

# Throw Perlin Noise Out

*Sean Butler*

## I DON'T LIKE PERLIN NOISE.

There I said it. Gosh I feel better already! Though, this improved feeling does perhaps come at the risk of committing procedural blasphemy\*. Many procedural developers would say Perlin Noise is incredibly useful and, of course, they'd be right.

As a multilayered, scaled, random-ish waveform Perlin Noise can be put to use in many ways. Clouds, Waves, Mountains, Islands, Caves, Lava Lamps, Economies, etc. Perlin & other similar noise systems generate these kinds of structures easily because they can generate an output with a repeating, but never quite the same pattern. One of the valuable aspect of the algorithm's output is that the variations in the data can all be of a similar scale. This similarity of scale allows us to use the output as a source for rolling hills or islands or other features which should all be different, but are created by the same processes so should be of similar scale.

In their vanilla form, the landscapes generated from Perlin noise tend to have a characteristic shape. Luckily for us we can influence that shape. So, most developers put in additional manipulations to create the geographic structures we all enjoy. One particular way is to provide a variable increasing scale on the y-axis. This has the desirable effect of making the higher points of the landscape more pointy. Which more accurately matches the profiles of mountains and hills in the world around us.



Public Domain, <https://commons.wikimedia.org/w/index.php?curid=4152613>



Webmaster.vinarice [CC BY-SA 4.0], via Wikimedia Commons

Removing the primary physical aspect of a gameplay challenge in making it a simulation based challenge means the space often needs to be amplified for fun to occur, consider the tracks of SSX in comparison with your own experience of skiing. Perhaps authentic artefacts don't always make for great video game gameplay?

Inauthentic artefacts can remind the player they are in a simulation which might not be what you as a designer want. Normalisation, conventions and suspension of disbelief go some way to ameliorate this.

So, at one end we have arbitrary algorithms and heuristics which generate forms useful in a variety of domains. At the other end we can have simulations whose internal factors accurately and mathematically replicate the internal states and dynamics of the system found in reality whose aspects we find valuable for our game.

Recently a student I supervised built a system to generate levels for a real-time strategy game. Along with a variety of other techniques, they used a cellular automata to distribute the placement of trees within little forests across the map. While inspired by biology the mathematical features of the cellular automata have little or nothing in common with the systems of pressures on tree populations in the countryside.

Consider Chris Ofili's use of elephant dung, one could argue that the exact shades he needs for the supports and other round features in his paintings could more easily and cheaply be produced with other materials. Conversely reflect on Damien Hirst's spot paintings, some of which are shipped to their buyer as

a series of pots of paint and instructions. Hurst himself states the best spot painting to be had was painted by an assistant. So from the perspective of the artist, and art market the necessity for authenticity is a movable requirement.

Academic or scientific approaches to procedural generation of sand dunes often has the goal of generating natural shaped landscapes.

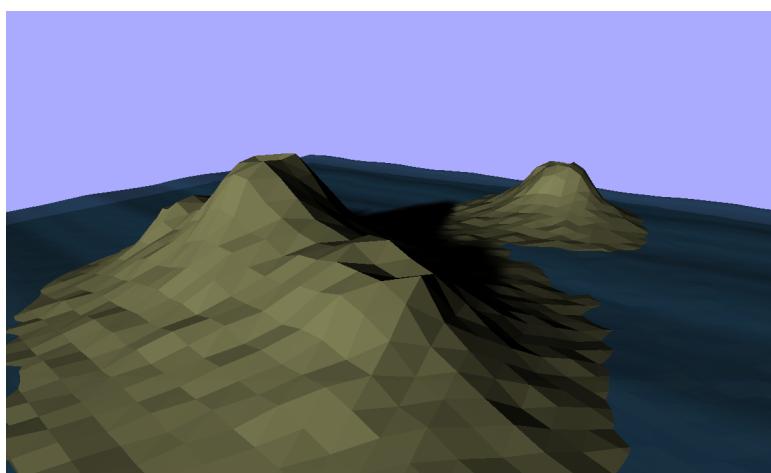
Meteor Storm Escape acted as a MVP/prototype for a desert dune racing game. At every stage we compromised authenticity for a particular challenging and exhilarating player experience.



## Tutorial

As we've already seen that due to plate tectonics the thin parts of the earths crust where volcanos can form is often shaped as a long wiggly line. Simulations of plate tectonics exist, I've yet to hear of a game which uses it to determine gameplay. If anyone knows a bit about it, get in-touch, perhaps we can work together? Its simple enough with a move, turn, loop to manipulate a sequence of vectors to steer a random walk in a particular direction. Parameterising the length of the vectors and the range and shape of random numbers for orientation, we can create a more or less linear (chaotic) path to represent the place where two parts of the world's crust are moving together or apart.

## Results





*ISS Crew Earth Observations experiment and the Image Science & Analysis Group, Johnson Space Center. Public domain via Wikimedia Commons*

In each case the islands generated show convex and concave features around their edges, this is in keeping with authentic natural landscapes, and depending on scale and designer goals could be instrumental in anchoring detracting from gameplay goals.

The algorithm demonstrated is primarily for the generation of shape, so as such the islands aren't rendered with specialist shaders including water's edge, greenery, rock and snow.

## Discussion

The aesthetics of space and visuals in games is different from that in the traditional visual arts. As game developers we must balance the needs of the visual form against the needs of the gameplay form.

In big studio production this is often done by having white/orange boxing and by having gameplay collision mesh built in parallel with the gameplay design and tested against that gameplay as a separate process from the (later) art generation.

In procedural generation the management of this dual nature has to be handled differently. Lots of current procedural generation seems to closely tie or consider identical the visuals and gameplay space.

## Conclusion

Authentic and interesting natural structures need so much addition work on top of the source Perlin Noise that we should consider alternatives.

Researchers have developed many high end algorithms for producing authentic natural landscapes. These can be complex to implement and computationally intensive

As we have seen in the tutorial above, passable approximations are achievable which are lightweight customisable, flexible and easy to understand.

- 
- A better kind of procedural blasphemy might be an ‘electric sinner’. Use logic and NLP to codify the core rule set from a religious text. Build a machine to break those rules (in thought, by visualising them, then perhaps automatically tweeting the images, by communication by inciting others to commit sin and finally in action, would need a robot for this one). Is it possible to carry out any of these activities without actually committing a sin oneself? I suppose we could consider it educational or somewhat like a morality play. (some might say that video games are already doing this)
-