Alec Walker

Complex Games Assessment Task 2:

Integration

**Document Overview:**

This document aims to guide the end user through the integration of the Procedural Planet Generator into their project, as well as the features and functionality of the complex system.

A screenshot of a computer program

Description automatically generated

**Initial Setup:**

Make sure when creating a project within Unity it utilizes the Universal Render Pipeline as there are certain hardcoded features that require this.

Once you’re in a new or existing URP compatible project simply drag and drop the “proceduralplanetgeneration.unitypackage” file into your **Assets** window, then click **Import**.



A screenshot of a computer

Description automatically generated

That’s it… the system has now been added to your Unity project! The rest of this document describes features and how to use the system to generate planets.

**Setting Up Your First Planet:**

1. To start seeing anything you’ll have to create an empty game object within your scene **Hierarchy** (You can call “GameObject” whatever you like).

A screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

1. Add the “Planet” script component to the empty game object.

A screenshot of a computer program

Description automatically generated

*Upon adding the “Planet” script you’ll notice some errors in the console. This will be fixed in step* ***3****.*

1. Create two “settings” files anywhere in your **Assets** window, one “Colour Settings” and one “Shape Settings”. You can call these whatever you like.

A screenshot of a computer

Description automatically generated

*These files are used to store a planets shape a colour data. Each different planet you create will require its own shape and colour settings.*

Once you have your settings files, connect them to the “Planet” component.

A screen shot of a computer

Description automatically generated

*The console errors from connecting the Shape Settings will be resolved in step* ***4****.*

1. After step **3** you should have 2 drop down fields under the “Planet” component.

A screen shot of a phone

Description automatically generated

Start by added a layer of noise to the “Noise Layers” under Shape Settings 🡪 Noise Layers.

A screenshot of a computer

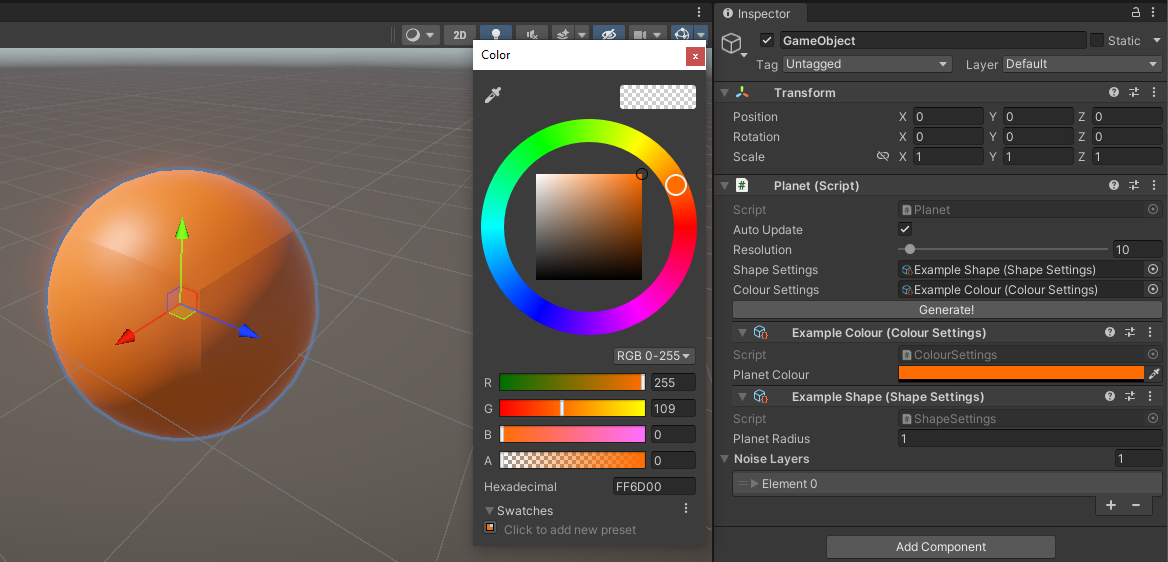
Description automatically generated

You should now see your planet in the scene!

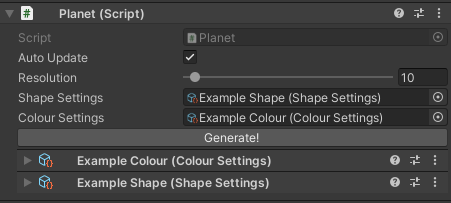
A screenshot of a computer

Description automatically generated

1. You can change the colour of this planet by editing the “Colour Settings” scriptable object, either directly in the “Planet” component or from wherever you saved it under **Assets**.



**Things to Know About the Planet Component:**



**3**

**2**

**1**

1. **“Auto Update”**

Auto Update usually only needs to be disabled when editing a planet with immense detail, as the higher a planets quality is, the more lag is induced with every change.

1. **“Resolution”**

Resolution describes how many vertices are along a given “face” of the planet. This is because the planet made up of 6 separate faces like a cube.

|  |  |
| --- | --- |
| Planet with resolution of 5: | Planet with resolution of 50: |

1. **“Generate!”**

The generate button only serves a purpose when “Auto Update” is unchecked. This is used to see results of editing when rendering changes live is causing too much lag.

**Using the Shape Settings Scriptable Object:**

The rest of this document will go into detail about the “Shape Settings” portion of the editor as this is what creates natural looking planets.

**Planet Radius:** describes the distance (in Unity units) from the centre of the planet to its surface.

**Noise Filters:** used to store multiple layers of noise with differing values to create desired results.

A screenshot of a computer

Description automatically generatedExpanding a “Noise Filter” element will show a few more options:

**Enabled:** dictates whether this noise layer is rendered to planet.

**First Filter Should Mask:** dictates whether the first filter of “Noise Filters” (Element 0) should mask this layer.

**Noise Settings:**

Within each layer of noise is a “Noise Settings” dropdown. This field contains all the variables and types of noise to use when generating the planet.

**Noise Type:**

There are currently two types of noise, but the system is setup for new types of noise to be seamlessly integrated. These noise types are:

|  |  |
| --- | --- |
| * **Simple**   Raw 3D Simplex Noise values. | * **Inverse**   Inversed 3D Simplex Noise values. |

The following examples are all using the “Simple” noise type.

**Strength:** multiplies the final noise value with this float.

*Below is the same noise configuration with different strength values:*

|  |  |
| --- | --- |
| Strength: 0.1 | Strength: 0.5 |

**Roughness:** multiplies the frequency of noise within subsequent “Noise Layer” layers with this float.

*This is only visible when there are multiple “Noise Layers” in a single “Noise Filter” element.*

*Below is the same noise configuration with different roughness values:*

|  |  |
| --- | --- |
| Roughness: 1 | Roughness: 5 |

**Base Roughness:** the frequency of all noise layers is this value.

*Below is the same noise configuration with different base roughness values:*

|  |  |
| --- | --- |
| Base Roughness: 1 | Base Roughness: 5 |

**Persistence:** represents the amount the previous “Noise Layer” should affect this noise layer.

*The hardcoded value this effects is 1 meaning a persistence value of 0.5 would result in 50% of the previous noise layer affecting this layer.*

*Below is the same noise configuration with different persistence values:*

|  |  |
| --- | --- |
| Persistence: 0.25 | Persistence: 1 |

**Min Value:** represents the minimum value of noise displacement before beginning to render noise.

*Below is the same noise configuration with different min values:*

|  |  |
| --- | --- |
| Min Value: 0  A brown ball with a grid  Description automatically generated with medium confidence | Min Value: 1.25 |

**Noise Layers:** the number of layers of noise that should affect this instance of the filter.

*Below is the same noise configuration with different numbers of noise layers:*

|  |  |
| --- | --- |
| Noise Layers: 2  A brown ball with a grid  Description automatically generated with medium confidence | Noise Layers: 8 |

**Origin:** a vector 3 that represents displacement of noise. This will not affect how the noise looks just how it appears on the sphere.

*The best way to visualize this is to have “Auto Update” checked and to drag the “x, y or z” components.*

*Below is the same noise configuration with different origin points:*

|  |  |
| --- | --- |
| Origin: 0, 0, 0  A brown ball with a grid  Description automatically generated with medium confidence | Origin: 0, 0, 5 |

**Inverse Noise Type:** this is an example of the same noise configuration with different noise types.

|  |  |
| --- | --- |
| Noise Type: Simple | Noise Type: Inverse |

**Conclusion:**

With these variables you can create wide variety of planets, however due to the nature of this system it is hard to predict what tweaking a given variable will achieve. The easiest way to make what you imagine is to just play around with settings until you have a better understanding of how these change the result!