

---

# INSTALLING AND USING THE ROCK GENERATOR ADD-ON FOR BLENDER 2.5

---

Version 1.2 - July 11, 2011

Paul “brikbot” Marshall

## TABLE OF CONTENTS

---

<b>INTRODUCTION .....</b>	<b>3</b>
RANDOM NUMBER GENERATION .....	3
<b>INSTALLATION .....</b>	<b>6</b>
REQUIREMENTS .....	6
INSTALLING NUMPY.....	6
INSTALLING ROCK GENERATOR.....	7
<b>USING THE ROCK GENERATOR .....</b>	<b>8</b>
SIZE CONTROLS.....	8
MODIFIER CONTROLS.....	9
MATERIAL CONTROLS .....	9
RANDOM SEED CONTROLS .....	10
PRESETS .....	10
ADVICE .....	11
<b>SUPPORT &amp; BUG REPORTING.....</b>	<b>12</b>
ROCK GENERATOR.....	12
NUMPY.....	12
<b>SAMPLE RENDERES .....</b>	<b>14</b>
DEMO RENDER: WATERSIDE .....	14

---

## INTRODUCTION

---

The Rock Generator script for Blender 2.5 was launched on seeing a need for a way to quickly produce finished rock objects inside of Blender. This requires that the add-on use a few input parameters to generate a base mesh, apply modifiers for detail, create and modify textures for displacement mapping, and be able to create and modify materials and procedural textures for the materials. In addition, to better reflect nature the generator required an element of randomness be built in to each aspect of the generator.

This add-on was spurred by a personal project, which at the beginning of which I also considered alternate, pre-existing methods for rock generation. The primary alternative is the ANT Landscape generator, set to generate a sphere. I will note that if you only need one or two rocks, the ANT Landscape generator is simpler to use and is a part of the official Blender distribution. However, it has some weaknesses:

1. The ANT Landscape generator does not provide the detail level that this add-on can provide.
2. The ANT Landscape generator does not allow the flexibility that this add-on allows.
3. The ANT Landscape generator does not generate materials for the resulting rock(s).
4. The ANT Landscape generator does not allow for the generation of multiple rocks.
5. The ANT Landscape generator is limited to a sphere based shape. This add-on uses multiple shapes as the basis for the resulting rock(s).

The base methodology of the add-on is based on Andrew “BlenderGuru” Price’s asteroid tutorial found at <http://www.blenderguru.com/how-to-make-a-realistic-asteroid/>. Additionally, aspects of Sascha Henrichs’ tutorial for creating a rock with procedural textures in 3ds Max were integrated. That tutorial can be viewed at <http://saschahenrichs.blogspot.com/2010/03/3dsmax-environment-modeling-1.html>.

This document will walk you through installing and using the add-on. Along the way it will provide insights on some of the design decisions that were made in its creation and some of the technical features that have been implemented. Finally, this document will remain in a fluid state to reflect the most recent publicly released version of the script.

---

## RANDOM NUMBER GENERATION

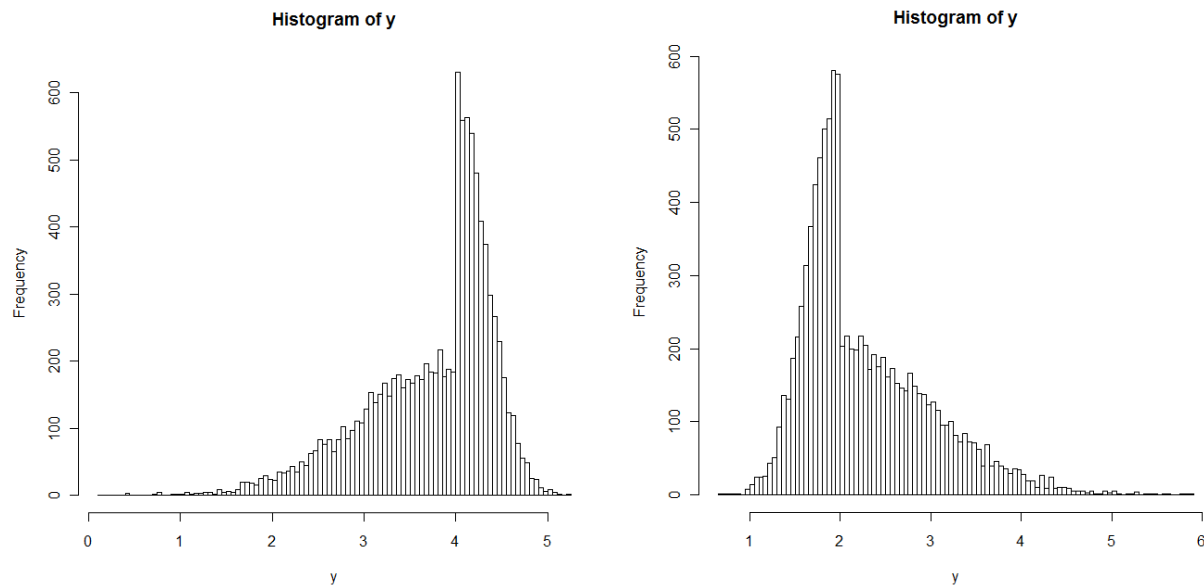
---

Because of the add-on’s heavy reliance on random number generation and the requirement to quickly generate a large number of rocks, much consideration was put into what generator to use. The requirements were:

- The generator was to be distribution based, with the most random values being generated near a mean value with fewer being generated further from the given mean.
- The generator must use a mean and variance as the basis for input.
- The generator had to be able to work with a soft upper and lower boundary (95% values within the boundaries).
- The generator had to be able to output a skewed distribution.

- Be efficient with unique sequential calls to the generator.

There are two main distributions that support these requirements: a Gaussian (normal) distribution or a Beta distribution. Python 3.2 has both built into it, or the package NumPy also provides both. Therefore, before starting the project the four possible distributions were profiled as they would be used in the add-on. To summarize the results, NumPy's beta distribution based random number generator was more than 10 times faster than Python's, and NumPy's Gaussian distribution based random number generator was more than twice as fast as Python's. Between the two NumPy based generators, the beta was slower than the Gaussian when called once before the parameters changed, but over twice as fast when called several times without having to re-calculate alpha and beta values. Because the implementation would better reflect the first scenario, NumPy's Gaussian provides the best performance. There are two significant drawbacks to using NumPy's Gaussian based random number generator. First, this requires the installation of an external package not included with Blender. While it is simple to install, some users may find it daunting. Secondly, the skewing function does not result in a continuous distribution curve. Rather, it creates a piecewise function divided about the mean with a large jump when crossing to the skewed side. Below are two graphs demonstrating the resulting distributions. Both used an upper limit of 5 and a lower limit of 1. Both had a standard deviation of 1 ( $\sigma^2 = 1$ , 95% CI = 3). For the right skewed distribution the mean  $\mu = 4$  and for the left skewed  $\mu = 2$ . Both used 10,000 samples:



Below is the R source code used after the appropriate values were set to generate the above histograms, which reflects the Python code used in the addon:

```

x = rnorm(n, mu, sigma)
y = rnorm(n, mu, sigma)
i = 1
mid = (lower + upper) / 2
for (j in x) {
  if (x[i] < mu && mu < mid) {
    y[i] = ((mu - lower) / (3 * sigma)) * x[i] + ((mu * (lower
- (mu - 3 * sigma))) / (3 * sigma))
  } else if (x[i] > mu && mu > mid) {
    y[i] = ((mu - upper) / (3 * -sigma)) * x[i] + ((mu * (upper
- (mu + 3 * sigma))) / (3 * -sigma))
  } else {
    y[i] = x[i]
  }
  i = i + 1
}
hist(y, breaks=n/100)

```

---

## INSTALLATION

---

Installing Rock Generator consists of two main steps: installation of NumPy to Blender 2.5's bundled Python, and the installation of the add-on itself. If you are curious why you need to install NumPy, first read the proceeding section titled "Random Number Generation". The reason I am using NumPy is because it provides a significant performance increase over Python's random number generators. This is significant because each rock has to generate about 50 random numbers and there could be anywhere from a single rock to over a thousand rocks being generated (50 – 50000+ random number calls). On the low end the difference is insignificant, but on the high end it could mean a time savings of minutes.

---

## REQUIREMENTS

---

The rock generator addon has the following requirements:

- Blender 2.58 API r38025 or newer
- Python 3.2. This is bundled with Blender.
- NumPy for Python 3.2 is recommended.

There are no specific hardware requirements, but note that the script will benefit significantly from a faster processor. At high rock counts it will also benefit from a large amount of system RAM.

---

## INSTALLING NUMPY

---

Before installation, it is necessary to download NumPy. I have personally performed a NumPy build for Windows and verified that it works with Blender for both 32-bit based systems and 64-bit based systems. The version of NumPy that you need is dependent not on your system, but your version of Blender. If you are using a 32-bit version of Blender, then you must use a 32-bit build of NumPy. If you are using a 64-bit version of Blender, then you must use a 64-bit build of NumPy. My builds can be downloaded at the below links:

- For 32-bit versions: <http://www.mediafire.com/file/6c6y56fj16q98sd/numpy-x86.zip>
- For 64-bit versions: <http://www.mediafire.com/file/y6x7xdd3d37ool9/numpy-x64.zip>

Additionally, it is possible to download a build directly from SciPy.org. These builds may be more recent but have not been verified to work with Blender. To do so, download the appropriate version from the SciPy.org site: <http://www.scipy.org/Download>.

As I work from Windows based installations of Blender, this guide will be written from that perspective. If you have installation issues, I request that you PM me on BlenderArtists at <http://blenderartists.org/forum/member.php?26655-BrikBot>. I will do my very best to assist. Also, I am only writing this with the assumption that you have downloaded the appropriate build that I have provided.

Installing NumPy first requires unzipping the appropriate version of NumPy to the Blender installation directory. By default, this is "[C:\Program Files\Blender Foundation\Blender\](#)". If you have installed a 32-bit version of Blender on a 64-bit version of Windows, then your default path will instead be "[C:\Program Files \(x86\)\Blender Foundation\Blender\](#)". Finally, if you have

installed Blender by unzipping it as a portable installation, then you will need to locate the appropriate directory yourself. Once you have located the appropriate directory, navigate to “[\2.58\scripts\modules\](#)”. Inside this folder create another folder called “numpy”. This folder is the target folder you will use when unzipping NumPy.

To verify that NumPy has been installed correctly, open Blender. Then from the bar at the top, change the window view from “Default” to “Scripting”. In one of the two Python consoles, type “import numpy” and press enter. If it runs the command without issue then NumPy has been installed correctly. If it says that “numpy is not found” then there has been an issue with the installation. Double check the installation to make sure it was done properly (such as was there a typo in naming the folder “numpy”?). If problems continue, please see “Support and Bug Reporting” on page 12.

Currently it appears the Numpy is not compatible with Blender’s Python implementation on Mac OS X. The script does not require Numpy, so the script will still run. Unfortunately, the script will not benefit from the performance optimizations that Numpy has.

---

## INSTALLING ROCK GENERATOR

---

Now that you have located and have been working in Blender’s installation directory, the first part of installing the rock generator will be fairly simple. Starting from the install directory, this time navigate to “[\2.58\scripts\addons\](#)”. Simply extract the “add\_mesh\_rocks” folder to this location. Make sure the folder is placed in this location, not the contents of the folder.

Now that the script can be located by Blender, it needs to be enabled. Open Blender. If you still have Blender open from testing the NumPy installation, you will need to restart Blender. Once Blender is open again, open User Preferences. You can either use a Blender panel, open it with “Ctrl + Alt + U”, or by “File > User Preferences”. Go to the add-ons section. To make it easier to find, use the “Add Mesh” filter on the left. Once you have found the add-on, enable it. If you want the add-on enabled by default, select “Save as Default” at the bottom. Otherwise just close the window.

At this point, the add-on should be fully installed and ready for use. If you would like to check the installation, pull up the Add Mesh menu, go to Mesh, and at the bottom should be an item called “Rocks”. Select this. After a slight delay you will get a rock in the center of the Blender world and a panel will be activated on the tools at the right side of the window.

---

## USING THE ROCK GENERATOR

---

The add-on panel has been divided into five main sections. The first only contains the control for the number of rocks to generate. The second modifies the XYZ size controls. Third are the shape and modifier controls. The fourth box contains the material and texture controls. Finally, the fifth box is for controlling the random seed.

---

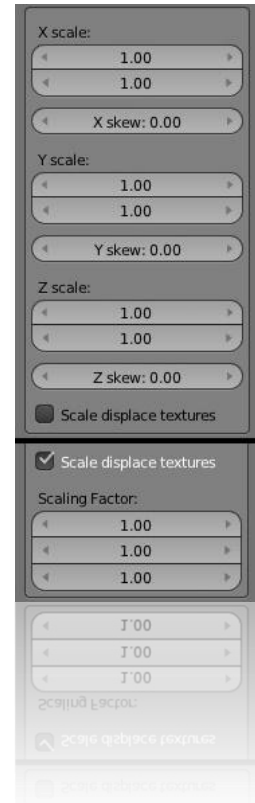
### SIZE CONTROLS

---

The size controls consist of a trio of controls for the X, Y, and Z dimensioning. The first two controls control the boundaries of the generated rocks with the top controlling the lower bound and the lower controlling the upper bound. These bounds are not hard bounds: it is possible to have rocks generated that fall outside these bounds. They simply define where most rocks will fall. If the two are the same, no randomness will be used on that axis when generating the rocks.

The mean value for the rocks is controlled the third control labeled skew. Adjusting this value up and down moves the mean proportionally between the upper and lower bounds of the distribution. The possible range of skew values is -1 to 1. When the skew is set to -1, the mean value of the distribution will be the lower bound and when the skew is set to 1, the mean falls instead on the upper bound. Setting the skew to zero results in a normal distribution. Internally, the skew is acting like a percentage. However, mathematically you can't have -100% to 100%, so the add-on takes the raw skew value, divides it by two, then shifts it up by 50%. This means that internally the add-on sees -1 as 0% and 1 as 100%, with zero being seen as 50%.

The final control is a check box to scale the displacement textures. This is an option because it changes the way the mesh is built which changes how the displacement textures look. If left unchecked, the displacement textures will not be scaled. This will not be a problem the size of the rocks are not on an extreme (between 1 and 2). For really small rocks the displacement texture may be much too large or with really large rock the displacement texture may be much too small. In such cases it is advisable to use scaled displacement textures. However, the textures are scaled independently in the X, Y, and Z directions so the resulting textures may not be desirable if one of the dimensions is very different from the others. In such cases it is advisable to use scaled displacement textures. However, the textures are scaled independently in the X, Y, and Z directions so the resulting textures may not be desirable if one of the dimensions is very different from the others. To help with this, as of version 1.2, it is possible to manually set the scaling factor for the X, Y, and Z directions individually. Additionally, it is possible to use the scaling to distort the textures to create other effects such as the wind-blown sandstone look (see Sandstone preset for example).





## MODIFIER CONTROLS

These controls control the various modifiers that are applied to the rocks. The modifier stack that is generated consists of two subsurf modifiers, four displacement modifiers, and an optional smooth modifier. There are six sliders for controlling the modifiers, with two controls for each type of modifier. The top two modifiers control the deform modifiers, the middle two control the subsurf modifiers, and the bottom two control the smooth modifier.



The deformation slider controls the first two displacement modifiers. These two modifiers focus on the base shape of the rock. This slider is in actuality the mean value for a Gaussian distribution, with the variance of the distribution linearly related to the mean such that the higher the deformation, the greater the variance. The second of the two top sliders is the roughness slider. Here, roughness controls the last two displacement modifiers for the smaller details of the rock. As with the deformation slider, the roughness control is also acting as the mean value of a Gaussian distribution with the variance generated from the same roughness value.

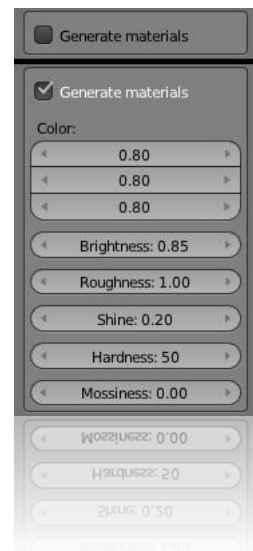
The two subsurf modifiers are identical and are both controlled from the middle two controls. The upper of the two controls is the render detail level. This links to the render slider found under subdivisions in the modifiers. Below the detail slider is the display detail control which controls the display detail shown in the 3D viewport. This directly links to the view slider under subdivisions in the modifiers. These two settings do not have any randomness involved.

The final two sliders control the optional smooth modifier. When the sliders are both set to zero, the smooth modifier is disabled and not added to the modifier stack. If both sliders are greater than zero, a smooth modifier is added. Likewise if both are set back to zero, the smooth modifier will be removed. The smooth factor control provides a mean value for a Gaussian random number generator and the variance is linearly related factor value such that a greater factor value results in higher variance. The generated random number sets the factor value of the smooth modifier. The smooth iterations control is directly linked to the smooth modifier's iterations slider.

## MATERIAL CONTROLS

The rock generator add-on provides the option to generate materials and textures for the generated rocks. By default, this is not enabled (top box). When material generation is enabled, the user is presented with the ability to adjust the base RGB values, the brightness of the materials, the roughness of the materials, how shiny the materials appear, how hard the materials appear, and the apparent mossiness of the rock (bottom box).

The color control is composed of three sliders representing RGB values: the first slider controls red, the second controls green, and the last controls blue. Each component is linked independently to a Gaussian distribution with the input values serving as the mean. As the RGB values approach zero, the variance also approaches zero because smaller differences between the different shades become more noticeable at very small values.



Brightness and roughness both control the diffuse shader for the materials. Brightness serves as a mean value for a Gaussian distribution, and provides a basis for the variance of the distribution. Because of this linking the variance approaches zero as the brightness nears zero. Brightness does not affect the shade of the color, only the brightness of the shader. Roughness acts the same way, except it is first lineally scaled first. The control provided has a range of zero to five, but the actual resulting range is scaled from zero to  $\pi$ . Roughness also controls the normal values for the textures. Here, roughness is not scaled, though it still provides a mean and variance for a Gaussian distribution.

Shine and hardness both correlate to the specular shader, with shine providing the mean and variance for a Gaussian distribution for the specular intensity. The hardness likewise provides the mean and variance for a Gaussian distribution, but this instead maps to the hardness values of the shader.

The add-on has a basic, texture based implementation for moss. The two moss textures are always generated, and the color influence can be controlled with the mossiness slider. A value of zero disables the textures completely and the max value of one will cause the two textures to have a strong color influence.

---

## RANDOM SEED CONTROLS

---

Because the script extensively uses random number generation, it must be seeded. The first option, which is also the default, is to use a “random” seed (top box). In actuality, the seed is not really random but is based on time. Because of this, it will always result in a unique seed. The alternate is to specify a seed for the generator (bottom box). This will result in a predictable sequence of outputs.

Also seen in the image is the preset drop-down list.



---

## PRESETS

---

The rock generator add-on has support for preset values. To load a preset, simply select it from the drop-down list. The specified preset values will then be loaded.

The presets are stored in an XML file called “add\_mesh\_rocks.xml” which is located in the same directory as the add-on. You may add or edit preset values in the XML file, but you must follow the format exactly or the add-on will crash. If you have irrevocably modified “add\_mesh\_rocks.xml”, simply delete the file and on the next run the add-on will attempt to create a new XML file with the original settings. Two things to note: all your customizations to the original will be lost, and if you are working on Windows and Blender is located in Program Files then you will need to run Blender as an administrator the following run to give Blender permission to create the file. Do not attempt to modify any other file: if you do so it is at your own risk.

The preset values specified are as follows:

- Default: this is the clean slate that the add-on initializes to.
- Asteroid: creates larger asteroid-looking rocks

- River Rock: generates rocks similar to the classic river pebble often used in landscaping.
- Sandstone: this setting generates rocks that have a very smooth, horizontally worn appearance.

---

## ADVICE

---

Some dos/do not's on using the rock generator:

1. Do remember that this generator randomizes almost everything to some extent. This is great of lots of variety but is unpredictable.
2. Do use a user seed while adjusting the settings. This will make it easier to see how changing a value affects the resulting rocks.
3. Do use a random seed for your final product. The purpose after all is a truly random system of rocks.
4. Do use a few rocks (5 or so) while adjusting the settings to get a better feel for how the majority of rocks will actually look like. If you just use one, you might be making adjustments based on an outlier.
5. Do increase the number of rocks last because the length of time it takes for the script to run is directly proportional to the number of rocks being generated. It takes a bit longer to generate a hundred rocks vs. ten rocks.
6. Do use the render detail level for the display detail while adjusting the settings, then feel free to lower the display detail before generating the final number of rocks.
7. Do generate more rocks than you need. You may not like some of the outliers that might be produced, and it is trivial to delete extra rocks.
8. Do not use extreme skew values, unless you know this reflects the system you are modeling. A high skew value will not reflect how nature typically acts (which is to default to a normal distribution).

---

## SUPPORT & BUG REPORTING

---

I will do my best to provide support for the add-on. However, I can only guarantee full support for Windows. I may be able to help with alternate operating systems if I can get a virtual machine running, but I may be limited by my lack of knowledge and familiarity with the system.

Before submitting a problem, please read the entire BlenderArtists thread for the rock generator add-on at <http://blenderartists.org/forum/showthread.php?216271-Rock-generator-script>. At this point it is very probable that you may find an answer to your problem by doing so.

---

### ROCK GENERATOR

---

This script is fully cross-platform so issues regarding the add-on itself I will be able to provide full support for. If further help or explanation on installing or using the rock generator add-on, there are several methods I can be contacted at for support. If you encounter what you believe to be a bug or think of an improvement or tweak to the rock generator add-on, please report them using these same methods:

1. The add-on's thread in the BlenderArtists forum. I would prefer that requests for support be made here so other users will have access the information also. The thread can be found at: <http://blenderartists.org/forum/showthread.php?216271-Rock-generator-script>
2. By leaving a comment on the add-on's tracker page. This method is also preferred because it also provides the solution in a public setting for others to access. The tracker page is currently located at: <http://projects.blender.org/tracker/index.php?func=detail&aid=27314>
3. By a personal message on the BlenderArtists forum. This can be done through my profile page on the forums at: <http://blenderartists.org/forum/member.php?26655-BrikBot>
4. Finally, I can be contacted via e-mail. I do not advise this method as I do not check this e-mail account regularly. The e-mail address is [brikbob@gmail.com](mailto:brikbob@gmail.com)

---

### NUMPY

---

NumPy is cross-platform, with some limitations. NumPy relies on some compiled .pyd and C/H files that are not cross-platform compatible. As such the add-on does not rely on the NumPy package but will use it if possible. If I do not have a version of NumPy built that works with your operating system and you are not comfortable building your own, I recommend notifying me so that I can look into creating a compatible build. Please use the same methods as for the add-on:

1. The add-on's thread in the BlenderArtists forum. I would prefer that requests for support be made here so other users will have access the information also. The thread can be found at: <http://blenderartists.org/forum/showthread.php?216271-Rock-generator-script>
2. By leaving a comment on the add-on's tracker page. This method is also preferred because it also provides the solution in a public setting for others to access. The

tracker page is currently located at:

<http://projects.blender.org/tracker/index.php?func=detail&aid=27314>

3. By a personal message on the BlenderArtists forum. This can be done through my profile page on the forums at:  
<http://blenderartists.org/forum/member.php?26655-BrikBot>
4. Finally, I can be contacted via e-mail. I do not advise this method as I do not check this e-mail account regularly. The e-mail address is [brikbot@gmail.com](mailto:brikbot@gmail.com)

Additional information about NumPy can be found at the project's homepage <http://numpy.scipy.org/>, and the source code for NumPy is downloadable from the NumPy SourceForge page at <http://sourceforge.net/projects/numpy/files/NumPy/>.

## SAMPLE RENDERS

---

The following is a collection of renders featuring rocks generated by the script. The purpose of this section is to give an idea of what this add-on can and cannot do.

### DEMO RENDER: WATERSIDE

---



This is a sample rendering featuring two separate rock generations. The first was of 50 rocks set with a XYZ lower bound of 0 and a XYZ upper bound of 2. Mossiness was increased to 0.5. All other values were left as default. The second generation was of 30 rocks with the same settings as above with the exception of the XYZ upper bound was set to 1.