and at a certain point in the complex plane. This means there are a number of variables that can be used to determine what the final pixel colour should be. The pre-set formulas use these variables, optionally applying trigonometric functions to them to produce some stunning effects. Experimentation is the best policy!

The Preview Image

The inspector also shows a fractal preview image, which shows the current fractal based on the selected parameters. The preview can be small, medium or large (smaller sizes allow the preview image to be displayed at the same time as the parameters, depending on screen resolution), and responds to mouse input as follows:

- The left mouse button can be used to drag the position of the fractal. This will change its **Centre** property.
- The mouse wheel can be used to zoom in or out of the fractal. This will change its **Scale** property.