



# PROCEDURAL ISLAND GENERATOR

User Guide

How to create an immersive low-poly terrain with least time and effort

Pinwheel Studio  
<http://pinwheel.studio>

## OVERVIEW

Nowadays, procedural content is one of the hottest subjects in game development field. Any game which is promoted with this term usually get higher attention from players, and its content also varies across levels. The term “procedural” is applied to several field of game assets creation, including map, terrain, animal species, or other environmental elements like rock, cloud, etc. The main purpose of procedural content is to create a variety of assets with minimum effort and time, by a process called “parameterize”.

As an indie/ soloed developer, you often spend a lot of time to create a gorgeous looked terrain, from modelling, to UV mapping, and texturing, then spend hours to place every single tree and rock in place. Stop! You don’t have to do that anymore.

Procedural Island Generator is a tool to help you to quickly create high quality, immersive looked terrains in just seconds. The land is fully created using parameters, no modelling skill required. The tool is also packed with a Foliage Spawning System, quickly spawn trees and rocks with the most natural look.

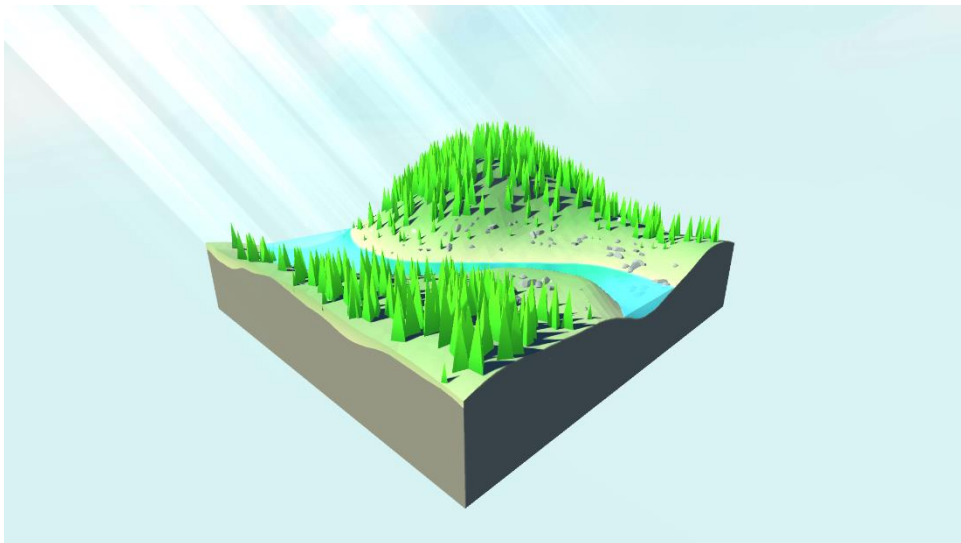
Below are some highlights of what it can do:

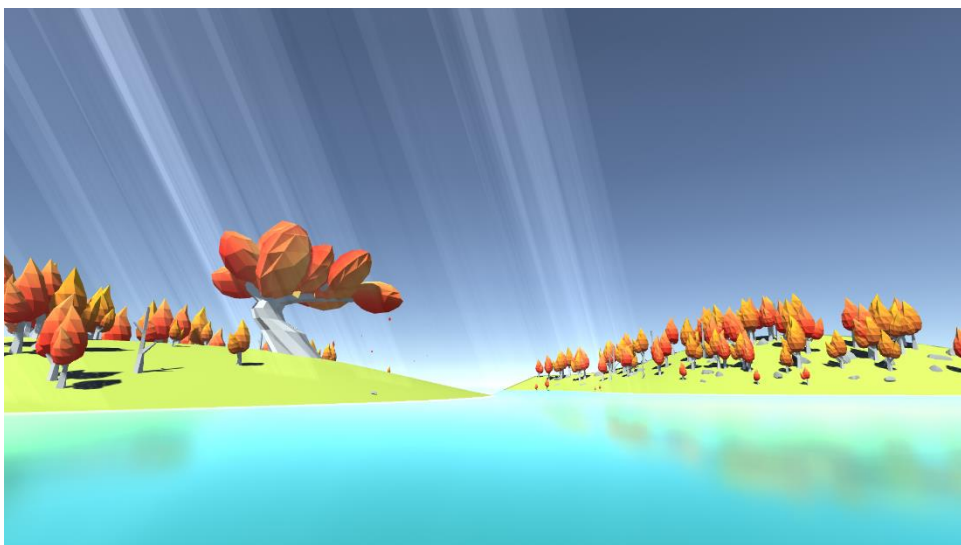
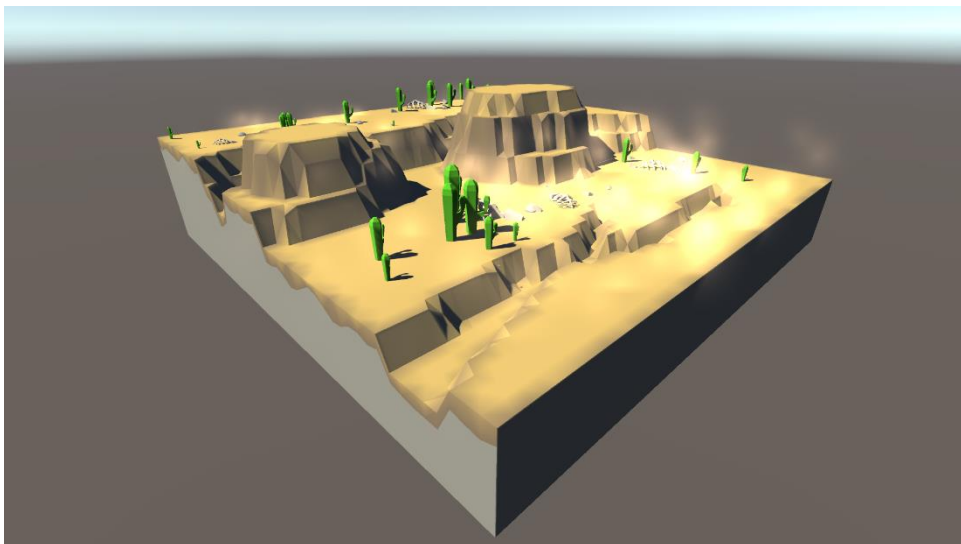
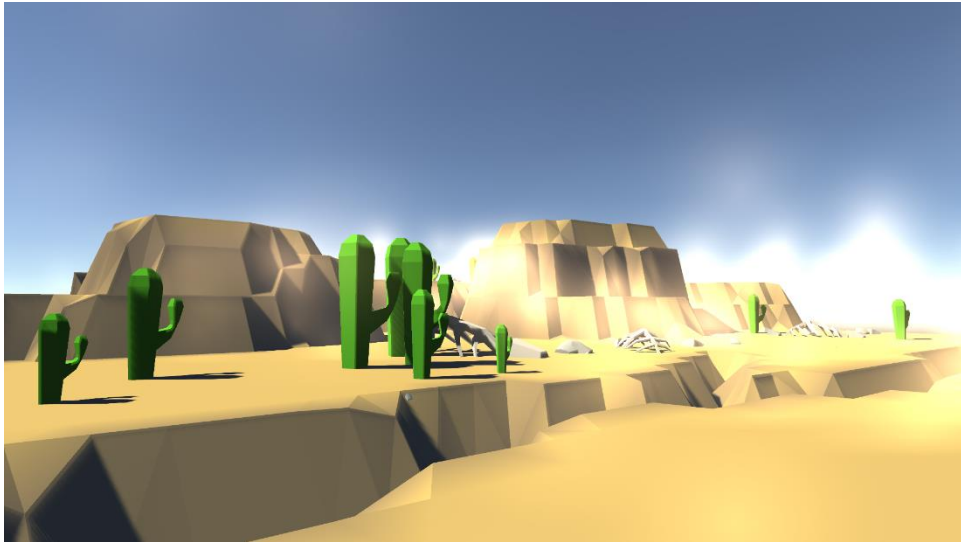
- Super easy to use.
- Real-time update for faster workflow
- Re-generate the exact terrain with the same parameters (pseudo-randomness)
- Vertex displacement using height map and Perlin noise
- The land looks like a slice of earth surface, not a boring bumping plane
- Fully control over how the underground looks like, from the surface thickness to the deep.
- Coloring using gradients, no colored texture required.
- Support for both flat and smooth shading.
- Stripping off unnecessary process.

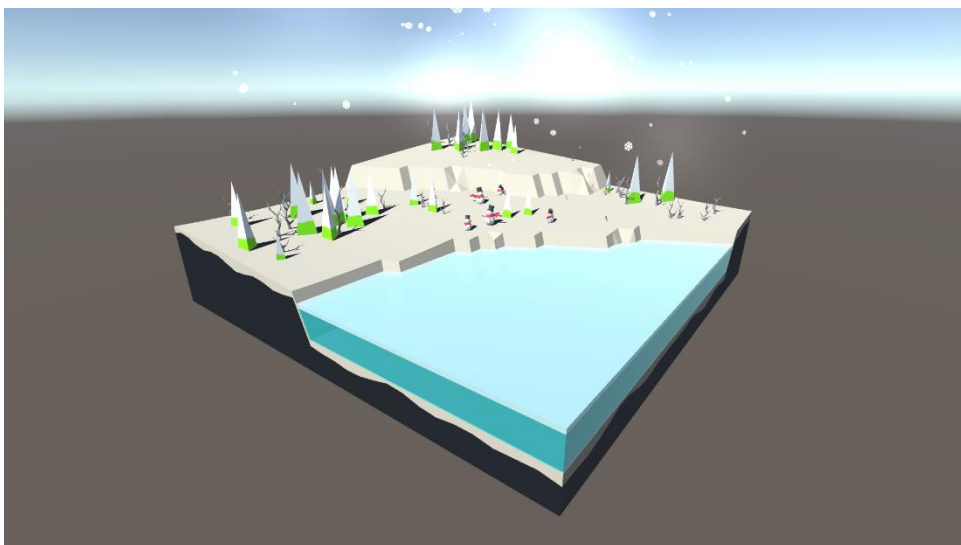
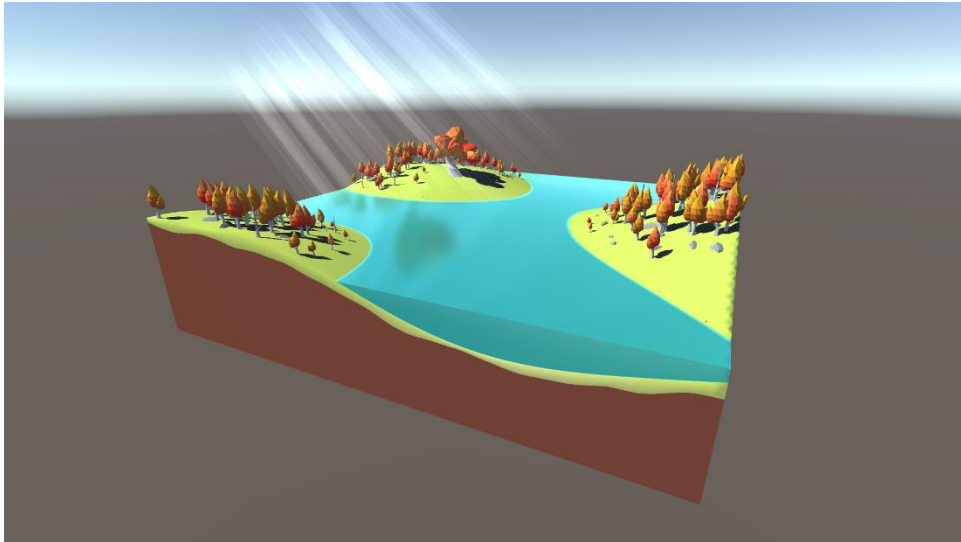


- Highly optimize for best performance.
- Export UV layout to a PNG file.
- Spawn tree and combine them into one large tree group.

Can't wait to see the result? Look at the pictures below:







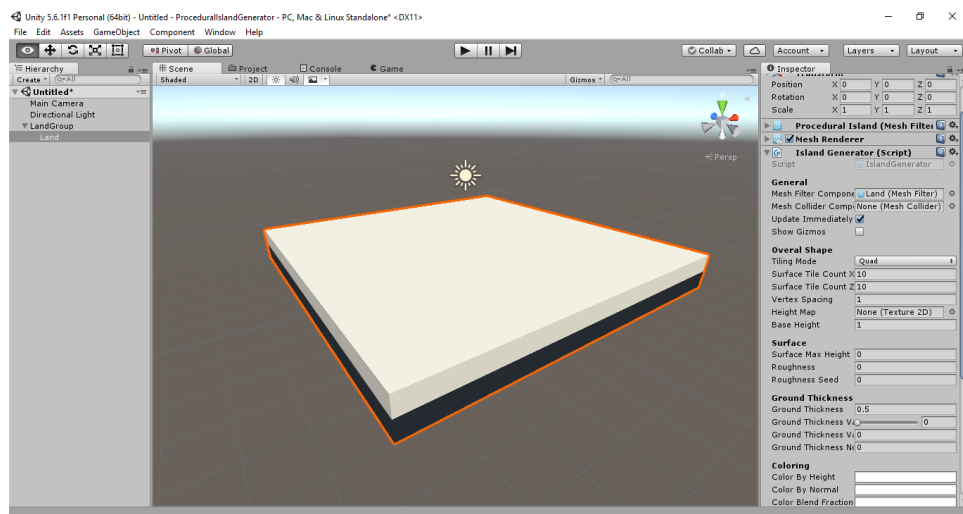
# HOW TO USE IT

## 1. Setting up the scene

In the current scene, create a new game object, let's call it "LandGroup", then create another game object as the LandGroup's child, then reset its Transform component, let's call the second game object "Land".

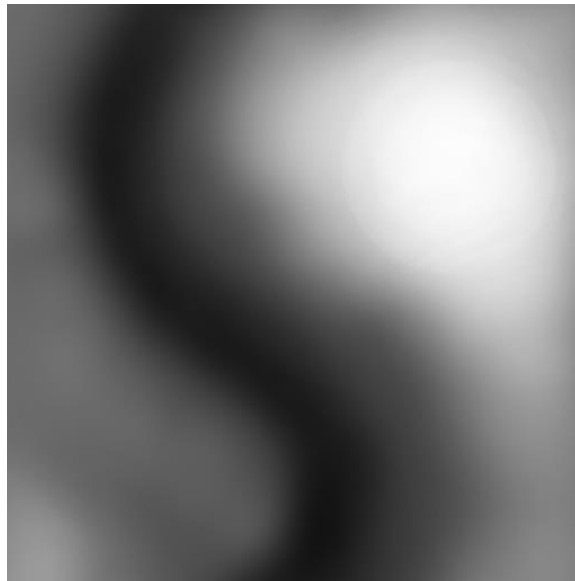
Attach a MeshFilter and MeshRenderer component to the Land, then assign the Material slot with the Terrain material located in Assets/IslandGenerator/Materials folder.

Then, attach the IslandGenerator component to the Land. After that, it should look like this:



## 2. Preparing your height map

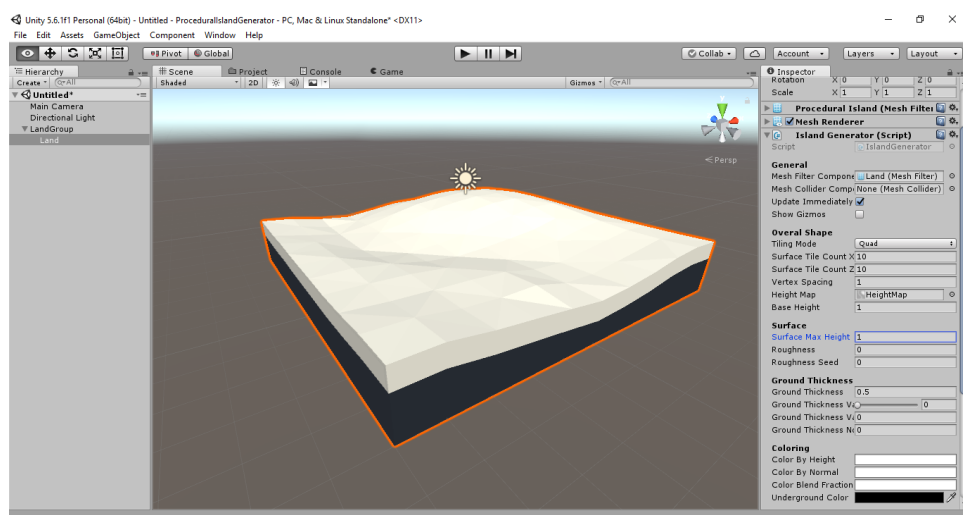
The height of your terrain is determined by a height map, which only contain grayscale pixels, where black is the lowest and white is the highest altitude. For example, we have this map, used in the Spring demo scene:



After sketching your height map, import it into Unity and tweak these following options:

- Alpha Source: set to **From Gray Scale**
- Read/Write Enable: set to **True**
- Wrap Mode: set to **Clamp**

Then, try assign the map into the HeightMap slot, set SurfaceMaxHeight to 1 and see the result:



### 3. Tweaking parameters

You can tweak these parameters to control how the terrain looks like:

**General:**

- Mesh Filter Component: hold the reference to the Mesh Filter, required.
- Mesh Collider Component: hold the reference to the Mesh Collider, assign it when your player has to walk on the terrain.
- Update Immediately: should the terrain update every time you change a parameter? Uncheck if your terrain's dimension is too large.
- Show Gizmos: should the editor draw a cube icon at every vertex?

**Overall Shape:**

- Tiling Mode: control how vertices connect together.
- Surface Tile Count X: number of tiles on the x-axis.
- Surface Tile Count Z: number of tiles on the z-axis.
- Vertex Spacing: distance between 2 vertices.
- Height Map: a map to control the elevation rate of the terrain.
- Base Height: height of the base (underground part).

**Surface:**

- Surface Max Height: the maximum height of terrain surface.
- Roughness: control surface bumpiness.
- Roughness Seed: a floating point number to generate a different look of roughness.

**Ground Thickness:**

- Ground Thickness: control the thickness of the surface layer.
- Ground Thickness Variation: determine how strong the thickness randomize is.
- Ground Thickness Variation Seed: a floating point number to generate a different look of ground thickness variation.
- Ground Thickness Noise Step: smooth out the ground thickness variation.

**Coloring:**

- Color By Height: determine color of each vertex by its height.





- Color By Normal: determine color of each vertex by its normal vector. The gradient fraction is the dot product of normal vector and global up vector.
- Color Blend Fraction: a fraction to blend between Color By Height and Color By Normal, where  $\alpha = 0$  is fully by height, and  $\alpha = 1$  is fully by normal.
- Underground Color: color of the underground part.

**Optional:**

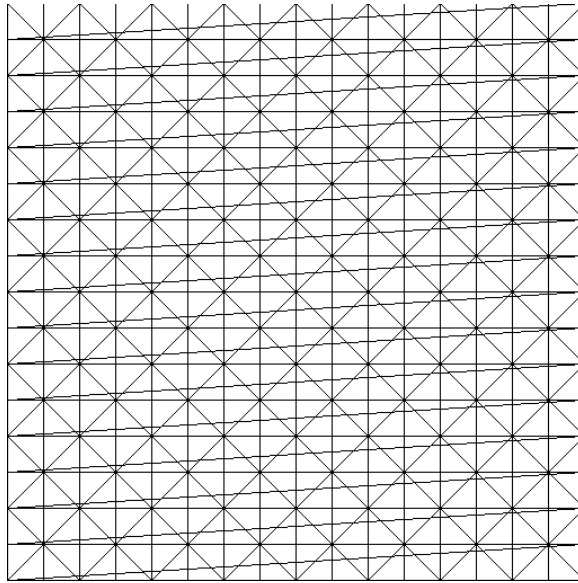
- Use Flat Shading: toggle between flat and smooth shading, the number of vertices is triple when turning on.
- Use Vertex Color: determine if it should generate vertex color data or not. Turn off if you use your custom shader that doesn't use this data.
- Should Recalculate Bounds: Check to recalculate mesh bounds. Should always on.
- Should Recalculate Normals: Check to recalculate mesh normal. Check on if your shader uses normal vectors.
- Should Recalculate Tangents: Check to recalculate mesh tangents. Check on if your shader uses tangent vectors.
- Should Unwrap Uv: Check to unwrap UV for the mesh. Should always on.
- Should Enclose Bottom Part: Check to enclose the bottom part of the mesh. Check on if your terrain should be visible from the bottom.

#### 4. Export UV layout

The UV layout is important to create a precise foliage map. To export UV layout, attach the UvExporter component to the Land, fill in appropriate parameters, then hit Export.

Your UV layout will be save in a png picture, located in Assets folder.

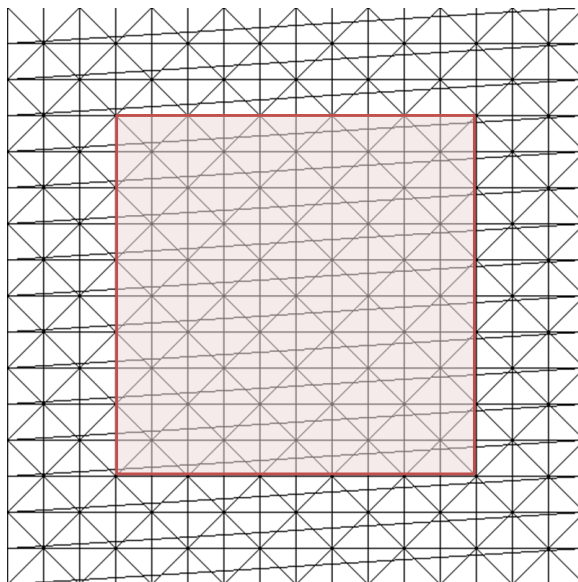




## 5. Create foliage distribution map

This kind of map is something similar to height map. It contains grayscale pixels where black is the sparsest and white is the densest region of tree.

One thing to remember, the UV coordinate of the surface does not cover the whole texture, but scale down a little bit, like the picture below (the first three rows and columns, last three rows and columns is used for underground vertices).



After creating your foliage distribution map, import it into Unity, remember to check the options as describe when we prepare height map.



Then, create a new game object as the LandGroup's child, attach the FoliageSpawner component to it, fill in appropriate parameters, hit Populate, you will see the trees grow.

Too many trees in a scene may affect performance. You can combine them into a larger group to reduce draw call and number of shadow caster. To do so, simply click Combine Meshes.

## Thanks!

Thank you for using the Procedural Island Generator, hope you like it.

For more interesting product, please visit: <http://pinwheel.studio>

Help us to develop more great product: <https://www.patreon.com/tntam>

