

MANUAL

PROCEDURAL

PLANETS

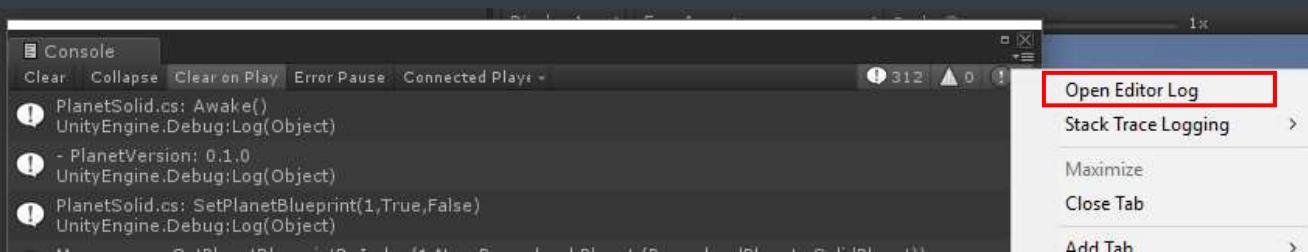
- **VERSION 0.1.1 ALPHA**
 - THIS IS AN ALPHA VERSION OF THE PROCEDURAL PLANETS ASSETS
 - THERE WILL BE BUGS AND NOT ALL FEATURES HAVE BEEN IMPLEMENTED
- BUGS REPORTS - YOU CAN REGISTER AND FOLLOW PROGRESS ON BUGS HERE:
 - <https://www.imphenzia.com/mantisbt>

ALPHA VERSION

DISCLAIMER

OR EMAIL THEM TO SUPPORT@IMPHENZIA.COM

1. PLEASE SET DEBUG LEVEL TO “BASIC” IN THE BOTTOM OF THE INSPECTOR OF THE PROCEDURALPLANETMANAGER MANAGER COMPONENT.
2. REPRODUCE THE PROBLEM (INCLUDE DESCRIPTION IN THE BUG REPORT IF POSSIBLE)
3. CLICK ON DROPODOWN OF THE CONSOLE AND SELECT OPEN EDITOR LOG
4. INCLUDE THE CONTENT OF THE EDITOR LOG IN THE BUG REPORT



KNOWN ISSUES

THIS IS AN ALPHA RELEASE THERE ARE KNOWN ISSUES TO BE AWARE OF.

VERY IMPORTANT:

- CHANGING BLUEPRINT ORDER, BLUEPRINT CONFIGURATION, ADDING / REMOVING BLUEPRINTS, AND CHANGING PROCEDURAL MATERIAL ARRAYS WILL CHANGE APPEARANCE OF PLANETS EVEN IF THE SAME SEED IS USED. THIS IS THE NATURE OF RANDOMNESS. PLANETS EXPORTED TO JSON-STRINGS, HOWEVER, INCLUDE ALL VALUES AND CAN FORCE PROPERTY OVERRIDE SO PLANETS LOOK THE SAME TO SOME EXTENT.
- APPEARANCE OF PLANETS CREATED DURING THE ALPHA RELEASE WILL NOT LOOK THE SAME (OR MAY NOT WORK) AS IN THE FINAL RELEASE SINCE MATERIALS AND PROPERTIES WILL BE MODIFIED AND ADDED/REMOVED.

GOOD TO KNOW:

- GAS PLANETS ARE NOT YET IMPLEMENTED.
- UNITY RESET BUTTON (FOUND UNDER THE COG WHEEL ICON) IN INSPECTOR CANNOT BE USED TO RESET A PLANET.
- PLANETS WILL INITIALLY APPEAR IN A SOLID COLOR A GRADUAL BUILDUP OF TEXTURES UNTIL THE PLANET IS FULLY CREATED.
- EDITING PREFABS OF PLANETS THAT ARE NOT INSTANTIATED IS NOT SUPPORTED.
- NOT ALL UNITY SUPPORTED PLATFORMS ARE SUPPORTED, E.G. WEBGL
- LINEAR COLOR SPACE LOOKS DIFFERENT FROM GAMMA COLOR SPACE (UNAVOIDABLE – CHOOSE A COLOR SPACE TO YOUR LIKING)
- PLANETARY RINGS CANNOT BE ROTATED AROUND ANY AXIS YET – THIS IS BECAUSE THEY NEED TO AUTO ALIGN WITH THE CAMERA FOR TRANSPARENCY SORTING PURPOSES.
- **WARNINGS IN 2017.3 AND NEWER:**
 - SOME WARNINGS WILL BE DISPLAYED IN UNITY 2017.3 AND NEWER BECAUSE "PROCEDURAL MATERIALS" WILL BE DEPRECATED. THE PROCEDURAL MATERIALS ARE BASED ON TECHNOLOGY BY ALLEGORITHMIC AND FOR UNITY 2018.x ALLEGORITHMIC WILL RELEASE THEIR OWN PLUGIN INSTEAD TO ALLOW MORE RAPID IMPLEMENTATION OF FEATURES COMPARED TO WHEN IT IS INTEGRATED IN UNITY AS IT IS TODAY. ALTHOUGH THIS WILL CREATE A DEPENDENCY ON ANOTHER COMPONENT IT MAY COME WITH THE BENEFIT OF ALLOWING HIGHER RESOLUTION TEXTURES, GPU PROCESSING FOR TEXTURES, LIKE IN OTHER GAME ENGINES THAT USE ALLEGORITHMIC'S OWN PLUGIN. I WILL ACTIVELY BE WORKING TO GET UNITY 2018 TO WORK WITH THIS ASSET AS SOON AS POSSIBLE.

COLOR SPACE

“WORKING IN LINEAR COLOR SPACE GIVES MORE ACCURATE RENDERING THAN WORKING IN GAMMA COLOR SPACE.”

- **LINEAR OR GAMMA WORKFLOW?**
 - DESIGNED FOR LINEAR COLOR SPACE
 - WORKS IN GAMMA COLOR SPACE (BUT LOOKS DIFFERENT)
- **MORE INFORMATION:**
 - [HTTPS://DOCS.UNITY3D.COM/MANUAL/LINEARRENDERING-LINEARORGAMMAWORKFLOW.HTML](https://docs.unity3d.com/Manual/LinearRendering-LinearOrGammaWorkflow.html)

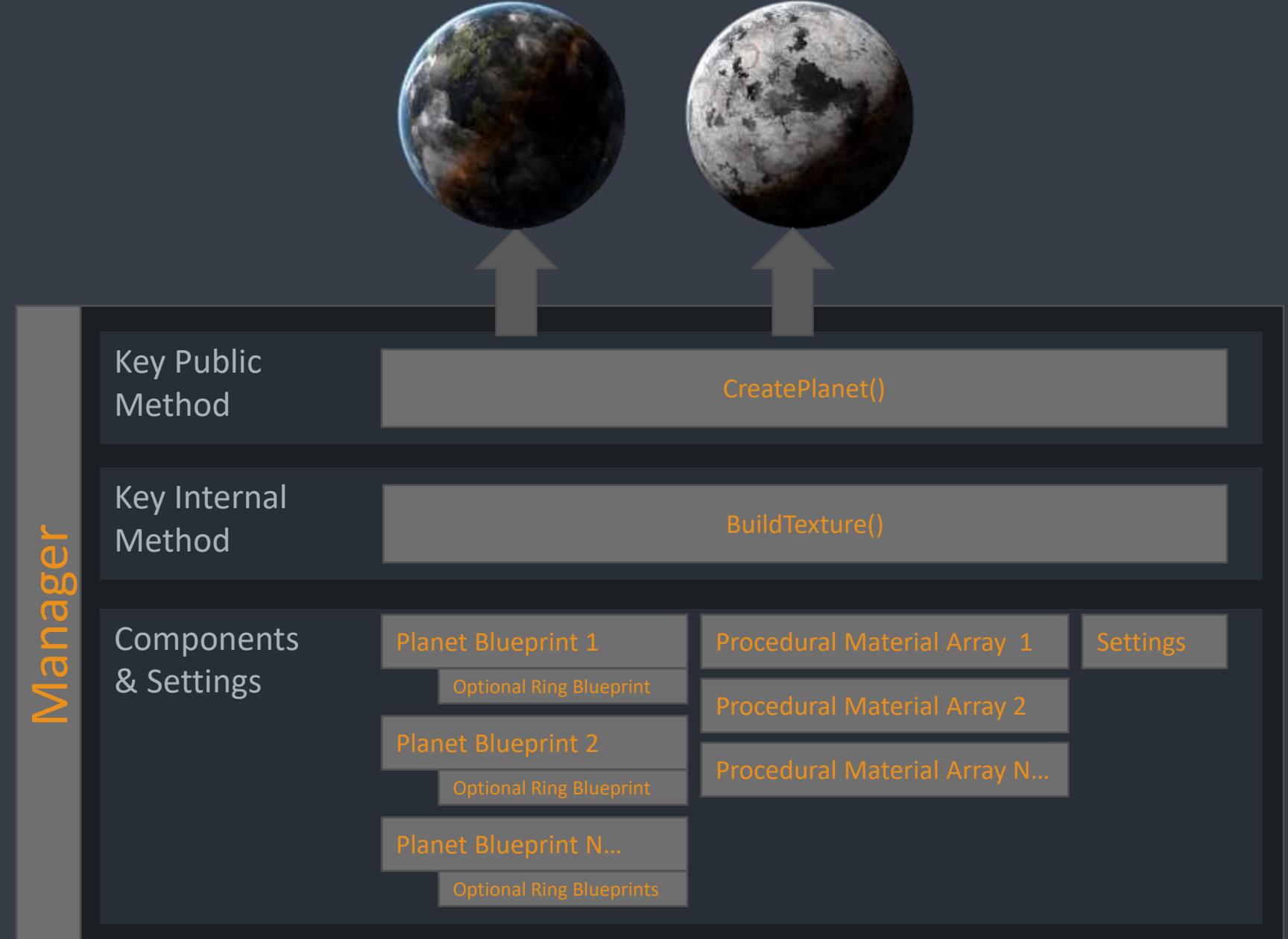
GENERAL INFORMATION

- **PROGRAMMING LANGUAGE**
 - ALL SCRIPTS ARE WRITTEN IN C#
- **NAMESPACE**
 - TO AVOID NAME CONFLICTS ALL SCRIPTS USE THE NAMESPACE “PROCEDURALPLANETS”
 - USE “USING PROCEDURALPLANETS;” DIRECTIVE IN SCRIPTS TO ACCESS THE NAMESPACE.
- **PERSISTENT MANAGER COMPONENT**
 - A PERSISTENT GAME OBJECT WITH THE “MANAGER” COMPONENT MUST EXIST IN EACH UNITY SCENE THAT HAS PROCEDURAL PLANETS.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using ProceduralPlanets;
```

ARCHITECTURE OVERVIEW

THIS IS ONLY A ARCHITECTURAL CONCEPT PICTURE AND THE FOLLOWING PAGES WILL DESCRIBE SOME OF THE IMPORTANT CONCEPTS, METHODS, AND COMPONENTS.



ARCHITECTURE OVERVIEW

MANAGER COMPONENT

Manager

Manager is an essential key component that must persistently exist in all scenes for the planets to work.

What does the Manager component do?

The Manager maintains a repository of **Planet Blueprints** that allow you to create random planets of specific types, e.g. terrestrial (earth-like) planets, ice planets, dusty planets, etc. The Manager also keeps track of the probability of a planet type is to be randomly created so that you can make derelict rocky planets more common than populated water worlds.

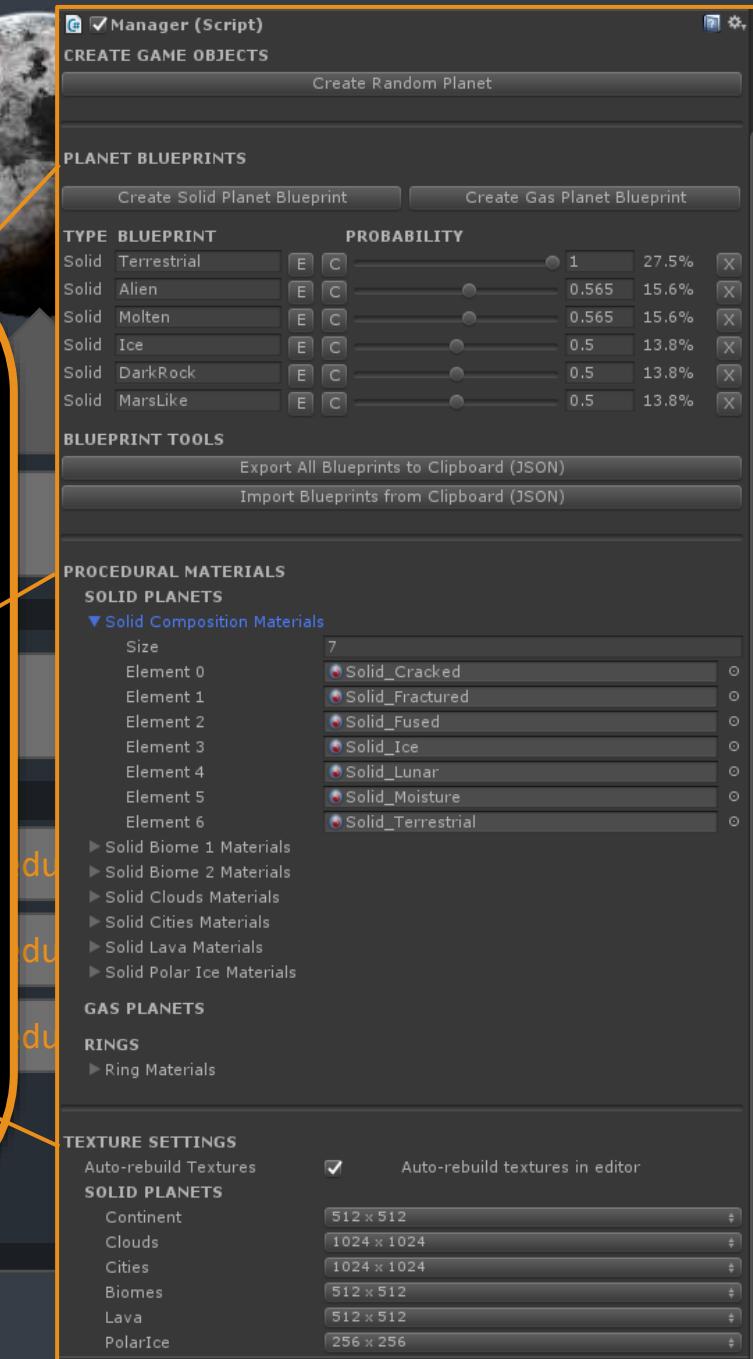
The Manager maintains a repository of **Procedural Material arrays**. These arrays can be filtered by blueprints so dusty planets only allow dusty surface materials. This also allows for more procedural materials for planet composition, biomes, polar ice, clouds, and lava to be added at a later stage.

The Manager is in charge of **building planet textures** and sending copies of the textures to planets once they have been built. Why it works this way is described in more detail later on.

The Manager controls the **resolution of textures**.

The Manager can be called via scripts to **create planets during runtime**.

The Manager inspector can be used to create planet and ring blueprints, random planets, or planets of a specific blueprint type.



ARCHITECTURE OVERVIEW

KEY PUBLIC METHODS

Manager

Key Public Methods

CreatePlanet()

K

CreatePlanet() is a public method that you can access through scripts to create planets during runtime.

You can call it like this to create a random planet at a Vector3 position in the current scene:

```
Manager.Instance.CreatePlanet(<Vector3 position>);
```

You can also specify a specific planet seed, like this:

```
Manager.Instance.CreatePlanet(<Vector3 position>, <integer seed>);
```

If you want to override and force a specific blueprint, e.g. a terrestrial planet, you can do this:

```
Manager.Instance.CreatePlanet(<Vector3 position>, <integer seed>, <string blueprint name>);
```

Finally, you can also create a very specific planet using a JSON-string, like this:

```
Manager.Instance.CreatePlanet(<Vector3 position>, <string JSON settings>);
```



ARCHITECTURE OVERVIEW

KEY INTERNAL METHODS

Manager

Key Public Methods

`CreatePlanet()`

Key Internal Methods

`BuildTexture()`

BuildTexture() is an important internal method that you don't need to do anything with but you need to be aware of its existence and what it does.

Procedural Materials in Unity overwrite textures that they generate and in order to have multiple planets visible at any given time – the Manager must sequentially build textures and allocate them to planets one by one.

Planets automatically call the `BuildTexture()` method of their manager when their `Awake()` method is executed AND whenever properties of planets are changed that require procedural textures to be rebuilt. The request is put in a queue and processed in the order they arrive to the Manager. Once a texture is rebuilt the texture is copied in the GPU and sent back to the planet so it changes its appearance without affecting any other planets.

The `BuildTexture()` executes both in Play mode and in the editor and textures are built asynchronously so it doesn't halt any other processes or game events.



ARCHITECTURE OVERVIEW

COMPONENTS & SETTINGS

The Manager component in the Inspector shows the following settings:

- CREATE GAME OBJECTS**: Create Random Planet
- PLANET BLUEPRINTS**:
 - Create Solid Planet Blueprint
 - Create Gas Planet Blueprint
- TYPE BLUEPRINT**:

Type	Blueprint	E	C	Probability	Probability (%)
Solid	Terrestrial	<input type="radio"/>	<input type="radio"/>	1	27.5%
Solid	Alien	<input type="radio"/>	<input type="radio"/>	0.565	15.6%
Solid	Molten	<input type="radio"/>	<input type="radio"/>	0.565	15.6%
Solid	Ice	<input type="radio"/>	<input type="radio"/>	0.5	13.8%
Solid	DarkRock	<input type="radio"/>	<input type="radio"/>	0.5	13.8%
Solid	MarsLike	<input type="radio"/>	<input type="radio"/>	0.5	13.8%
- BLUEPRINT TOOLS**:
 - Export All Blueprints to Clipboard (JSON)
 - Import Blueprints from Clipboard (JSON)

Planet Blueprints are templates with specific random ranges for parameters.

Why are planet blueprints needed?

Totally random planets tend to not look very good unless you are very lucky. Also, you may want to be able to specify a particular type of planet to be created, e.g. a terrestrial (earth-like) planet, ice planet, molten planet, or dusty planet.

How do you create blueprints?

- 1) Select the Manager component in the inspector and click, for example, "Create Solid Planet Blueprint".
- 2) Set the Probability value in the Manager inspector – this dictates how likely this blueprint is to be used when a random planet is created.
- 3) Highlight the newly created child object in the hierarchy
- 4) Rename the planet blueprint to a new unique descriptive name, e.g. Desert
- 5) Configure sliders in the inspector to filter use of only certain materials, minimum/maximum ranges of properties (e.g. if you want no molten lava on the planet you set the both Min and Max values of the slider to 0 and if always want at least half the planet to be covered by water you set Liquid Level Min to 0.5 and Max value to 1.0)
- 6) You can optionally also create a child ring blueprint and set the probability of a planet created sing this blueprint having rings.

Note: Creating blueprints will require you to have a fair understanding of all the properties so you may want to play around with a planet in the scene and pull all sliders to see what effect they have.

Manager

Components & Settings

Planet Blueprint 1

Optional Ring Blueprint

Planet Blueprint 2

Optional Ring Blueprint

Planet Blueprint N...

Optional Ring Blueprints

Procedural

Procedural

Procedural

The Blueprint Solid Planet (Script) component in the Inspector shows the following settings:

- PLANET BLUEPRINT TOOLS**:
 - Create Planet
 - Export Blueprint to Clipboard (JSON)
 - Import Blueprint from Clipboard (JSON)
- PLANET SETTINGS**:
 - Alienization: 0.00 - 0.00
 - Composition: Solid_Terrestrial (selected)
 - Specular Color:
 - Hue Range: 0.05
 - Saturation Range: 0.05
 - Brightness Range: 0.05
- CONTINENTS**:
 - Continent Size: 0.77 - 1.00
 - Continent Complexity: 0.21 - 0.43
- COAST LINES**:
 - Coastal Detail: 0.20 - 0.47
 - Coastal Reach: 0.25 - 0.57
- LIQUID**:
 - Liquid Level: 0.63 - 0.84
 - LiquidColor:
 - Hue Range: 0.02
 - Saturation Range: 0.082
 - Brightness Range: 0.134
 - Liquid Opacity: 1.00 - 1.00
 - Shallow Distance: 0.00 - 0.00
 - Specular Power: 0.52 - 0.59
- ICE**:
 - Polar Ice: 0.33 - 0.49
 - Polar Caps:
 - Hue Range: 0.1
 - Saturation Range: 0.1
 - Brightness Range: 0.1
- ATMOSPHERE**:
 - Atmosphere Color:
 - Hue Range: 0.025
 - Saturation Range: 0.13
 - Brightness Range: 0.2
 - External Size: 0.50 - 0.66
 - External Density: 0.66 - 0.82
 - Internal Density: 0.90 - 1.00
 - Twilight Color:
 - Hue Range: 0.05
 - Saturation Range: 0.2
 - Brightness Range: 0.2
- CLOUDS**:
 - Clouds Opacity: 1.00 - 1.00
 - Clouds:
 - Clouds Color: Everything

ARCHITECTURE OVERVIEW

COMPONENTS & SETTINGS

Manager

Key Public Methods

Key Internal Methods

Components & Settings



Procedural Material Arrays are lists of procedural materials associated with planet composition, biomes, clouds, lava, ice, and night lights.

The planet blueprints can have masked selection lists from these procedural material arrays so a planet type like terrestrial can mask out materials, like Dust for example, so it's never randomly selected when a planet is generated.

The asset is designed so more materials can be added although that will break existing planets random nature and force them to have all materials overridden.

Planet Blueprint 1

Optional Ring Blueprint

Planet Blueprint 2

Optional Ring Blueprint

Planet Blueprint N...

Optional Ring Blueprints

Procedural Material Array 1

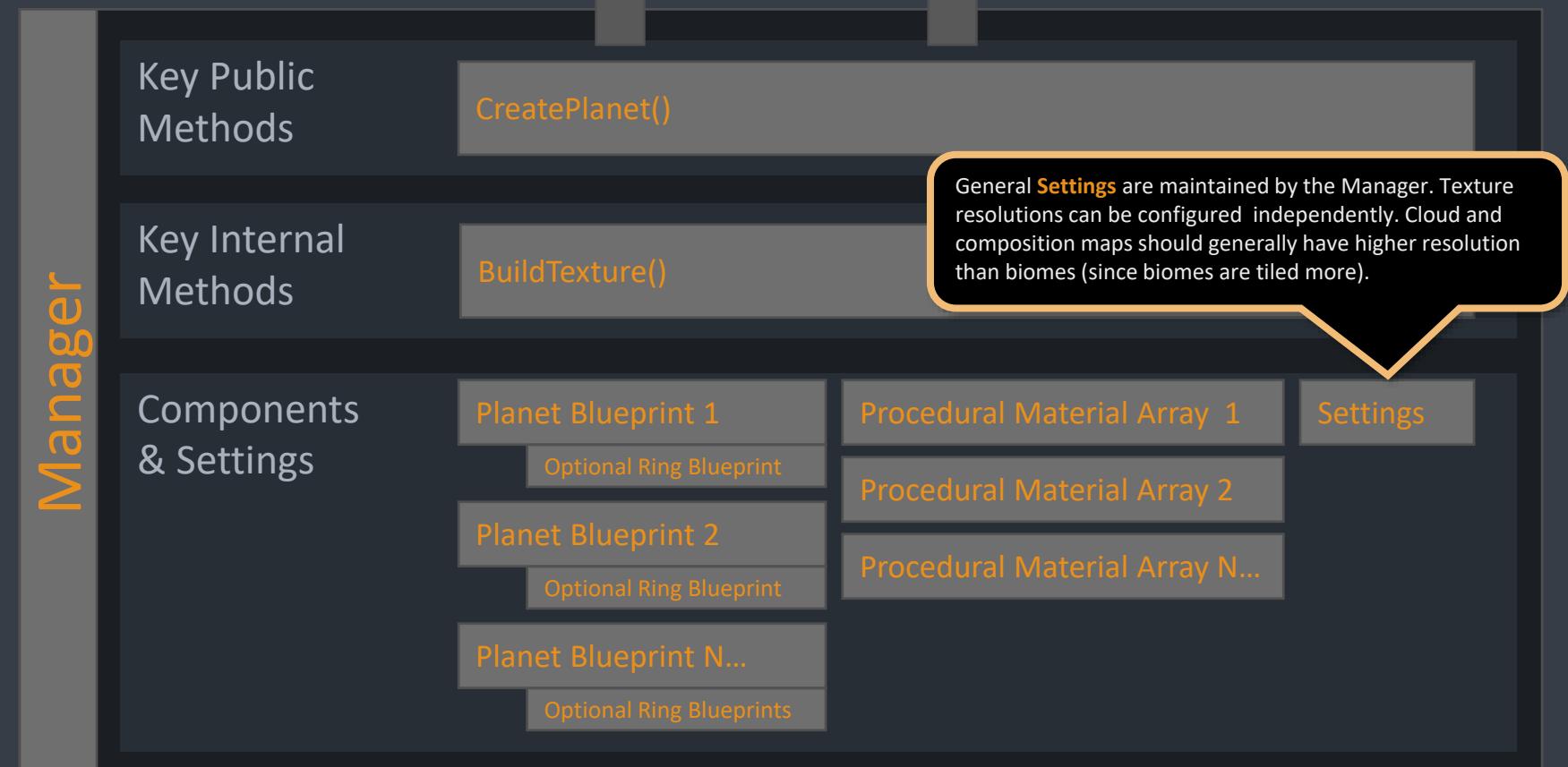
Procedural Material Array 2

Procedural Material Array N...

Settings

ARCHITECTURE OVERVIEW

COMPONENTS & SETTINGS



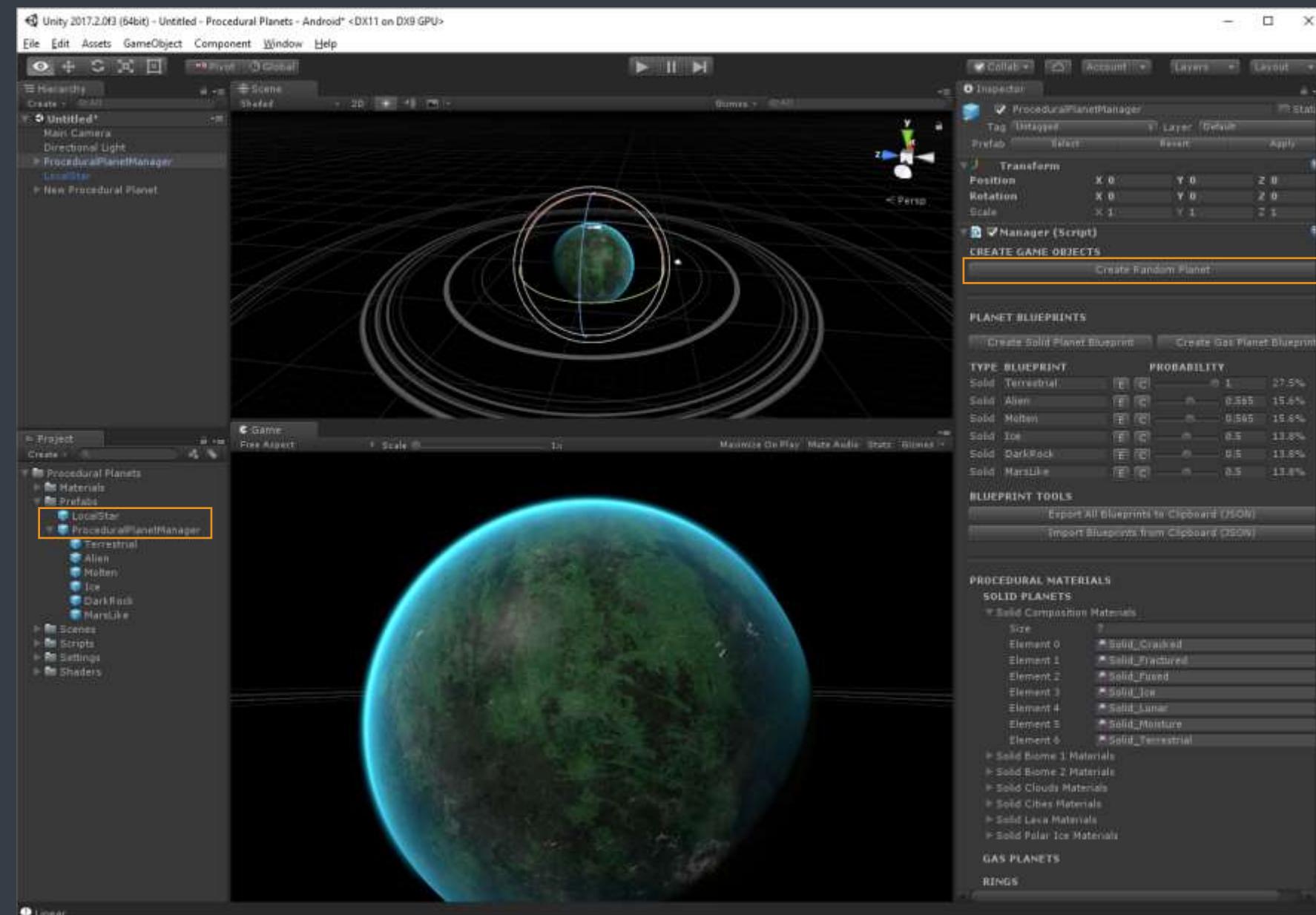
GETTING STARTED

PREPARATIONS

1. CREATE A NEW PROJECT
2. GO TO FILE | BUILD SETTINGS | PLAYER SETTINGS...
 1. UNDER OTHER SETTINGS CHANGE COLOR SPACE TO LINEAR
3. IMPORT THE PROCEDURALPLANETS ASSET
4. CHANGE THE MAIN CAMERA TO SOLID COLOR AND SET BACKGROUND COLOR TO BLACK

CREATE MANAGER, STAR AND A RANDOM PLANET

1. DRAG THE PREFABS\PROCEDURALPLANETMANAGER PREFAB TO THE HIERARCHY
2. DRAG THE PREFABS\LOCALSTAR PREFAB TO THE HIERARCHY
3. PROCEDURALPLANETMANAGER THE HIERARCHY
 1. CLICK ON CREATE RANDOM PLANET BUTTON TO CREATE YOUR FIRST PLANET



MODIFYING PLANET IN INSPECTOR

1. SELECT THE PLANET IN THE HIERARCHY
2. SCROLL DOWN IN THE INSPECTOR FOR THE PLANET AND LOCATE PLANET SETTINGS
3. USE SLIDERS, COLOR FIELDS AND DROP DOWN DIALOGS TO OVERRIDE PARAMETERS.

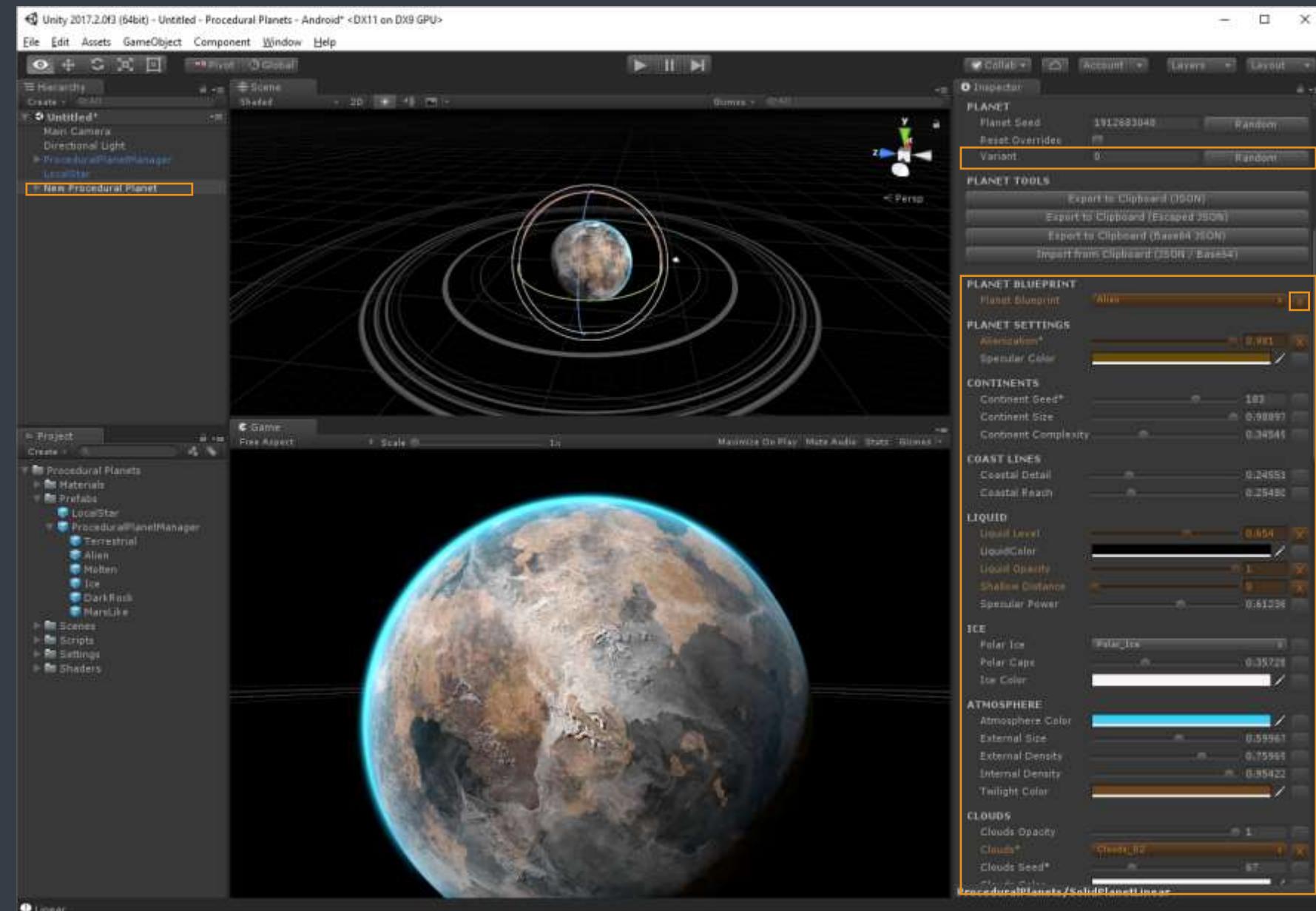
AS YOU CHANGE A PARAMETER IT WILL SWITCH FROM BEING RANDOMLY ALLOCATED TO AN OVERRIDDEN VALUE. THIS IS INDICATED BY CHANGING TO YELLOW COLOR IN THE INSPECTOR AND A YELLOW X BUTTON APPEARS (WHICH WILL REVERT TO THE ORIGINAL RANDOM VALUE IF CLICKED)

PLANET BLUEPRINT:

BY DEFAULT A PLANET WILL ALREADY HAVE THE PLANET BLUEPRINT SET TO AN OVERRIDDEN VALUE. THIS IS BECAUSE THE MANAGER SPAWNS TYPES OF PLANETS BASED ON PROBABILITY SO IT NEEDS TO FORCE A BLUEPRINT TYPE. YOU CAN MAKE A PLANET SELECTING A RANDOM BLUEPRINT BY CLICKING THE YELLOW X BUTTON NEXT TO THE PLANET BLUEPRINT.

VARIANT:

YOU CAN CHANGE TO A NEW VARIANT OF THE PLANET BY CHANGING THE VARIANT SEED. THIS WILL CHANGE RANDOM OF SOME SELECTED PARAMETERS TO A SMALL DEGREE WHICH SHOULD MAKE A VERY SIMILAR PLANET BUT WITH DIFFERENT CONTINENTS AND FEATURE FORMATIONS.



EXPORTING & IMPORTING PLANETS

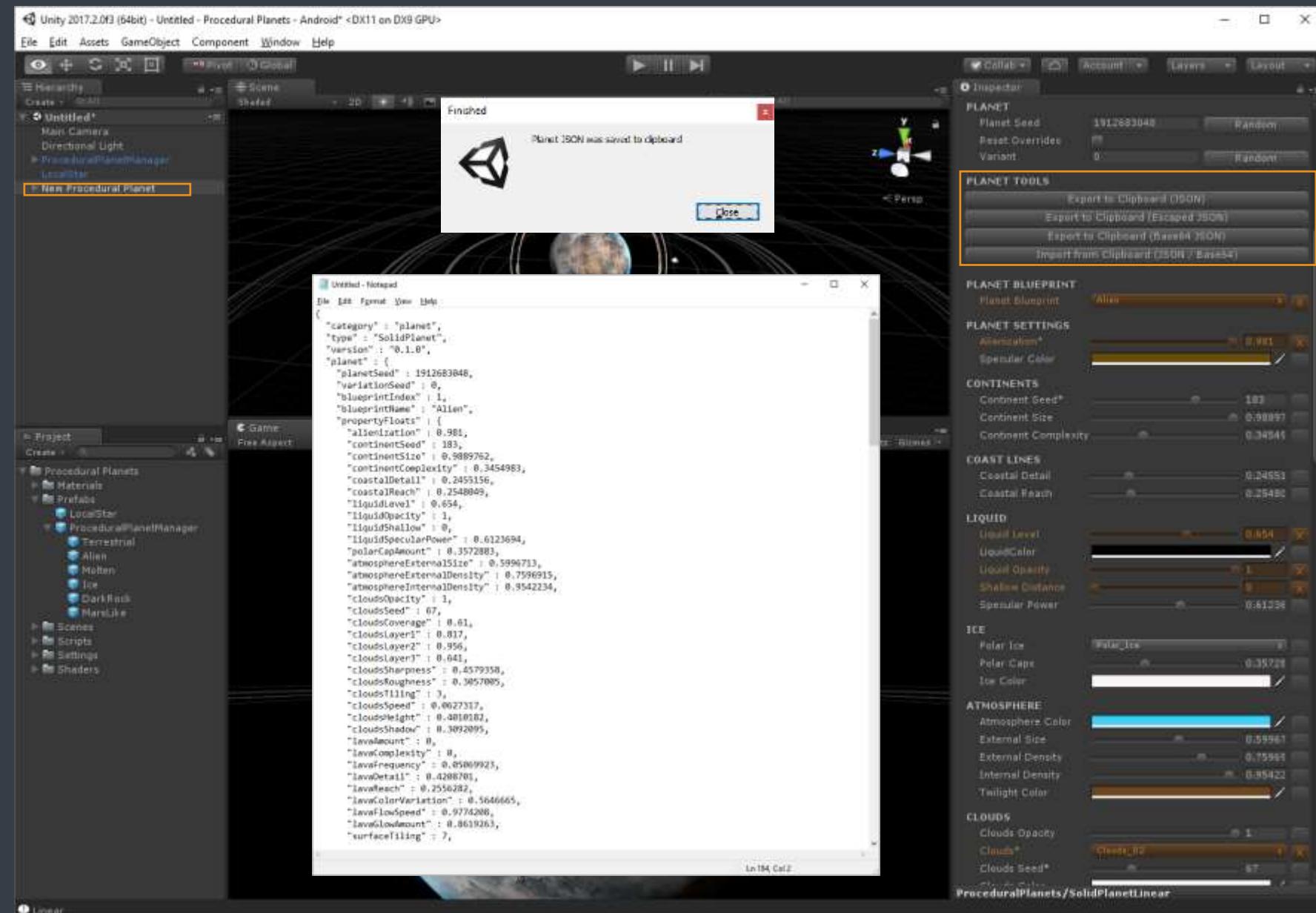
1. SELECT THE PLANET IN THE HIERARCHY
2. SCROLL DOWN IN THE INSPECTOR FOR THE PLANET AND LOCATE PLANET TOOLS
3. CLICK ON “EXPORT TO CLIPBOARD (JSON)” BUTTON

YOU NOW HAVE EASY TO READ INDENTED JSON IN YOUR CLIPBOARD. PASTE THE CONTENTS IN NOTEPAD (OR ANOTHER EDITOR) TO INSPECT.

IMPORTANT: NOTE THAT EVEN IF ONLY SOME PROPERTIES ARE OVERRIDDEN, ALL THE PROPERTIES ARE INCLUDED IN THE JSON EXPORT. THIS IS BECAUSE IF YOU CHANGE ANYTHING IN THE BLUEPRINTS, OR ADD/REMOVE TEXTURES, OR CHANGE THE ORDER OF BLUEPRINTS — THE RANDOM SEEDED NATURE OF PROPERTIES WILL NO LONGER BE THE SAME. IF THIS JSON DATA IS IMPORTED WHEN SUCH CHANGES HAVE BEEN MADE ALL THE PROPERTIES WILL BE OVERRIDDEN TO ENSURE THAT THE PLANET LOOKS LIKE IT DID WHEN IT WAS EXPORTED.

WHAT TO DO WITH THE JSON EXPORTED INFORMATION?

- YOU CAN STORE THE JSON STRINGS IN A TEXT FILE AS A BACKUP TO IMPORT AT A LATER STAGE.
- YOU CAN IMPORT IT AGAIN USING THE **IMPORT FROM CLIPBOARD (JSON/BASE64)** BUTTON.
- YOU CAN EXPORT AS ESCAPED JSON OR BASE64 JSON WHICH ARE BOTH SAFE TO STORE IN DATABASES OR IN SCRIPT CODE AND USE THE **CREATEPLANET()** METHOD IN THE MANAGER TO INSTANTIATE THE EXACT PLANET FROM A SCRIPT.



PLANET BLUEPRINTS OVERVIEW

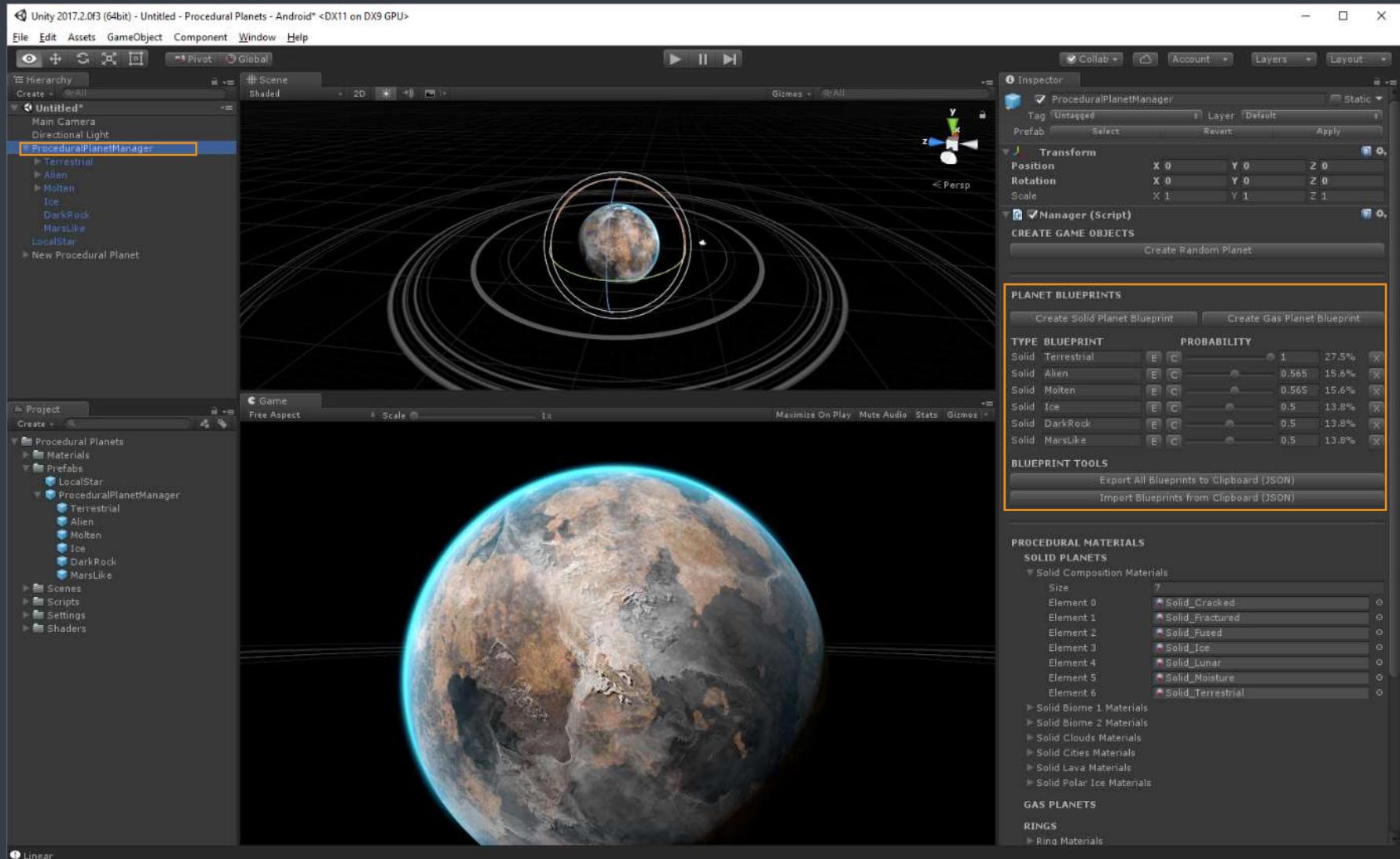
1. HIGHLIGHT THE MANAGER IN THE HIERARCHY

2. SEE THE PLANET BLUEPRINTS SECTION

CREATE SOLID/GAS PLANET BLUEPRINT BUTTONS

- CREATES A PLANET BLUEPRINT AND ADDS IT TO THE LIST OF BLUEPRINTS BELOW. A NEW CHILD OBJECT WITH A DERIVED PLANET BLUEPRINT COMPONENT WILL BE ADDED TO THE MANAGER WITH A UNIQUE NAME.
- YOU CAN RENAME BLUEPRINTS (NAMES MUST BE UNIQUE) BY EDITING THE TEXT FIELD IN THE INSPECTOR OR BY RENAMING THE CHILD GAME OBJECT DIRECTLY.
- THE [E] BUTTON IS A SHORTCUT TO SELECT AND EDIT THE PLANET BLUEPRINT IN THE EDITOR
- THE [C] BUTTON CREATES A PLANET USING A PARTICULAR BLUEPRINT.
- THE PROBABILITY SLIDER SETS HOW LIKELY THIS PLANET TYPE WILL BE USED WHEN A RANDOM PLANET IS CREATED.
- THE [X] BUTTON DELETES THE BLUEPRINT.

THE EXPORT/IMPORT BLUEPRINT BUTTONS TO/FROM CLIPBOARD WILL EXPORT AND IMPORT ALL BLUEPRINTS TO AND FROM JSON STRINGS. USE THIS TO BACKUP/RESTORE BLUEPRINTS OR WHEN MIGRATING BLUEPRINTS TO A NEW PROJECT ETC.



PLANET BLUEPRINT

1. HIGHLIGHT A CHILD OBJECT OF THE MANAGER IN THE HIERARCHY

2. SEE THE INSPECTOR

THE **CREATE PLANET** BUTTON CREATES A PLANET IN THE SCENE USING THIS BLUEPRINT AND WITH RANDOM PROPERTIES.

THE **EXPORT/IMPORT BLUEPRINT TO/FROM CLIPBOARD (JSON)** BUTTONS EXPORTS AND IMPORTS BLUEPRINTS TO/FROM THE CLIPBOARD. USE THIS TO BACKUP/RESTORE BLUEPRINTS OR TO MIGRATE BLUEPRINTS TO OTHER PROJECTS. YOU CAN STORE EXPORTED STRINGS IN A TEXT FILE.

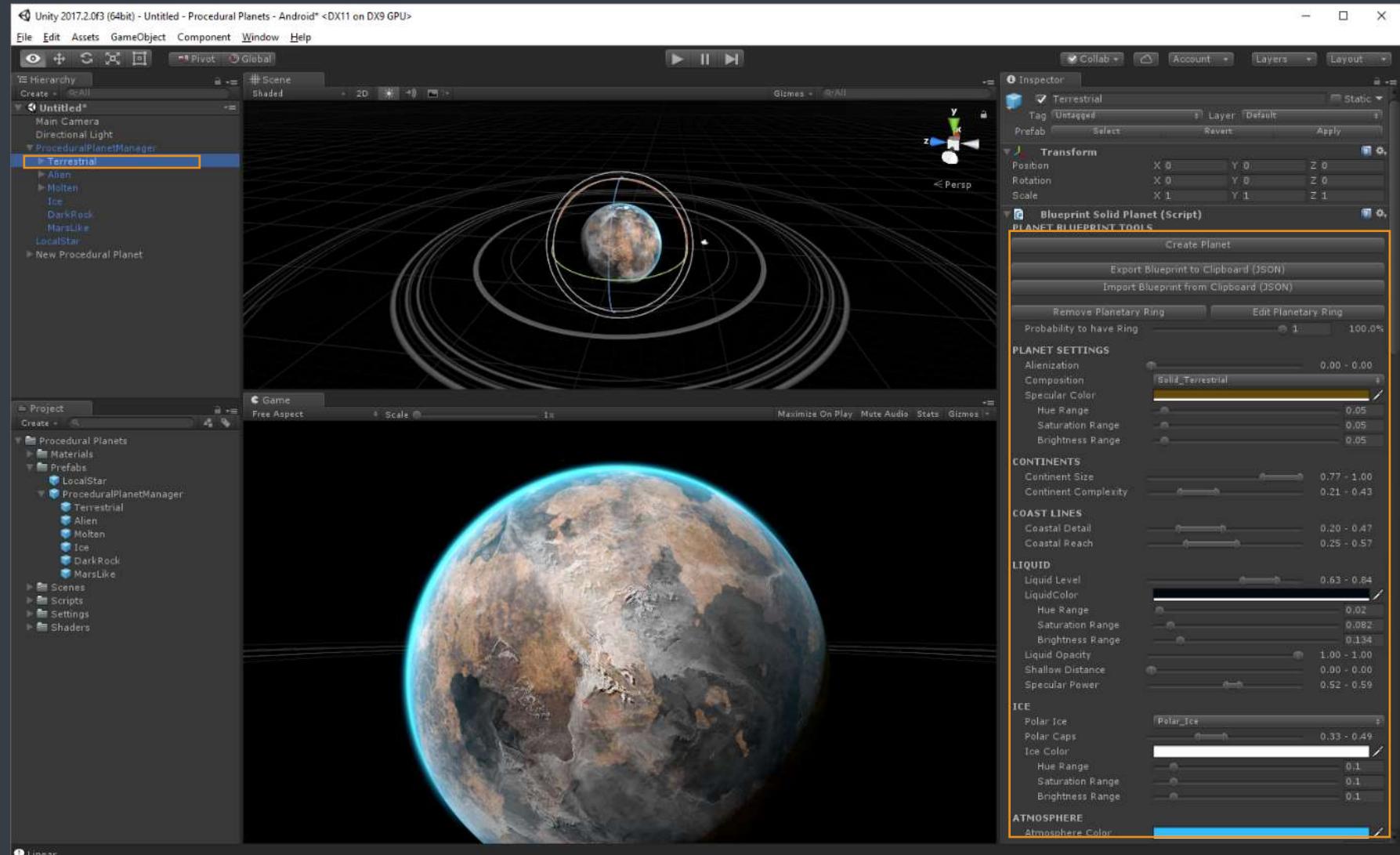
YOU CAN USE THE **CREATE PLANETARY RING** BUTTON TO ADD A RING TO THIS PLANET BLUEPRINT. ONCE ADDED YOU CAN USE BUTTONS TO REMOVE OR EDIT THE PLANETARY RING BLUEPRINT.

THE **PROBABILITY TO HAVE RING** SLIDER DICTATES HOW LIKELY A PLANET USING THIS BLUEPRINT WILL HAVE TO ACTUALLY HAVE A RING INSTANTIATED.

FLOAT PROPERTIES HAVE MIN/MAX SLIDERS WHICH ALLOWS YOU TO SET A RANGE FOR THE RANDOM VALUES TO BE WITHIN. YOU CAN SET SLIDERS WHERE BOTH MIN/MAX ARE ZERO TO FORCE IT TO ZERO, E.G. IF YOU DON'T WANT TO HAVE ANY LIQUID OR LAVA.

COLORS HAVE A BASE COLOR AND SLIDERS TO SET MAXIMUM DEVIATION FROM THE COLOR IN TERMS OF HUE, SATURATION, AND BRIGHTNESS VALUES.

MATERIALS HAVE A MASK THAT ALLOWS YOU TO RESTRICT RANDOM SELECTION TO A PREDETERMINED SELECTION OF MATERIALS. E.G. TO REMOVE DUST FROM TERRESTRIAL PLANETS.



TIP: IT MIGHT BE DIFFICULT TO CREATE BLUEPRINTS BEFORE YOU KNOW WHAT ALL THE PROPERTY SLIDERS DO AND WHAT IMPACT THE VALUES HAVE OF THE APPEARANCE OF A PLANET. I RECOMMEND THAT YOU PLAY AROUND WITH SLIDERS ON AN INSTANTIATED PLANET IN A SCENE TO SEE WHAT EFFECT EACH SLIDER HAS SO YOU CAN DETERMINE APPROPRIATE RANGES TO CREATE A CUSTOM BLUEPRINT.

PROCEDURAL MATERIALS OVERVIEW

PLANETS USE PROCEDURAL MATERIALS TO GENERATE ALL TEXTURES. UNITY USES SO CALLED "SUBSTANCES" FROM "ALGORITHMIC" AND IN THE PROCEDURAL PLANETS ASSET THERE ARE A NUMBER OF PROCEDURAL MATERIALS FOR PLANET COMPOSITION, SURFACE ("BIOME") TEXTURES, LAVA, CLOUDS, ICE, AND NIGHT CITY LIGHTS.

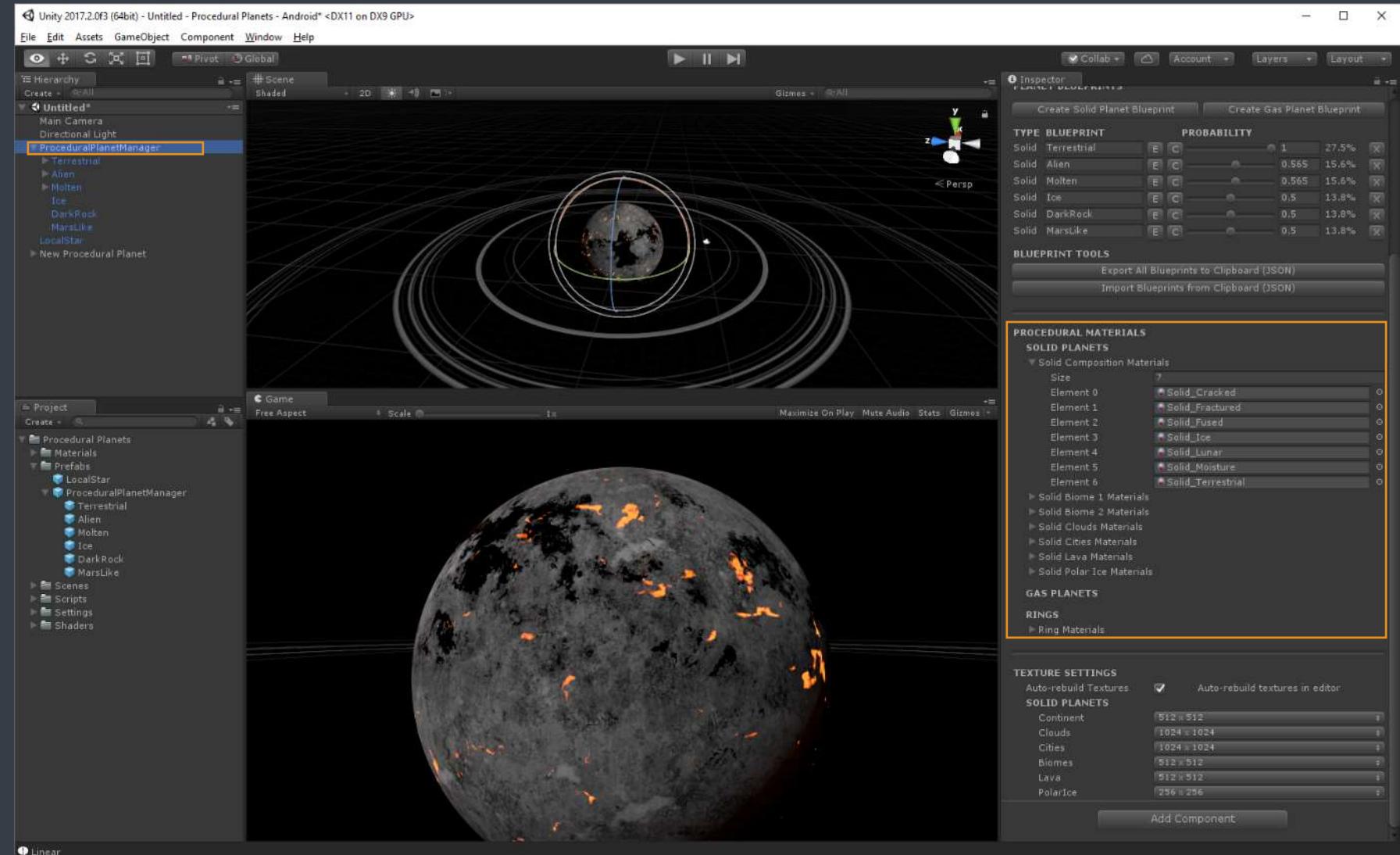
IN THE MANAGER ARRAYS OF MATERIALS HAVE BEEN CONFIGURED. THESE ARRAYS CAN ALSO BE EXPANDED IN THE FUTURE FOR ADDITIONAL MATERIALS.

YOU WILL NOT NEED TO MAKE ANY CHANGES HERE, JUST BE AWARE OF THEIR EXISTENCE. PLATFORM SUPPORT, PERFORMANCE, AND LIMITATIONS ARE DEPENDENT ON HOW UNITY HANDLES PROCEDURAL MATERIALS.

BE AWARE THAT IF ANY CHANGES ARE MADE HERE PREVIOUSLY CREATED PLANETS WILL NOT USE THE SAME TEXTURES SINCE THE RANDOM SEED WILL POINT TO A DIFFERENT TEXTURE.

THESE ARE THE "ONLY" INSTANCES OF MATERIALS THAT EXIST SO WHEN A PLANET MODIFIES A TEXTURE THESE ARE THE PROCEDURAL MATERIALS THAT ARE UPDATED AND A COPY OF THE TEXTURE IS SENT TO THE PLANET.

NOTE: THERE ARE NO GAS PLANET MATERIALS IN THIS RELEASE.



TEXTURE SETTINGS & RESOLUTIONS

TEXTURES RESOLUTIONS CAN BE SET INDEPENDENTLY.

FOR PLANETS THAT ARE VERY SMALL ON SCREEN, AND WITHOUT CLOSEUPS, YOU MAY GET AWAY WITH FAIRLY LOW TEXTURE RESOLUTIONS SUCH AS 128x128 – 512x512.

IF YOU NEED HIGHLY DETAILED PLANETS FOR LARGE SCREENS WITH HIGH DEFINITION AND OR FOR CLOSEUPS OF PLANETS YOU MAY WANT TO CONSIDER 1024x1024 – 2048x2048 TEXTURES.

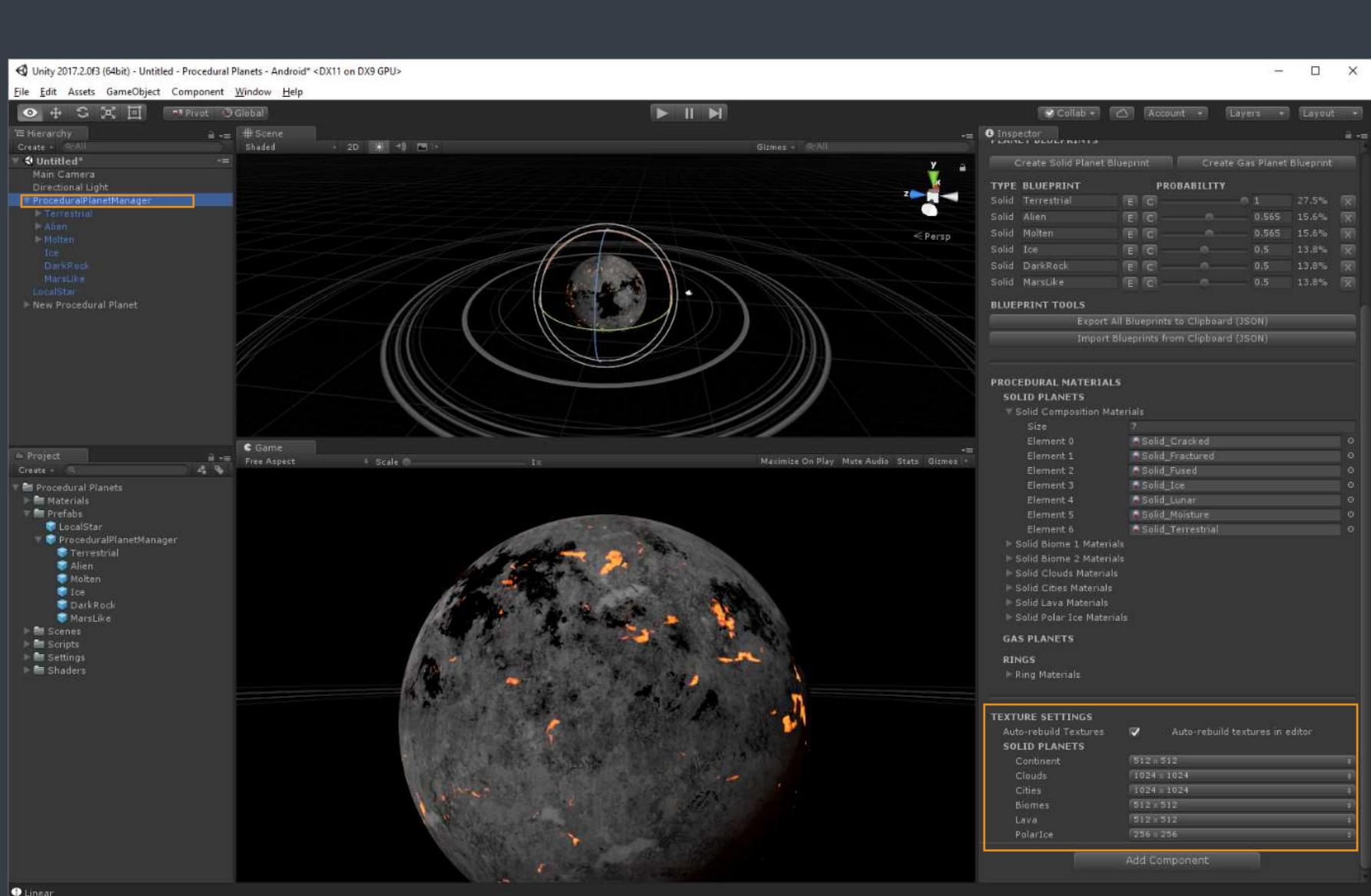
PLANETS USE 7 TEXTURES:

- COMPOSITION/CONTINENT
- CLOUDS
- CITIES
- BIOMES 1-2
- LAVA
- POLAR ICE

GENERALLY YOU WANT TO KEEP COMPOSITION/CONTINENT AND CLOUDS TEXTURE RESOLUTIONS HIGHER BECAUSE THEY TILE FEWER TIMES AROUND THE PLANET.

WHEN RESOLUTION SETTINGS ARE CHANGED TEXTURES WILL BE REBUILT AUTOMATICALLY.

NOTE: TEXTURES ARE CROSSFADED AND BLENDED BY THE SHADER SO THE VIRTUAL RESOLUTION COMPARED TO A SINGLE SPHERICAL TEXTURE IS MUCH HIGHER.



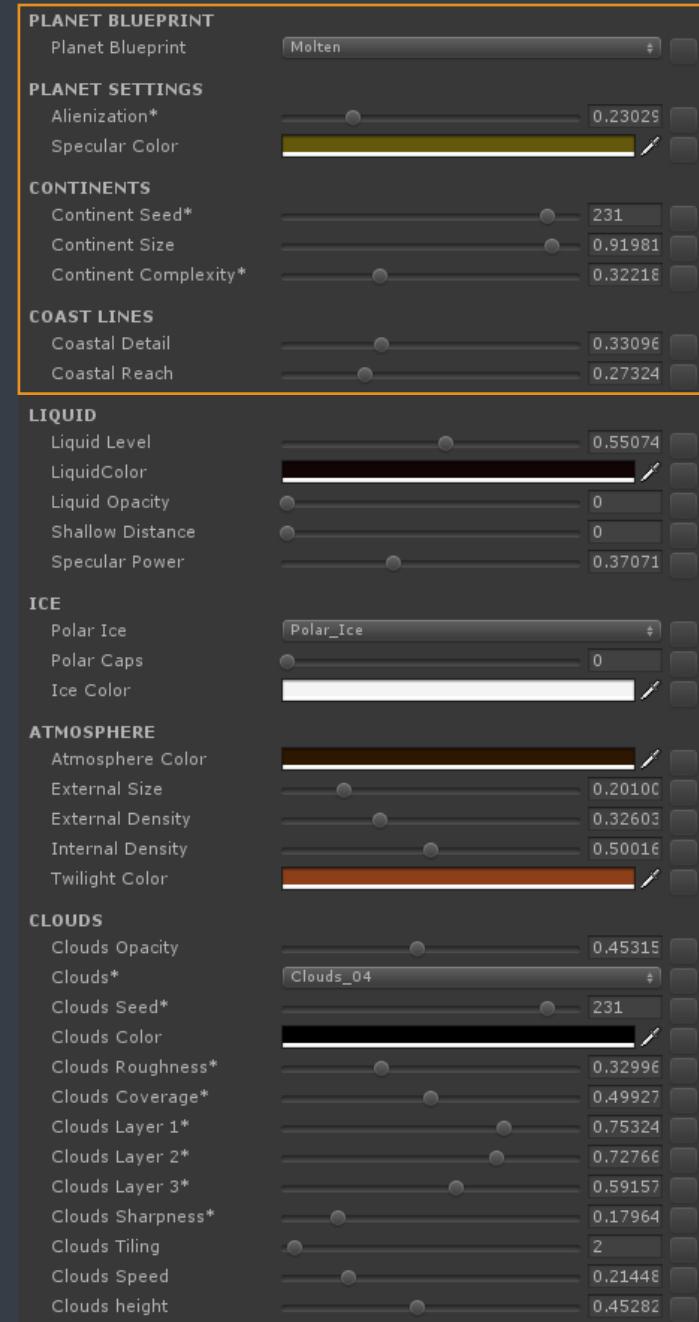
PLANET PROPERTIES

PAGE 1/7

PROPERTIES MARKED “(SHADER ONLY)” CAN BE CHANGED WITHOUT ANY PERFORMANCE IMPACT AND REQUIRES NO TEXTURE REBUILD. AN ASTERISK (*) INDICATES THAT A LOOKUP TEXTURE NEEDS TO BE REGENERATED BUT THAT IS A FAIRLY LOW COST OPERATION.

PROPERTIES MARKED “(REBUILDS XYZ TEXTURE)” REQUIRES ONE OR MORE PROCEDURAL TEXTURES TO BE REBUILT WHICH IS EXPENSIVE FROM A PERFORMANCE PERSPECTIVE. CHANGING JUST ONE SINGLE PROCEDURAL TEXTURE ON A SINGLE PLANET AT LOWER RESOLUTIONS CAN BE DONE IN NEAR REAL-TIME ON HIGH END HARDWARE.

- **PLANET BLUEPRINT** (REBUILDS EVERYTHING)
 - THE BLUEPRINT A PLANET USES TO GET VALID RANGES FOR RANDOM PROPERTIES. A BLUEPRINT DOES NOT RESTRICT YOU FROM OVERRIDING PROPERTIES TO ANY EXTENT.
- **ALIENIZATION** (REBUILDS BIOME 1 & BIOME 2 TEXTURES)
 - A HUE SHIFTER AND THE HIGHER IT IS THE MORE DIFFERENT THE COLORS ARE. E.G. A FOREST IS GREEN BUT USING THE ALIENIZATION SLIDER YOU CAN MAKE IT ORANGE AND PURPLE.
- **SPECULAR COLOR** (SHADER ONLY)
 - THE COLOR TINT OF SPECULAR REFLECTION ON LIQUID.
- **CONTINENT SEED** (REBUILDS COMPOSITION TEXTURE)
 - THE RANDOM SEED USED FOR CONTINENT FORMATION. CHANGE THIS IF YOU WANT TO HAVE A PLANET WITH DIFFERENT SHORE LINE AND CONTINENT FORMATION.
- **CONTINENT SIZE** (SHADER ONLY)
 - SETS THE SIZE OF CONTINENTS WHERE 0 IS SMALL AND 1 IS LARGE.
- **CONTINENT COMPLEXITY** (REBUILDS COMPOSITION TEXTURE)
 - WARPS THE COMPOSITION MAP TO CREATE MORE COMPLEX FEATURES AND COAST LINES.
- **COASTAL DETAIL** (SHADER ONLY)
 - SETS THE SMOOTHNESS AND COMPLEXITY OF SHORES AND COASTLINES. 0 IS SMOOTH AND 1 IS DETAILED AND COMPLEX.
- **COASTAL REACH** (SHADER ONLY)
 - AFFECTS THE ARCHIPELAGO, I.E. NUMBER OF LITTLE ISLANDS AROUND COASTLINES.



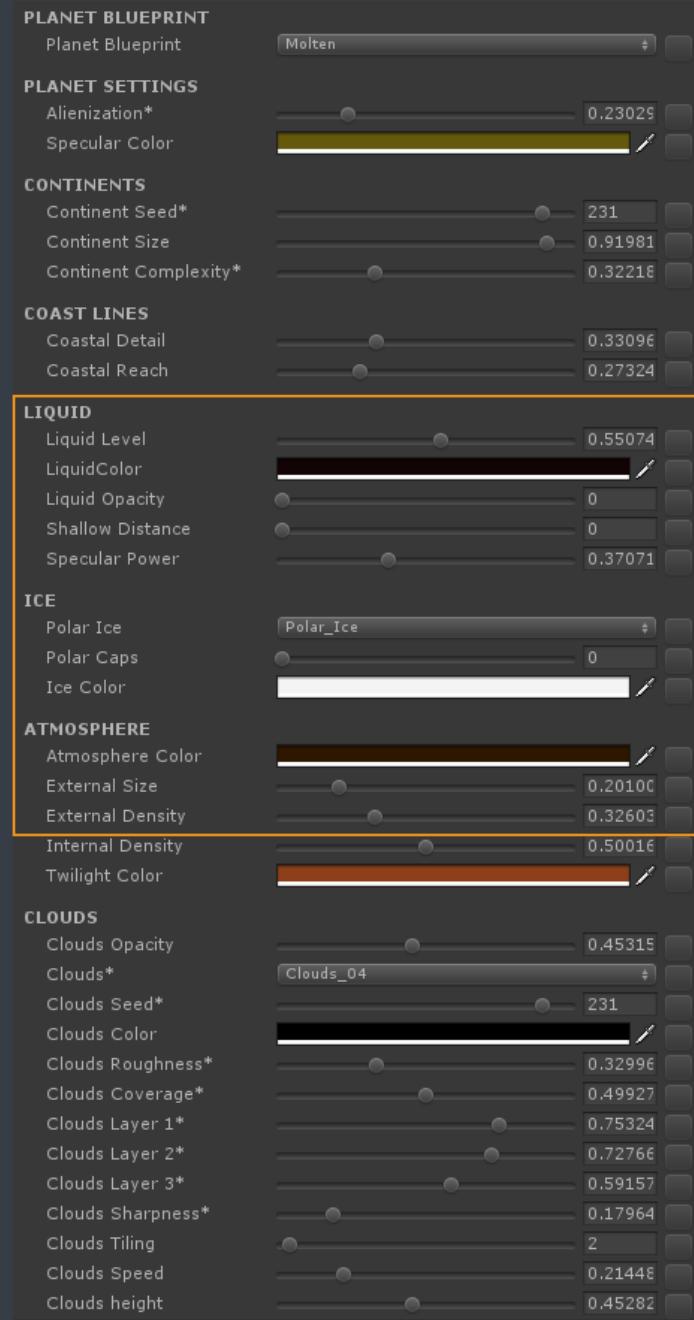
PLANET PROPERTIES

PAGE 2/7

PROPERTIES MARKED “(SHADER ONLY)” CAN BE CHANGED WITHOUT ANY PERFORMANCE IMPACT AND REQUIRES NO TEXTURE REBUILD. AN ASTERISK (*) INDICATES THAT A LOOKUP TEXTURE NEEDS TO BE REGENERATED BUT THAT IS A FAIRLY LOW COST OPERATION.

PROPERTIES MARKED “(REBUILDS XYZ TEXTURE)” REQUIRES ONE OR MORE PROCEDURAL TEXTURES TO BE REBUILT WHICH IS EXPENSIVE FROM A PERFORMANCE PERSPECTIVE. CHANGING JUST ONE SINGLE PROCEDURAL TEXTURE ON A SINGLE PLANET AT LOWER RESOLUTIONS CAN BE DONE IN NEAR REAL-TIME ON HIGH END HARDWARE.

- **LIQUID LEVEL (SHADER ONLY*)**
 - LIQUID / WATER COVERAGE OF PLANET. 0 = NO WATER, 1 = ONLY WATER.
- **LIQUID COLOR (SHADER ONLY)**
 - COLOR OF LIQUID / WATER. USUALLY VERY DARK BLUES LOOK GOOD.
- **LIQUID OPACITY (SHADER ONLY)**
 - TRANSPARENCY OF LIQUID / WATER. 0 = FULLY TRANSPARENT, 1 = FULLY OPAQUE. TIP: LIQUID OPACITY SET TO 0 CAN BLOCK OUT AREAS OF LAVA.
- **(LIQUID) SHALLOW DISTANCE (SHADER ONLY*)**
 - TRANSPARENCY CROSSFADE REGION BETWEEN LIQUID AND LAND. 0 = SHARP COASTLINES, 1 = LONG CROSSFADE DISTANCE.
- **(LIQUID) SPECULAR POWER (SHADER ONLY)**
 - SPECULAR REFLECTION COVERAGE. 0 = LARGE AREA, 1 = SMALL AREA.
- **POLAR ICE (REBUILDS POLAR ICE TEXTURE)**
 - MATERIAL USED FOR POLAR CAPS.
- **POLAR CAPS (SHADER ONLY*)**
 - SIZE OF POLAR CAPS, HOW FAR POLAR ICE REACHES FROM POLES TOWARDS EQUATOR. 0 = NO POLAR CAPS, 1 = COVERAGE TO POLAR CIRCLE.
- **ICE COLOR (SHADER ONLY)**
 - COLOR TO REPLACE WHERE POLAR CAPS COVER LIQUID.
- **ATMOSPHERE COLOR (SHADER ONLY)**
 - COLOR OF ATMOSPHERE.
- **(ATMOSPHERE) EXTERNAL SIZE (SHADER ONLY)**
 - SIZE OF EXTERNAL ATMOSPHERE, HOW FAR OUT IT REACHES FROM PLANET.
- **(ATMOSPHERE) EXTERNAL DENSITY (SHADER ONLY)**
 - DENSITY / THICKNESS OF EXTERNAL ATMOSPHERE.



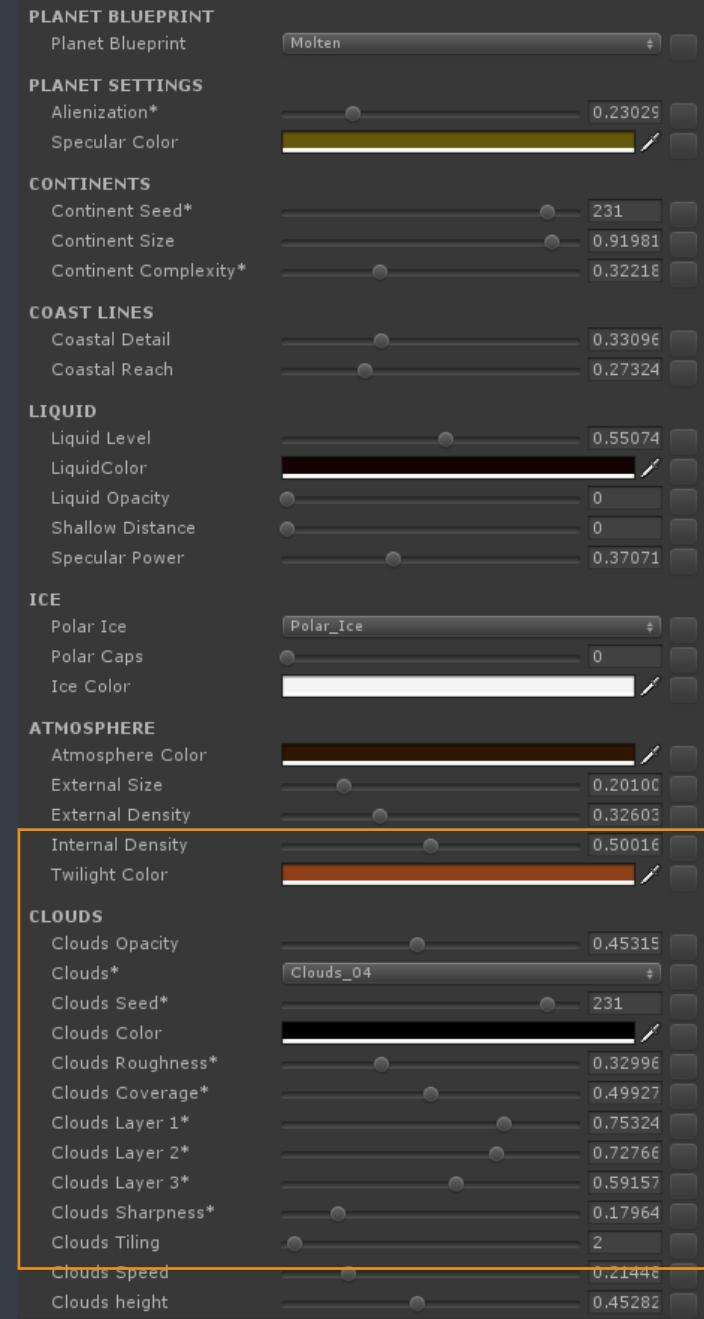
PLANET PROPERTIES

PAGE 3/7

PROPERTIES MARKED “(SHADER ONLY)” CAN BE CHANGED WITHOUT ANY PERFORMANCE IMPACT AND REQUIRES NO TEXTURE REBUILD. AN ASTERISK (*) INDICATES THAT A LOOKUP TEXTURE NEEDS TO BE REGENERATED BUT THAT IS A FAIRLY LOW COST OPERATION.

PROPERTIES MARKED “(REBUILDS XYZ TEXTURE)” REQUIRES ONE OR MORE PROCEDURAL TEXTURES TO BE REBUILT WHICH IS EXPENSIVE FROM A PERFORMANCE PERSPECTIVE. CHANGING JUST ONE SINGLE PROCEDURAL TEXTURE ON A SINGLE PLANET AT LOWER RESOLUTIONS CAN BE DONE IN NEAR REAL-TIME ON HIGH END HARDWARE.

- **(ATMOSPHERE) INTERNAL DENSITY (SHADER ONLY)**
 - THICKNESS OF ATMOSPHERE THAT COVERS THE PLANET, I.E. RIM LIGHTING.
- **TWILIGHT COLOR (SHADER ONLY)**
 - COLOR TINT OF REGION BETWEEN NIGHT AND DAY TO SIMULATE DUSK/DAWN TWILIGHT ZONES.
- **CLOUDS OPACITY (SHADER ONLY)**
 - TRANSPARENCY OF CLOUDS. 0 = INVISIBLE (AND DISABLED), 1 = FULL OPACITY. NOTE CLOUD OPACITY IS APPLIED AFTER CLOUD LAYERS AND CLOUD COVERAGE SO EVEN WITH OPACITY SET TO 1 CLOUDS CAN BE INVISIBLE IF COVERAGE OR LAYERS ARE SET LOW.
- **CLOUDS SEED (REBUILDS CLOUDS TEXTURE)**
 - RANDOM SEED USED FOR CLOUD NOISE GENERATORS.
- **CLOUDS COLOR (SHADER ONLY)**
 - COLOR OF CLOUDS.
- **CLOUDS ROUGHNESS (REBUILDS CLOUDS TEXTURE)**
 - NORMAL/BUMP MAP OF CLOUDS.
- **CLOUDS COVERAGE (REBUILDS CLOUDS TEXTURE)**
 - OPACITY OF CLOUDS AFTER CLOUDS LAYER 1 + 2 + 3 HAVE BEEN COMBINED.
- **CLOUDS LAYER 1-3 (REBUILDS CLOUDS TEXTURE)**
 - INDIVIDUAL LAYERS OF DIFFERENT STYLES OF CLOUDS. LAYER 1-3 ARE BLENDED TOGETHER AND IS THEN Affected BY CLOUDS COVERAGE AND CLOUDS OPACITY.
- **CLOUDS SHARPNESS (REBUILDS CLOUDS TEXTURE)**
 - SHARPNESS, CONTRAST, DETAIL OF CLOUDS.
- **CLOUDS TILING (SHADER ONLY)**
 - HOW MANY TIMES CLOUD TEXTURE IS TILED AROUND THE PLANET. TIP: USE 2-6 TIMES FOR PLANETS FULLY VISIBLE AND INCREASE TO HIGHER VALUES IF VIEWED FROM CLOSE DISTANCE TO AVOID REPETITION.



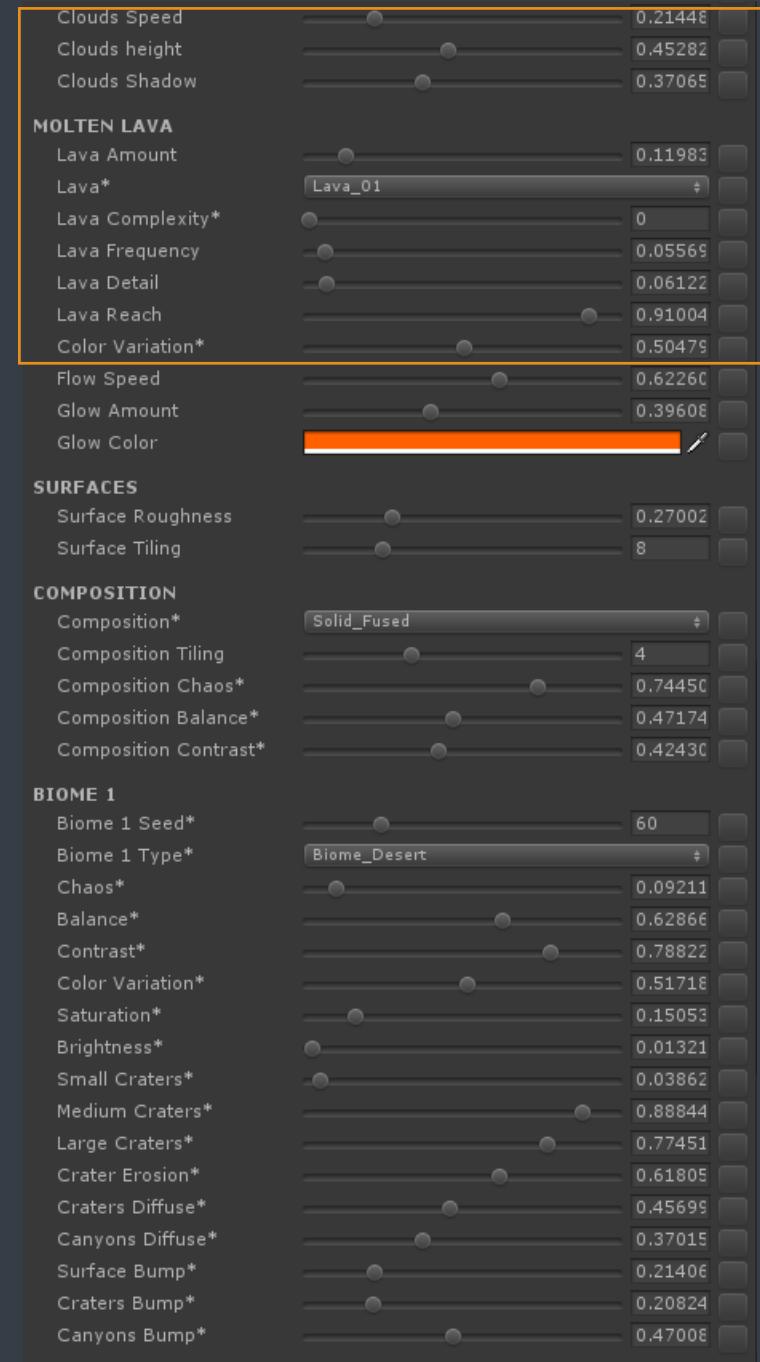
PLANET PROPERTIES

PAGE 4/7

PROPERTIES MARKED “(SHADER ONLY)” CAN BE CHANGED WITHOUT ANY PERFORMANCE IMPACT AND REQUIRES NO TEXTURE REBUILD. AN ASTERISK (*) INDICATES THAT A LOOKUP TEXTURE NEEDS TO BE REGENERATED BUT THAT IS A FAIRLY LOW COST OPERATION.

PROPERTIES MARKED “(REBUILDS XYZ TEXTURE)” REQUIRES ONE OR MORE PROCEDURAL TEXTURES TO BE REBUILT WHICH IS EXPENSIVE FROM A PERFORMANCE PERSPECTIVE. CHANGING JUST ONE SINGLE PROCEDURAL TEXTURE ON A SINGLE PLANET AT LOWER RESOLUTIONS CAN BE DONE IN NEAR REAL-TIME ON HIGH END HARDWARE.

- **CLOUDS SPEED** (SHADER ONLY)
 - HOW FAST THE CLOUDS ROTATE AROUND THE PLANET.
- **CLOUDS HEIGHT** (SHADER ONLY)
 - HOW MUCH THE CLOUD SHADOWS SHOULD BE OFFSET FROM CLOUDS.
- **CLOUDS SHADOW** (SHADER ONLY)
 - STRENGTH OF SHADOW (SHADOW DISTANCE IS AFFECTED BY CLOUD HEIGHT).
- **LAVA AMOUNT** (SHADER ONLY)
 - HOW MUCH MOLTEN LAVA SHOULD EXISTS. FOR SMALL CRACKS, USE VERY LOW NUMBERS, E.G. 0.001 – 0.01. IF YOU WANT TO BREAK UP THE PATTERN, CONSIDER USING A FULLY TRANSPARENT LIQUID WITH NO SHALLOW DISTANCE TO BLOCK AREAS WHERE LAVA WOULD OTHERWISE BE.
- **LAVA** (REBUILDS LAVA TEXTURE)
 - TEXTURE TO BE USED FOR LAVA.
- **LAVA COMPLEXITY** (REBUILDS COMPOSITION TEXTURE)
 - WARPING/COMPLEXITY MODIFIER FOR LAVA. THE COMPOSITION TEXTURE CONTAINS A CHANNEL FOR LAVA COVERAGE.
- **LAVA FREQUENCY** (SHADER ONLY)
 - AFFECTS TILING OF THE LAVA TEXTURE RESULTING IN 0 = LARGE CRACKS AND 1 = SMALL CRACKS.
- **LAVA DETAIL** (SHADER ONLY)
 - AFFECTS STRAIGHTNESS OF CRACK EDGES 0 = STRAIGHT, 1 = WARPED.
- **LAVA REACH** (SHADER ONLY)
 - AFFECTS RANGE OF EDGES WITH HARDENED ROCK ISLANDS.
- **(LAVA) COLOR VARIATION** (REBUILDS LAVA TEXTURE)
 - HUE COLOR SHIFTING OF LAVA TEXTURE. 0.5 = NO CHANGE, 0 = SHIFT HUE LEFT AND 1 = SHIFT HUE RIGHT.



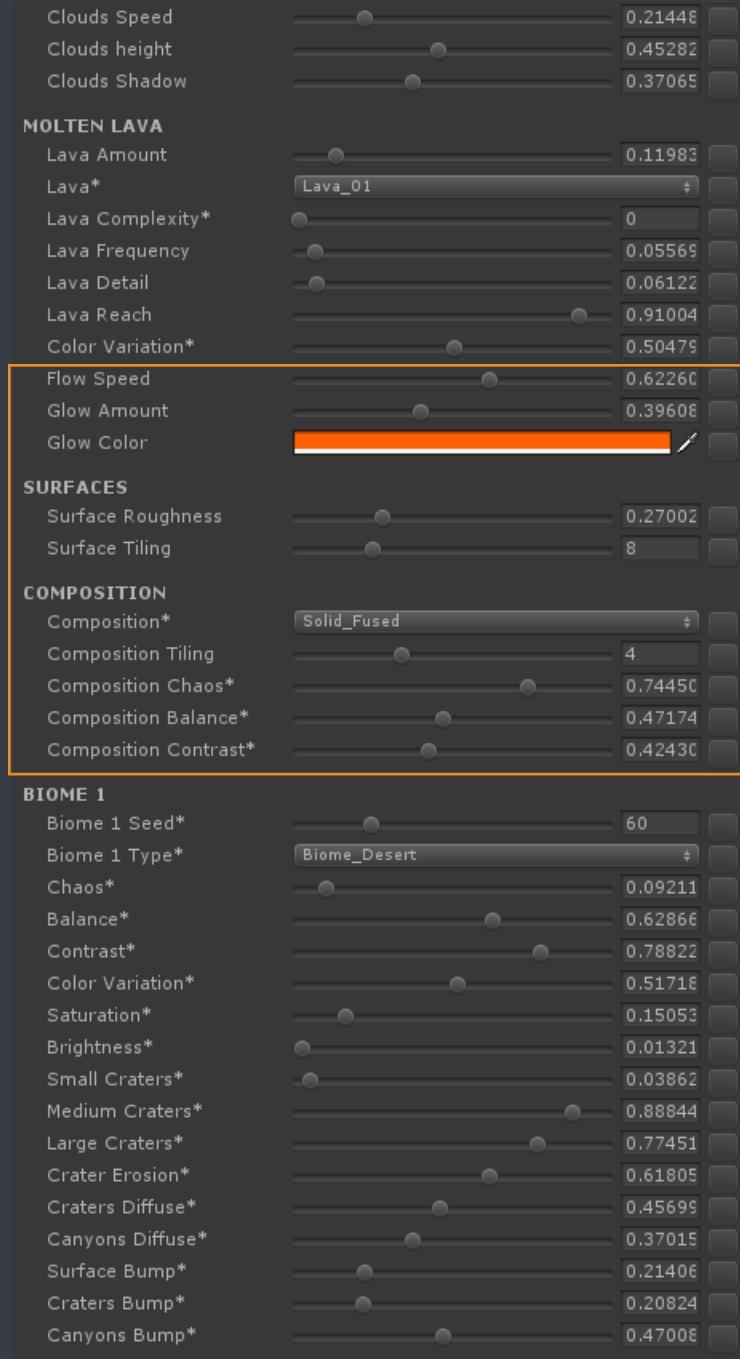
PLANET PROPERTIES

PAGE 5/7

PROPERTIES MARKED “(SHADER ONLY)” CAN BE CHANGED WITHOUT ANY PERFORMANCE IMPACT AND REQUIRES NO TEXTURE REBUILD. AN ASTERISK (*) INDICATES THAT A LOOKUP TEXTURE NEEDS TO BE REGENERATED BUT THAT IS A FAIRLY LOW COST OPERATION.

PROPERTIES MARKED “(REBUILDS XYZ TEXTURE)” REQUIRES ONE OR MORE PROCEDURAL TEXTURES TO BE REBUILT WHICH IS EXPENSIVE FROM A PERFORMANCE PERSPECTIVE. CHANGING JUST ONE SINGLE PROCEDURAL TEXTURE ON A SINGLE PLANET AT LOWER RESOLUTIONS CAN BE DONE IN NEAR REAL-TIME ON HIGH END HARDWARE.

- **(LAVA) FLOW SPEED (SHADER ONLY)**
 - HOW FAST THE LAVA ANIMATION FLOWS.
- **(LAVA) GLOW AMOUNT (SHADER ONLY)**
 - GLOWING EFFECT AROUND LAVA. 0 = NO GLOW, 1 = MAX GLOW.
- **(LAVA) GLOW COLOR (SHADER ONLY)**
 - COLOR OF LAVA GLOW EFFECT.
- **SURFACE ROUGHNESS (SHADER ONLY)**
 - NORMAL / BUMP MAP STRENGTH OF LAND AREAS. ALSO AFFECTS CLOUDS.
- **SURFACE TILING (SHADER ONLY)**
 - TILING AMOUNT OF SURFACE BIOME TEXTURES. THIS CAN OFTEN BE QUITE HIGH, 10-20, ESPECIALLY IF PLANETS ALSO HAVE LIQUID AND CLOUD COVERAGE.
- **COMPOSITION (REBUILDS COMPOSITION TEXTURE)**
 - MATERIAL TO BE USED FOR PLANET COMPOSITION, I.E. CONTINENTS AND BIOMES ARE MIXED TOGETHER.
- **COMPOSITION SEED (REBUILDS COMPOSITION TEXTURE)**
 - SEED USED FOR COMPOSITION TEXTURE THAT AFFECTS COMPOSITION MAP FOR BIOME BLENDING.
- **COMPOSITION TILING (SHADER ONLY)**
 - TILING AMOUNT OF COMPOSITION TEXTURE. FOR PLANETS VIEWED FROM A DISTANCE, USE A LOW VALUE, 2-5 AND FOR CLOSE DISTANCE WHERE YOU SEE PARTS OF PLANETS YOU CAN HAVE HIGHER TILING VALUES.
- **COMPOSITION CHAOS (REBUILDS COMPOSITION TEXTURE)**
 - CHAOS / WARP EFFECT OF COMPOSITION TEXTURE. AFFECTS HOW BIOME BLENDING EDGES APPEAR.
- **COMPOSITION BALANCE (REBUILDS COMPOSITION TEXTURE)**
 - BALANCE BETWEEN BIOME 1 AND BIOME 2 SURFACE TEXTURE WHERE 0.5 IS A 50/50 MIX, 0 = ONLY BIOME 1 AND 2 = ONLY BIOME 2.
- **COMPOSITION CONTRAST (REBUILDS COMPOSITION TEXTURE)**
 - CONTRAST / CROSSFADE BETWEEN BIOME 1 AND BIOME 2 SURFACE MATERIALS. 0 = SHARP TRANSITION, 1 = HEAVILY CROSSFADED TRANSITION.



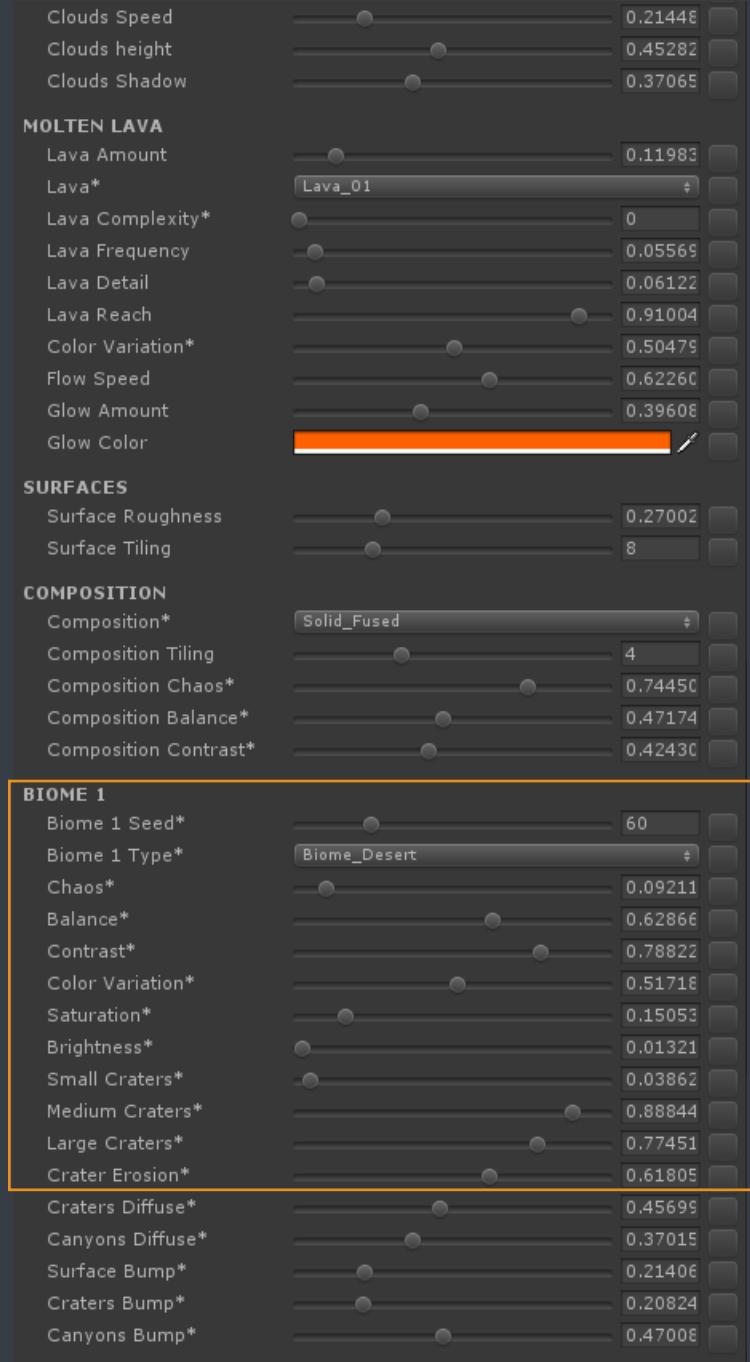
PLANET PROPERTIES

PAGE 6/7

PROPERTIES MARKED “(SHADER ONLY)” CAN BE CHANGED WITHOUT ANY PERFORMANCE IMPACT AND REQUIRES NO TEXTURE REBUILD. AN ASTERISK (*) INDICATES THAT A LOOKUP TEXTURE NEEDS TO BE REGENERATED BUT THAT IS A FAIRLY LOW COST OPERATION.

PROPERTIES MARKED “(REBUILDS XYZ TEXTURE)” REQUIRES ONE OR MORE PROCEDURAL TEXTURES TO BE REBUILT WHICH IS EXPENSIVE FROM A PERFORMANCE PERSPECTIVE. CHANGING JUST ONE SINGLE PROCEDURAL TEXTURE ON A SINGLE PLANET AT LOWER RESOLUTIONS CAN BE DONE IN NEAR REAL-TIME ON HIGH END HARDWARE.

- **BIOME 1-2 SEED** (REBUILDS BIOME 1-2 TEXTURES)
 - RANDOM SEED USED FOR TEXTURE GENERATION.
- **BIOME 1-2 TYPE** (REBUILDS BIOME 1-2 TEXTURES)
 - MATERIAL USED FOR BIOME 1-2.
- **BIOME 1-2 CHAOS** (REBUILDS BIOME 1-2 TEXTURES)
 - CHAOS / WARP EFFECT OF BIOME TEXTURE.
- **BIOME 1-2 BALANCE** (REBUILDS BIOME 1-2 TEXTURES)
 - BIOME TEXTURES HAVE TWO MAIN TEXTURES INTERNALLY – THE BALANCE BLENDS BETWEEN THE TWO.
- **BIOME 1-2 CONTRAST** (REBUILDS BIOME 1-2 TEXTURES)
 - BIOME TEXTURES HAVE TWO MAIN TEXTURES INTERNALLY, CONTRAST DICTATES FADING BETWEEN THE TWO TEXTURES WHERE 0 = SHARP AND 1 = HEAVILY CROSSFADED.
- **BIOME 1-2 COLOR VARIATION** (REBUILDS BIOME 1-2 TEXTURES)
 - SLIGHT HUE SHIFTING OF BIOME TEXTURE WHERE 0.5 IS ORIGINAL COLOR AND <0.5 SHIFTS COLOR HUE LEFT AND >0.5 SHIFTS COLOR HUE RIGHT.
- **BIOME 1-2 SATURATION** (REBUILDS BIOME 1-2 TEXTURES)
 - SATURATION OF BIOME TEXTURE WHERE 0.5 IS ORIGINAL SATURATION VALUE AND <0.5 DESATURATES TEXTURE AND >0.5 INCREASES SATURATION.
- **BIOME 1-2 BRIGHTNESS** (REBUILDS BIOME 1-2 TEXTURES)
 - BRIGHTNESS OF BIOME TEXTURE WHERE 0.5 IS ORIGINAL BRIGHTNESS AND <0.5 DARKENS TEXTURE AND >0.5 BRIGHTENS TEXTURE.
- **BIOME 1-2 SMALL/MEDIUM/LARGE CRATERS** (REBUILDS BIOME 1-2 TEXTURES)
 - STRENGTH OF DIFFERENT SIZED CRATERS ON THE BIOME TEXTURE.
- **BIOME 1-2 CRATER EROSION** (REBUILDS BIOME 1-2 TEXTURES)
 - EROSION STRENGTH OF CRATERS WHERE 0 = NO EROSION AND 1 = VERY ERODED.



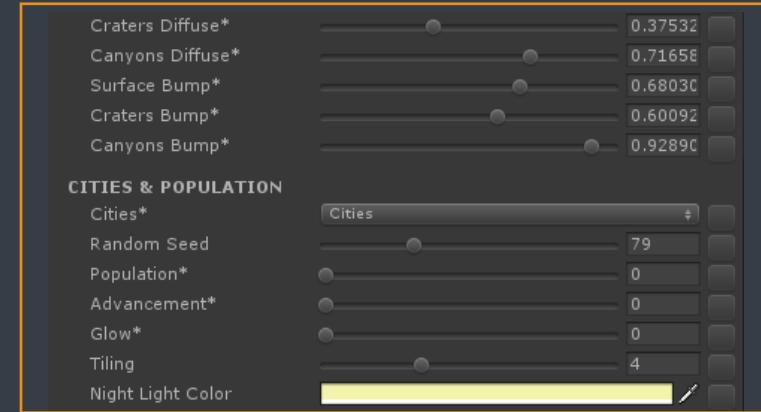
PLANET PROPERTIES

PAGE 7/7

PROPERTIES MARKED “(SHADER ONLY)” CAN BE CHANGED WITHOUT ANY PERFORMANCE IMPACT AND REQUIRES NO TEXTURE REBUILD. AN ASTERISK (*) INDICATES THAT A LOOKUP TEXTURE NEEDS TO BE REGENERATED BUT THAT IS A FAIRLY LOW COST OPERATION.

PROPERTIES MARKED “(REBUILDS XYZ TEXTURE)” REQUIRES ONE OR MORE PROCEDURAL TEXTURES TO BE REBUILT WHICH IS EXPENSIVE FROM A PERFORMANCE PERSPECTIVE. CHANGING JUST ONE SINGLE PROCEDURAL TEXTURE ON A SINGLE PLANET AT LOWER RESOLUTIONS CAN BE DONE IN NEAR REAL-TIME ON HIGH END HARDWARE.

- **BIOME 1-2 CRATERS DIFFUSE** (REBUILDS BIOME 1-2 TEXTURES)
 - COLOR VARIATION OF CRATERS WHERE 0 = NO CHANGE IN COLOR.
- **BIOME 1-2 CANYONS DIFFUSE** (REBUILDS BIOME 1-2 TEXTURES)
 - COLOR VARIATION OF CANYONS WHERE 0 = NO CHANGE IN COLOR.
- **BIOME 1-2 SURFACE BUMP** (REBUILDS BIOME 1-2 TEXTURES)
 - NORMAL / BUMP MAP STRENGTH OF BIOME / SURFACE TEXTURES.
- **BIOME 1-2 CRATERS BUMP** (REBUILDS BIOME 1-2 TEXTURES)
 - NORMAL / BUMP MAP STRENGTH OF CRATERS.
- **BIOME 1-2 CANYONS BUMP** (REBUILDS BIOME 1-2 TEXTURES)
 - NORMAL / BUMP MAP STRENGTH OF CANYONS.
- **CITIES** (REBUILDS CITY TEXTURES)
 - MATERIAL USED FOR NIGHT CITY LIGHTS.
- **(CITIES) RANDOM SEED** (REBUILDS CITIES TEXTURES)
 - RANDOM SEED FOR CITY TEXTURE GENERATOR.
- **(CITIES) POPULATION** (REBUILDS CITIES TEXTURE)
 - NUMBER OF CITY LIGHTS ON THE NIGHT SIDE OF A PLANET.
- **(CITIES) ADVANCEMENT** (REBUILDS CITIES TEXTURE)
 - STRENGTH OF “ADVANCED CIVILIZATIONS” LARGE CITIES.
- **(CITIES) GLOW** (REBUILDS CITIES TEXTURES)
 - GLOW STRENGTH OF CITY LIGHTS.
- **(CITIES) TILING** (SHADER ONLY)
 - AMOUNT OF TILING OF CITY TEXTURE. USUALLY LOWER VALUES 2-6.
- **NIGHT LIGHT COLOR** (SHADER ONLY)
 - COLOR OF NIGHT CITY LIGHTS.



ANTICIPATED QUESTIONS

CREATING PLANETS

HOW DO I CREATE A RANDOM PLANET?

- **IN THE INSPECTOR**
 - HIGHLIGHT THE MANAGER IN THE HIERARCHY, AND CLICK THE “CREATE RANDOM PLANET” BUTTON
- **IN SCRIPTS (USING PROCEDURALPLANETS NAMESPACE)**

```
ProceduralPlanets.Manager.Instance.CreatePlanet(<Vector3 position>);
```

HOW DO I CREATE A PLANET OF A SPECIFIC TYPE/BLUEPRINT

- **IN THE INSPECTOR**
 - HIGHLIGHT THE MANAGER IN THE HIERARCHY, AND CLICK THE [C] BUTTON FOR A SPECIFIC PLANET BLUEPRINT.
 - OR
 - HIGHLIGHT A PLANET BLUEPRINT UNDER THE MANAGER IN THE HIERARCHY AND CLICK BUTTON “CREATE PLANET”
- **IN SCRIPTS (USING PROCEDURALPLANETS NAMESPACE):**

```
ProceduralPlanets.Manager.Instance.CreatePlanet(<Vector3 position>, <integer seed>, <blueprint name>);
```

- TIP: YOU CAN USE -1 AS SEED FOR A RANDOM SEED.

HOW DO I CREATE A SPECIFIC (EXPORTED) PLANET IN A SCRIPT?

```
ProceduralPlanets.Manager.Instance.CreatePlanet(<Vector3 position>, <JSON string>);
```

- IF YOU CUSTOMIZE A PLANET IN THE INSPECTOR AND OVERRIDE A NUMBER OF PARAMETERS YOU CAN CLICK ON THE “EXPORT TO JSON” BUTTONS TO EXPORT THAT EXACT PLANET CONFIGURATION.
- THE EXPORTED STRING CAN BE USED IN PLACE OF <JSON STRING> IN THE LINE OF CODE EXAMPLE ABOVE. IT’S BEST TO USE EITHER THE “BASE64” OR “ESCAPED” EXPORTED JSON STRING FORMATS IF YOU PLAN TO INSTANTIATE THEM VIA SCRIPT (OR VIA TEXT FILES OR WEB URLs) BECAUSE THOSE FORMATS DO NOT CONTAIN CHARACTERS SUCH AS BACKSLASH AND QUOTATION MARKS THAT WOULD BREAK THE STRING HANDLING IN SCRIPTS.

EXAMPLE (BASE64 ENCODED JSON):

```
_planet = ProceduralPlanets.Manager.Instance.CreatePlanet(Vector3.zero,  
"eyJjYXRlZ29yeSI6InBsYW5ldCIisInR5cGUiOiJTb2xpZFBsYW5ldCIisInZlcnNpb24iOiIwLjE..");
```

NOTE: THE STRING IS IN REALITY 4000+ CHARACTERS LONG SINCE IT CONTAINS ALL PROPERTY VALUES INCLUDING THOSE THAT ARE NOT OVERRIDDEN IF RANDOM STATE CHANGES IN THE CONFIGURATION.

ANTICIPATED QUESTIONS

OVERRIDING PROPERTIES FROM SCRIPTS

HOW DO I OVERRIDE A PROPERTY FROM A SCRIPT?

- GET THE COMPONENT OF THE PLANET AND USE THE OVERRIDEPROPERTYFLOAT() METHOD TO OVERRIDE A PROPERTY WITH A NEW FLOAT. THE PROPERTIES USE “CAMELCASE” WITH A SMALL INITIAL LETTER AND CAPITALIZED LETTERS FOR FOLLOWING WORDS WITHOUT SPACES.
- EXAMPLE:

```
// You need to have a reference to the Planet component:  
ProceduralPlanets.Planet _planet;  
// You can get the component when a planet is created (or use GetComponent at a later stage)  
_planet = ProceduralPlanets.Manager.Instance.CreatePlanet(Vector3.zero, -1, "Terrestrial");  
_planet.OverridePropertyFloat("liquidLevel", 0.5f);  
_planet.OverridePropertyFloat("cloudOpacity", 0f);  
// You can also get a value of a property  
_planet.GetPropertyFloat("liquidLevel");  
_planet.OverridePropertyFloat("liquidLevel", _planet.GetPropertyFloat("liquidLevel") + 0.1f);
```

SEE PROCEDURALPLANETS_PROPERTIES.PDF FOR ALL PARAMETER CAMELCASE NAMES AND DETAILS.

ANTICIPATED QUESTIONS

FINDING OUT WHEN A PLANET HAS FINISHED BUILDING TEXTURES

HOW DO I FIND OUT WHEN A PLANET HAS FINISHED GENERATING TEXTURES IN A SCRIPT?

- YOU CAN SUBSCRIBE TO A UNITY SENDMESSAGE EVENT TO RECEIVE A MESSAGE EVERY TIME A PLANET HAS FINISHED BUILDING TEXTURES.
- YOU CALL THE METHOD **SUBSCRIBEMESSAGEONTUREBUILDCOMPLETE(GAMEOBJECT <RECEIVER>)** OF THE PLANET.
- ONCE A PLANET HAS FINISHED BUILDING TEXTURES, IT WILL CALL THE **ONTEXTUIREBUILDCOMPLETE(FLOAT _TIME)** METHOD VIA SENDMESSAGE. THE RECEIVING GAME OBJECT MUST HAVE THE METHOD DEFINED FOR IT TO BE CALLED BUT THERE WILL BE NO ERROR IF THE METHOD DOES NOT EXIST.
- YOU DO NOT NEED TO UNSUBSCRIBE TO THE EVENT – IF THE GAME OBJECT NO LONGER EXISTS IT WILL BE REMOVED AS A SUBSCRIBER BY THE PLANET NEXT TIME IT HAS REBUILT ANY TEXTURES.

EXAMPLE:

```
void Start()
{
    // You can get the component when a planet is created (or use GetComponent at a later stage)
    ProceduralPlanets.Planet _planet;
    _planet = ProceduralPlanets.Manager.Instance.CreatePlanet(Vector3.zero, -1, "Terrestrial");
    // You subscribe to a SendMessage when a planet has finished generating textures
    _planet.SubscribeMessageOnTextureBuildComplete(gameObject);
}

void OnTextureBuildComplete(float _time)
{
    // _time argument contains number of seconds it took to generate all textures.
    Debug.Log("Planet done regenerating textures (" + _time + " seconds)");
    // Do something here, e.g. show planet if you hid it at time of creation.
}
```