

Low Poly Modular Terrain Pack v1.3



lmhpoly.com

Contacts

E-mail: justinas@lmhpoly.com

Website: <http://lmhpoly.com/contact/>

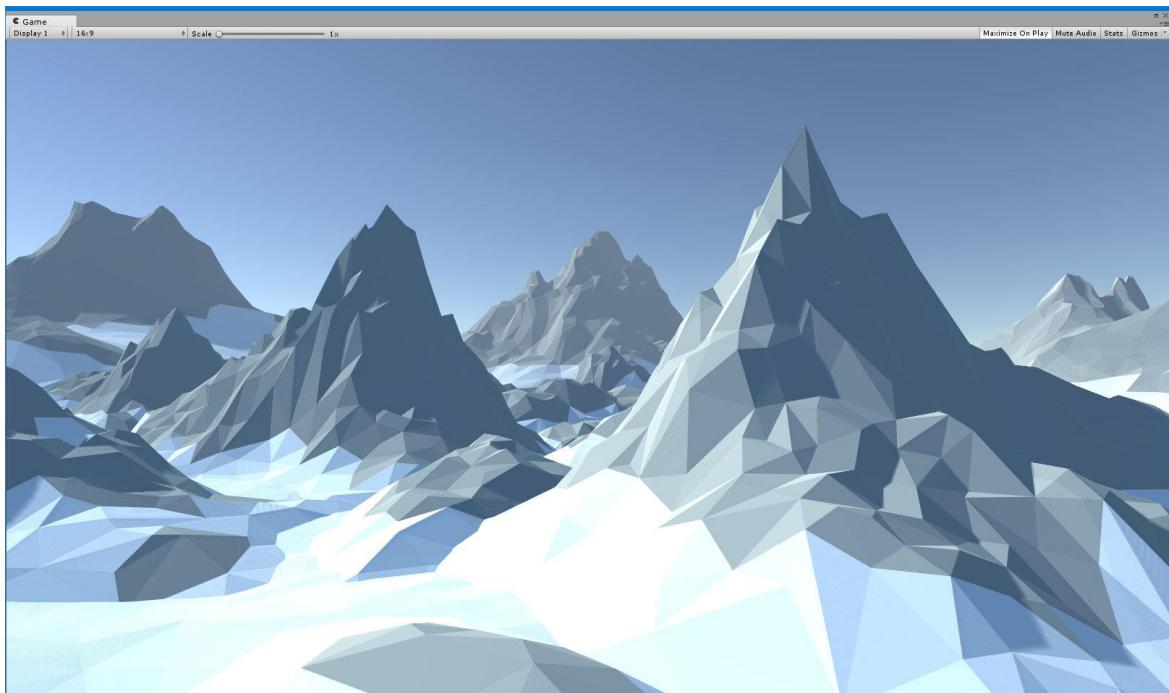
Follow me on **Twitter** to see what I'm working on right now:

<https://twitter.com/lmhpoly>

HOW TO SETUP DEMO SCENES (POST-PROCESSING) IN UNITY 2018.1 AND UP (FOR PC)	5
HOW TO SETUP DEMO SCENES IN UNITY 2018.1 AND UP (FOR ANDROID)	14
UNITY 2019.3 AND UP - UNIVERSAL RENDER PIPELINE (URP).....	23
UNITY 2018.1 AND UP - LIGHTWEIGHT RENDER PIPELINE (LWRP).....	27
UNITY 2018.1 AND UP - HIGH DEFINITION RENDER PIPELINE (HDRP).....	31
HOW TO USE "LOW POLY MODULAR TERRAIN PACK"	40
HOW TO CHANGE PREFAB COLOR / TEXTURE	46
CPT TERRAIN, MOUNTAINS, ISLANDS, RIVER.....	46
MT TERRAIN, MOUNTAINS, ISLANDS, RIVER.....	48
MT TERRAIN TEXTURE.....	49
U TERRAIN.....	51
HOW TO USE CPT TERRAIN TRANSITION PARTS	52
CHANGE TRANSITION TERRAIN COLOR	52
HOW TO PAINT VERTEX COLOR AND TEXTURES ON MT TERRAIN USING POLYBRUSH	55
ADDITIONAL INFO	59
NAMING CONVENTIONS.....	59
SCRIPTS	60
CONTACTS.....	61

Demo scenes

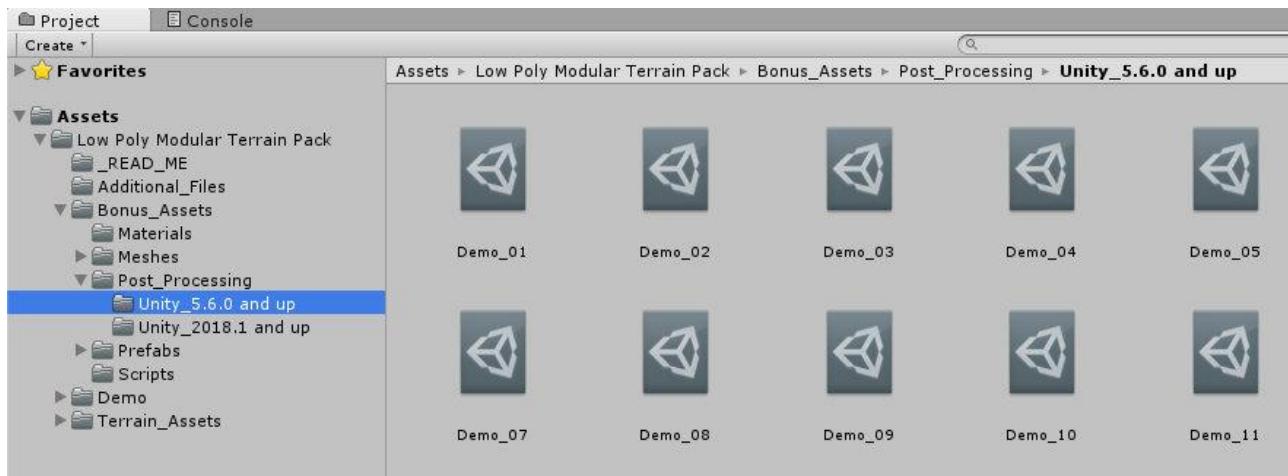
Now, as you have imported the whole “**Low Poly Modular Terrain Pack**” to your Unity project, go to *Low Poly Modular Terrain Pack > Demo > Demo_Scenes* - and open any Demo Scene (here is a **Demo_04** example). By default, the scene should look like this inside the **Game** view without any image effects applied. Scene by default, use **Gamma** Color Space.



To make it look like this:



you need to use **Post-Processing Profile** on each demo scene.



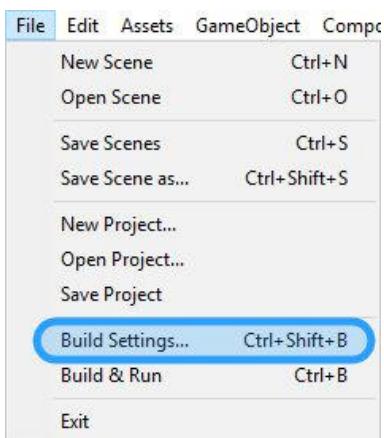
Follow the steps below to setup **Post-Processing** image effects for Demo Scenes!

- **Post-Processing in Unity 2018.1 and up**

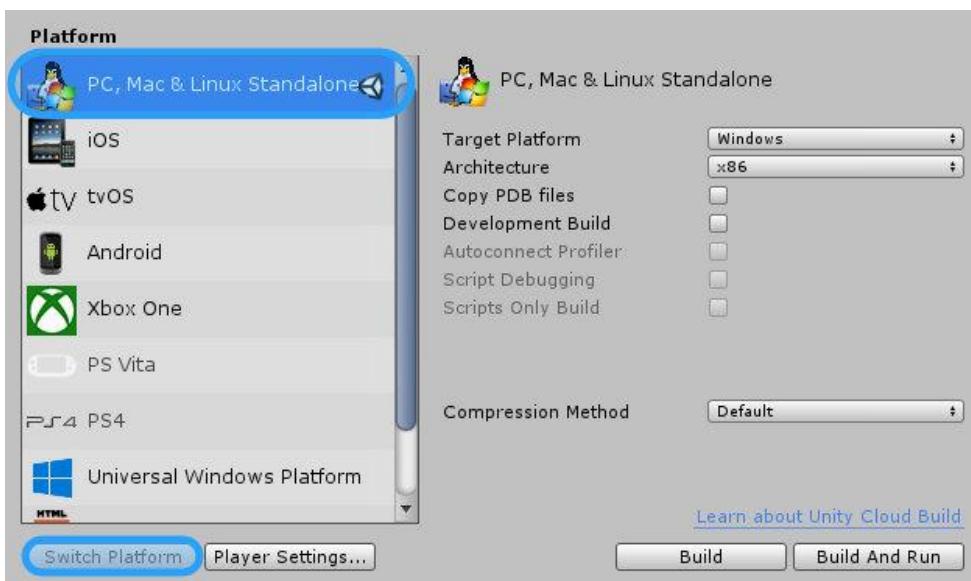
How to Setup Demo Scenes (Post-Processing) in Unity 2018.1 and up (For PC)

1. Make sure you are using PC, Mac & Linux Standalone!

Go to *File > Build Settings*



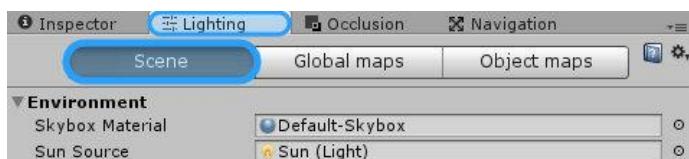
Select PC, Mac & Linux Standalone, and hit the Switch Platform button.



2. Clean GI Cache (Optional) – Skip this if you don't have any light baking errors!

Before you go to the next step, you need to disable the **Auto** build/bake feature.

You can find it in **Lighting** and select the **Scene** tab (If you don't see Lighting tab go to *Window > Lighting > Settings*).



At the bottom you will see this:

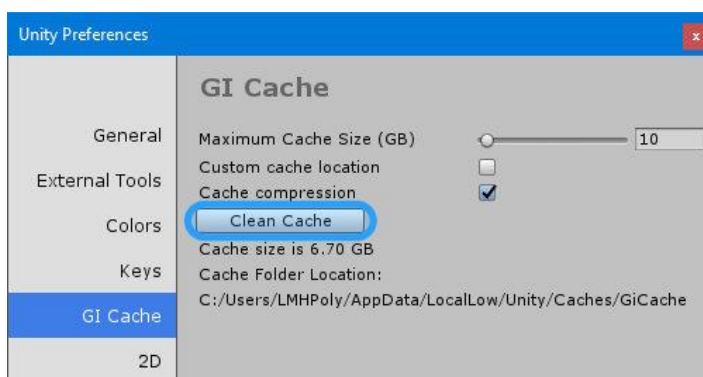


unchecked **Auto Generate**.

Go to *Edit > Preferences*



Select **GI Cache** tab and press on the **Clean Cache** button!



Enable **Auto Generate**/bake feature

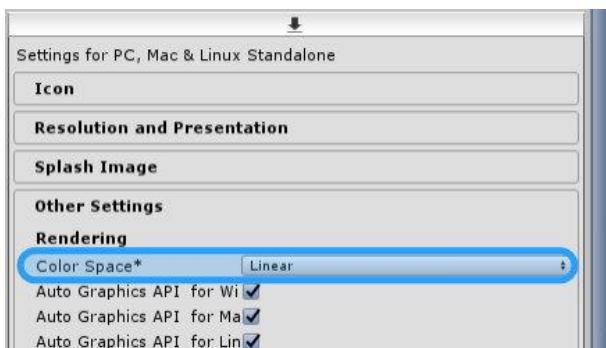


and wait until the generation is done (blue loading bar at the right bottom corner).

3. Make sure that **Color Space** is set to **Linear**.

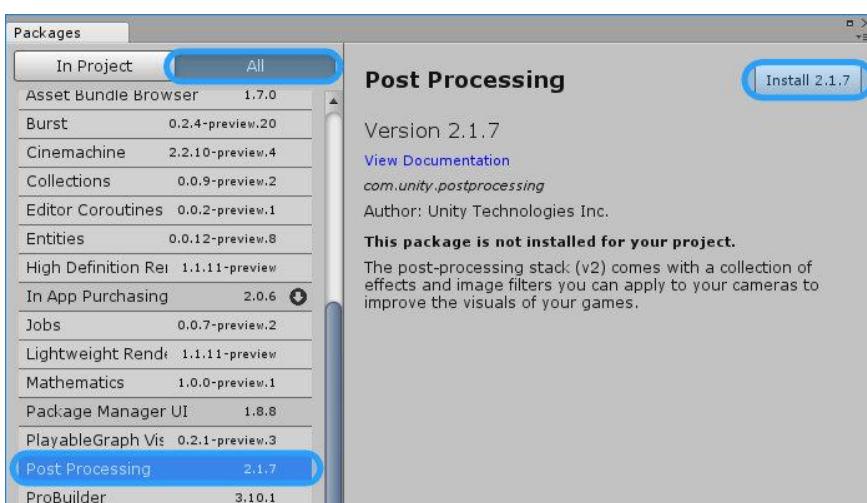
Go to *Edit > Project Settings > Player*

In the **Other Setting** tab, you will find **Color Space***, set it to **Linear**.



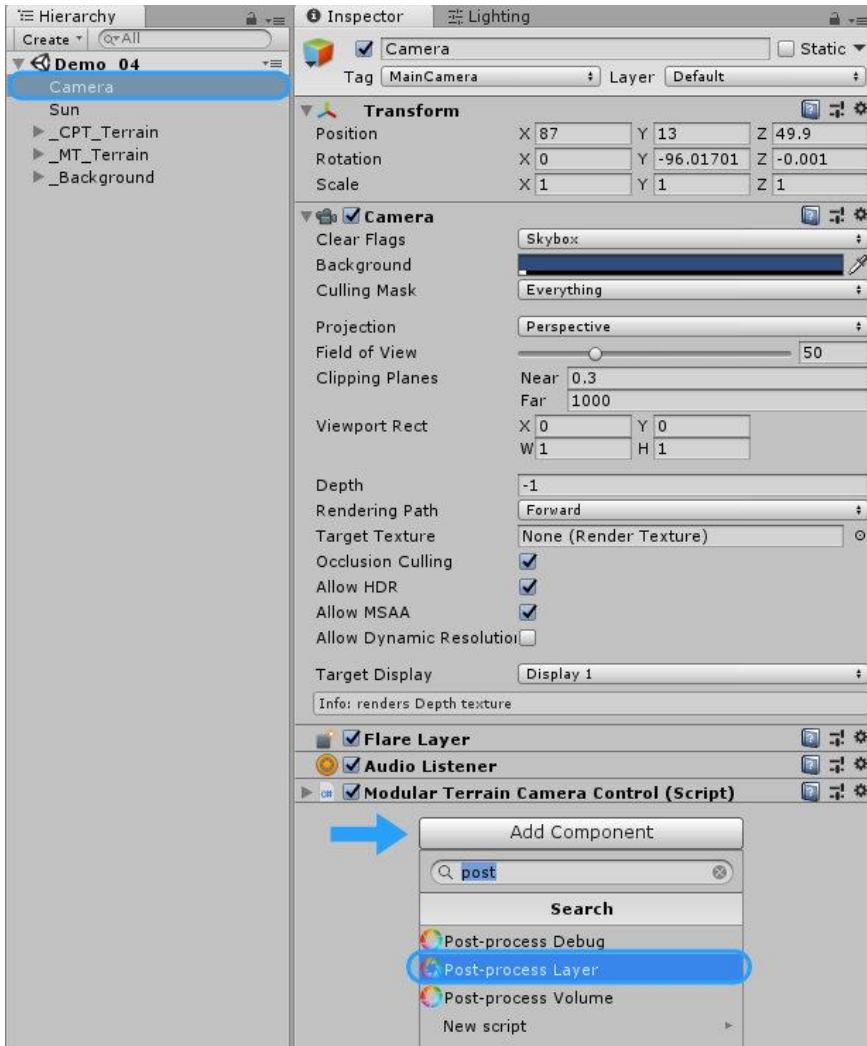
4. Install the **Post Processing** package from the **Package Manager**.

Go to *Window > Package Manager* - open the tab **All** and search for **Post Processing**. Select it and hit the **Install** button:



5. Apply Post-process Layer to the Camera.

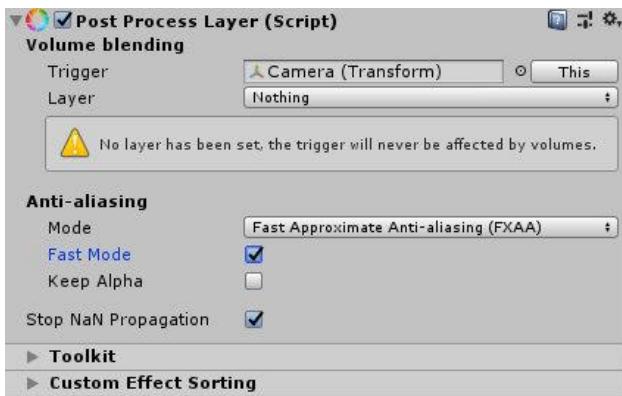
Select the **Camera** in the Hierarchy, press on **Add Component**, type **post** in the search window, and press on **Post-process Layer** to apply.



*Sometimes package installation gets corrupted, and you won't see those options (**Post-process Layer**, **Post-process Volume**, etc.). Follow these steps to fix that issue:

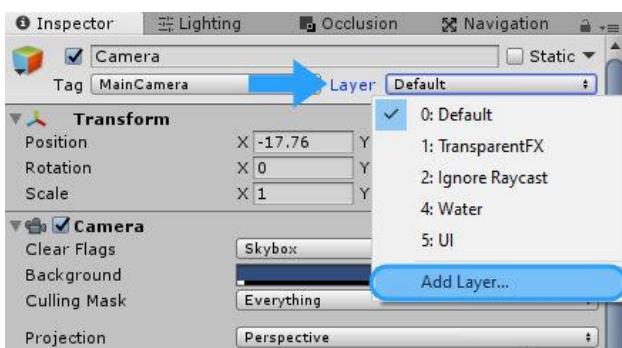
- Restart Unity.
 - If it still doesn't work, go to *Window > Package Manager*, remove the **Post Processing** package.
 - Restart Unity
 - Install the **Post Processing** package again. Now it should work.

Post Process Layer settings:

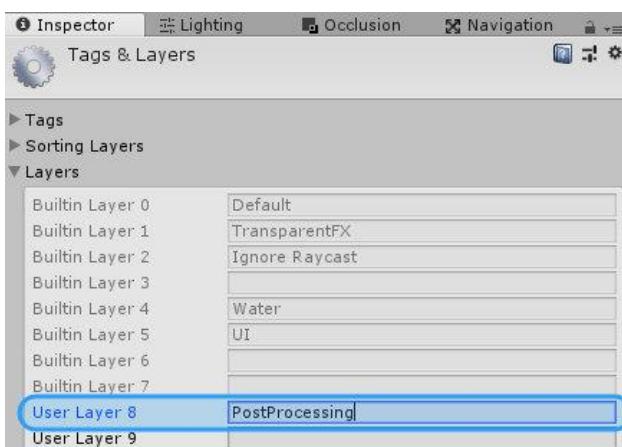


6. Create a **PostProcessing** layer and apply it to the **Camera**.

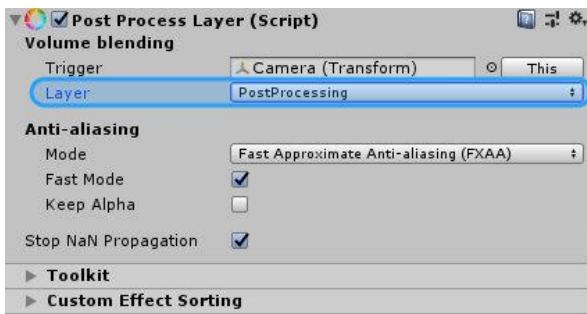
Press on **Layer Default > Add Layer...**



Create a new layer called **PostProcessing**:

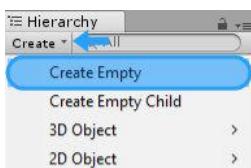


Select the **Camera** and inside the **Post Process Layer – Volume blending – Layer** apply **PostProcessing** layer we just created:

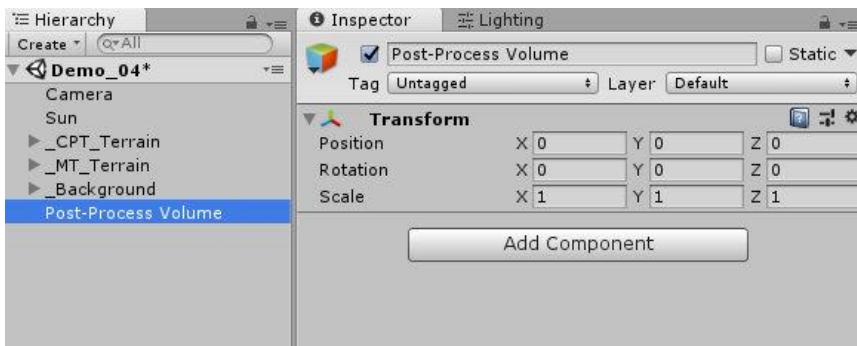


7. Create a Post-Process Volume.

Create Empty game object:



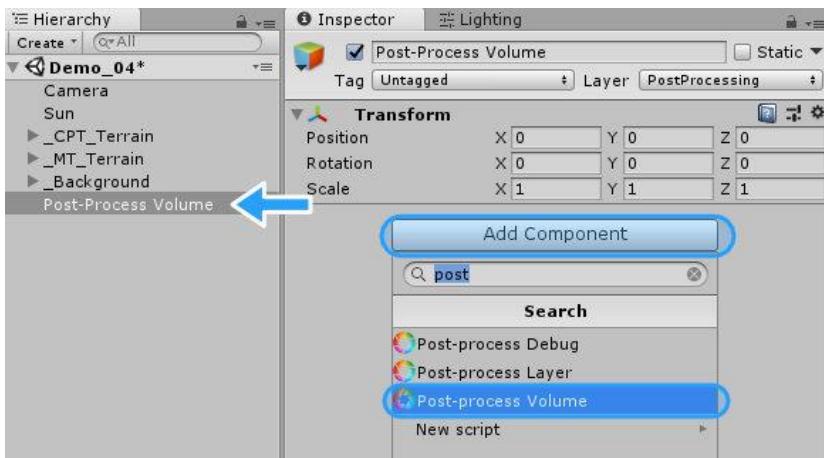
Rename it to something like **Post-Process Volume**:



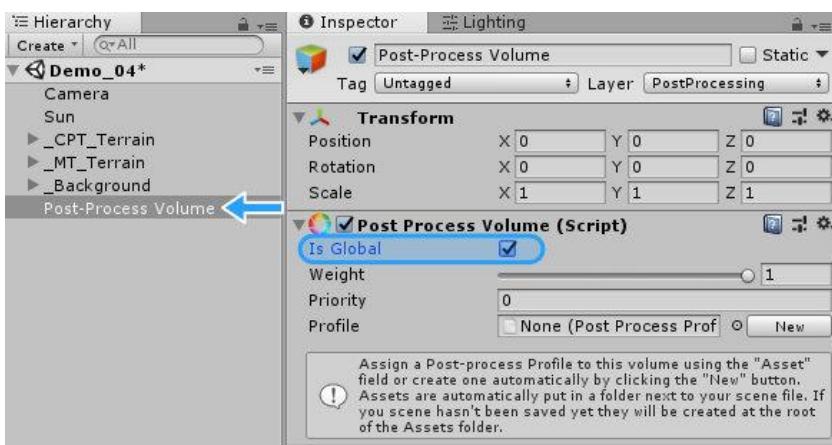
Set **Layer** to **PostProcessing**, the layer we just created before:



With **Post-Process Volume** selected press on **Add Component**, search for **post** and select **Post-process Volume** to apply:

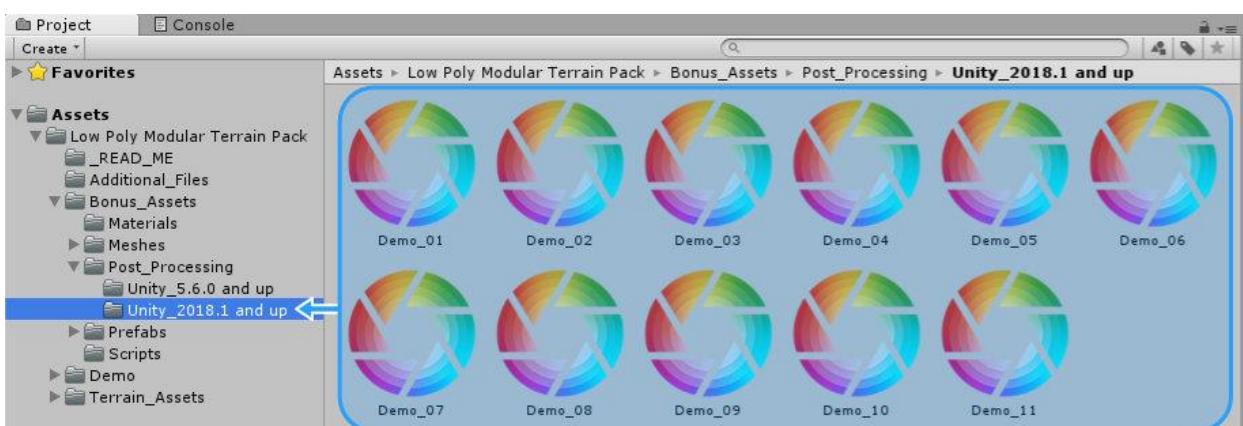


Enable **Is Global**:



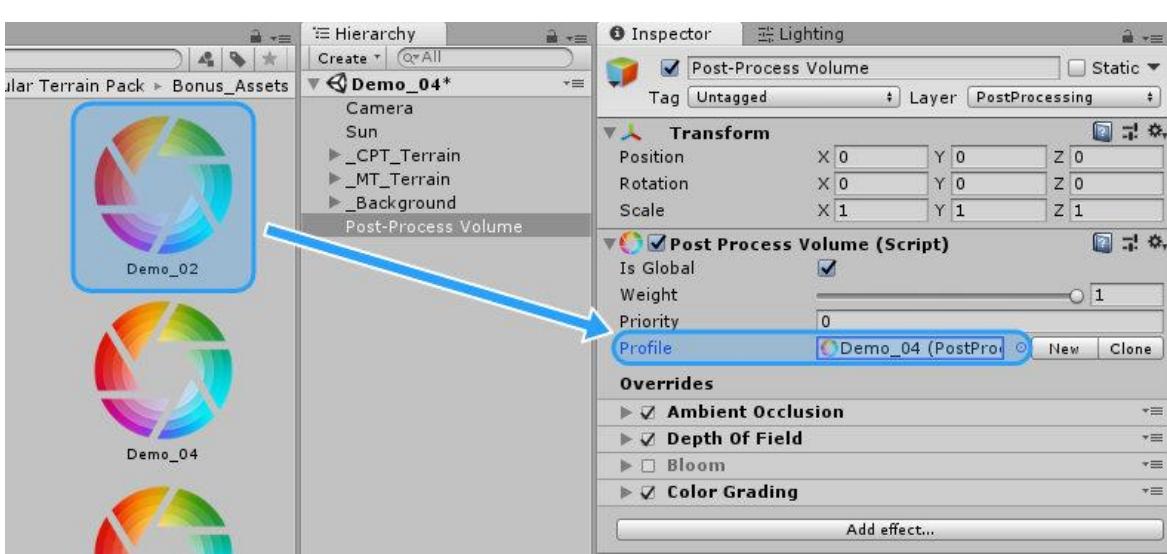
8. Apply Post-Processing Profile.

I've created 11 Post-Processing Profiles for 11 Demo scenes with different settings. Go to *Low Poly Modular Terrain Pack > Bonus Assets > Post_Processing > Unity_2018.1 and up*.



We have opened the **Demo_04** scene - so we will use the **Demo_04** Post-Processing Profile.

Drag and drop **Demo_04** Post-Processing Profile to **Profile** slot inside the **Post Process Volume**:



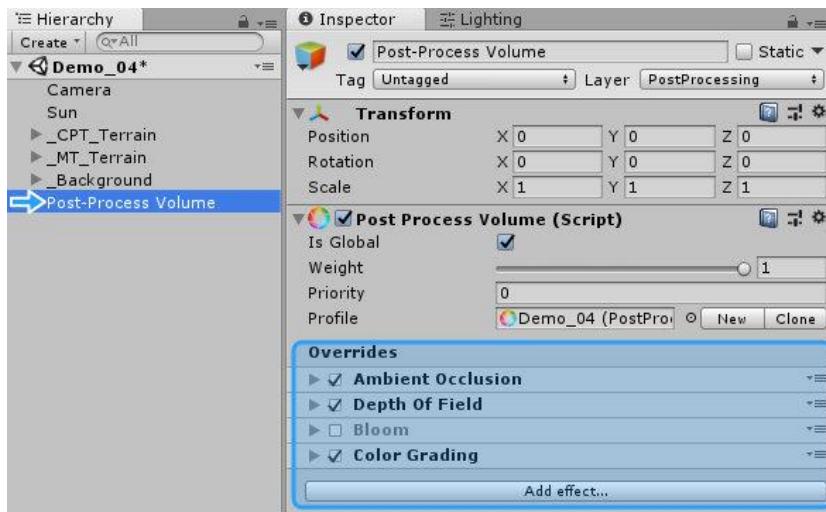
Now your scene should look like this (Demo_04):



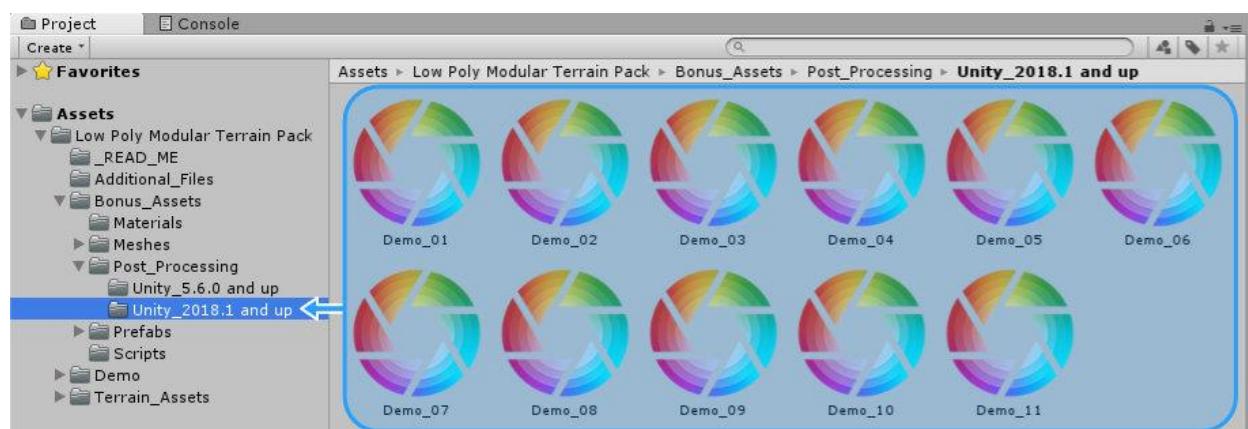
***For Low-End PC's** - if you hit play and it lags, try disabling Post-Processing effects one by one on the Post-Processing Profile settings!

9. Edit Post-Processing Profile.

Select **Post-Process Volume** game object in the **Hierarchy**, and Inside the **Post Process Volume**, you will see options like Ambient Occlusion, Depth of Field, etc. Open and edit them.



Or select any **Post-Processing Profile** inside the **Project** tab to edit.

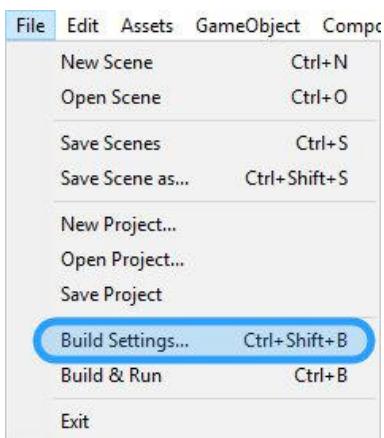


How to Setup Demo Scenes in Unity 2018.1 and up

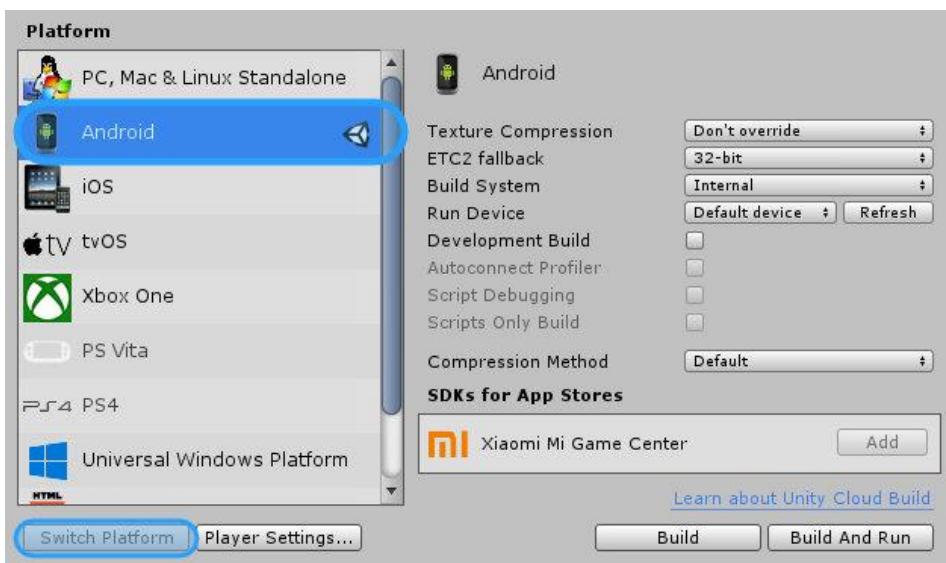
(For Android)

1. Make sure you are using **Android** build!

Go to *File > Build Settings*



Select **Android** and hit the **Switch Platform** button.



2. Clean GI Cache (Optional – Skip this if you don't have any light baking errors!)

Before you go to the next step, you need to disable the **Auto** build/bake feature.

You can find it in **Lighting** and select the **Scene** tab (If you don't see Lighting tab go to *Window > Lighting > Settings*).



At the bottom you will see this:

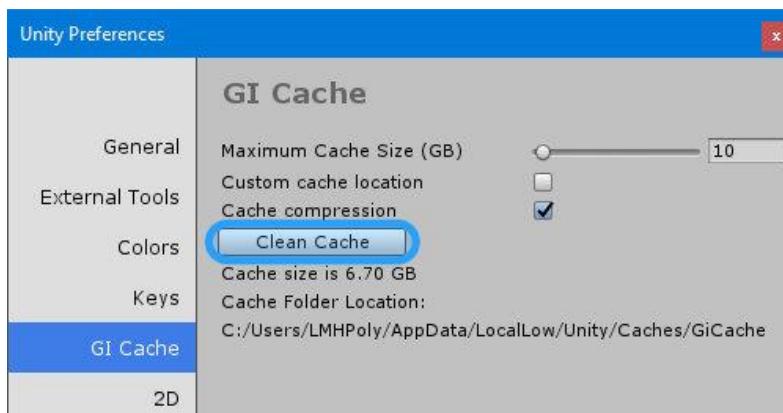


unchecked **Auto Generate**.

Go to *Edit > Preferences*



Select **GI Cache** tab and press on the **Clean Cache** button!



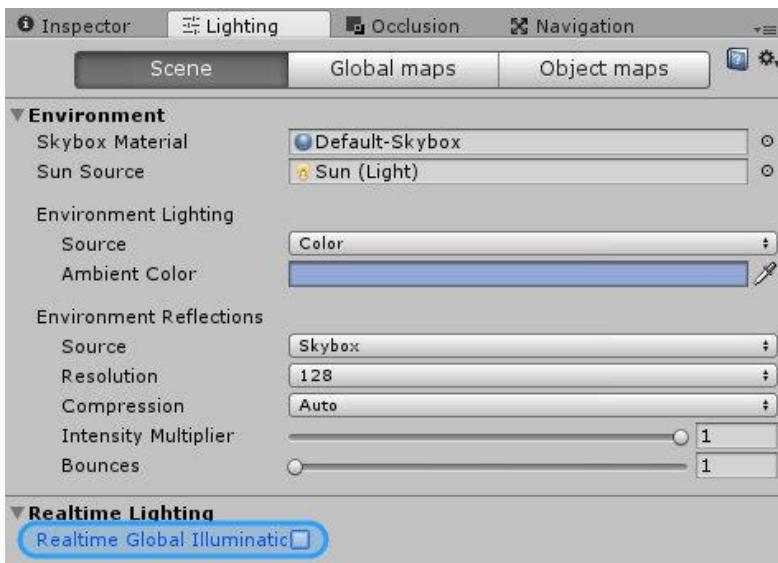
Enable **Auto Generate**/bake feature



and wait until the generation is done (blue loading bar at the right bottom corner).

3. Disable **Realtime Global Illumination** (Optional – for slightly better performance)

You can find it in **Lighting** and select the **Scene** tab (If you don't see Lighting tab go to *Window > Lighting > Settings*).

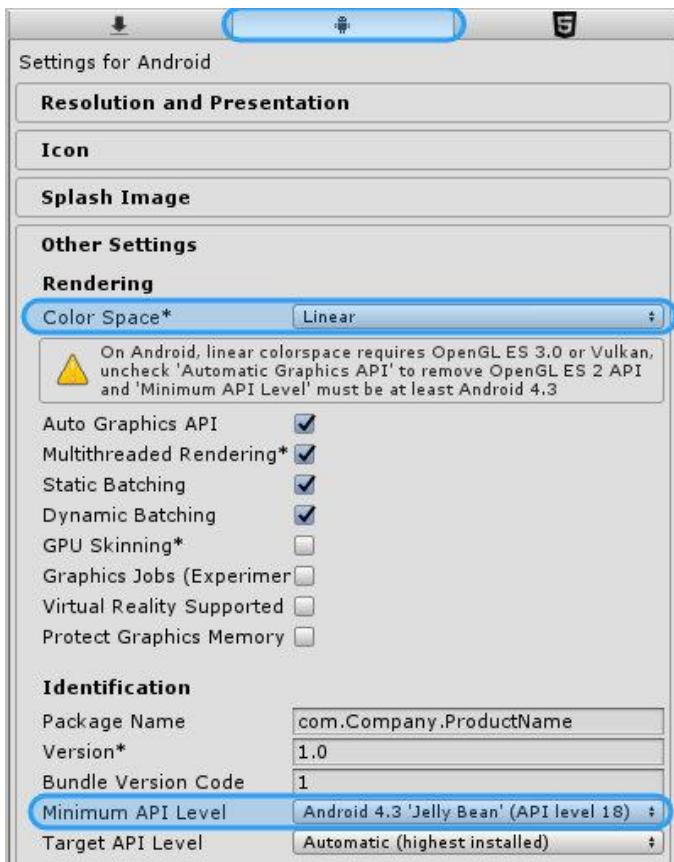


4. Make sure that **Color Space** is set to **Linear** (not all devices support it).

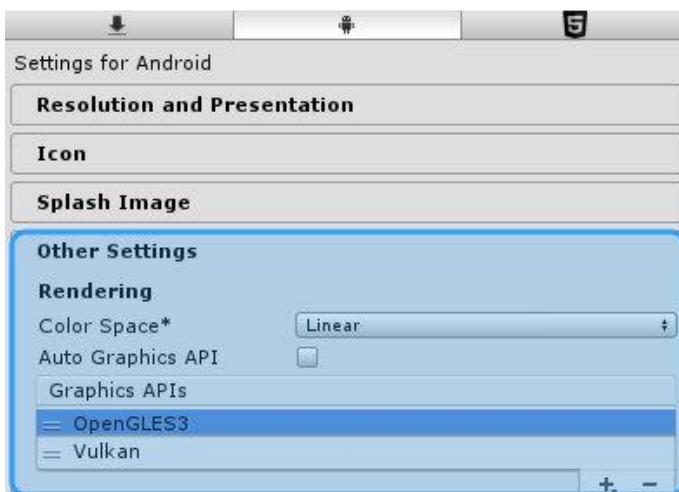
Go to *Edit > Project Settings > Player*

In the **Other Setting** tab, you will find **Color Space***, set it to **Linear**.

To use **Linear Color Space** on Android, you need to set the **Minimum API level** to at least **Android 4.3 (API level 18)** or higher!

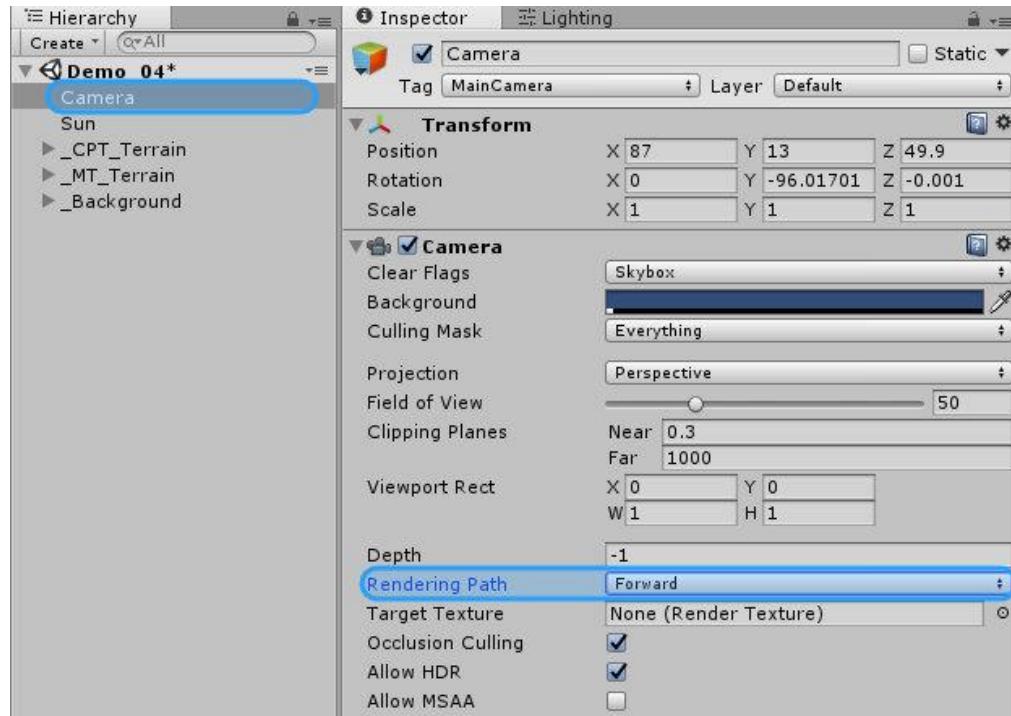


Also, uncheck **Auto Graphics API** and remove all Graphic APIs from the list, leave only **OpenGL ES3** and **Vulkan** (if you can't see it, press on **+** to add it). Make sure your Android device supports one of those graphic APIs!



5. Make sure that you are using **Forward Rendering**. (Use Forward Rendering instead of Deferred for better mobile performance).

Select the **Camera** and make sure that **Rendering Path** is set to **Forward**.

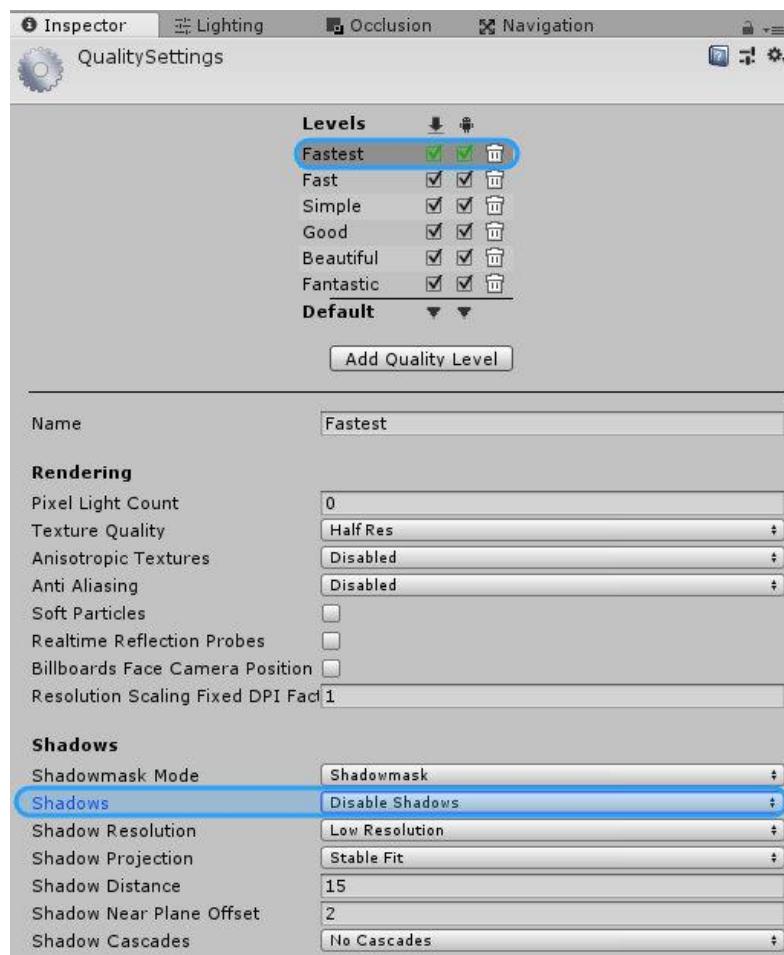


*If you set **Rendering Path** to **Deferred**, the game might slow down a lot on mobile!

6. Disable **Real-time Shadows** (Optional – for much better performance).

Go to *Edit > Project Settings > Quality*

Select Android quality level, which is in **Green Color**, for me, it's **Fastest**. And set **Shadows** to **Disable Shadows**.



*Realtime shadows are not recommended to use on mobile devices because they decrease the performance significantly. You should bake them instead. Or use them only on high-end devices.

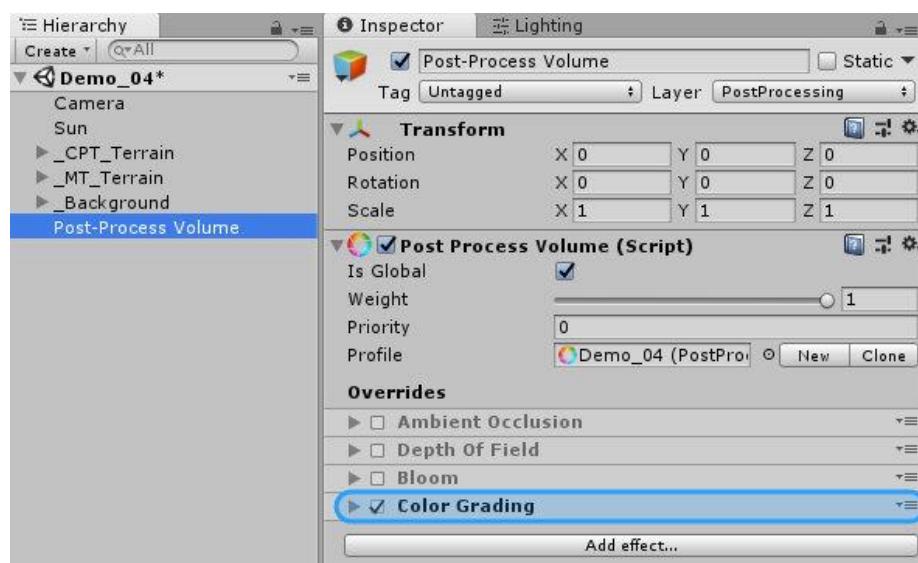
7. Import and enable **Post Processing** image effects (Optional – Big performance hit for mobile devices!).

Go to the part of the documentation: [Post-Processing in Unity 2018.1 and up](#)

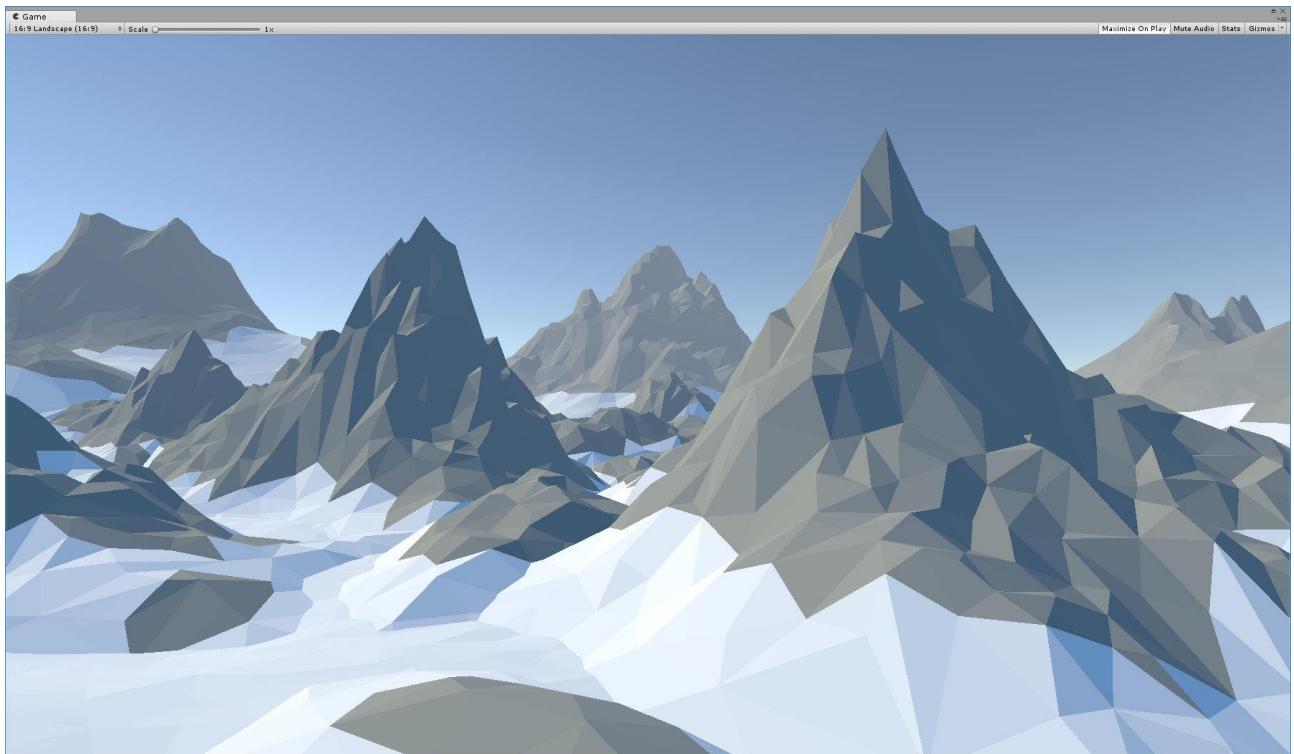
And follow those steps.

*I highly recommend not to use Post-Processing effects on mobile devices because it's a huge hit to performance!

If you will use **Post-Processing** effects, use **Color Grading** only, and leave everything else disabled. It will look nice, and it will work great on high-end devices (Tested on Google Pixel 2 XL).



Now your **Demo_04** scene should look like this (if you skipped all **Optional** steps, and with Realtime Shadows **Disabled**):

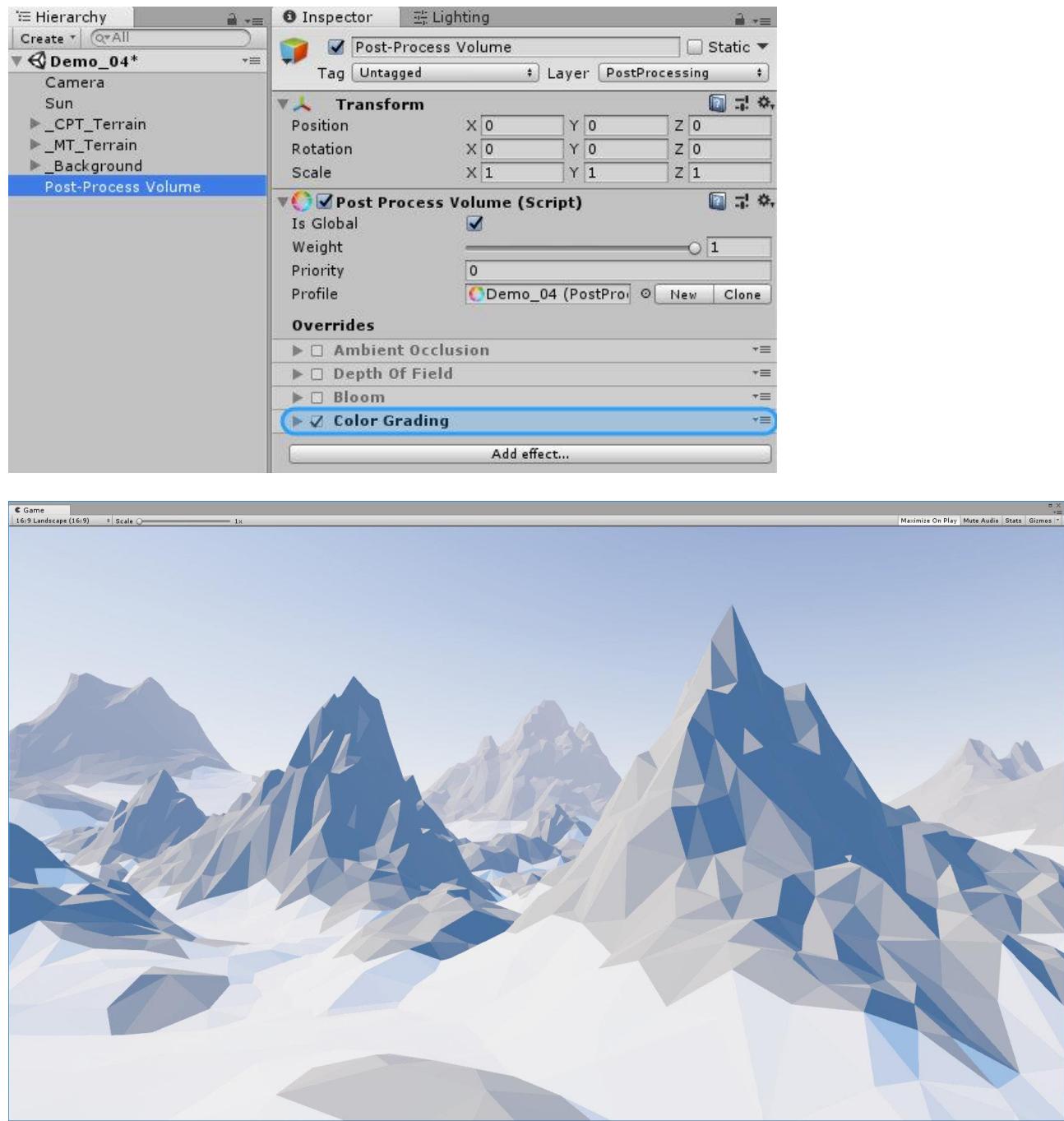


By using **Unity 5.5** and up + **Linear** lighting feature for **Android** and **iOS**, you can achieve much better results than using **Gamma** lighting!

All demo scenes, including **Demo_04**, has been tested on old Xperia Z Ultra (runs at solid 60FPS): without Post Processing effects, using Realtime GI, Linear Color Space, Forward Rendering Path and Real-time Shadows disabled.

*I don't have an **iOS** device, so I can't test it on that!

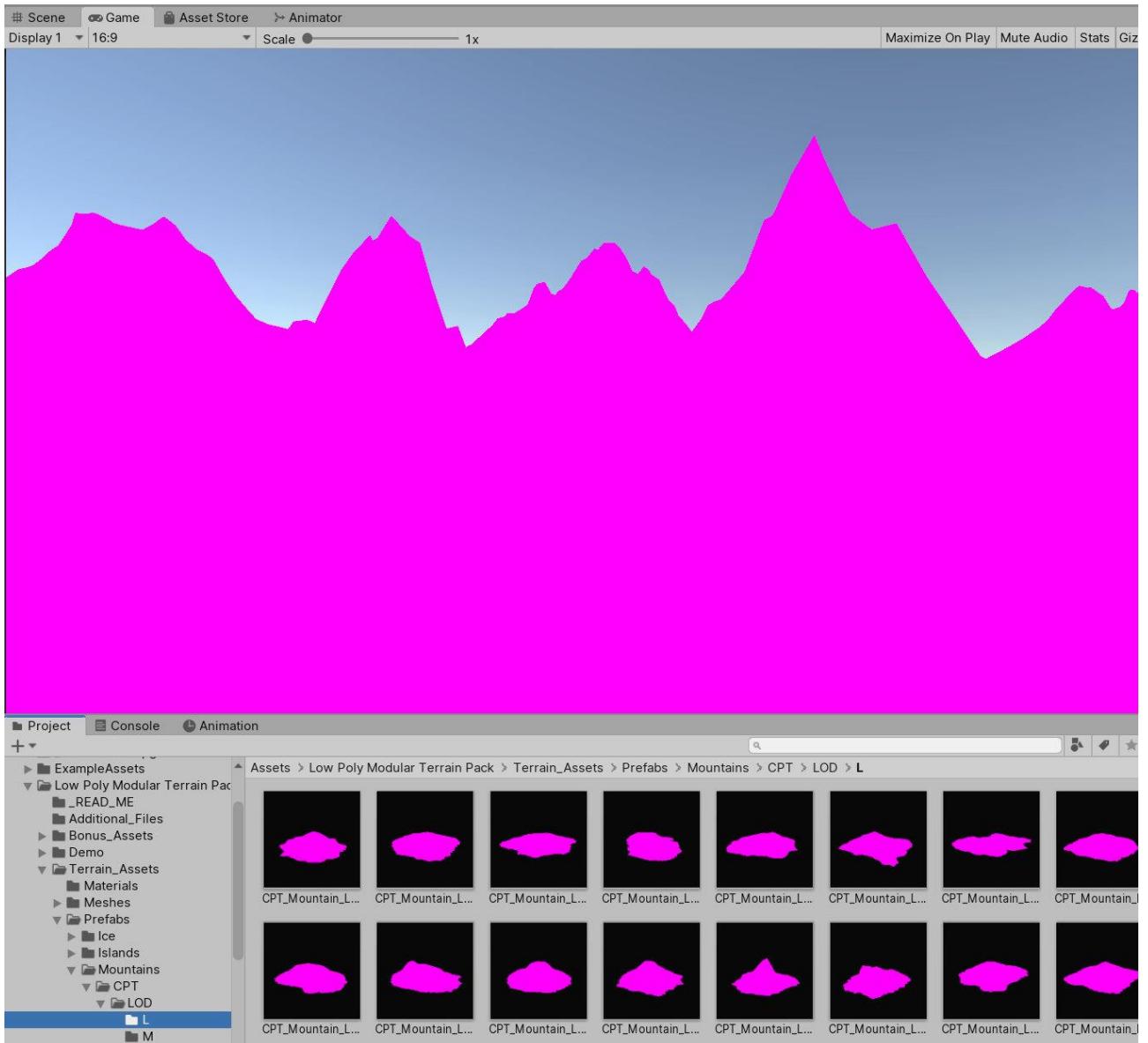
Demo_04 scene with the same settings + Post Processing (**Color Grading** enabled only)



Tested on Google Pixel 2 XL – runs at solid 60fps. Xperia Z Ultra drops to 30fps for using Color Grading.

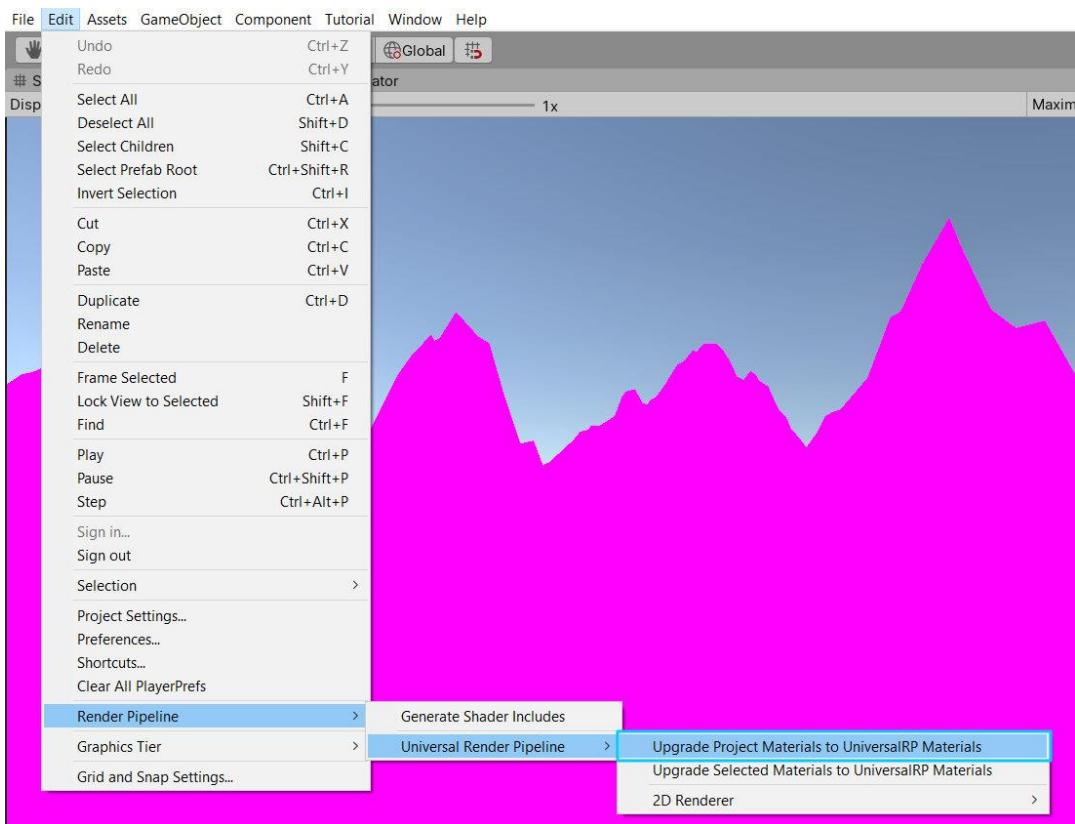
Unity 2019.3 and up - Universal Render Pipeline (URP)

You might encounter pink textures after importing **Low Poly Modular Terrain Pack** to your Unity project, which is using **Universal Render Pipeline (URP)**.

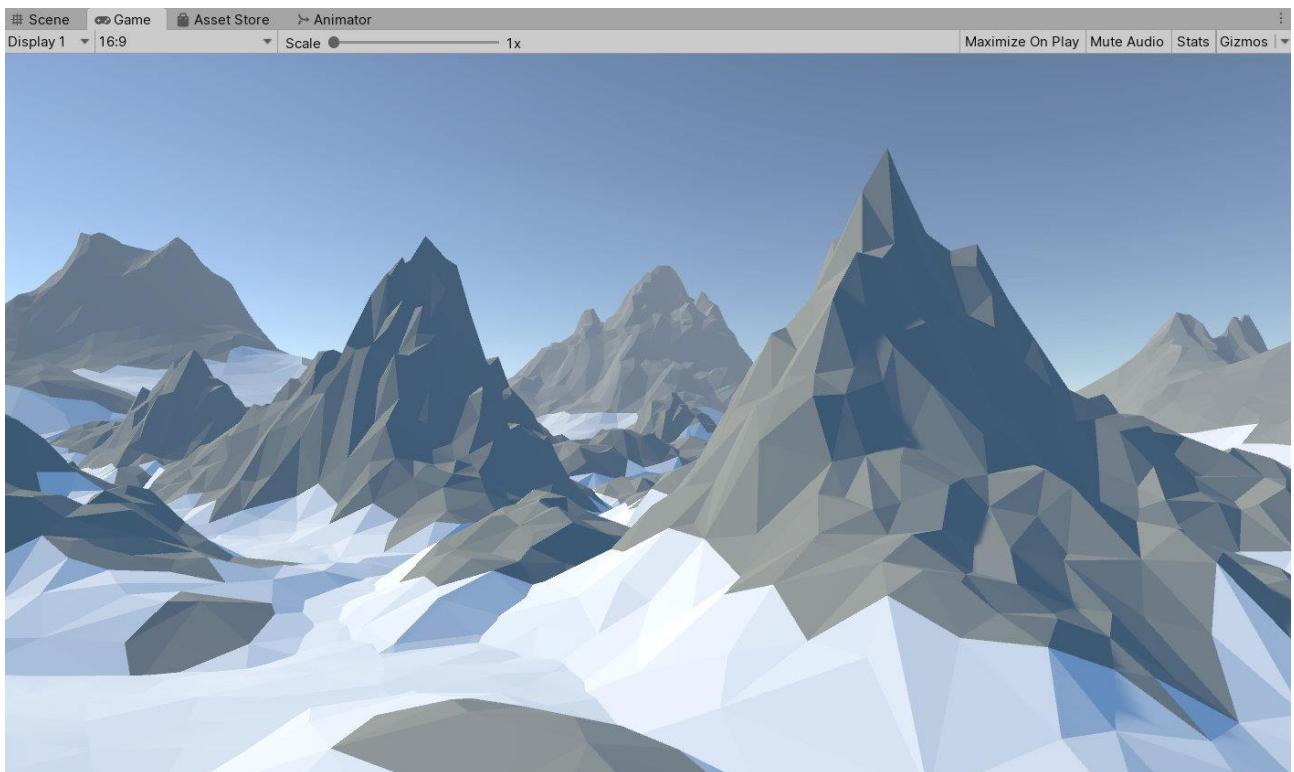


It's because all of **Low Poly Modular Terrain Pack** assets use material with a default **Standard Unity shader**. URP use different materials and shaders. So we need to change all materials from **Standard shader** to **Universal Render Pipeline/Lit shader**.

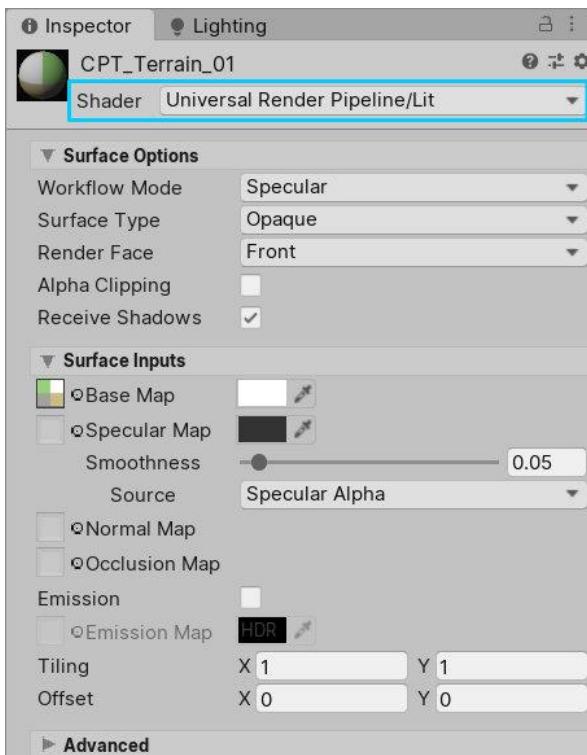
Go to *Edit > Render Pipeline > Universal Render Pipeline > Upgrade Project Materials to UniversalRP Materials*



And it's done!

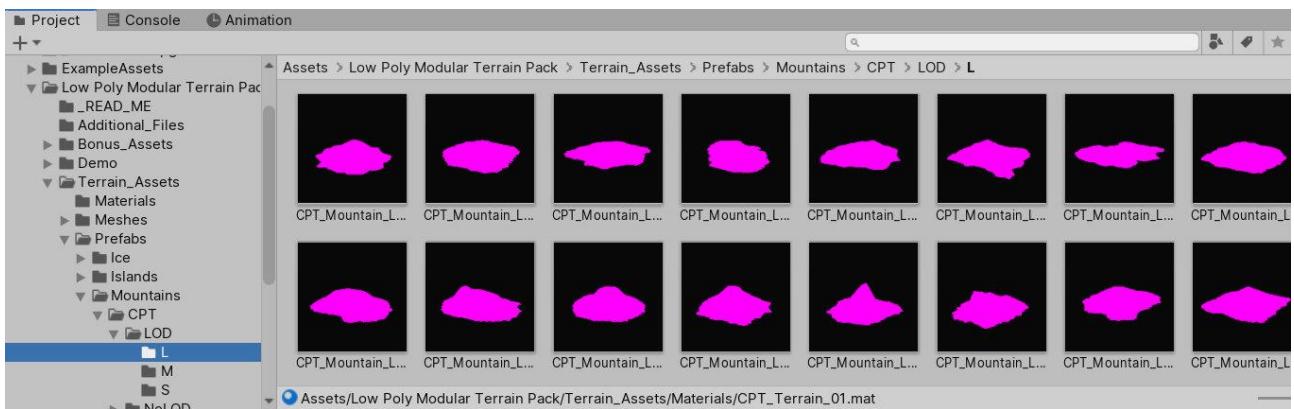


All project **Material** shaders were changed to **Universal Render Pipeline/Lit**

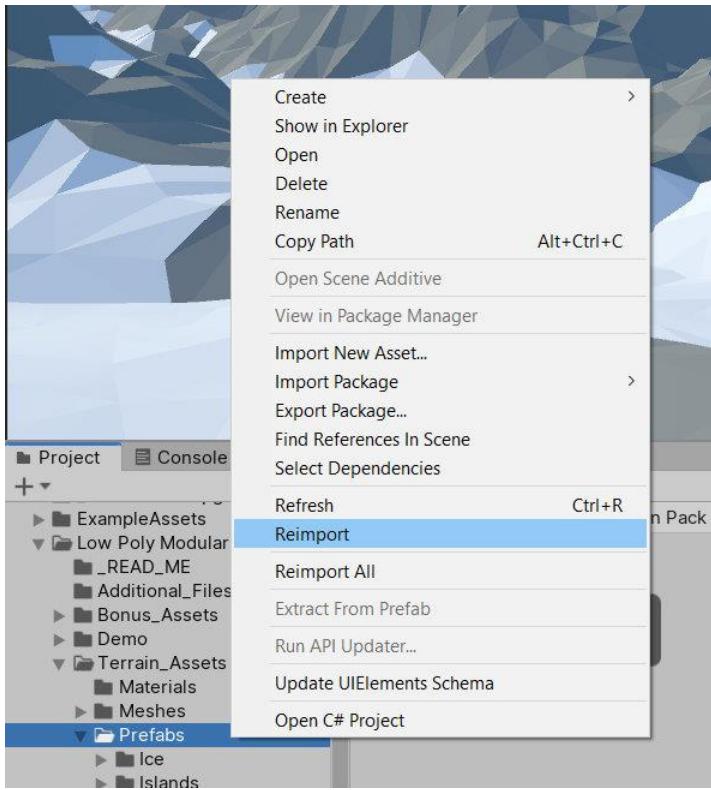


*You can do it manually by selecting **Material** and changing the **Shader** but it's much slower.

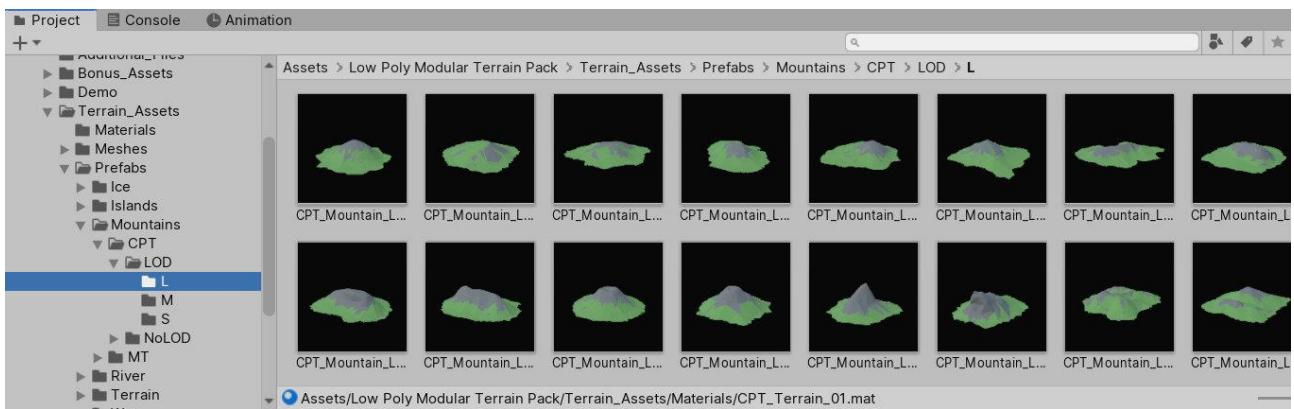
Now if you go to *Low Poly Modular Terrain Pack > Terrain_Assets > Prefabs > Mountains > CPT > LOD > L* - or inside any other prefab folder. You might see all of the prefabs in **Pink** color.



To fix that - press **RMouse** on the „**Prefabs**“ folder and select **Reimport**.

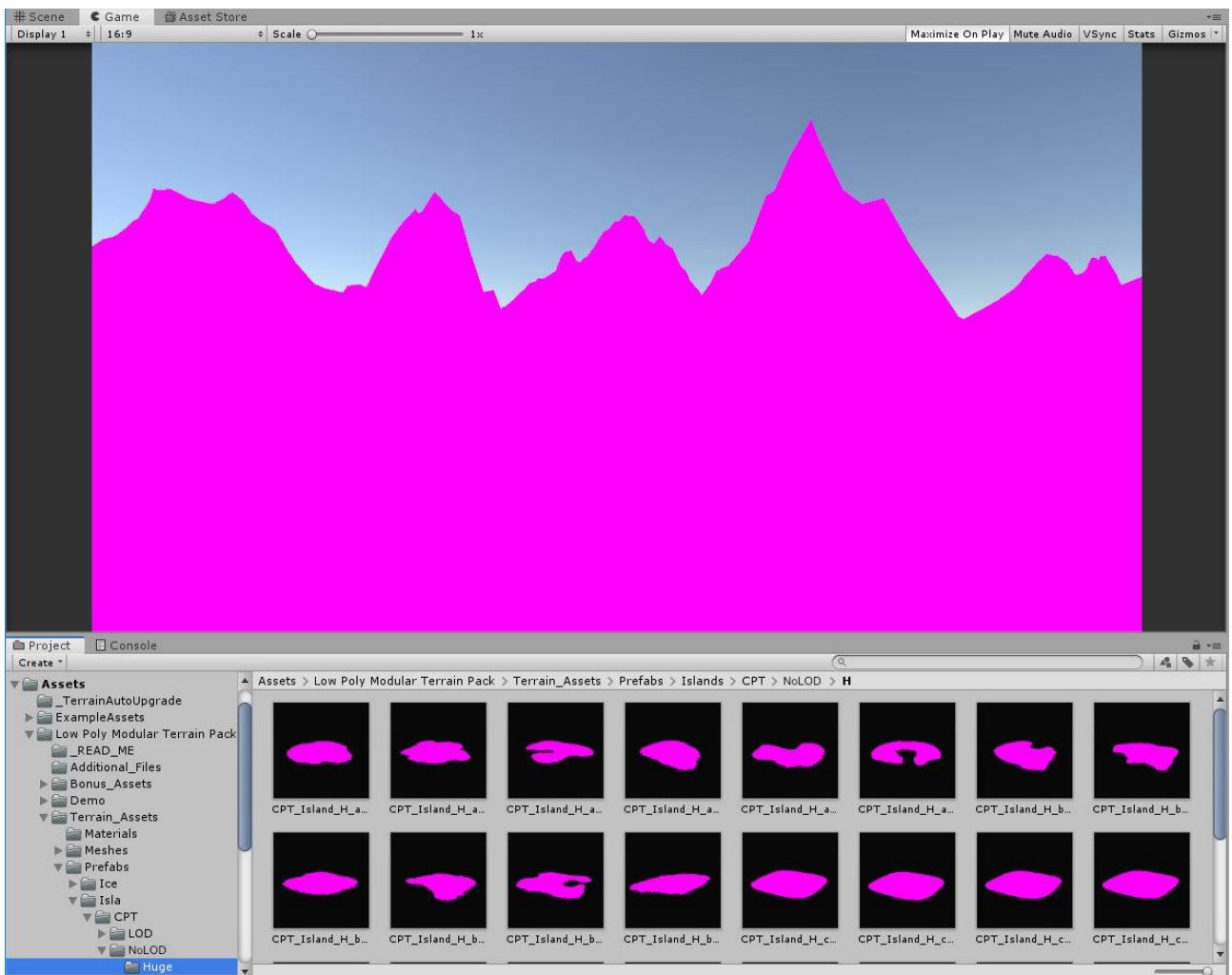


And it's fixed!



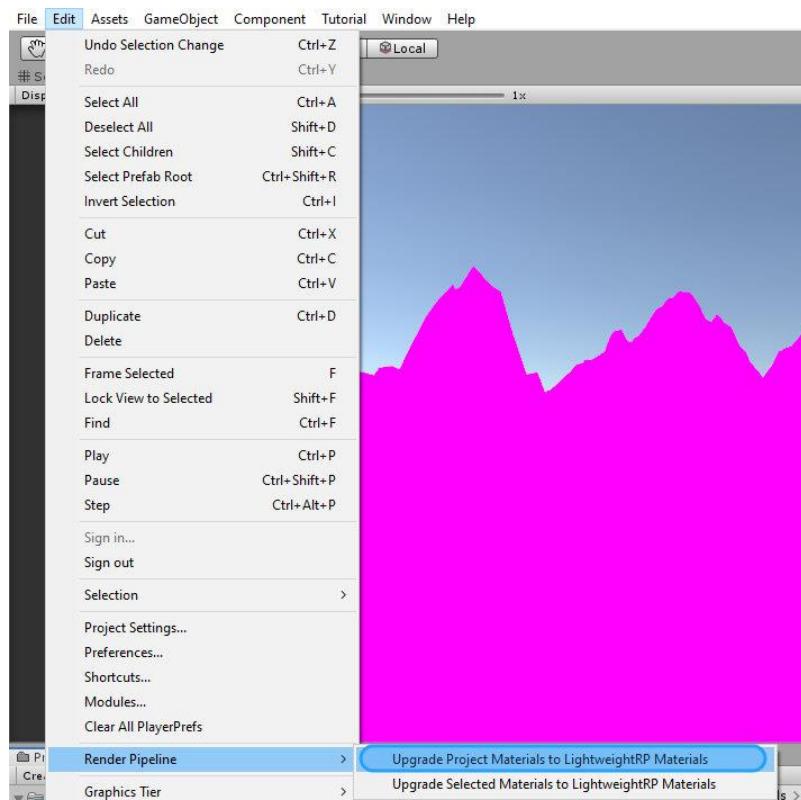
Unity 2018.1 and up - Lightweight Render Pipeline (LWRP)

You might encounter pink textures after importing **Low Poly Modular Terrain Pack** to your Unity project, which is using **Lightweight Render Pipeline (LWRP)**.

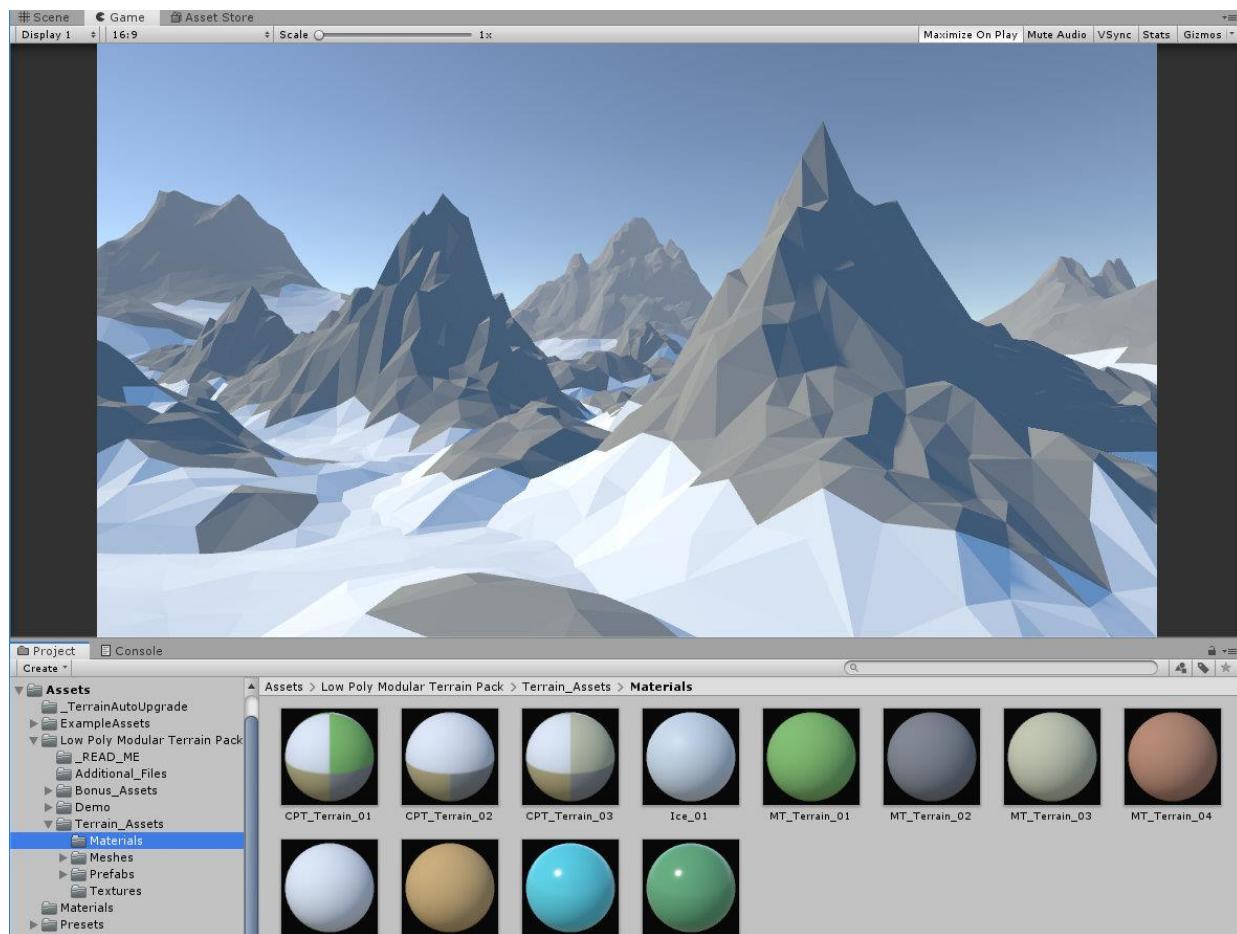


It's because all of **Low Poly Modular Terrain Pack** assets use materials with a default **Standard Unity shader**. LWRP uses different materials and shaders. So we need to change all materials from **Standard shader** to **LightweightPipeline shader**.

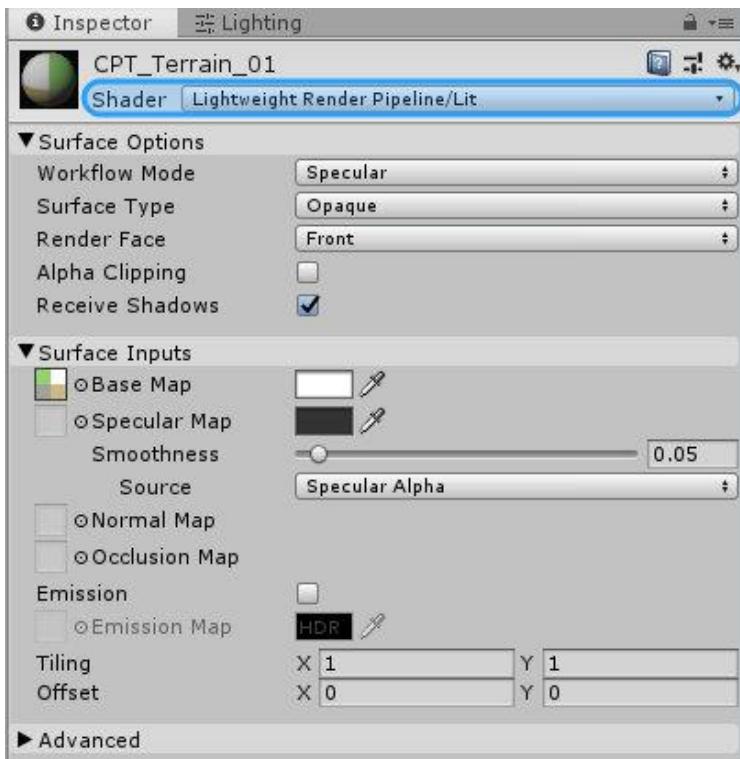
Go to *Edit > Render Pipeline > Upgrade Project Materials to LightWeight Materials*



And it's done!

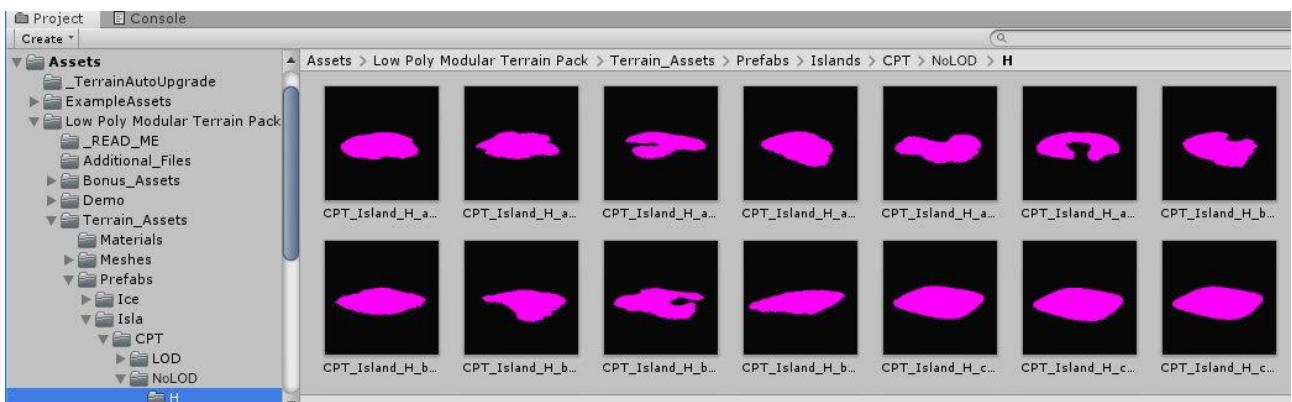


All project Material **Shaders** were changed to **Lightweight Render Pipeline/Lit**.

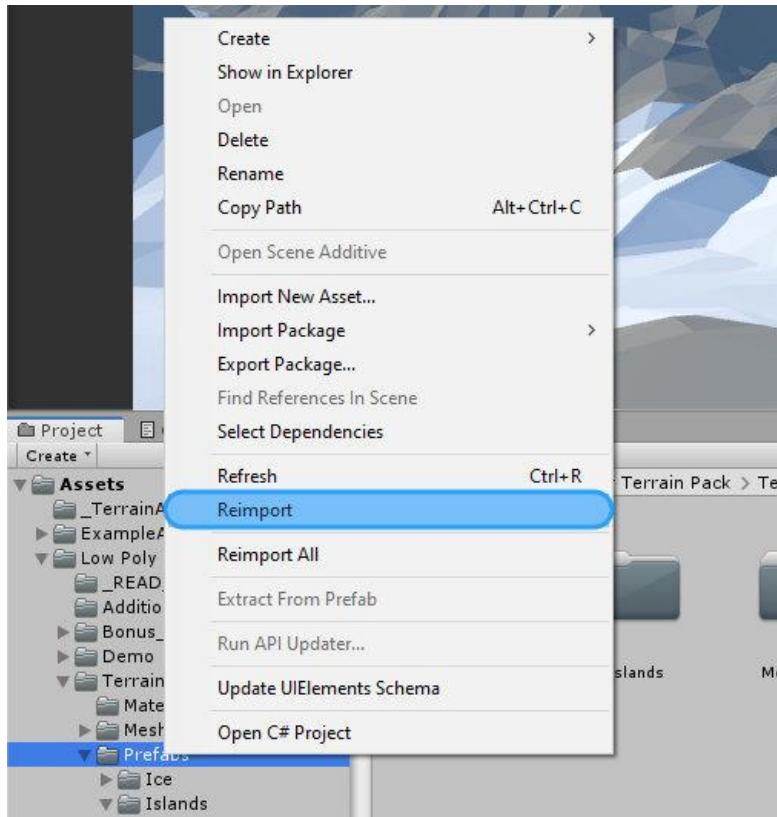


*You can do it manually by selecting **Material** and changing the **Shader**, but it's much slower.

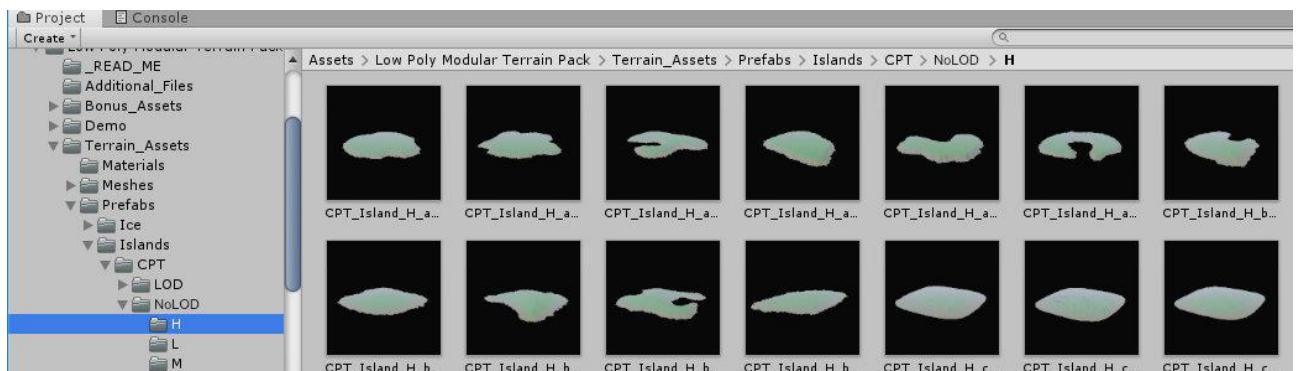
Now if you go to *Low Poly Modular Terrain Pack > Terrain_Assets > Prefabs > Islands > CPT > NoLOD > H* - or inside any other Terrain folder. You might see all of the prefabs in **Pink** color.



To fix, that press **Right Mouse Button** on the **Prefabs** folder and select **Reimport**.

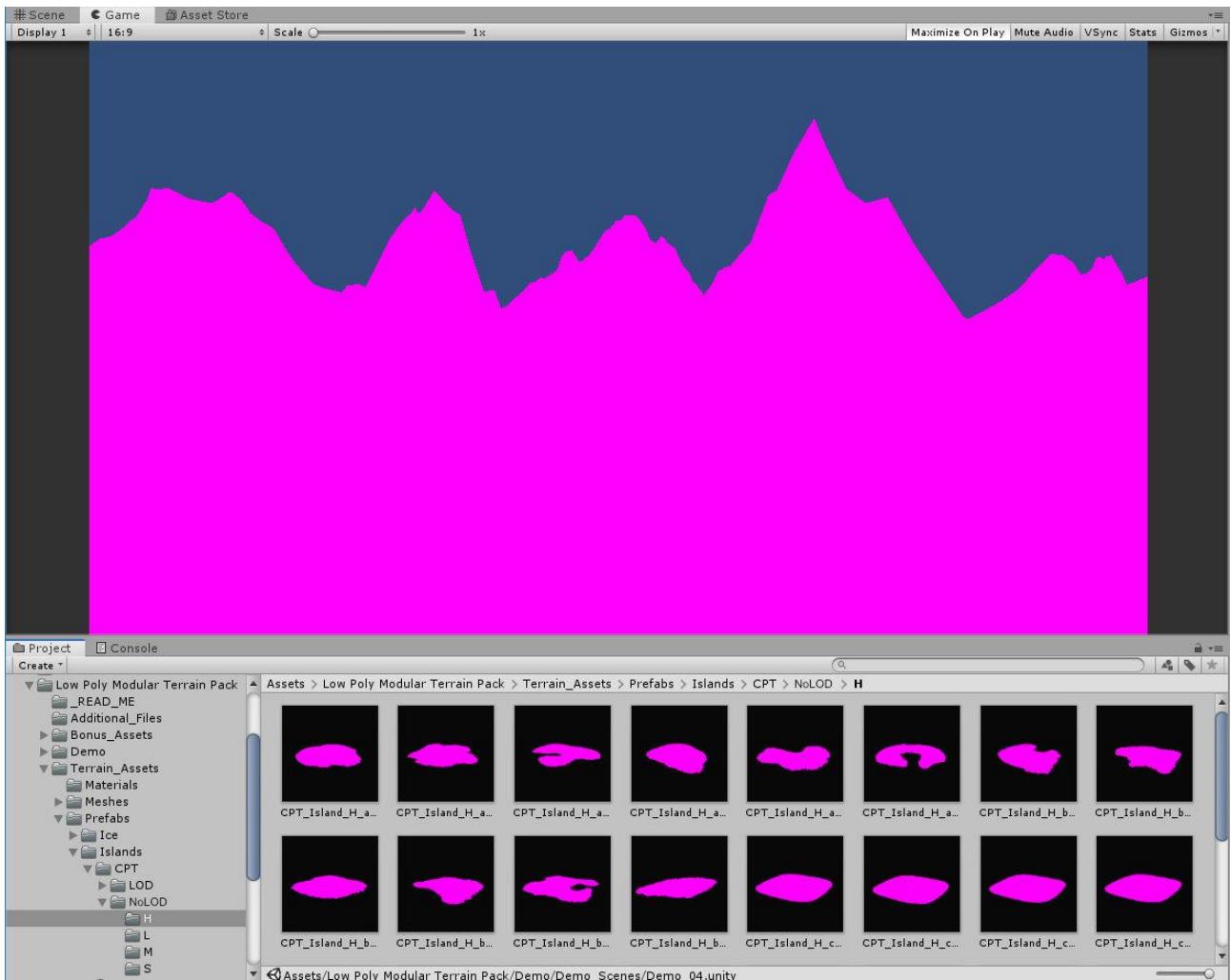


And it's fixed!



Unity 2018.1 and up - High Definition Render Pipeline (HDRP)

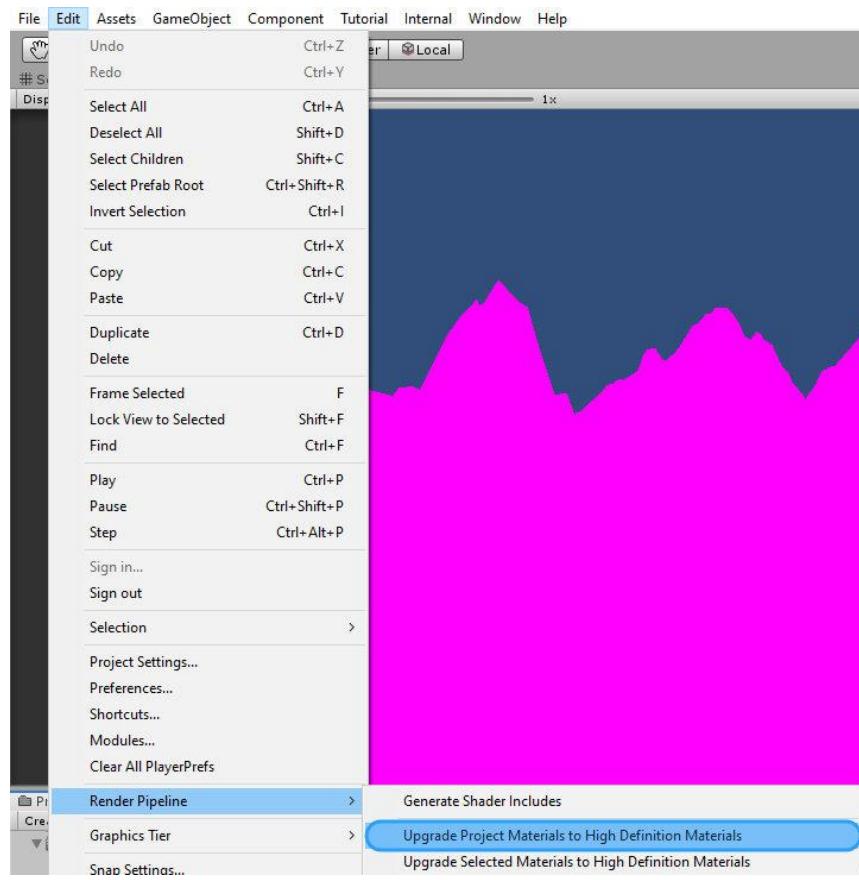
You might encounter pink textures after importing **Low Poly Modular Terrain Pack** to your Unity project, which is using **High Definition Render Pipeline (HDRP)**.



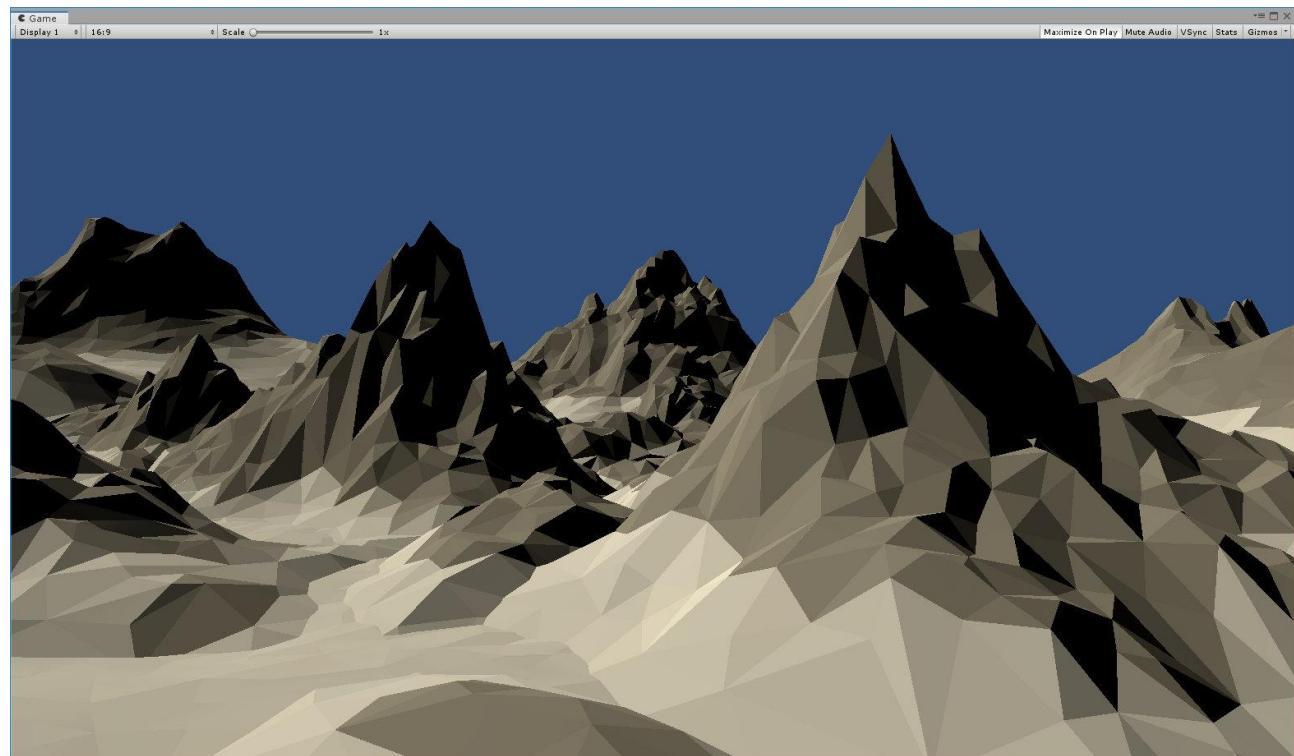
It's because all of **Low Poly Modular Terrain Pack** assets use materials with a default **Standard Unity shader**. HDRP uses different materials and shaders. So we need to change all materials from Standard shader to HDRenderPipeline shader.

1. Fix Purple Materials

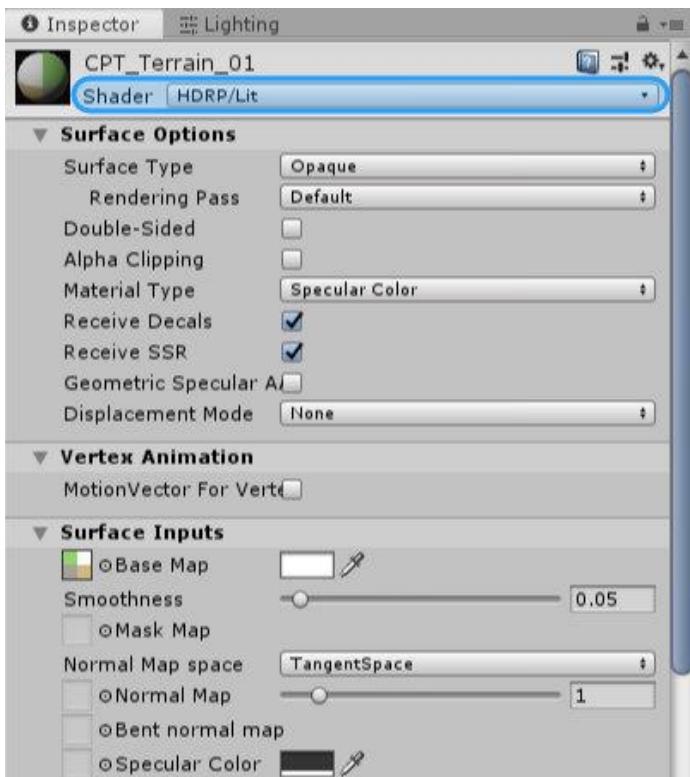
Go to *Edit > Render Pipeline > Upgrade Project Materials to High Definition Materials*



And it's done! Almost.

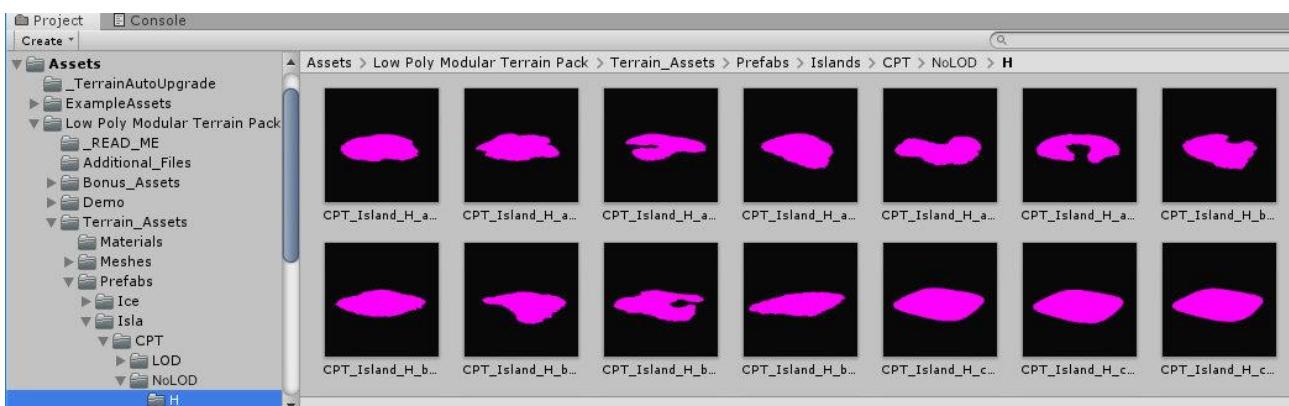


All project **Material** shaders were changed to **HDRP/Lit**.

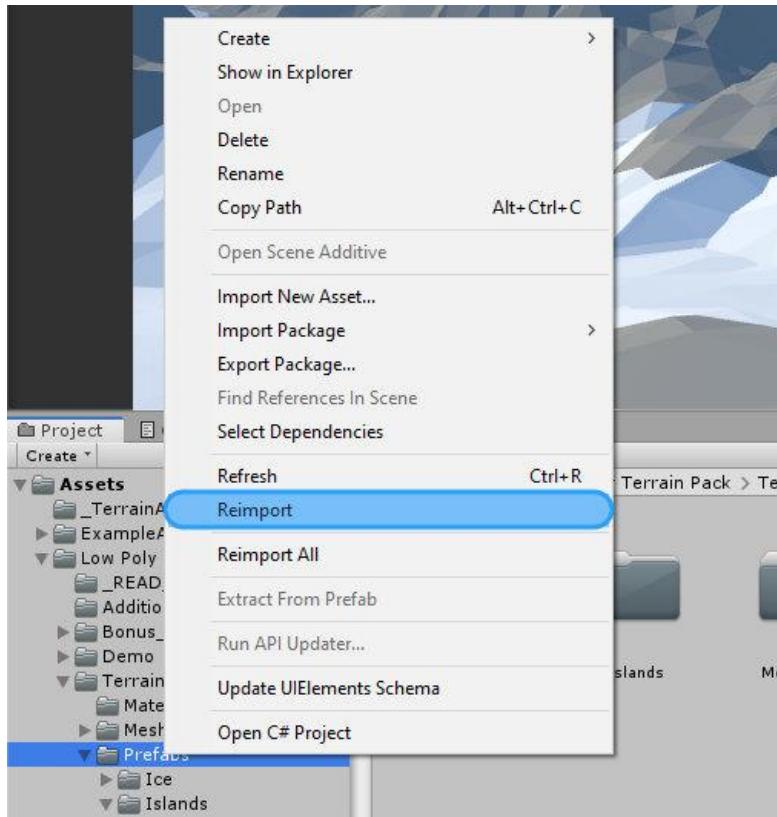


*You can do it manually by selecting **Material** and changing the **Shader**, but it's much slower.

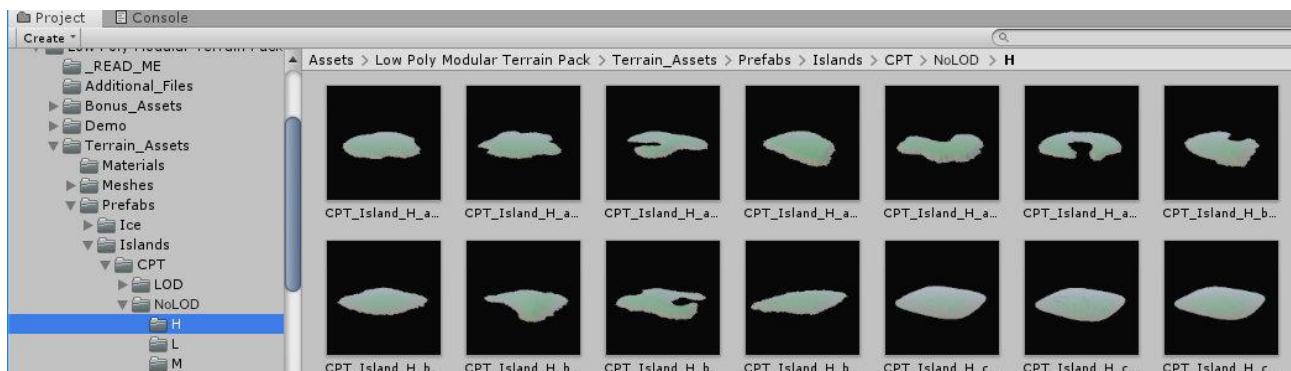
Now if you go to *Low Poly Modular Terrain Pack > Terrain_Assets > Prefabs > Islands > CPT > NoLOD > H* - or inside any other Terrain folder. You might see all of the prefabs in **Pink** color.



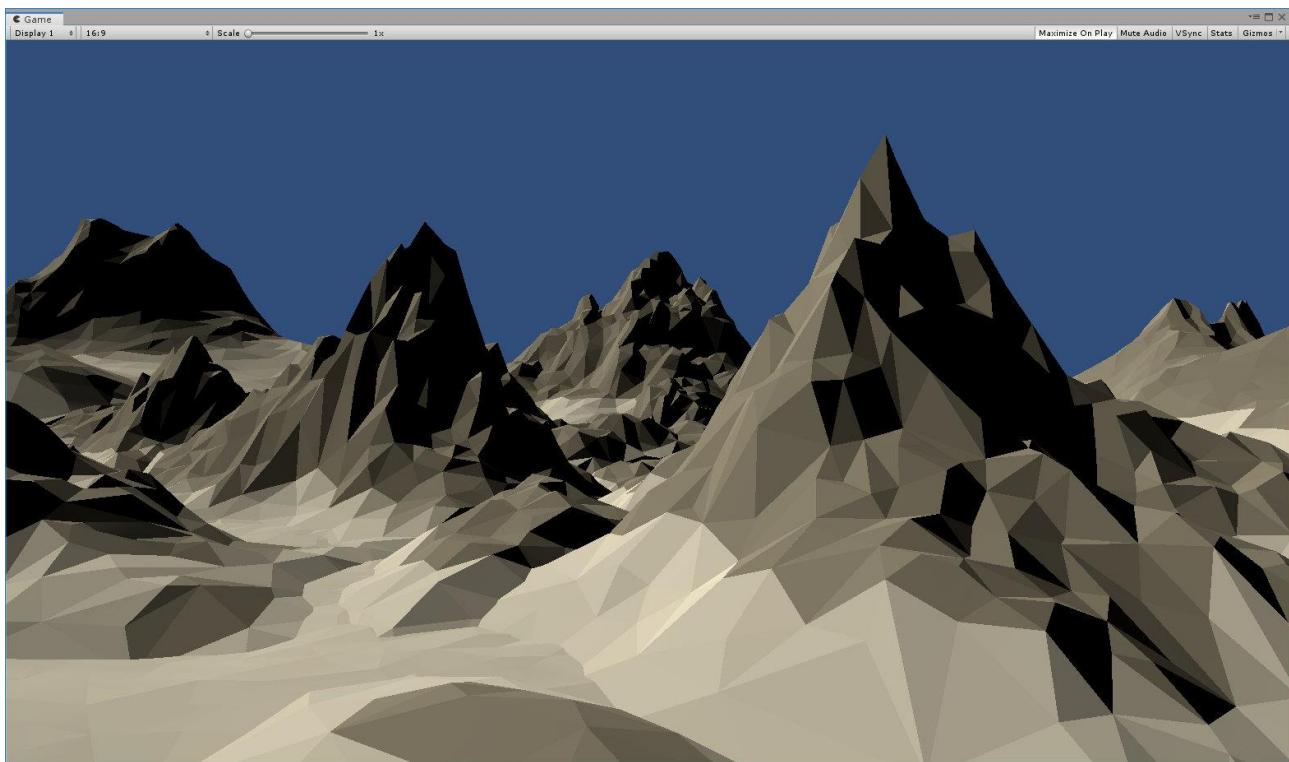
To fix that - press **Right Mouse Button** on the **Prefabs** folder and select **Reimport**.



And it's fixed!

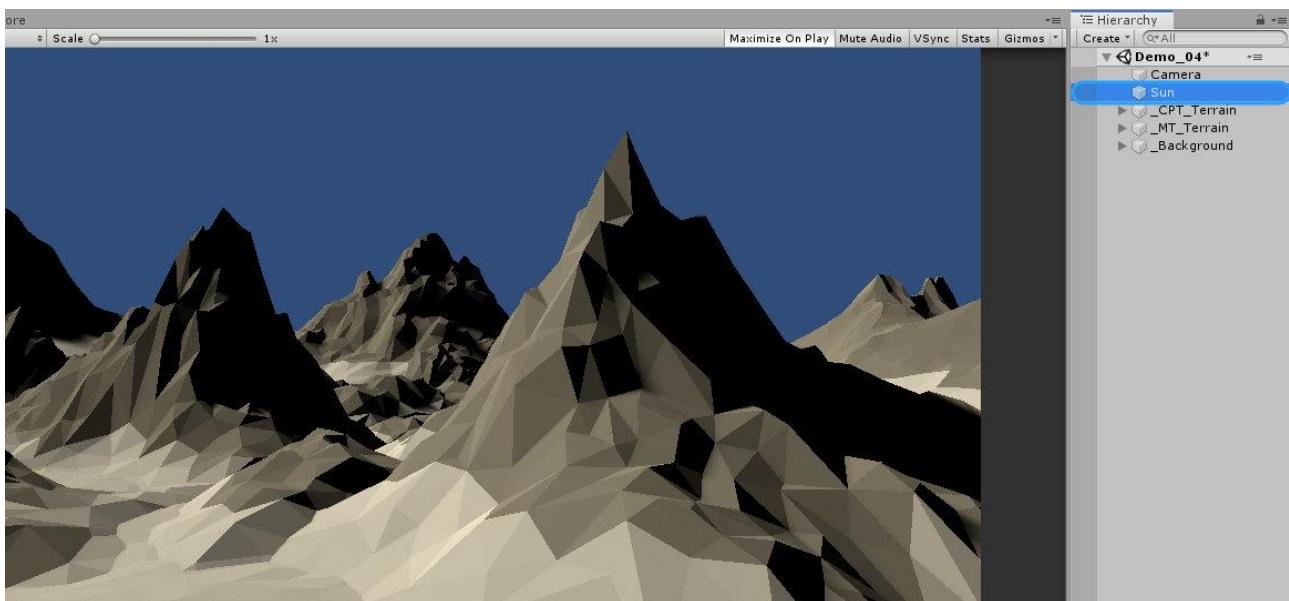


As you can see, the scene looks very dull. It has no skybox and proper lighting.



2. Fix Shadows and Lighting

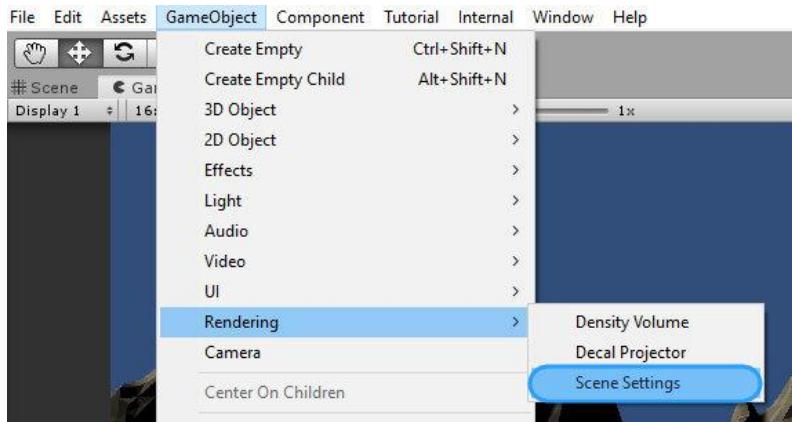
Just select the **Sun** in the **Hierarchy** for lighting and shadows to show up in the scene.



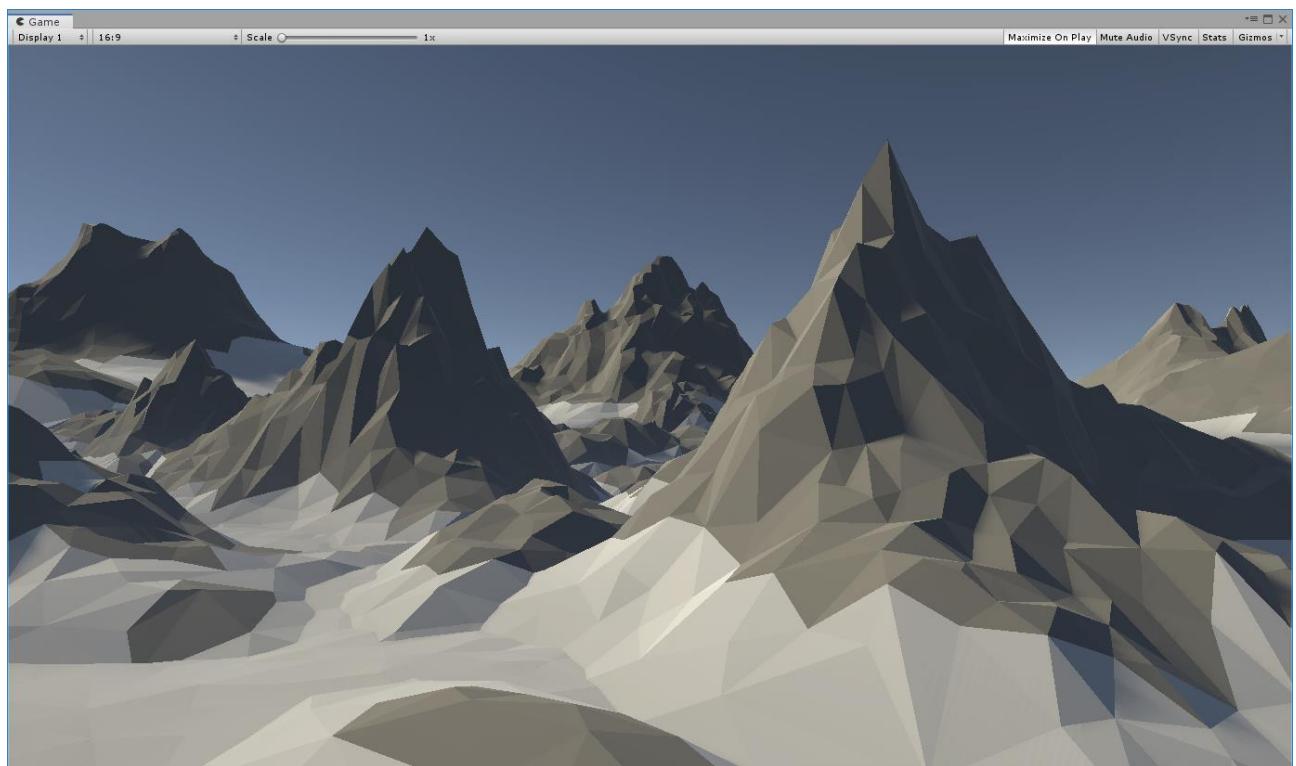
*If Unity freezes after selecting the light, upgrade your project to Unity 2019.1 or up.

3. Fix the **Skybox**

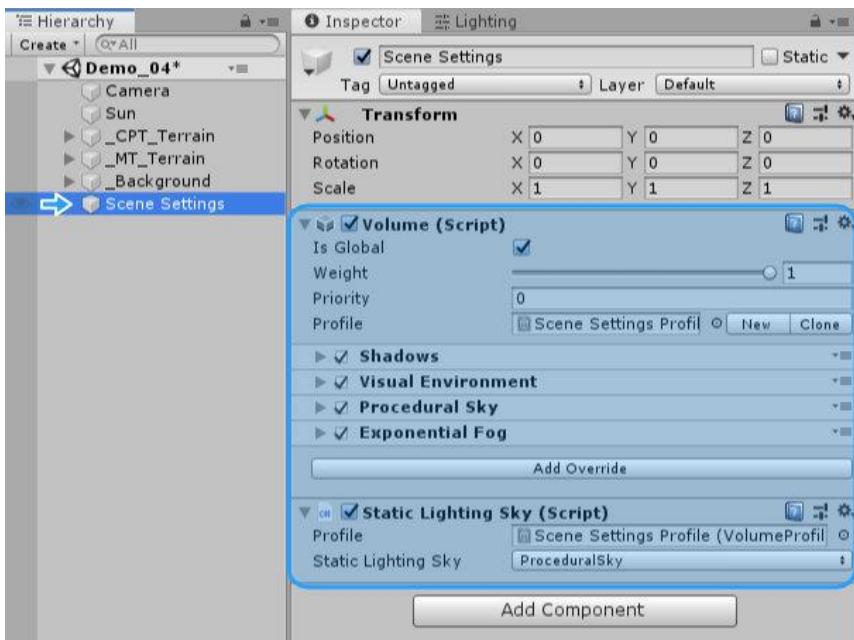
Using HDRP, you need to use **Scene Settings** - to change the **Skybox** and other scene settings. Go to *GameObject > Rendering > Scene Settings*



And you will see that the **Skybox** is applied to the scene right away.



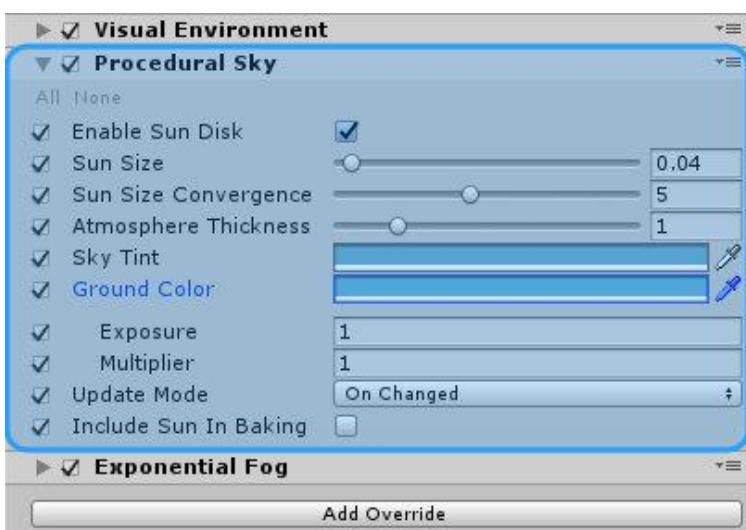
With **Scene Settings** selected, you can change a bunch of scene settings like (Shadows, Skybox, Fog, and much more).



You need to play a bit with all of those settings to achieve similar results, which you can get by default using Unity without HDRP.

4. Edit the **Procedural Sky** (Skybox)

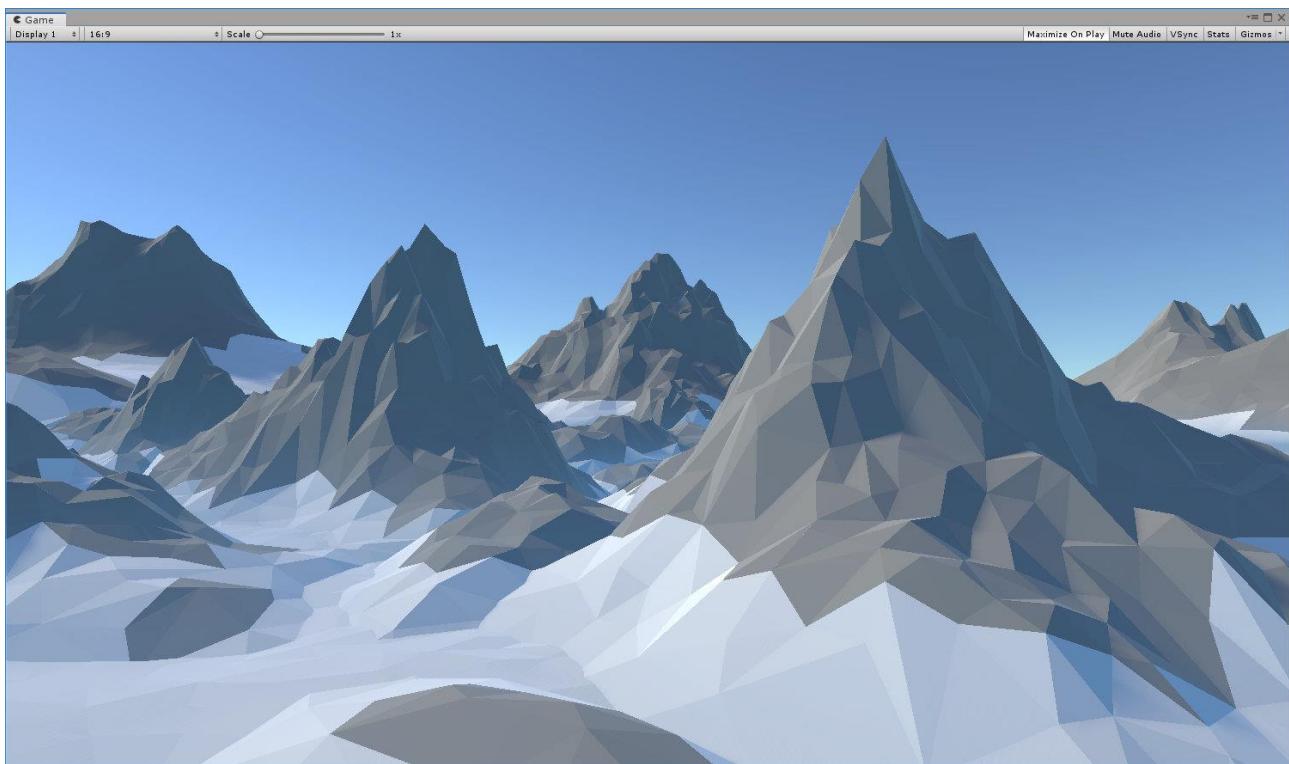
Use my **Procedural Skybox** settings:



Sky Tint (Color code): 68A4C3

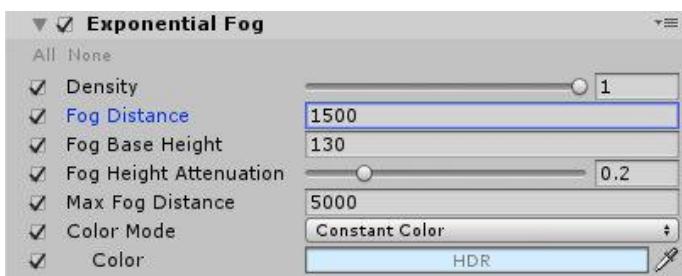
Ground Color (Color code): 5FAACF

to achieve this:



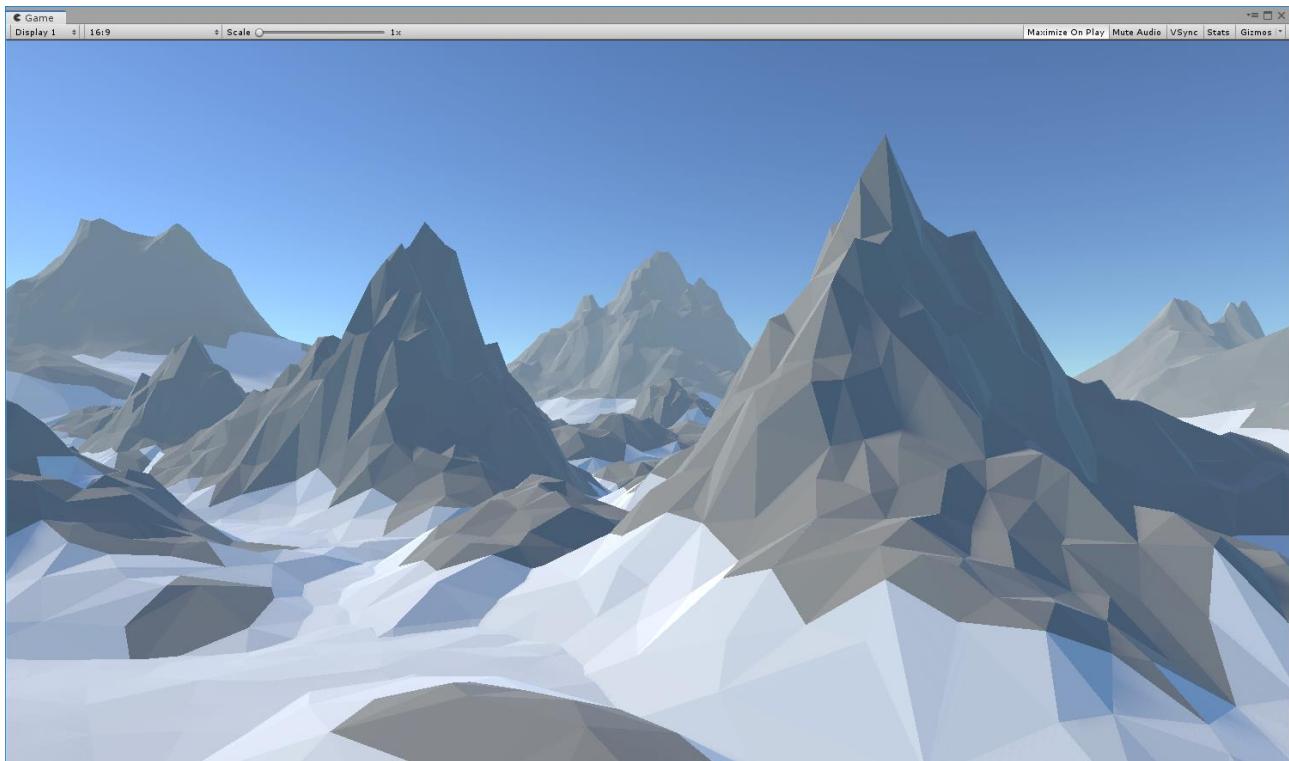
5. Edit the Exponential Fog

Use my **Exponential Fog** settings:



Set **Color Mode** to **Constant Color** and use this **Color** (R: 165; G:215; B:255)

Final result

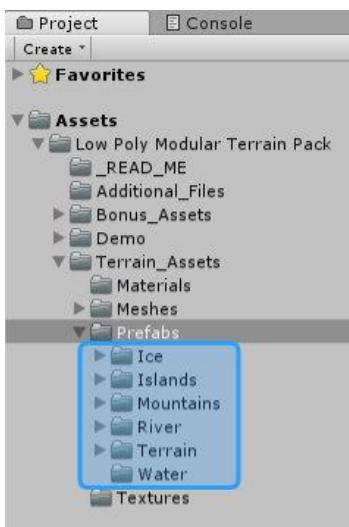


How to use “Low Poly Modular Terrain Pack”

Watch [VIDEO TUTORIAL](#)! Or follow the steps below.

Go to *Assets > Low Poly Modular Terrain Pack > Terrain_Assets > Prefabs*

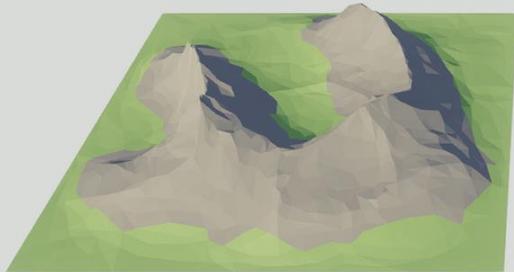
Choose which **Prefab** type you want to import to your scene:



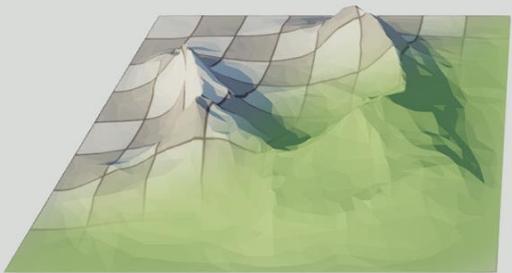
For example, open folder **Terrain**. You will see that you can choose between **3** types of Terrain:



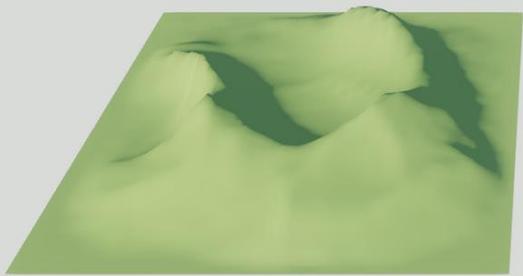
CPT – Color Palete Texture. All CPT prefabs use 1 material + 1 color palete texture atlas 64x64.



MT – Material & Texture. All MT prefabs use 1 material. You can also apply a seamless texture.



U – Unity Terrain. You can edit the terrain shape, paint textures, draw grass, trees, etc.



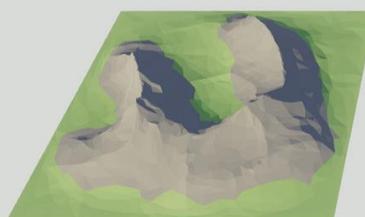
Open folder **CPT**. You can choose between **LOD** and **NoLOD**:



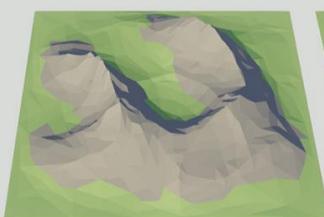
2 Versions of all Terrain prefabs

LOD - meshes with LODs

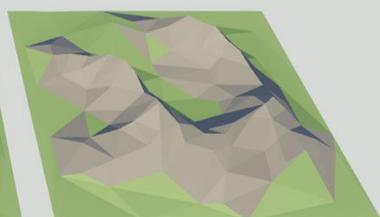
NoLOD - meshes without LODs



LOD0

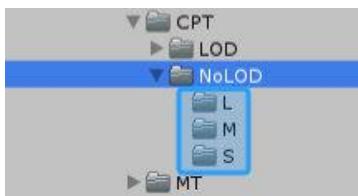


LOD1

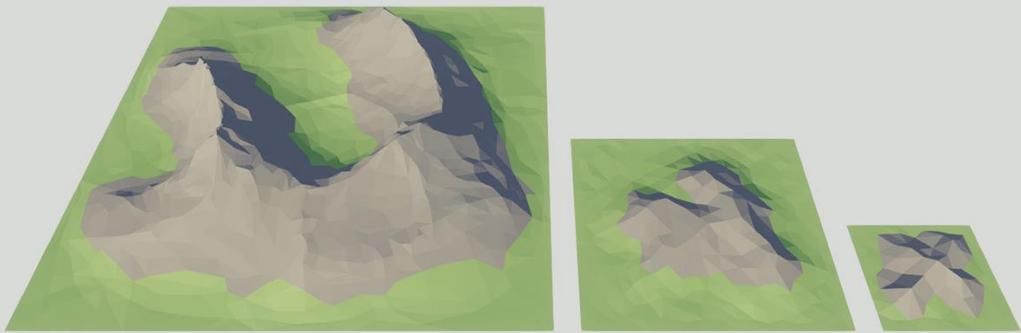


LOD2

Open folder **NoLOD**. You will see that you can choose between **L**, **M**, and **S**:



3 Terrain sizes



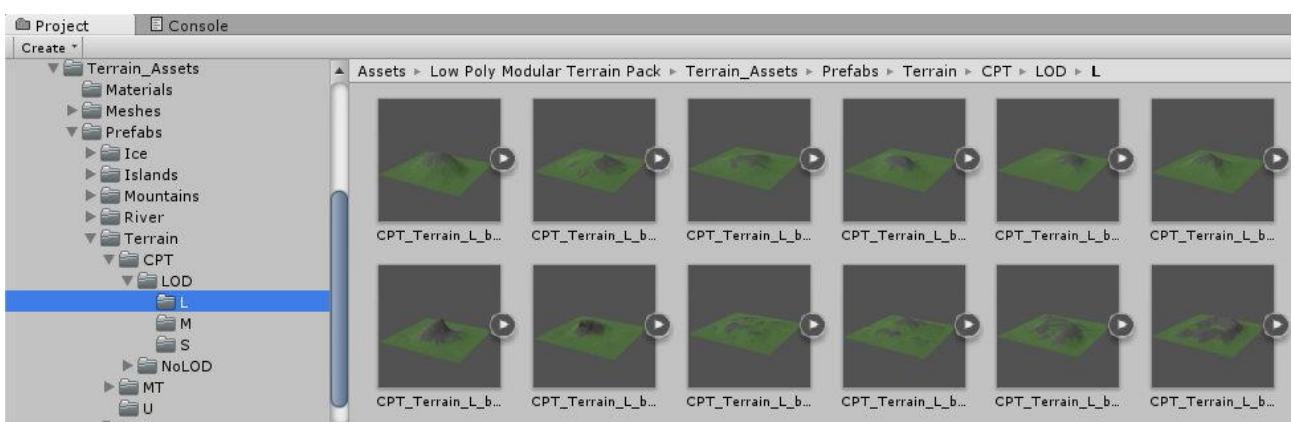
L - Large
100m x 100m

M - Medium
50m x 50m

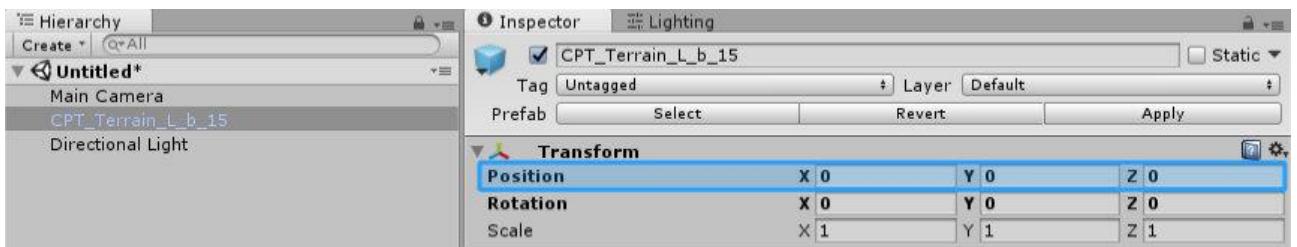
S - Small
25m x 25m

***L** – Large terrain is 100x100 meters (100x100 Unity units).

Let's open folder **L**, select and drag **Prefab** to your scene.



With a prefab selected in the Scene, **Reset Transform** (position to 0,0,0) so the Terrain will sit on the grid perfectly.



*I recommend drag and drop Prefabs straight to the **Hierarchy** tab. That way, you don't need to **Reset Transform** because it should be at **Position 0,0,0**.

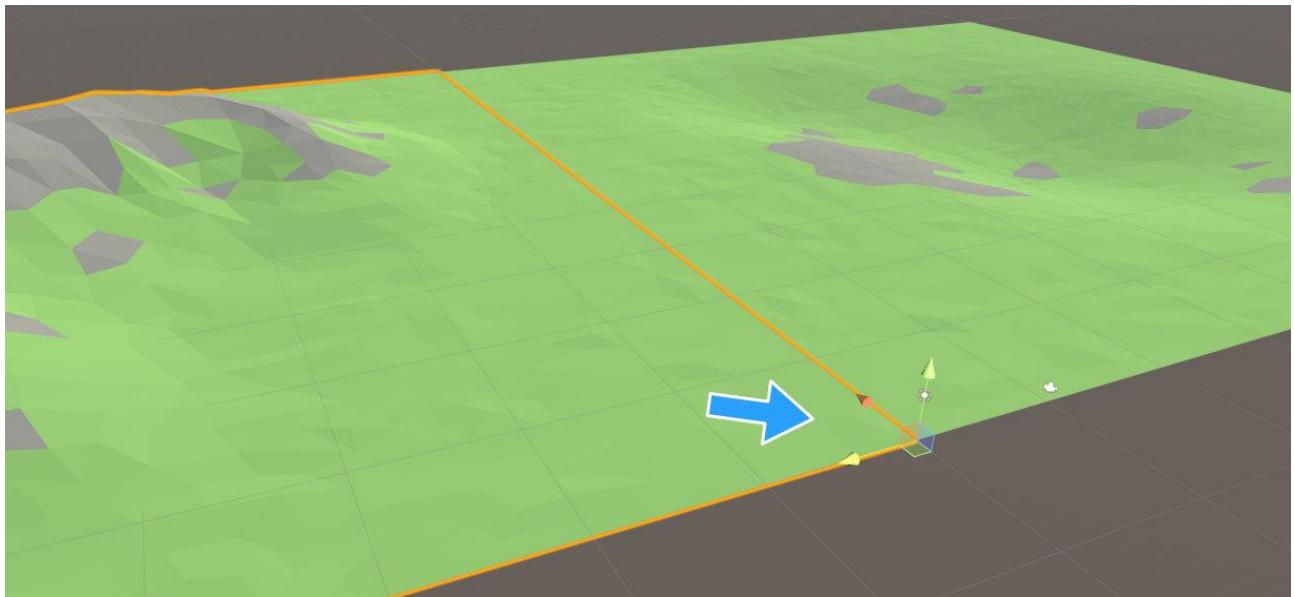
Every model pivot is at the bottom corner of the model, so you can quickly drop Terrain to the Scene, duplicate or add another one, move and snap to the grid.

*Use **Pivot** and **Global** settings for the best experience!

You can change it by tapping on the **buttons**, which are near Move, Scale tools.

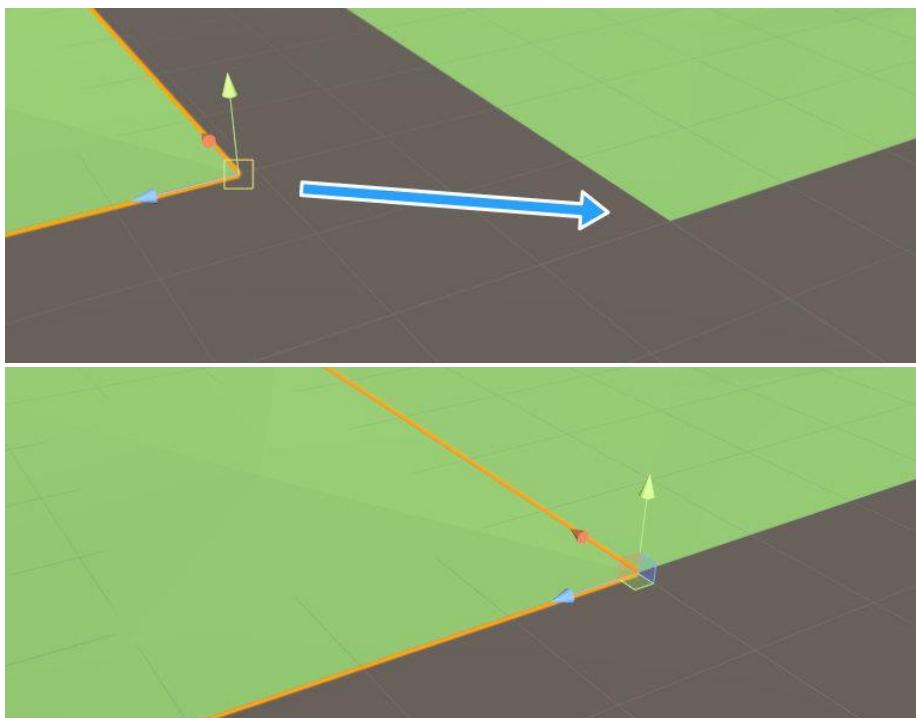


Hold CTRL + Grab and Drag the Transform arrow to snap Terrain to the Grid while moving it.



***L** – Large terrain is **100x100** units size, so you can easily change Position by **100** units to tile Terrain planes perfectly together.

You can also snap Terrain planes by using **V**. **Hold V** and hover the mouse cursor on the Terrain corner, and you will see a little **Yellow Square** (it shows which vertex of the mesh is selected for snapping). By **Holding V**, press and **Hold Left Mouse Button** and move it to the other Terrain corner to snap.



*I recommend using **Polybrush** for texture painting on **MT Terrain**. You can also draw prefabs (rocks, trees, grass, etc.) on any mesh terrain using Polybrush!

To use Bonus Assets

Go to *Assets > Low Poly Modular Terrain Pack > Bonus_Assets > Prefabs*

Select prefab you want and drag it to the scene.

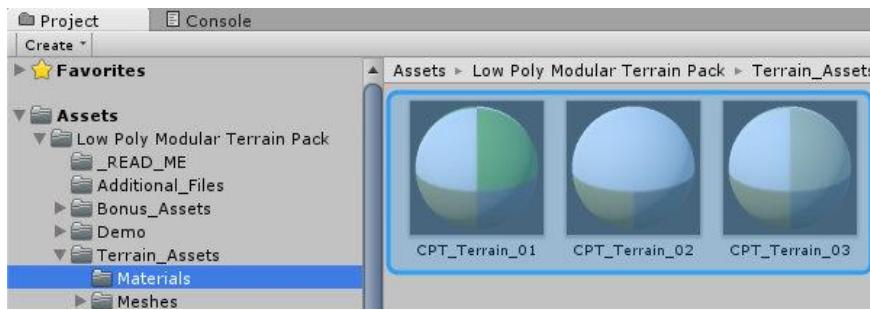
How to Change Prefab Color / Texture

CPT Terrain, Mountains, Islands, River

CPT – Color Palette Texture. All CPT prefabs use 1 material + 1 color palette texture atlas 64x64.

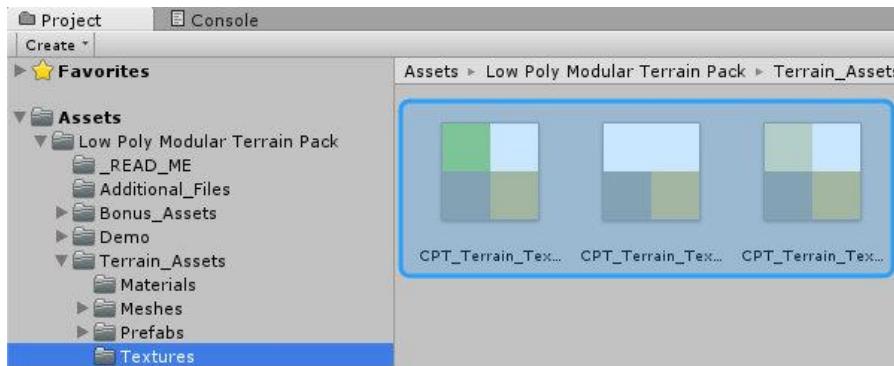
Watch [VIDEO TUTORIAL](#) at 05:17 or follow the steps below.

Go to *Low Poly Modular Terrain Pack > Terrain_Assets > Materials* - here, you will find **3 Materials** that are used for all **CPT** prefabs (CPT Terrain, Mountains, Islands, and River).



Material **CPT_Terrain_01** is applied to all **CPT** prefabs. To change the colors of CPT Prefabs, you can apply one of 3 CPT materials, or you can edit the Texture colors.

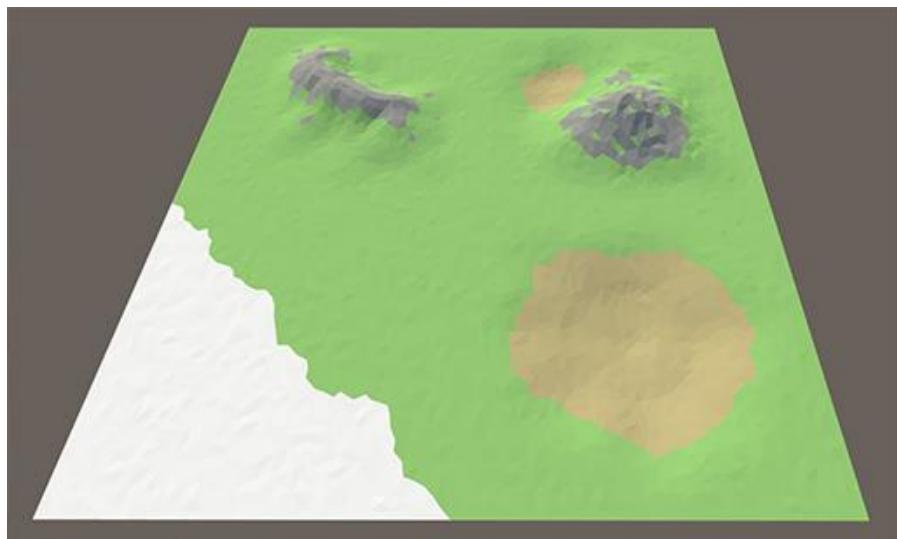
Go to *Low Poly Modular Terrain Pack > Terrain_Assets > Textures* - here, you will find **3 Textures**.



CPT_Terrain_Texture_Atlas_01.png is applied to **CPT_Terrain_01** Material. You can open this texture inside any image editing software and change the colors. There are only 4 colors on the texture:



These Colors are used like this:



There are 3 high-resolution **.PSD** textures included to edit it more easily. Go to *Low Poly Modular Terrain Pack > Additional_Files* - here, you will find the **CPT_ Textures.rar** file. Extract it, open any of 3 textures inside Photoshop, Affinity, or any other image editing software. Then save at a small resolution like 64x64 and import to your Unity project.

MT Terrain, Mountains, Islands, River

MT – Material & Texture. All MT prefabs use 1 material. You can also apply a seamless texture.

Watch [VIDEO TUTORIAL](#) at 11:35 or follow the steps below.

Go to *Low Poly Modular Terrain Pack > Terrain_Assets > Materials* - here you will find **6** materials which are used for **MT** Prefabs:



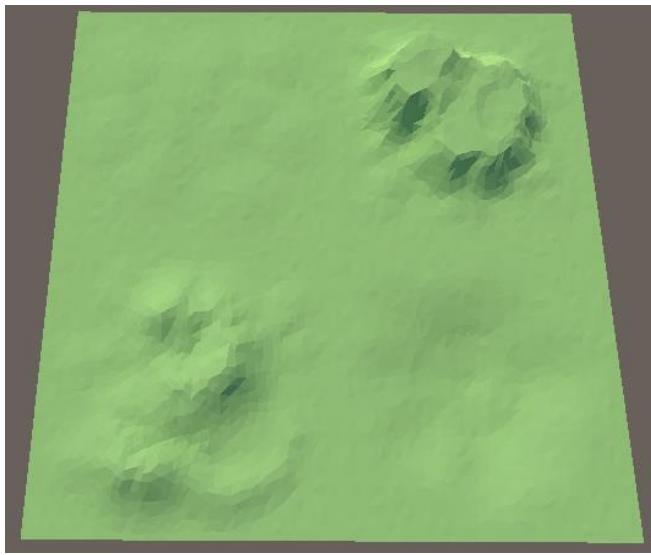
MT_Terrain_01 is applied to almost all **MT** Prefabs by default. To change the color of **MT Prefabs**, apply any of these materials, select material and change **Albedo** color:



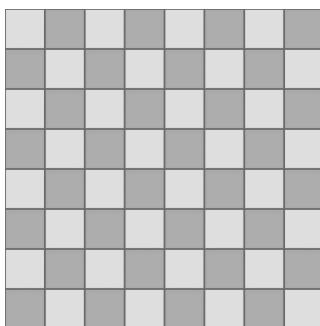
MT Terrain Texture

Watch [VIDEO TUTORIAL](#) at 12:40 or follow the steps below.

You can apply any tileable **Texture** to all **MT** Prefabs. Here are **4 MT_Terrain** Prefabs added to the Scene (Terrain use default **MT_Terrain_01** material):



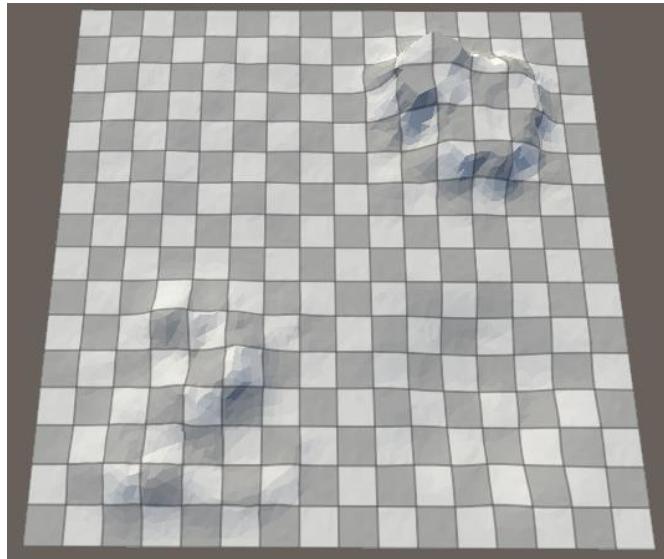
I will apply a simple Grid Texture to the **MT_Terrain_01** Material (**Albedo** slot):



And change material **Albedo** color to **White**:



Now Terrain looks like this:



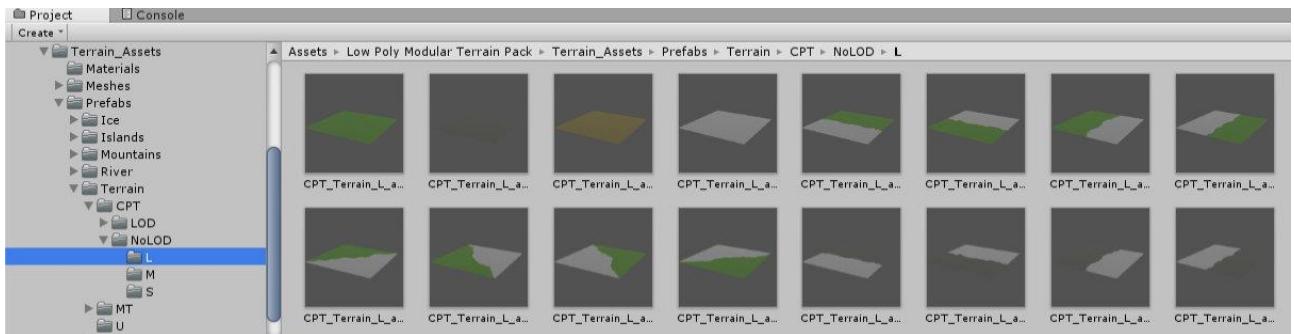
U Terrain

U – Unity Terrain. You can edit the terrain shape, paint textures, draw grass, trees, etc.

Watch [**VIDEO TUTORIAL**](#) on how to use it at **13:45**

How to Use CPT Terrain Transition Parts

If you go to *Assets > Low Poly Modular Terrain Pack > Terrain_Assets > Prefabs > Terrain > CPT > NoLOD > L* – you will see there are a bunch of terrain transition parts. They can be used to make a transition from a grass to a snow terrain, mud to grass, and so on.



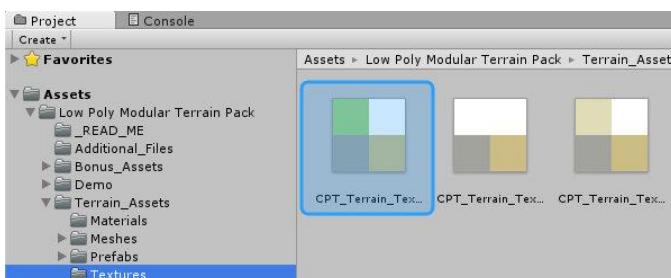
Here is an example of 3 MT Terrain prefabs in the scene. Terrain prefab on the left, transition Terrain in the center, and another Terrain on the right.



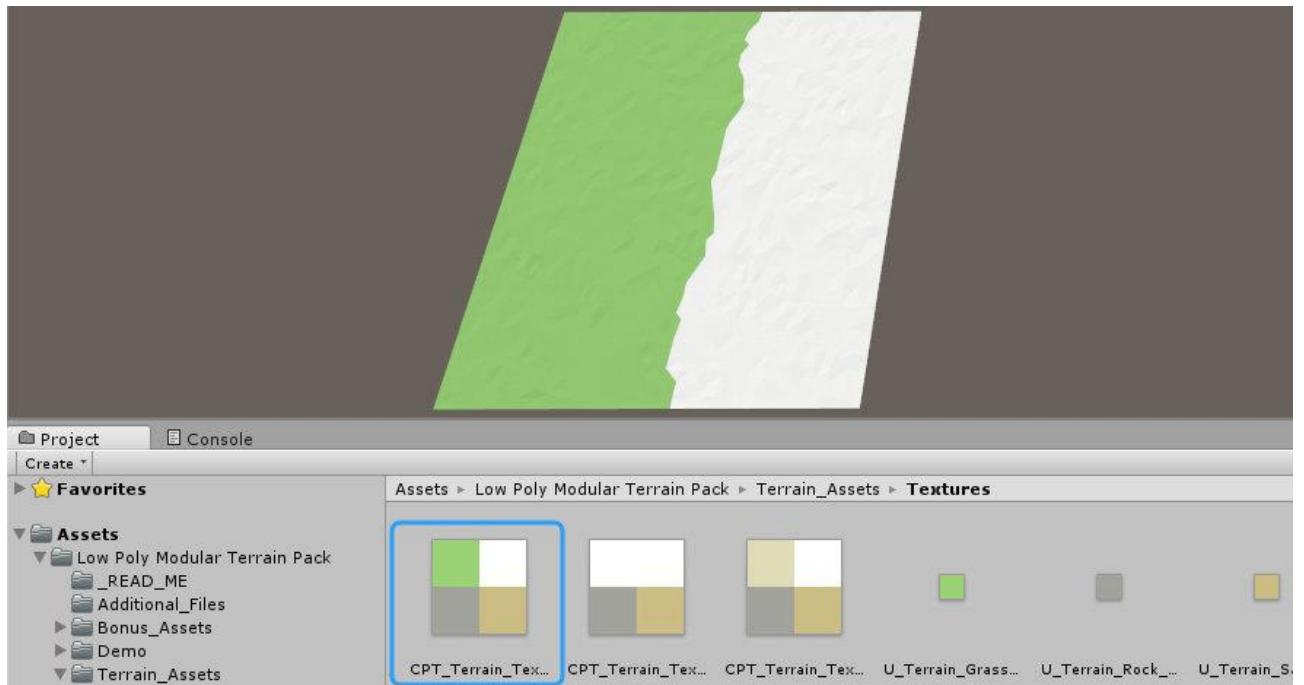
Change Transition Terrain Color

Go to *Assets > Low Poly Modular Terrain Pack > Terrain_Assets > Textures* –

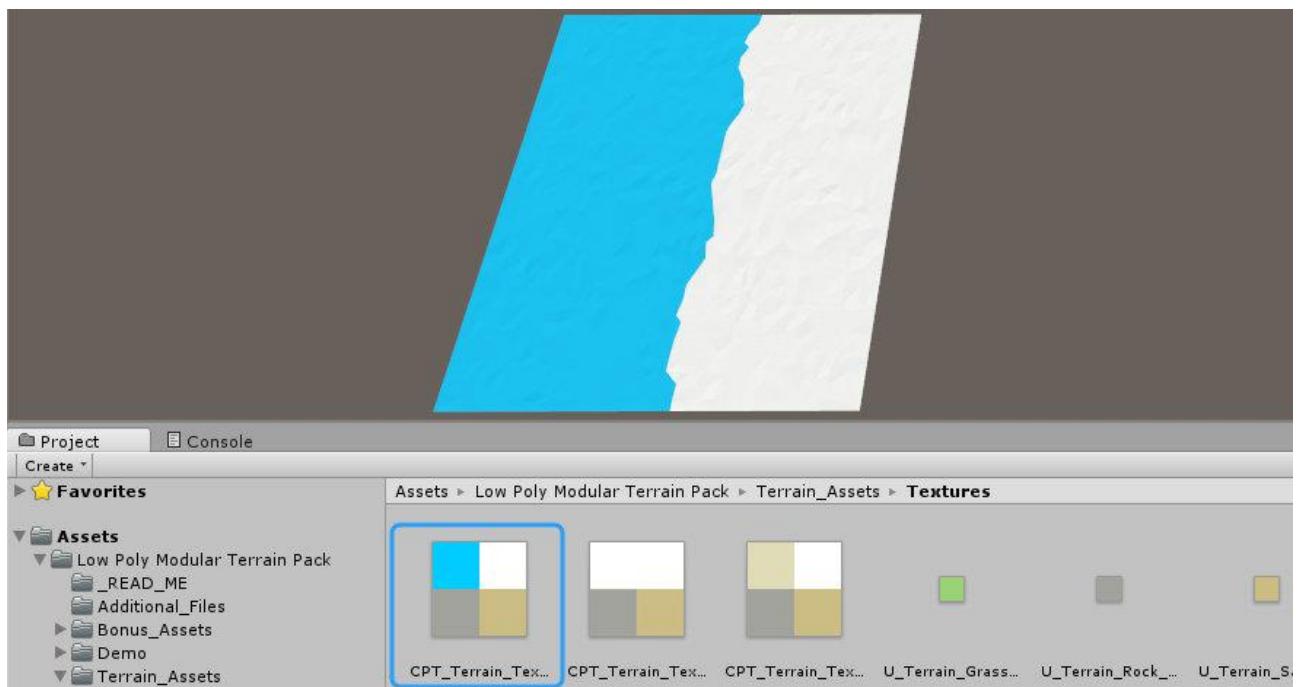
CPT_Terrain_Texture_Atlas_01.png is used for all Terrain transition parts by default.



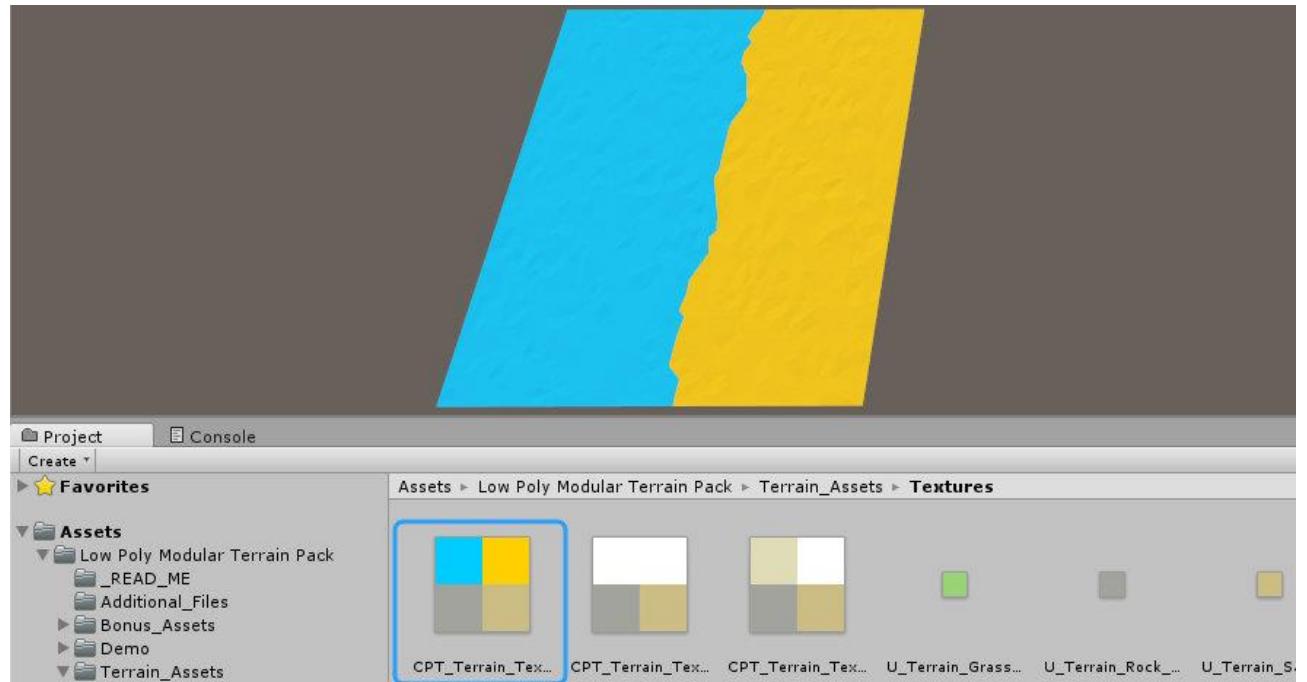
So open **CPT_Terrain_Texture_Atlas_01.png** in any image editing software and change the color you want. Here is the original color:



Changed the first color block to **Blue**:



Changed the second color block to **Yellow**:

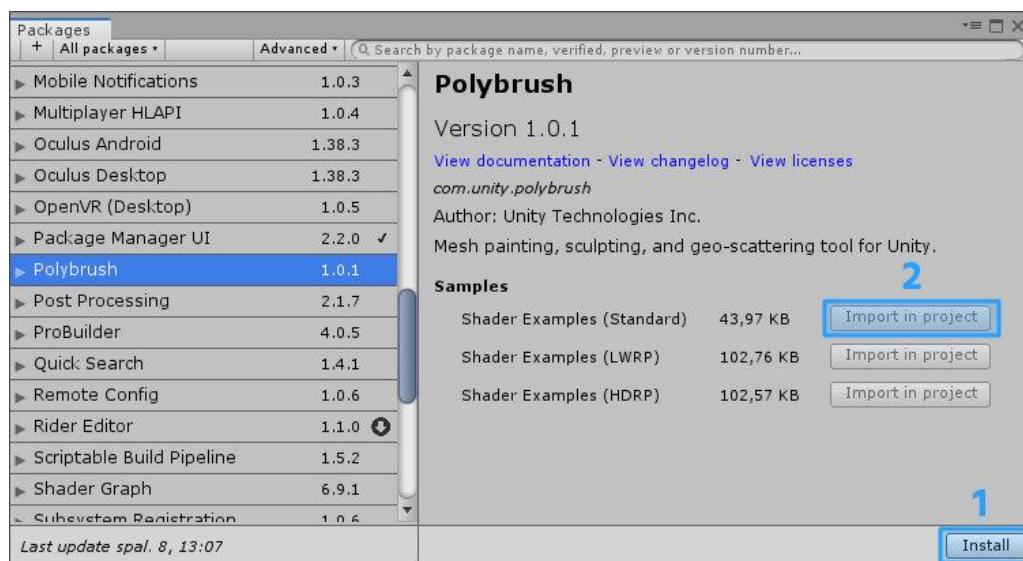


How to Paint Vertex Color And Textures on MT Terrain Using Polybrush

*To use Polybrush - you need at least **Unity 2017.1.4**

1. Import Polybrush.

If you are using **Unity 2019.1** and above, you can install **Polybrush** from the **Package Manager!** *Window > Package Manager – Polybrush*. Click on **Install**, after installing it, click on **Import in project** (to import example shaders we need to paint on terrain mesh).



(Optional) If you are using older Unity 2017.1.4 up to 2019.1, import Polybrush from the Asset Store.

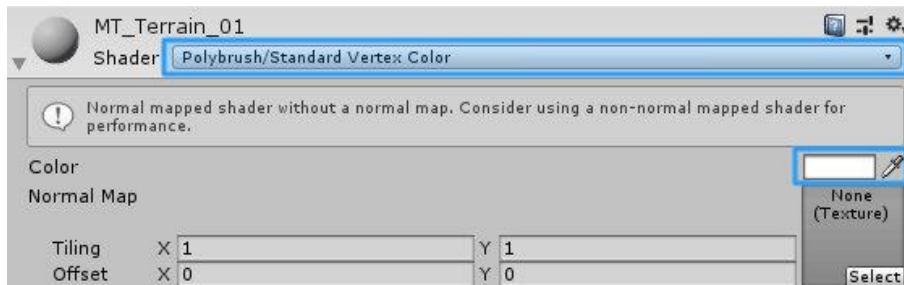
Go to *Window > Asset Store* and search for **Polybrush**:



Import it to your project.

2. Setup MT Terrain for Painting Vertex Color

Go to *Low Poly Modular Terrain Pack > Terrain_Assets > Prefabs > Terrain > MT > NoLOD > L -* and import any **MT terrain** to the scene. Select **MT Terrain** from your scene and change material Shader to **Polybrush/Standard Vertex Color** and set **Color** to **White**:

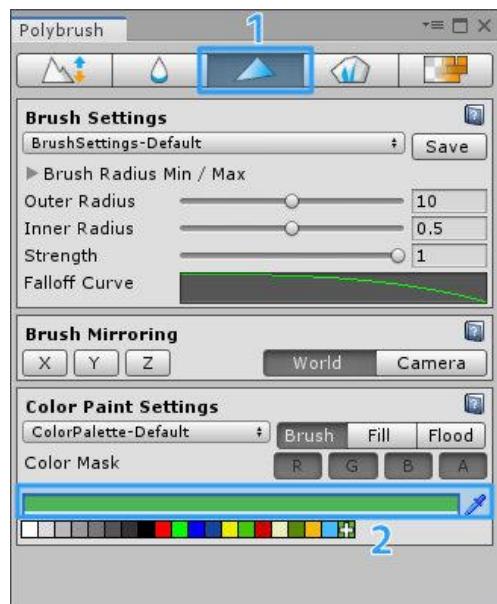


3. Setup and use Polybrush to paint Vertex Color on any MT prefab.

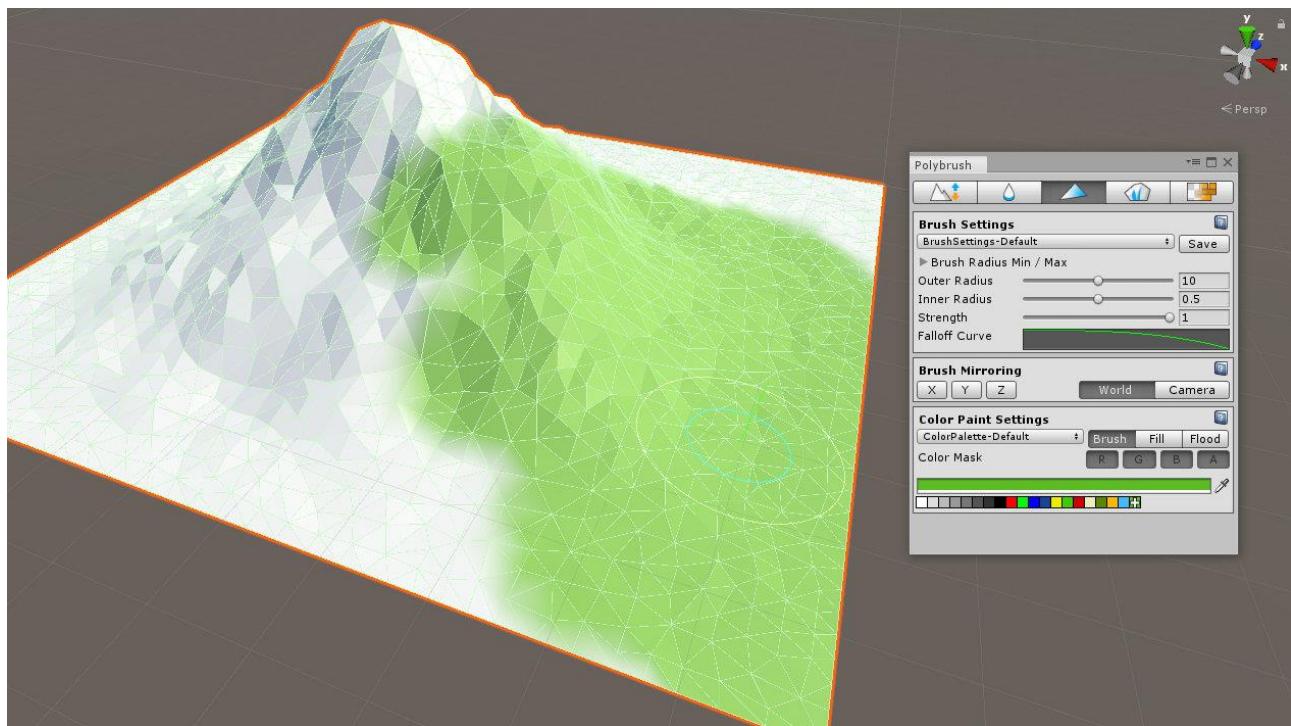
Go to *Tools > Polybrush > Polybrush Window*

You should see a **Polybrush** window.

1. Open Paint vertex colors on meshes tab
2. Choose a **Color** to paint

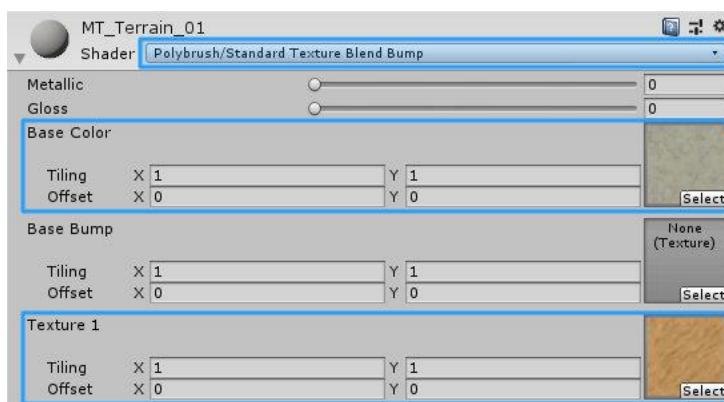


3. Select **MT Terrain** in the scene and **Paint**



4. Setup **MT Terrain** and use **Polybrush** to paint **Textures** on.

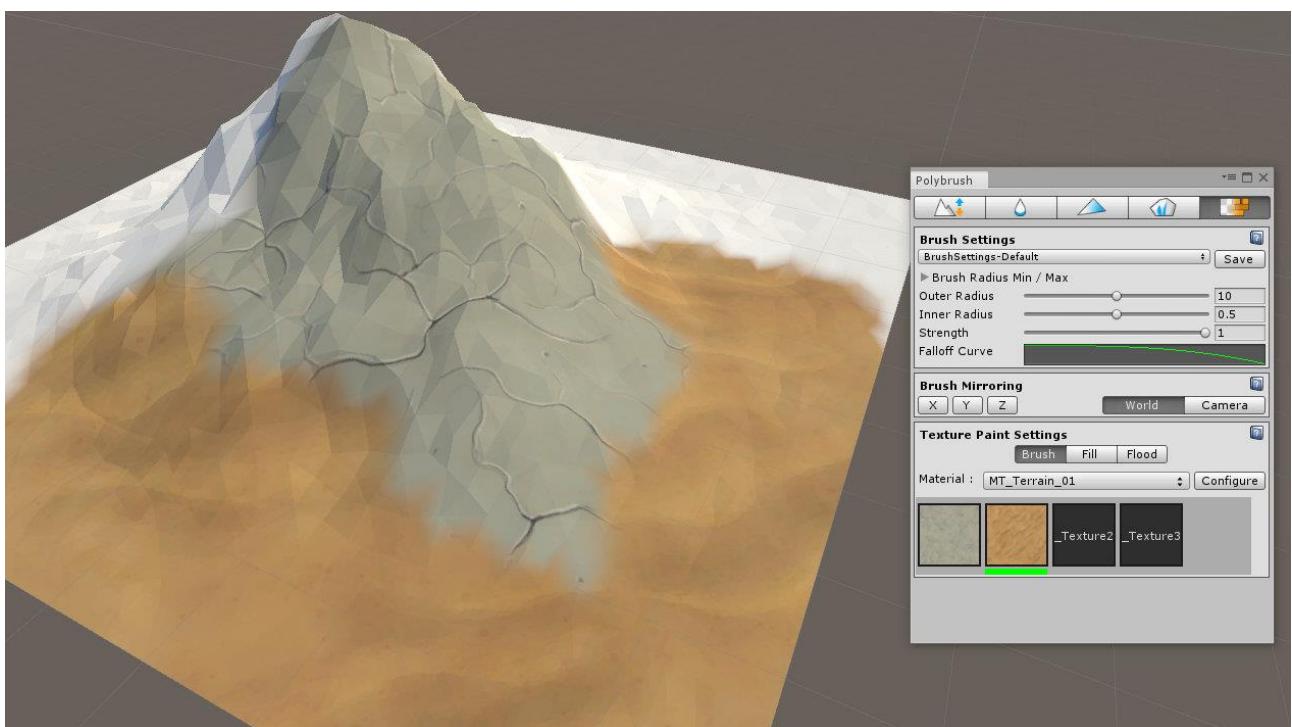
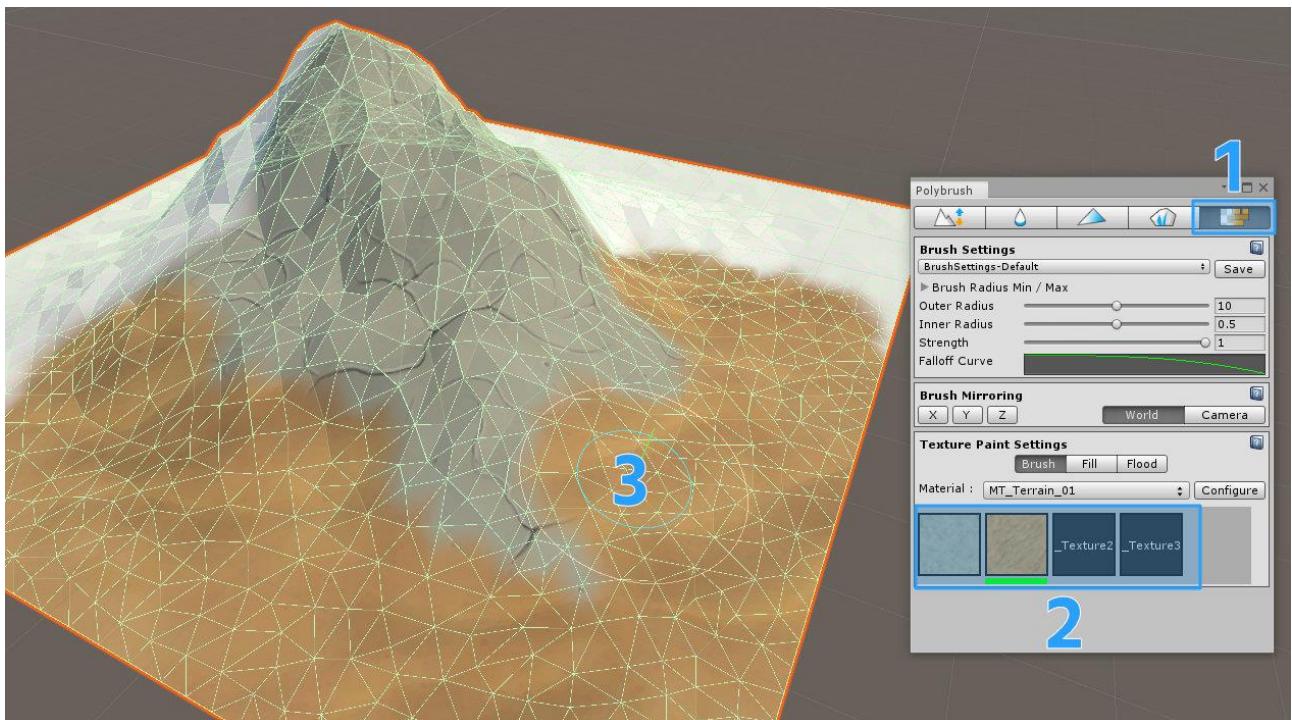
Select **MT Terrain** from your scene and change material Shader to **Polybrush/Standard Texture Blend Bump**. Apply **Textures** you want to paint as **Base Color**, **Texture 1**, **Texture 2**, etc.



*Textures I've used in this demo are not included in the package!

Now inside Polybrush

1. Open Paint textures on meshes tab
2. Choose the Texture to paint
3. Paint on the MT Terrain



Additional Info

Naming Conventions

Prefab name example 1: **CPT_Terrain_L_b_27**

- **CPT** – Color Palette Texture (All CPT Prefabs use 1 Material + 1 Color Palette Texture Atlas 64x64)
- **L** – Large 100m x 100m Terrain Size
- **b** – Just a Terrain type letter,
- **27** – Prefab number

Prefab name example 2: **Ice_H_BT_01**

- **H** – Huge size
- **BT** – With Bottom (ice has bottom faces and can be seen from both sides)

The same Terrain Prefabs comes in 3 different types:

- **CPT** – Color Palette Texture (All CPT Prefabs use 1 Material + 1 Color Palette Texture Atlas 64x64)
- **MT** – Material and Texture (All MT Prefabs use 1 Material. You can also add any seamless Texture to it!)
- **U** – Unity Terrain (You can edit the terrain shape, paint textures, draw grass, trees, etc.)

Almost all Prefabs come in 2 versions:

- **LOD** - Prefabs with 3 LOD levels: LOD0, LOD1, LOD2
- **NoLOD** –Mesh Prefabs without LODs

Prefabs come in 4 Sizes:

- **H** – Huge ~500x500m
- **L** - Large 100m x 100m Terrain Size
- **M** - Medium 50m x 50m Terrain Size
- **S** - Small – 25m x 25m Terrain Size

*Keep in mind that every terrain mesh is different, no matter is it small or large.

You can also find letter **R** at the end of the River part names. This means **Reversed!**

Scripts

Every scene **Camera**, **Directional Light**, and **_Clouds** (an empty game object which contains all clouds on the scene) have movement controls.

For, example, select **Camera** and on **Inspector** scroll down to the bottom, you will see **Modular Terrain Camera Control (Script)** attached to it. Here you can control **Camera Movement Speed** using sliders.



Same with **Direction Lights (Sun)** and **_Clouds**.

Contacts

If you have any questions, suggestions on what to improve or create. Maybe found any bugs, please send me an e-mail!

E-mail: justinas@lmhpoly.com

Website: <http://lmhpoly.com/contact/>

Follow me on **Twitter** to see what I'm working on right now:

<https://twitter.com/lmhpoly>



lmhpoly.com