

Low Poly Rocks 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:

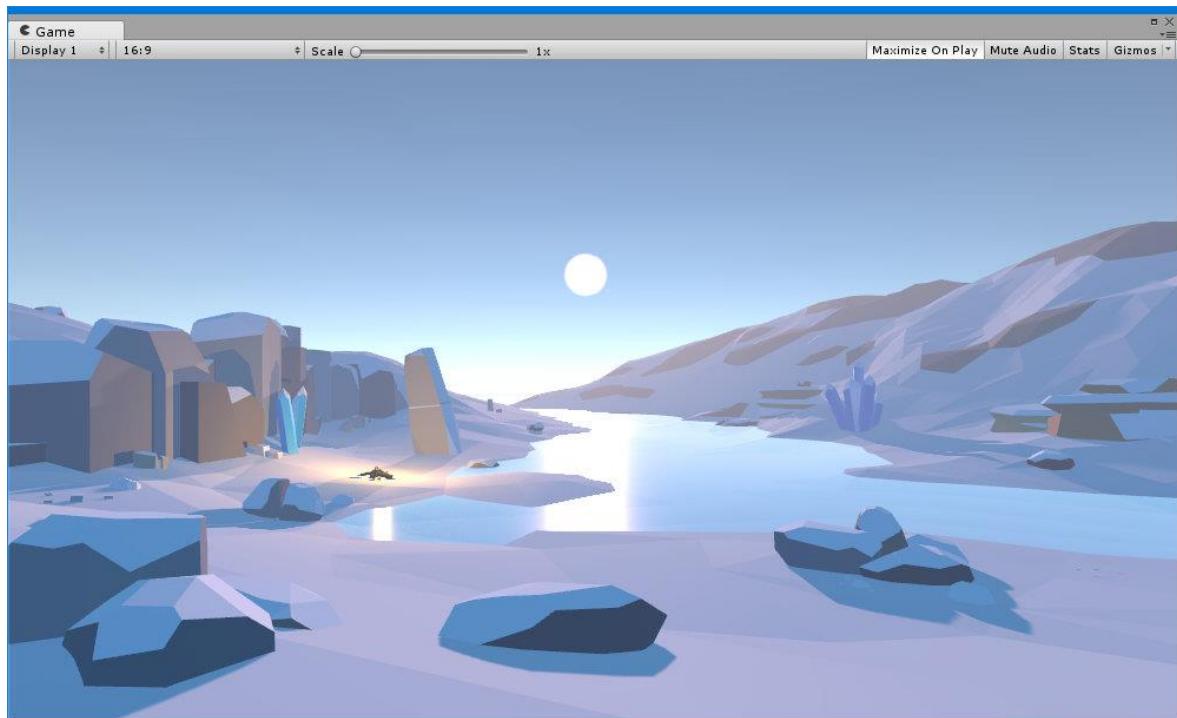
<https://twitter.com/lmhpoly>

Content

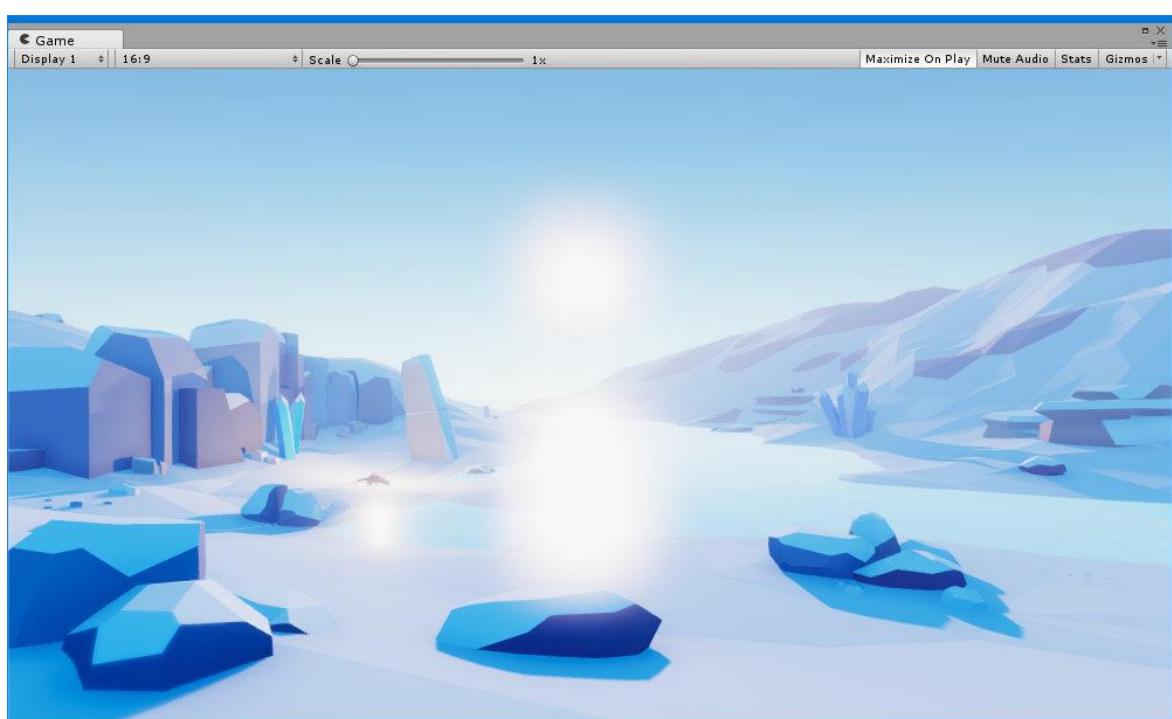
HOW TO SETUP DEMO SCENES (POST-PROCESSING) IN UNITY 5.6.0 - 2018.1 (FOR PC).....	5
HOW TO SETUP DEMO SCENES (POST-PROCESSING) IN UNITY 2018.1 AND UP (FOR PC).....	12
HOW TO SETUP DEMO SCENES IN UNITY 5.6.0 - 2018.1 (FOR ANDROID).....	21
HOW TO SETUP DEMO SCENES IN UNITY 2018.1 AND UP (FOR ANDROID).....	30
UNITY 2018.1 AND UP LIGHTWEIGHT RENDER PIPELINE (LWRP).....	39
UNITY 2018.1 AND UP HIGH DEFINITION RENDER PIPELINE (HDRP).....	43
HOW TO USE "LOW POLY ROCKS PACK"	52
HOW TO CHANGE PREFABS COLOR / TEXTURE.....	53
ROCKS / CRYSTALS.....	53
CHANGE CRYSTALS COLOR	53
CHANGE ROCKS COLOR.....	54
CHANGE THE SECOND COLOR FOR (2 COLOR - ROCK PREFABS)	55
CHANGE BONUS ASSETS COLOR.....	57
HOW TO DRAW ROCK PREFABS ON UNITY TERRAIN.....	59
HOW TO DRAW ROCK PREFABS ON MESH TERRAIN USING POLYBRUSH.....	61
ADDITIONAL INFO	64
NAMING CONVENTIONS.....	64
SCRIPTS.....	64
CONTACTS.....	66

Demo scenes

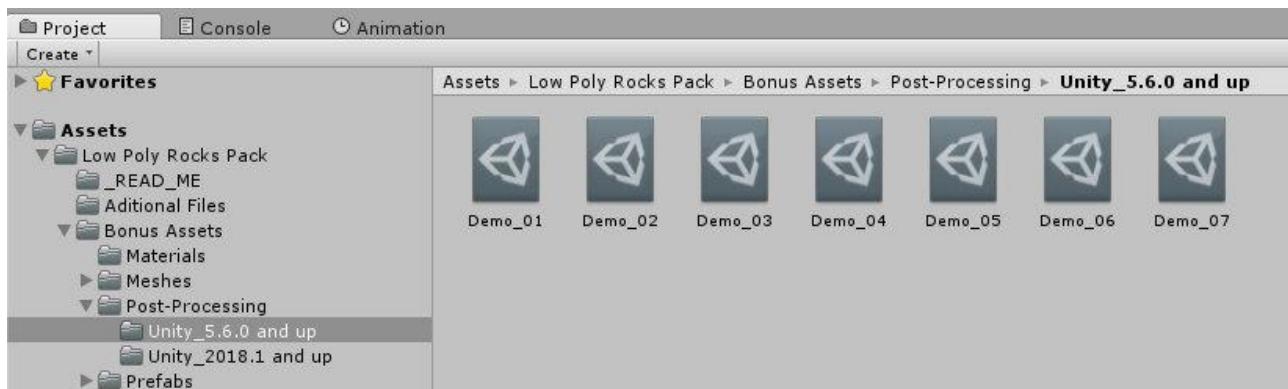
Now as you have imported the whole “**Low Poly Rocks Pack**” to your Unity project, go to *Low Poly Rocks Pack > Demo > Demo_Scenes* - and open any Demo Scene (here is a **Demo_07** example). By default, the scene should look like this inside the **Game** view without any image effects applied.



To make it look like this:



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



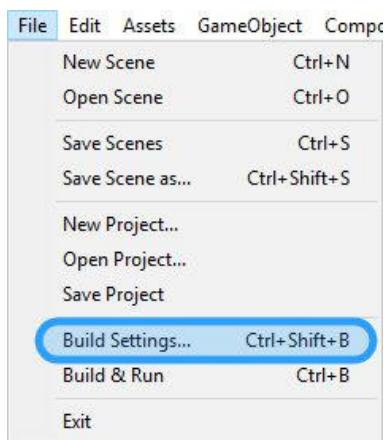
Follow the steps below to setup **Post-Processing** image effects for Demo Scenes!

- [Post-Processing in Unity 5.6.0 up to 2018.1](#)
- [Post-Processing in Unity 2018.1 and up](#)

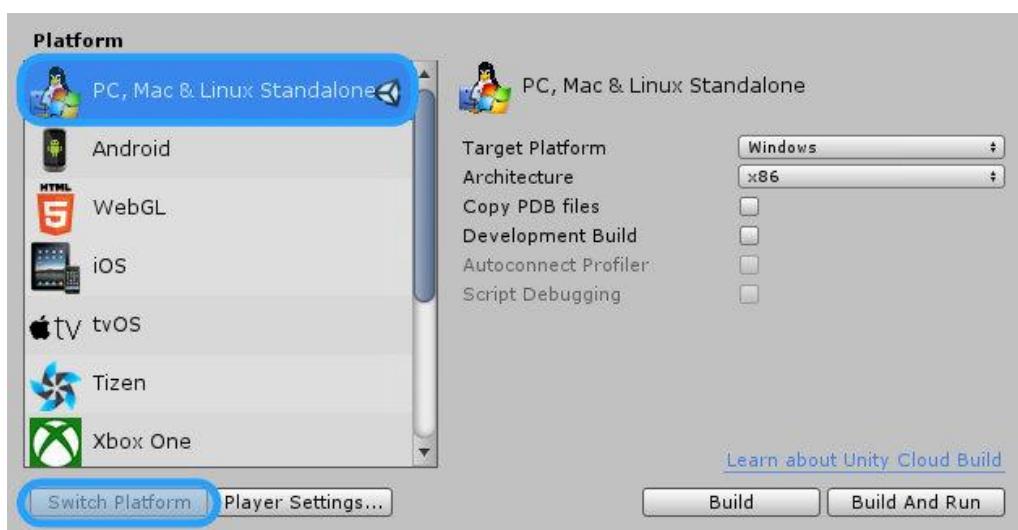
How to Setup Demo Scenes (Post-Processing) in Unity 5.6.0 - 2018.1 (For PC)

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

Go to *File > Build Settings*



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



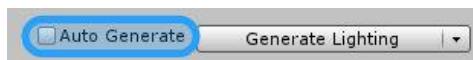
2. Clean GI Cache (Optional – Skip this if you don't have any light baking errors!).

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

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

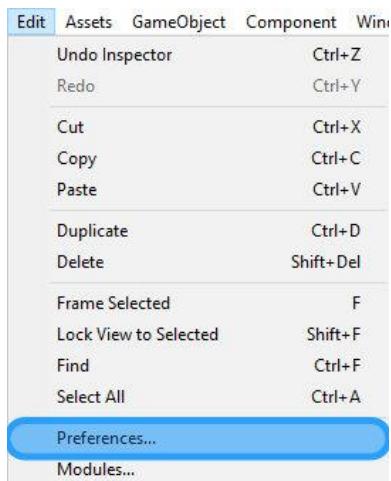


At the bottom you will see this:

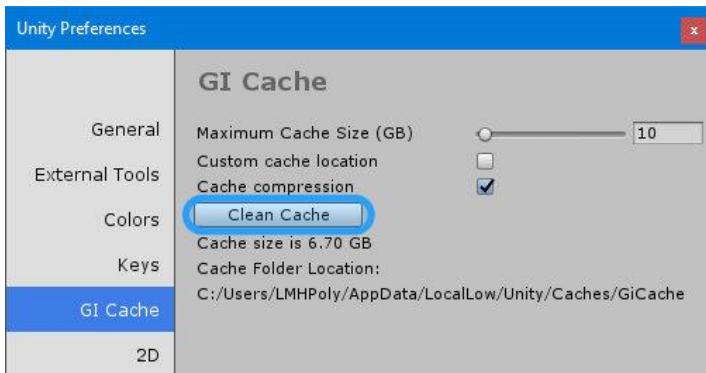


unchecked **Auto Generate**.

Go to *Edit > Preferences*



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



Enable **Auto Generate / bake** feature

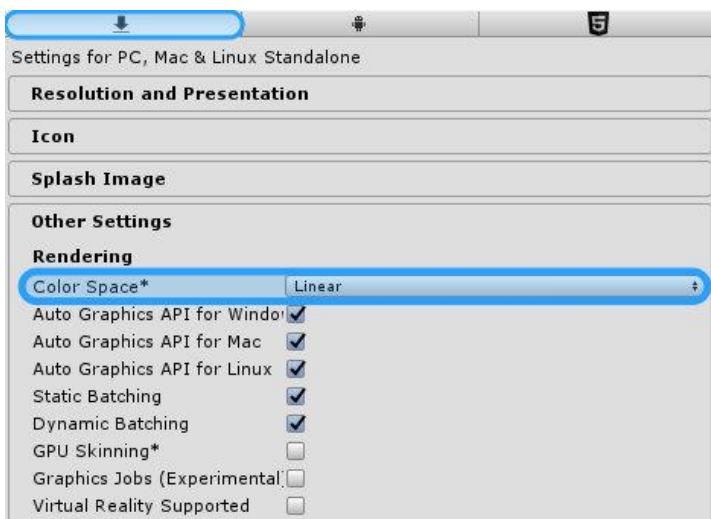


and wait until the generation is done (blue loading bar at the right bottom corner).

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**.

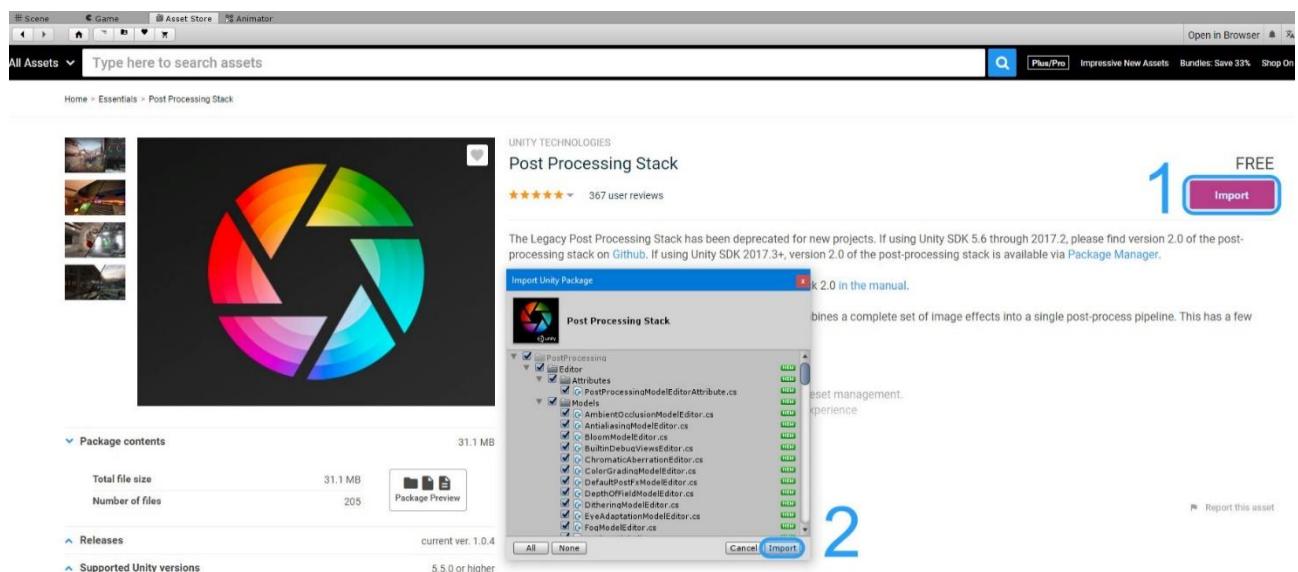


4. Import Post-Processing Stack.

Go to *Window > Asset Store* - and search for **Post Processing Stack**.

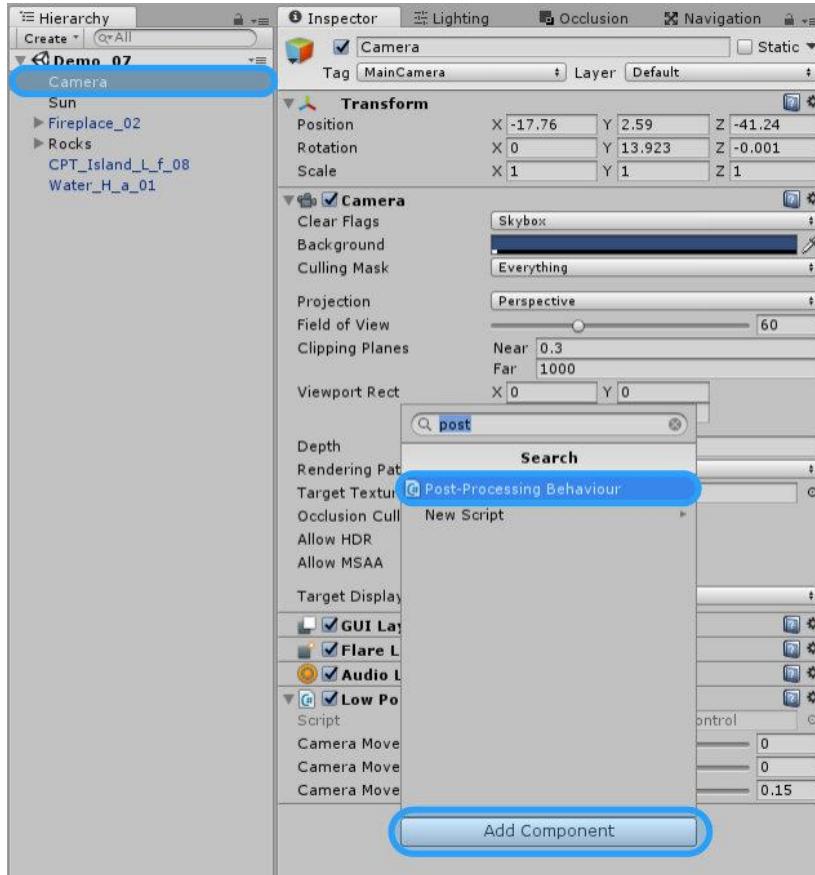


Download and Import it to your project.

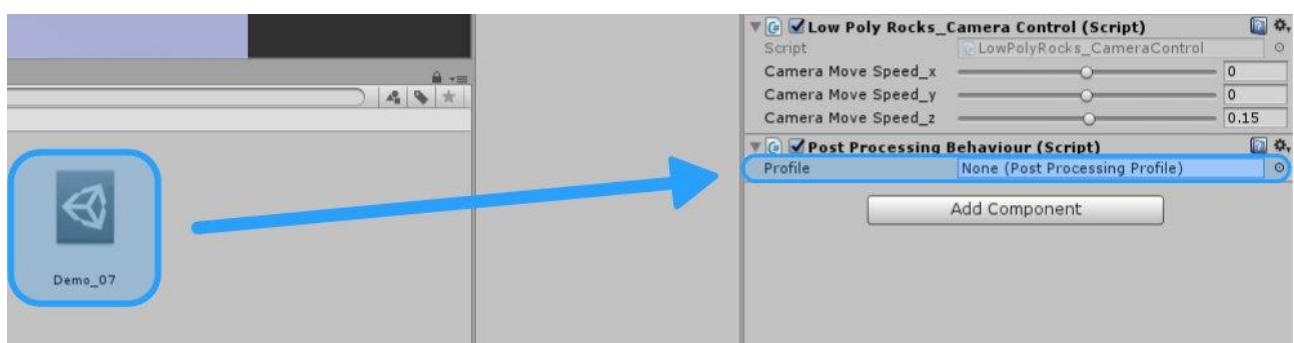


5. Enable Post-Processing Stack image effects.

Select the **Camera** of the demo scene in the Hierarchy and press on **Add Component**, search for **Post-Processing Behaviour**, and add it to the **Camera**.



Go to *Low Poly Rocks Pack > Bonus Assets > Post-Processing > Unity_5.6.0 and up*. Select the **Demo_07** post-processing profile (which I've created) and apply it to the **Post-Processing Behaviour** by dragging it and dropping to the **Profile** slot.



That's it! After that, you will see all the camera effects working as it should.

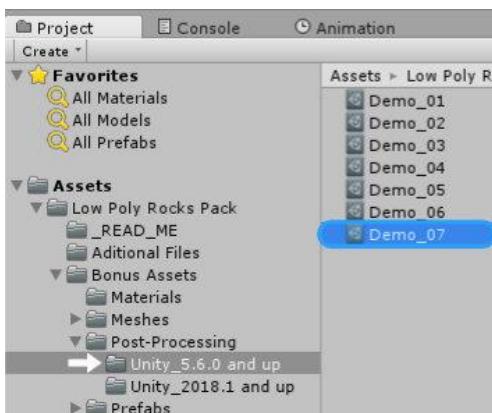
Now your scene should look like this (Demo_07):



Press Play and enjoy!

***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!

To edit **Post-Processing Profile** settings – go to *Low Poly Rocks Pack > Bonus Assets > Post_Processing > Unity_5.6 and up* - select **Demo** scene you want to edit:



Inside the **Inspector**, you will see effects like (Fog, Ambient Occlusion, Color Grading, etc.).

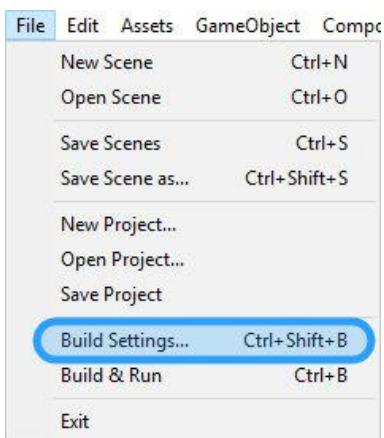
Open any tab and edit the setting how you like:



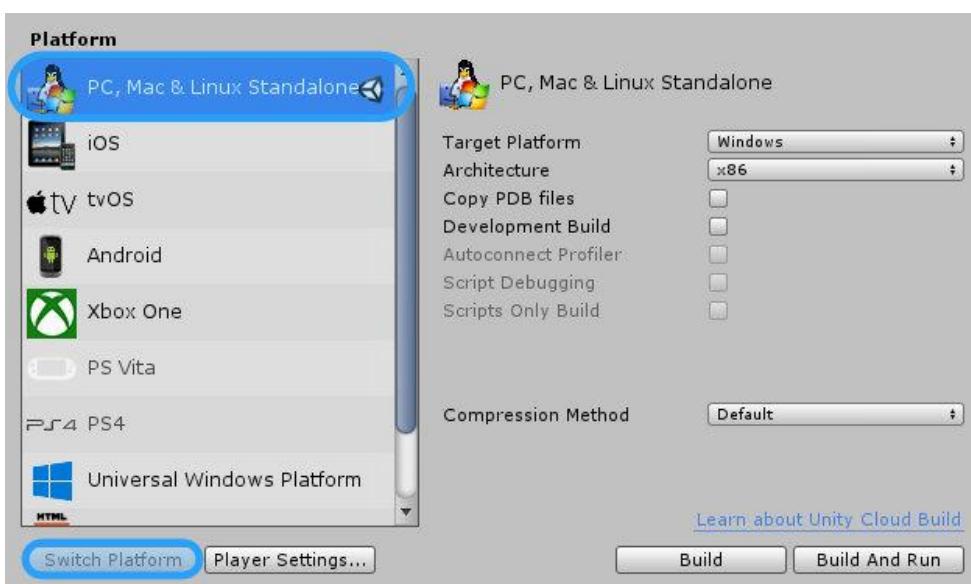
How to Setup Demo Scenes (Post-Processing) in Unity 2018.1 and up (For PC)

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

Go to *File > Build Settings*



Select **PC, Mac & Linux Standalone** and hit **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 **Auto** build/bake feature.

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



At the bottom you will see this:

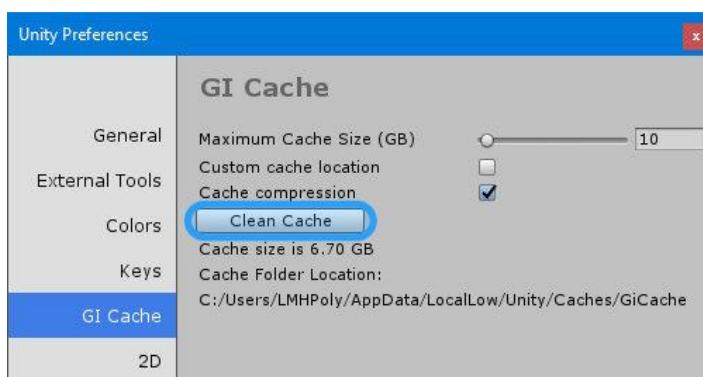


unchecked **Auto Generate**.

Go to *Edit > Preferences*



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



Enable **Auto Generate** / bake feature

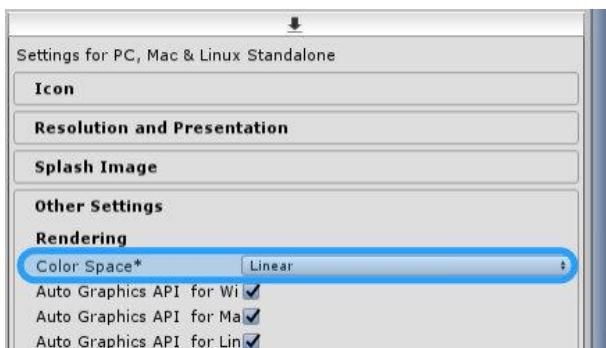


and wait until the generation is done (blue loading bar at the right bottom corner).

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

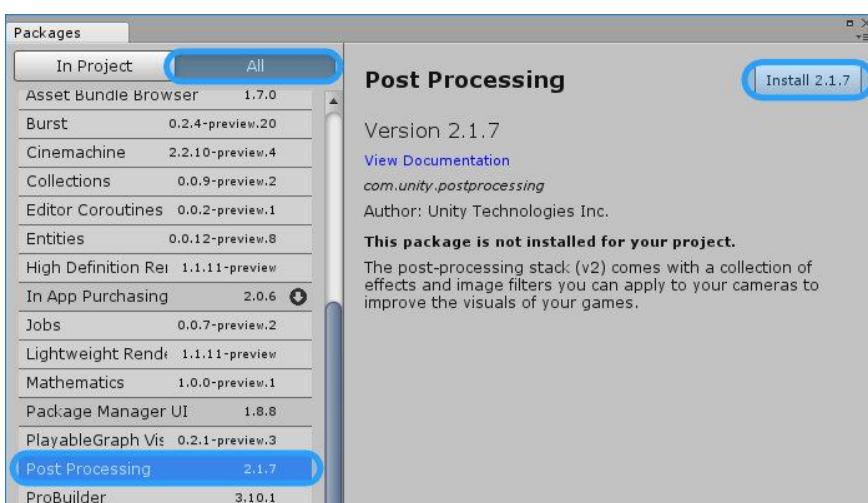
Go to *Edit > Project Settings > Player*

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



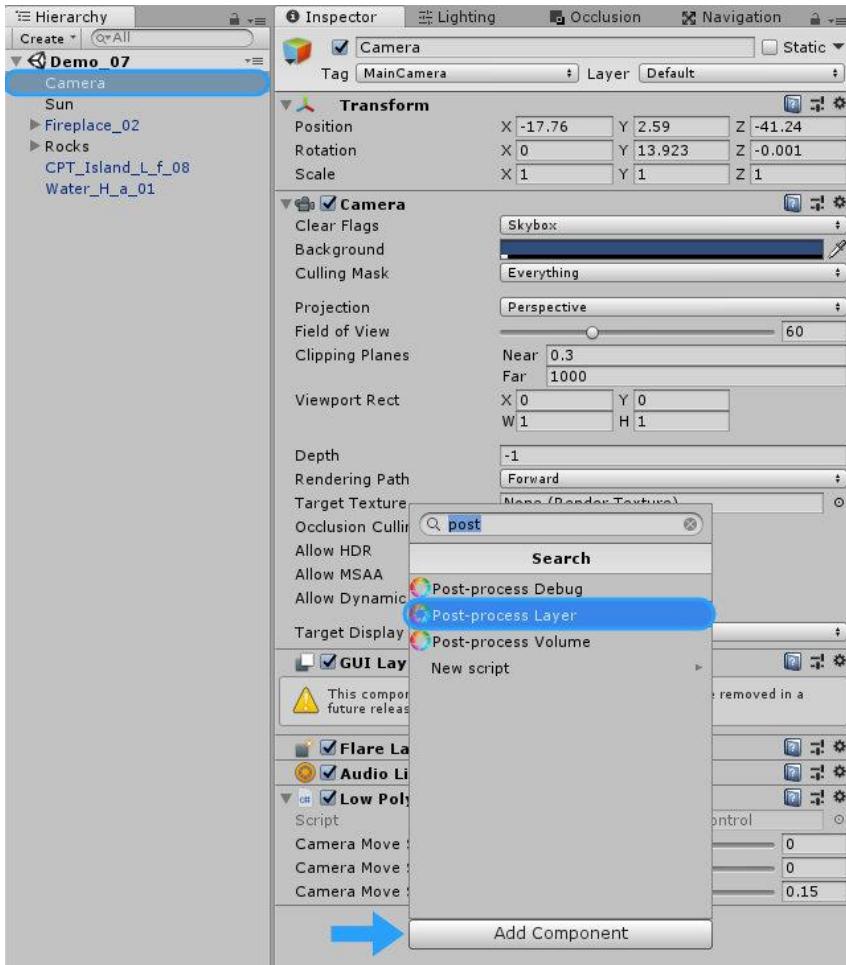
4. Install the **Post Processing** package from the **Package Manager**.

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.

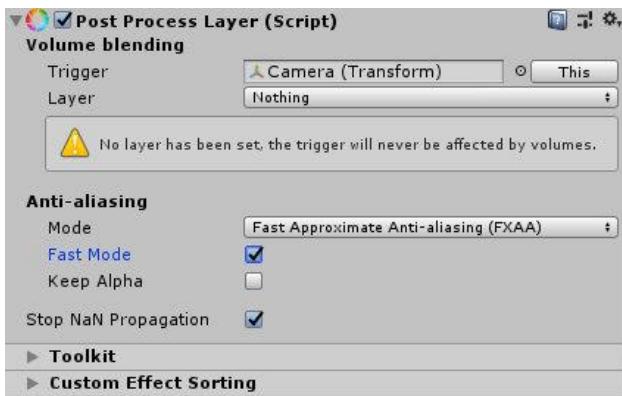
Select the **Camera** in the Hierarchy, press on **Add Component**, type **post** in the search window, and press on **Post-process Layer** to apply.



*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:

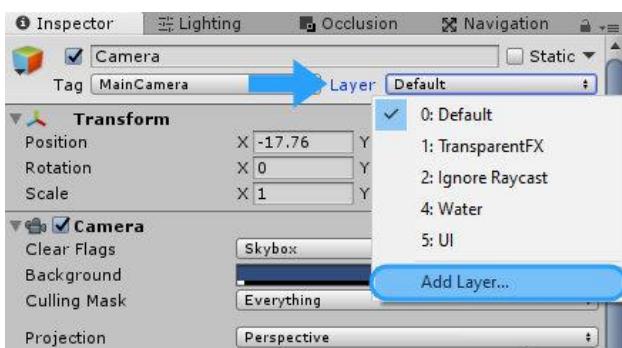
- Restart Unity.
 - If it still doesn't work, go to *Window > Package Manager*, remove **Post Processing** package.
 - Restart Unity
 - Install the **Post Processing** package again. Now it should work.

Post Process Layer settings:

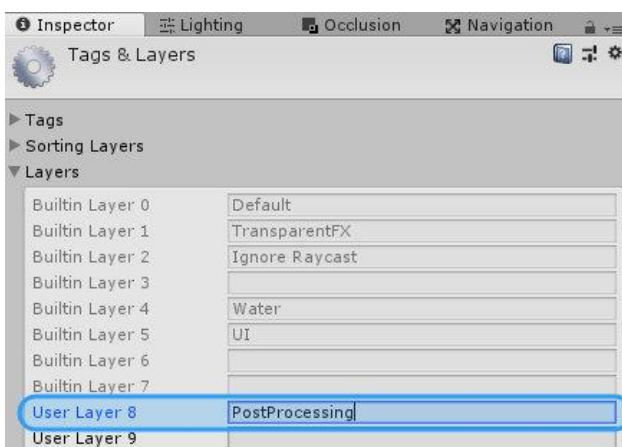


6. Create **PostProcessing** layer and apply it to the **Camera**.

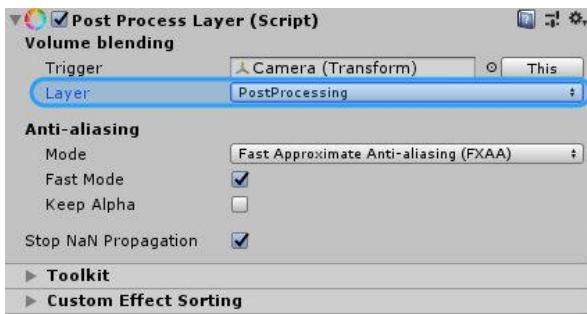
Press on **Layer Default > Add Layer...**



Create a new layer called **PostProcessing**:

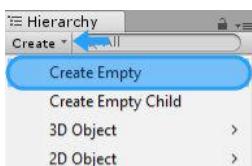


Select the **Camera** and inside the **Post Process Layer – Volume blending – Layer** apply **PostProcessing** layer we just created:

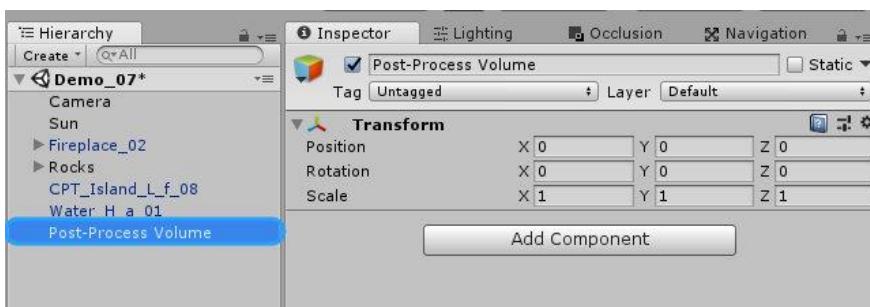


7. Create a Post-Process Volume.

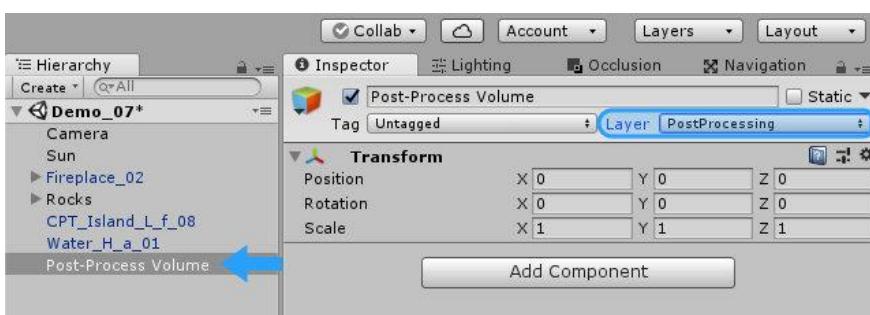
Create Empty game object:



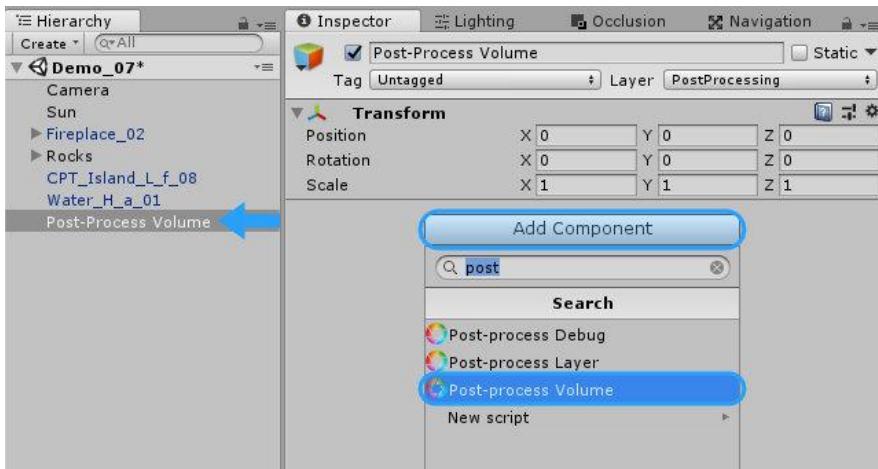
Rename it to something like **Post-Process Volume**:



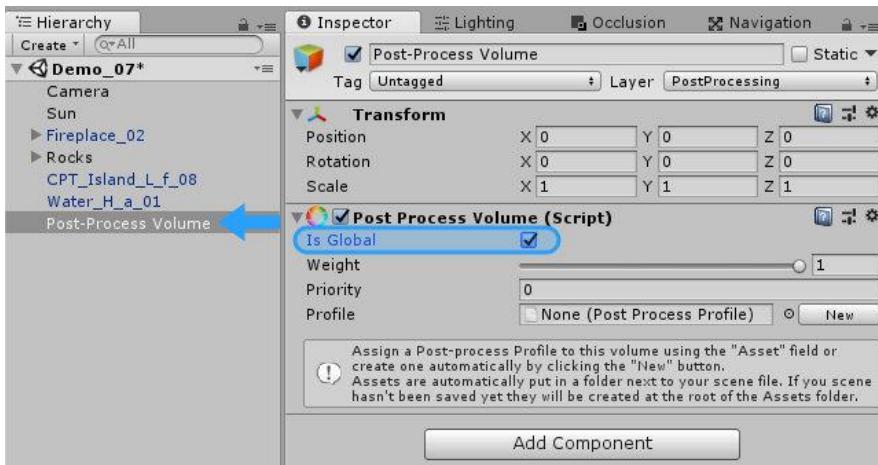
Set **Layer** to **PostProcessing**, the layer we just created before:



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

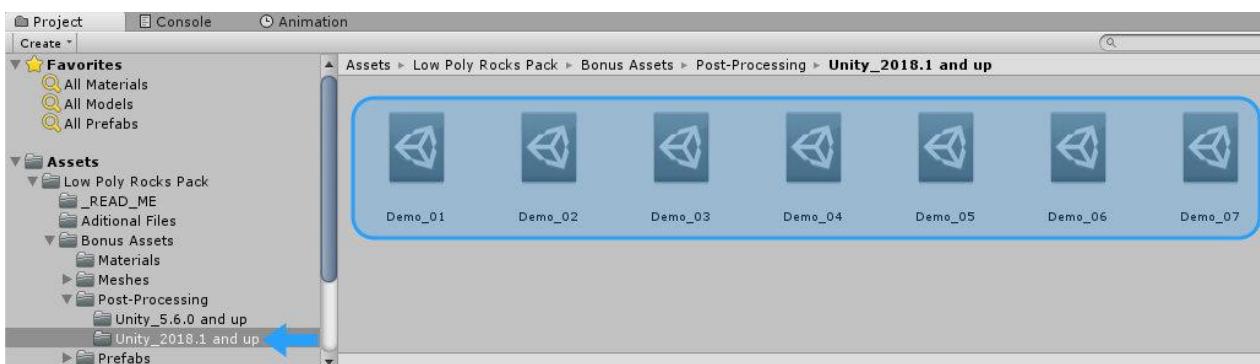


Enable **is Global**:

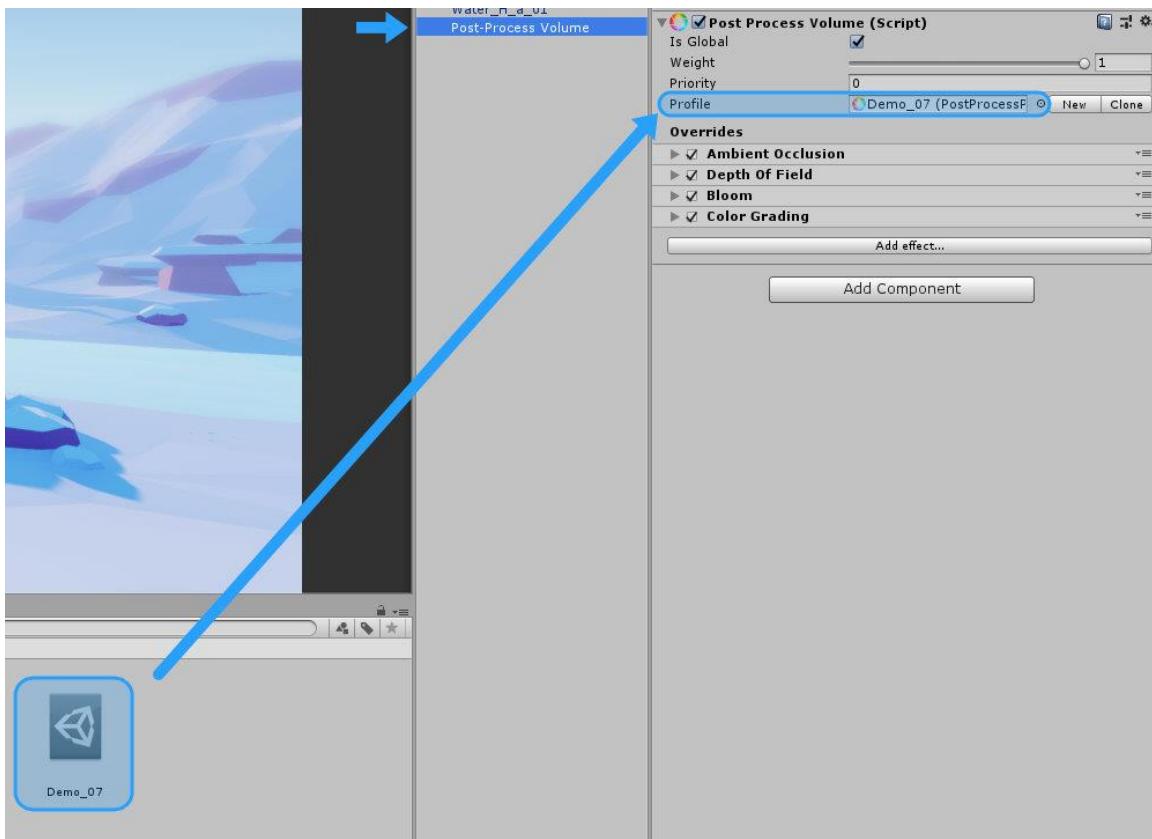


8. Apply Post-Processing Profile.

I've created 7 Post-Processing Profiles for 7 Demo scenes with different settings. Go to *Low Poly Rocks Pack > Bonus Assets > Post-Processing > Unity_2018.1 and up*.



We have opened **Demo_07** scene - so we will use **Demo_07** Post-Processing Profile. Drag and drop **Demo_07** Post-Processing Profile to **Profile** slot inside the **Post Process Volume**:



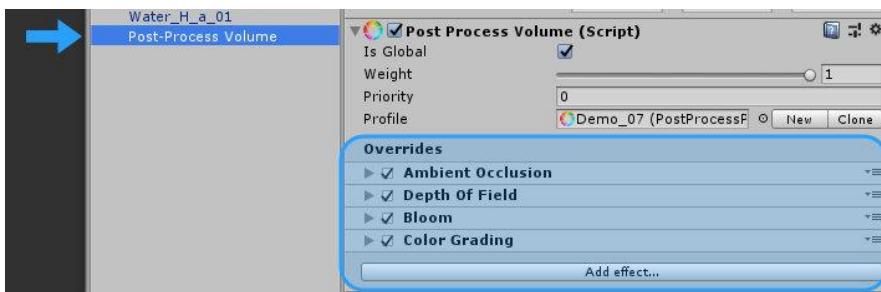
Now your scene should look like this (Demo_07):



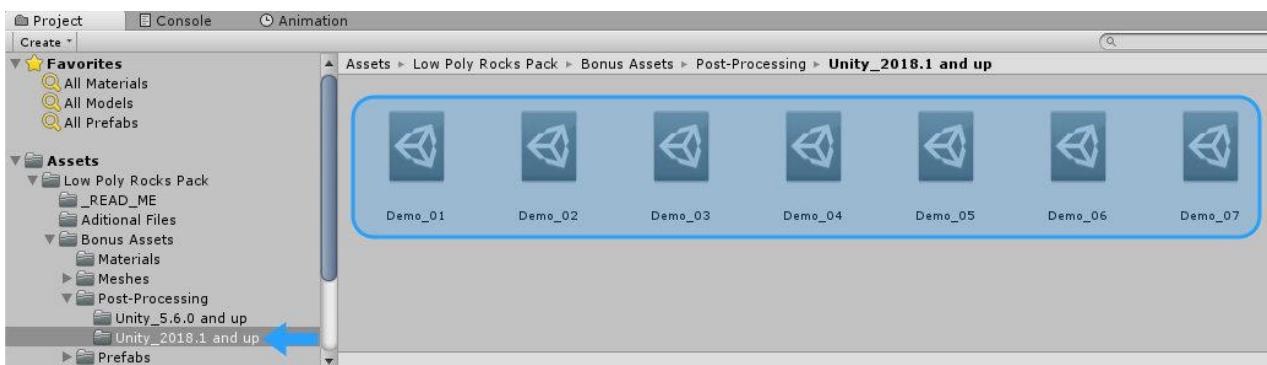
*For Low-End PC's - if you hit play and it lags, try disabling Post-Processing effects one by one on the Post-Processing Profile settings!

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.



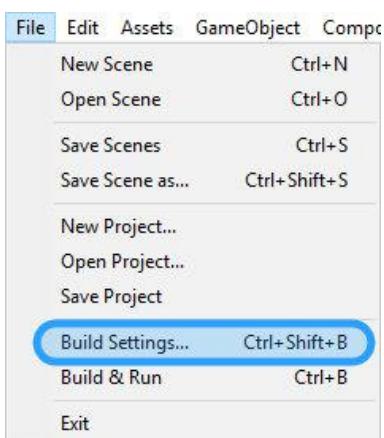
Or select any **Post-Processing Profile** inside the **Project** tab to edit.



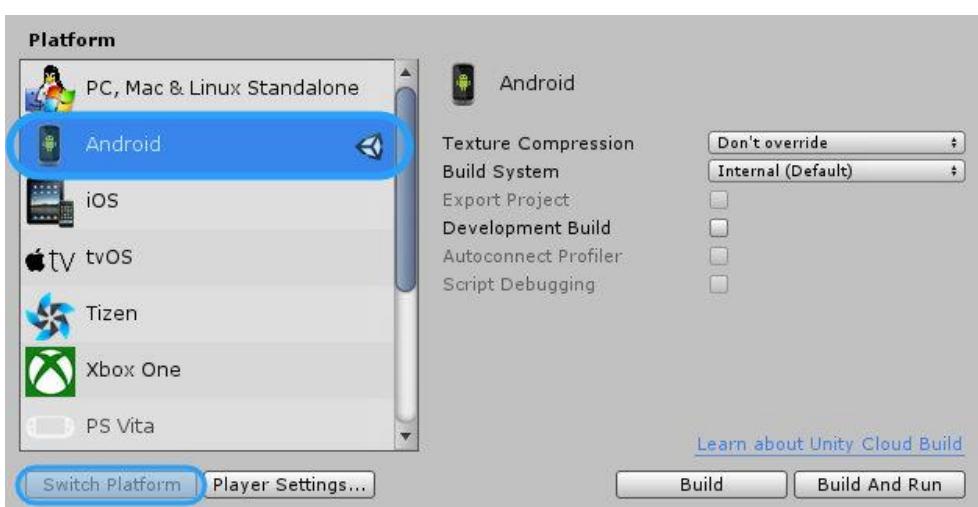
How to Setup Demo Scenes in Unity 5.6.0 - 2018.1 (For Android)

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

Go to *File > Build Settings*



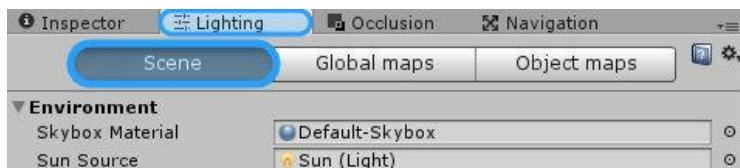
Select **Android** and hit the **Switch Platform** button.



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 **Auto** build/bake feature.

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

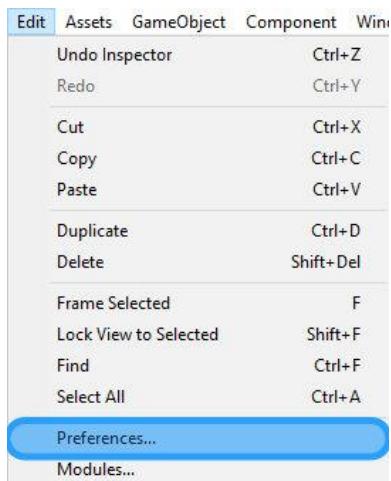


At the bottom you will see this:

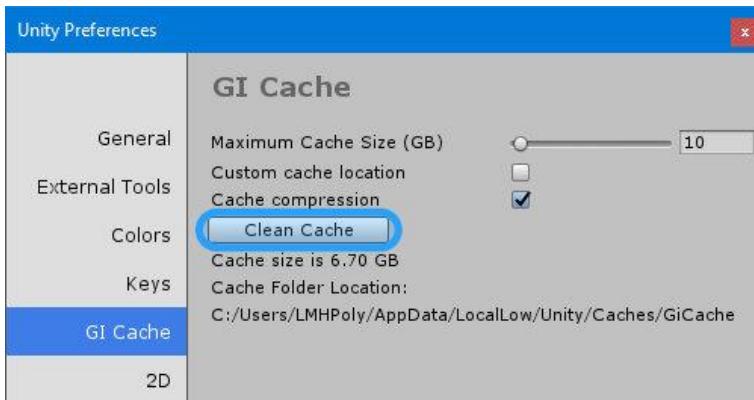


unchecked **Auto Generate**.

Go to *Edit > Preferences*



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



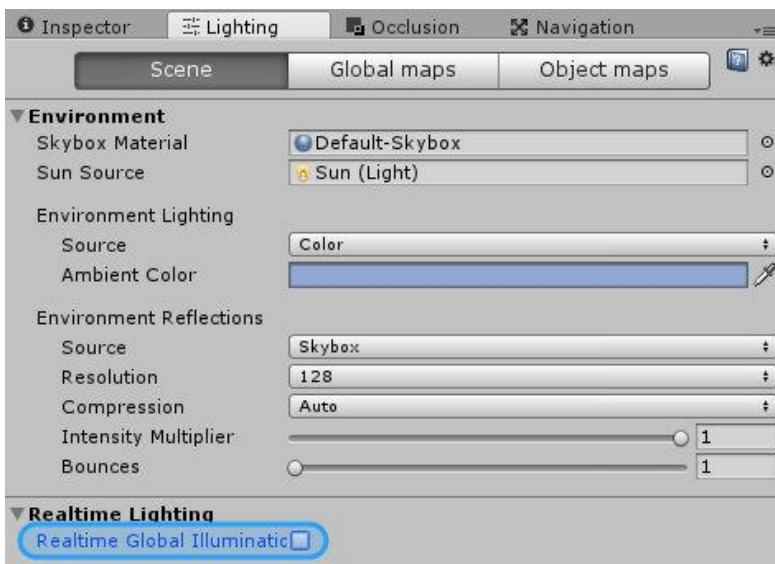
Enable **Auto Generate** / bake feature



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

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

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

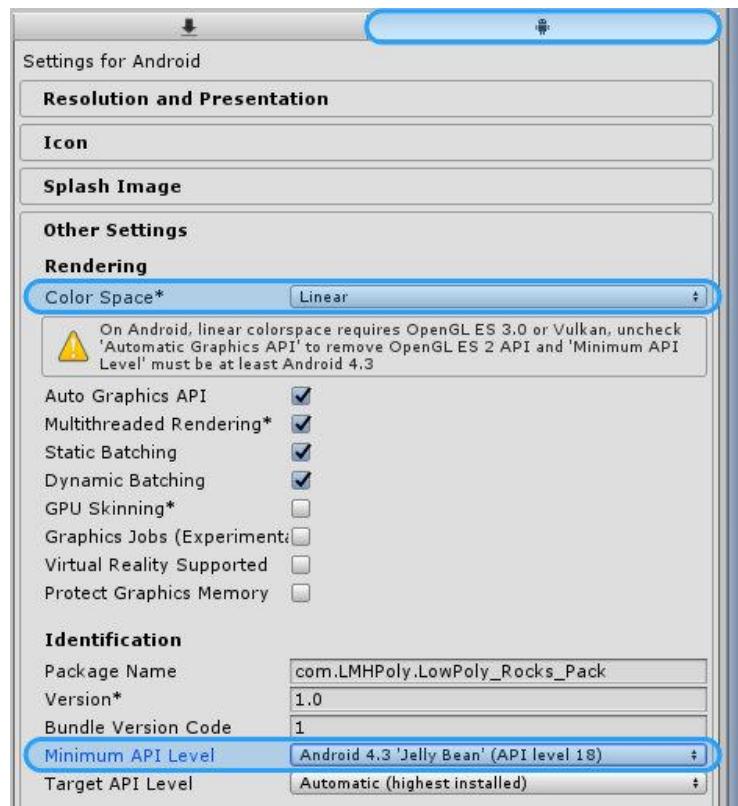


4. Make sure that **Color Space** is set to **Linear** (not all devices support it).

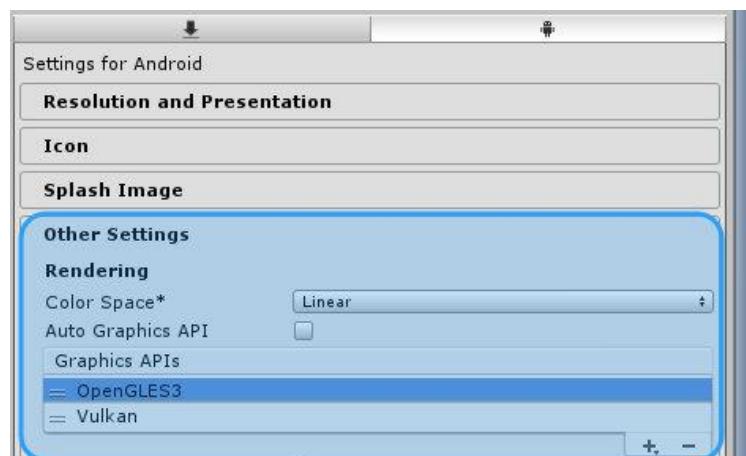
Go to *Edit > Project Settings > Player*

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 **Minimum API level** to at least **Android 4.3 (API level 18)** or higher!

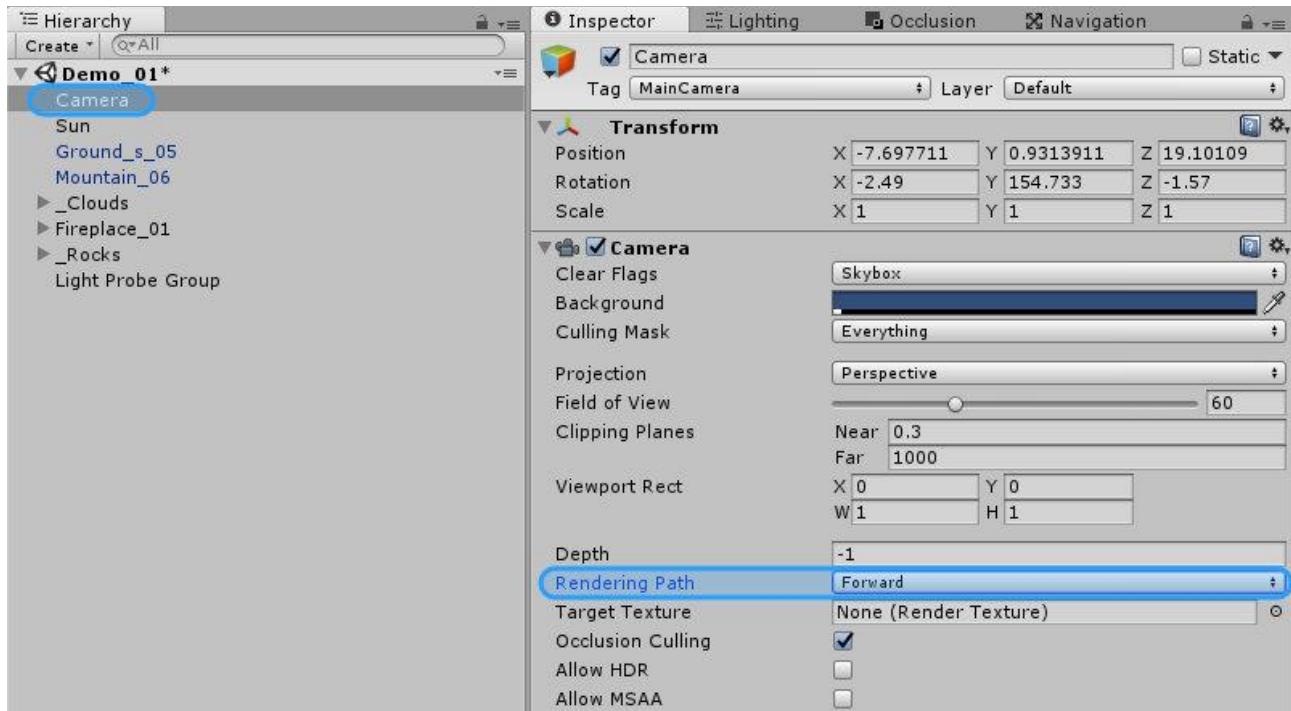


Also, uncheck **Auto Graphics API** and remove all Graphic APIs from the list, leave only **OpenGL ES3** and **Vulkan**(if you can't see it, press on **+** to add it). Make sure your Android device supports one of those graphics APIs!



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**.

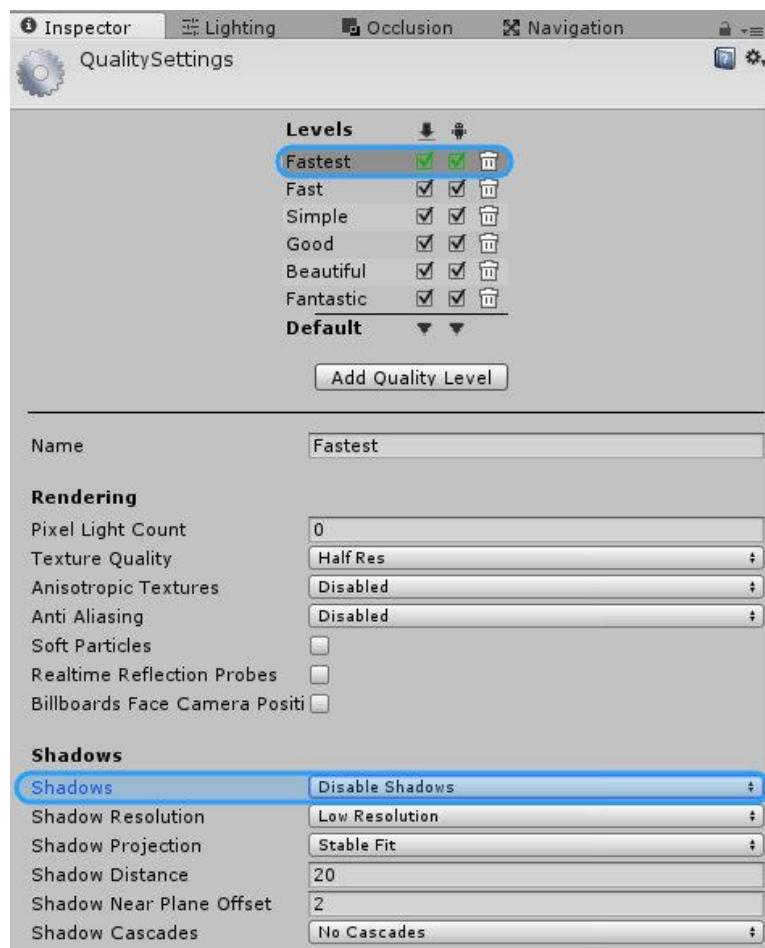


*If you set **Rendering Path** to **Deferred**, the game might slow down a lot on mobile!

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**.



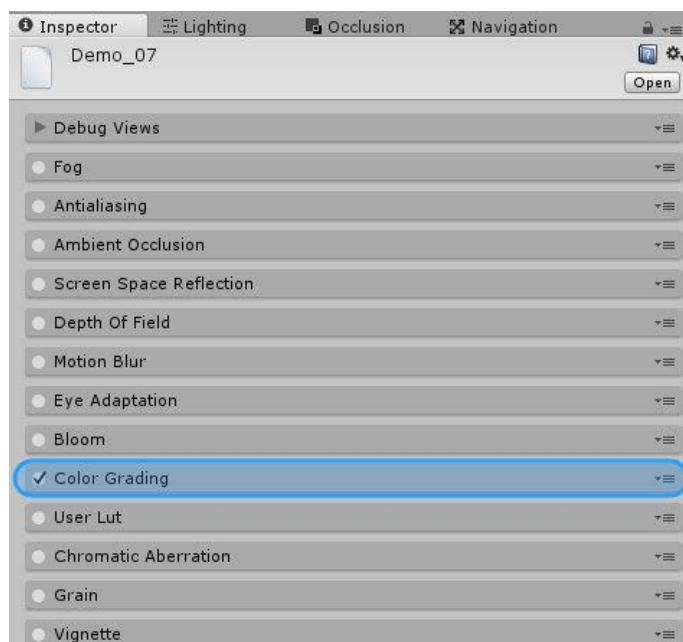
7. Import and enable **Post Processing Stack** image effects (Optional – Big performance hit for mobile devices).

Go to the part of the documentation: [Post-Processing in Unity 5.6 up to 2018.1](#)

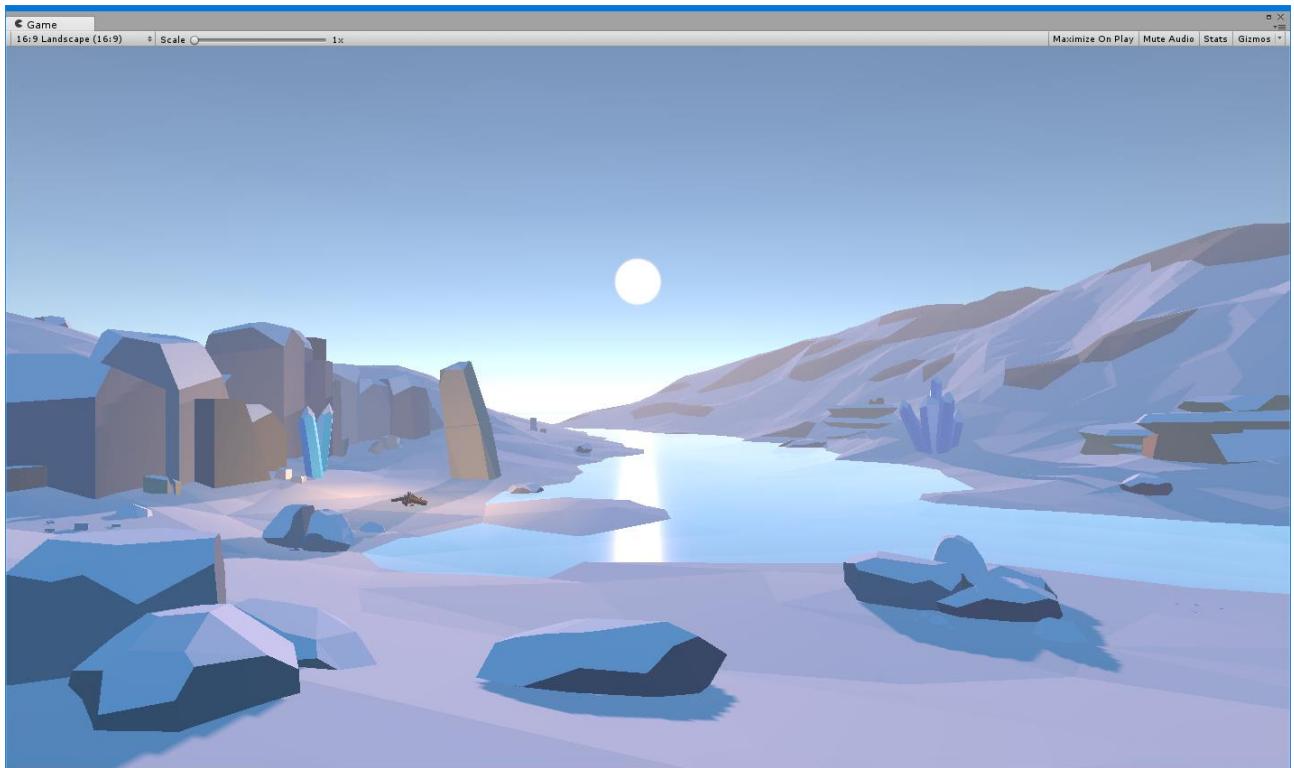
And follow those steps.

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

If you will use **Post-Processing Stack**, 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_07** scene should look like this (if you skipped all **Optional** steps):

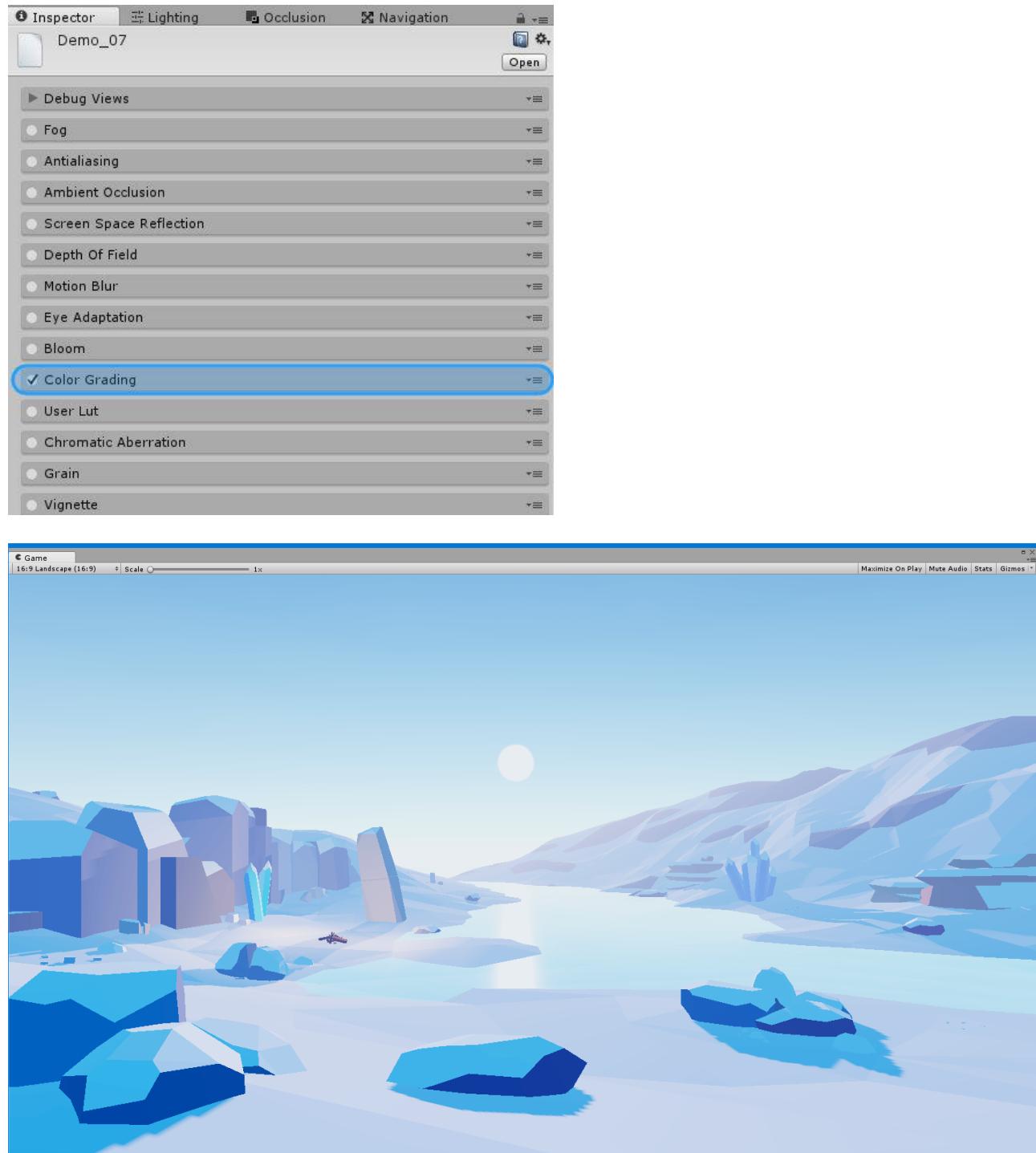


By using **Unity 5.5** and up + **Linear** lighting feature for **Android** and **iOS**, you can achieve much better results than using **Gamma** lighting!

All demo scenes including **Demo_07** has been tested on old Xperia Z Ultra (runs at solid 60FPS): without Post-Processing Stack effects, using Realtime GI, Linear Color Space, Forward Rendering Path and Real-time Medium Resolution Hard Shadows (Shadow Distance: 50, Shadow Projection: Close Fit).

*I don't have an **iOS** device to test this.

Demo_07 scene with the same settings + Post Processing Stack (**Color Grading** enabled only):



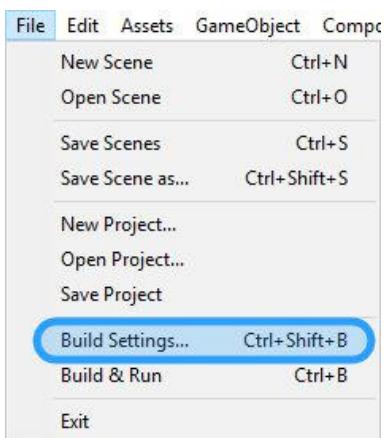
Tested on Google Pixel 2 XL – runs at solid 60fps. Xperia Z Ultra drops to 45fps just from using Color Grading.

How to Setup Demo Scenes in Unity 2018.1 and up

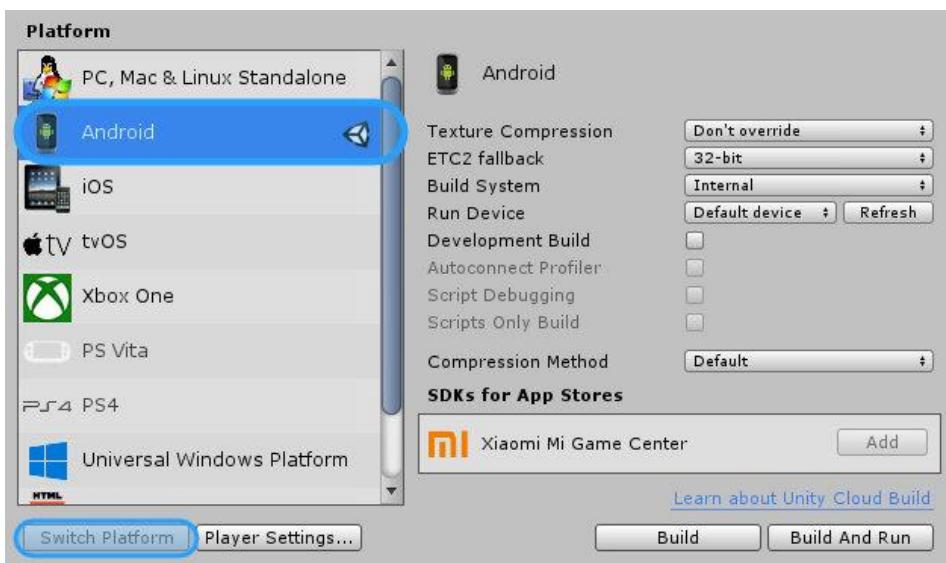
(For Android)

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

Go to *File > Build Settings*



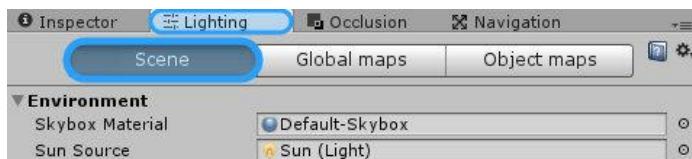
Select **Android** and hit the **Switch Platform** button.



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 **Auto** build/bake feature.

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

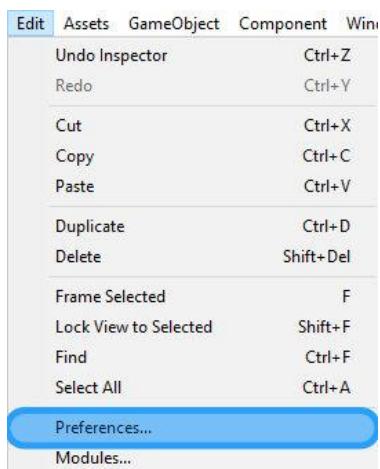


At the bottom you will see this:

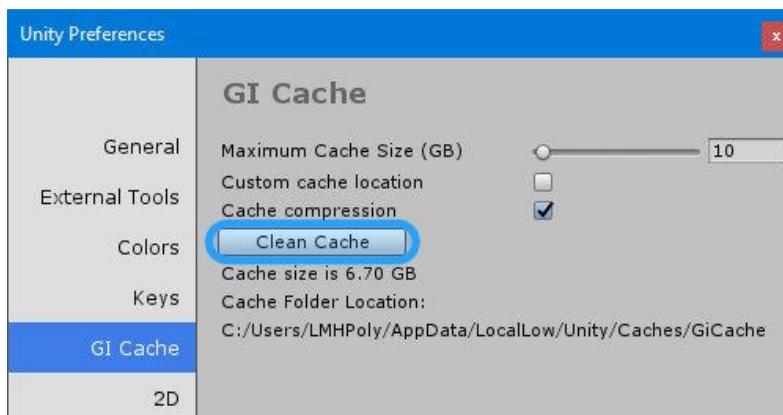


unchecked **Auto Generate**.

Go to *Edit > Preferences*



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



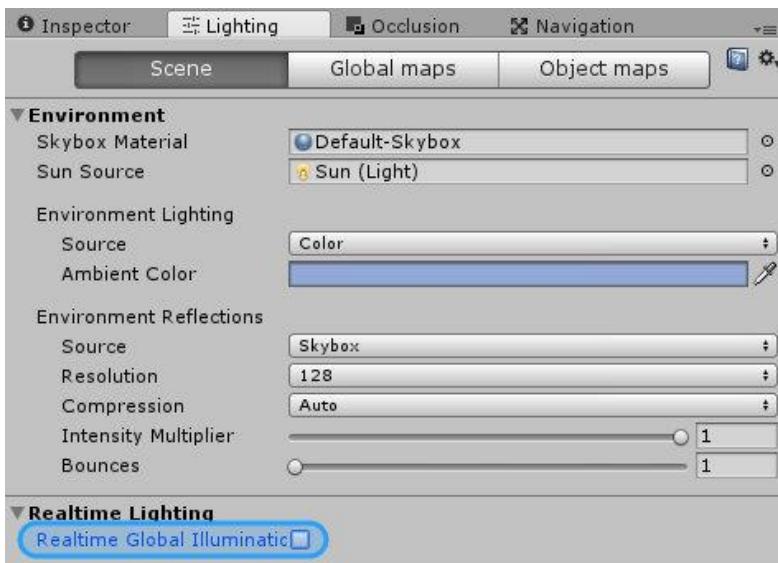
Enable **Auto Generate** / bake feature



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 **Scene** tab (If you don't see Lighting tab go to *Window > Lighting > Settings*).

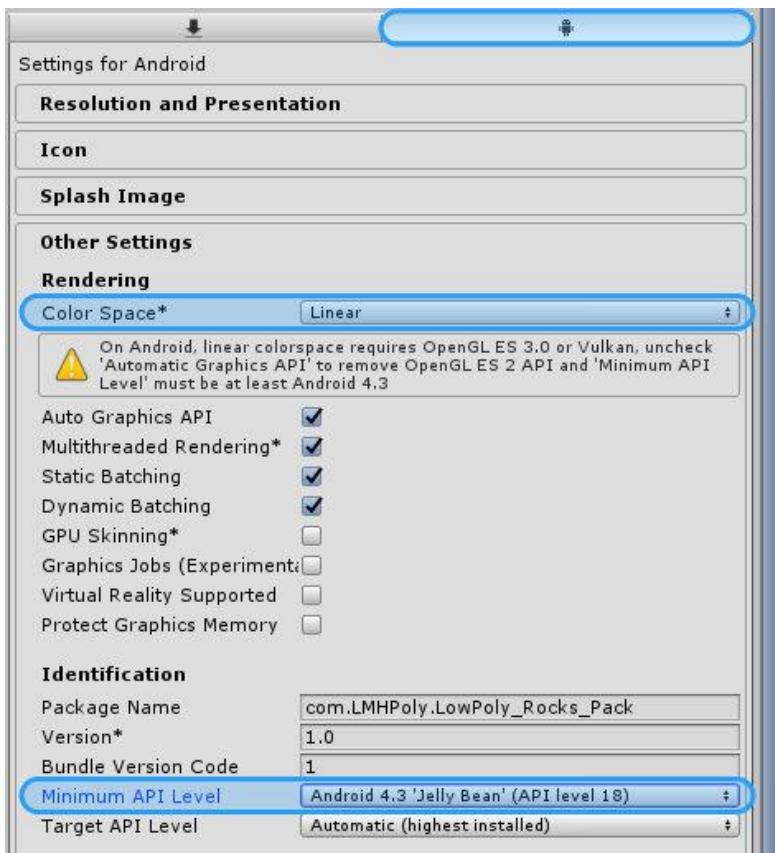


4. Make sure that **Color Space** is set to **Linear** (not all devices support it).

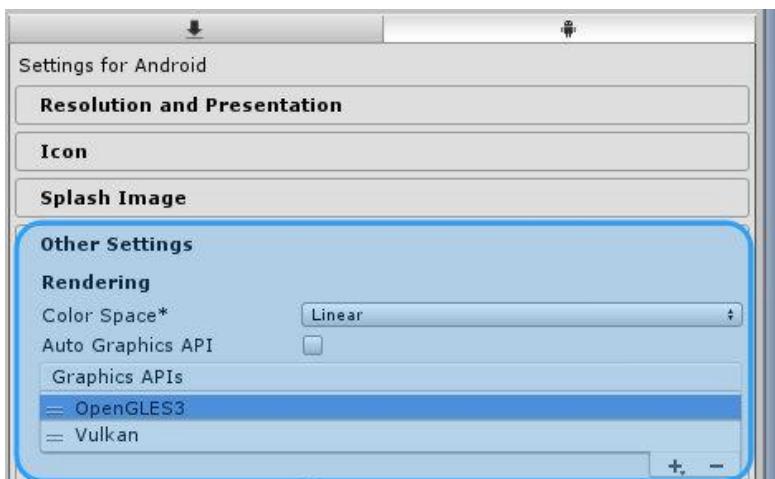
Go to *Edit > Project Settings > Player*

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 **Minimum API level** to at least **Android 4.3 (API level 18)** or higher!

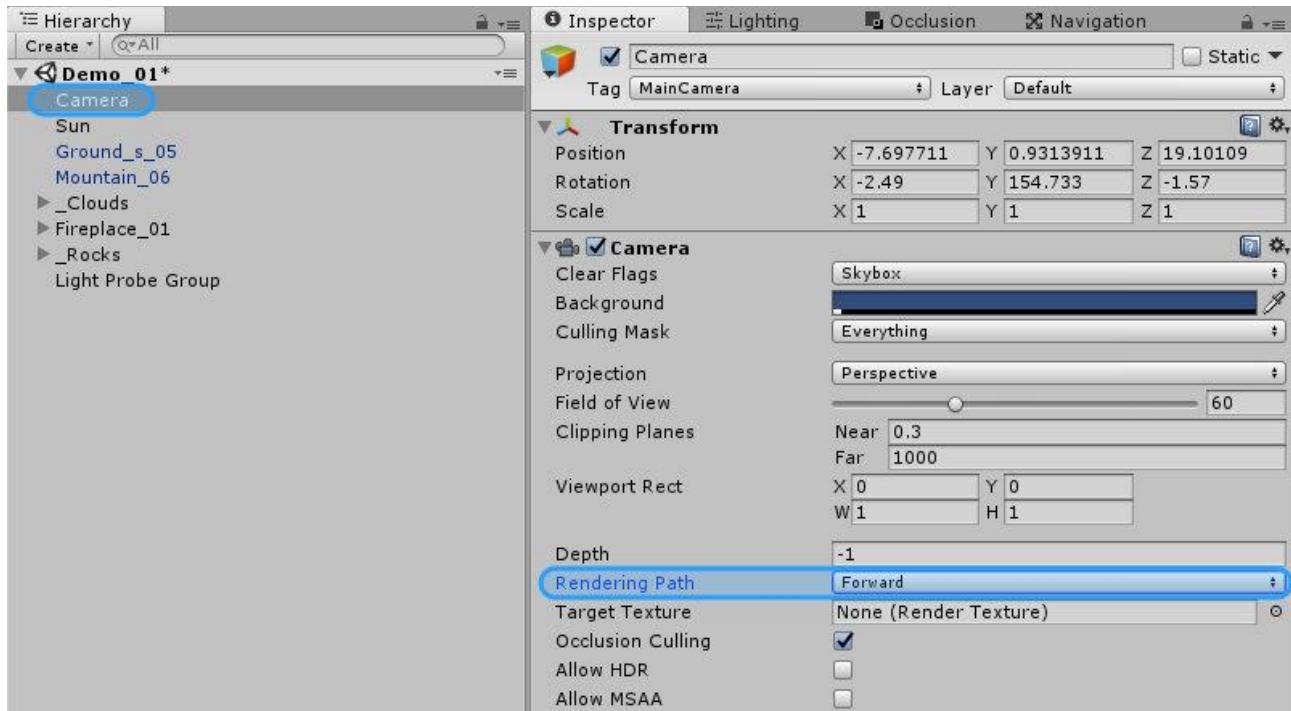


Also, uncheck **Auto Graphics API** and remove all Graphic APIs from the list, leave only **OpenGL ES3** and **Vulkan** (if you can't see it, press on **+** to add it). Make sure your Android device supports one of those graphic APIs!



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**.

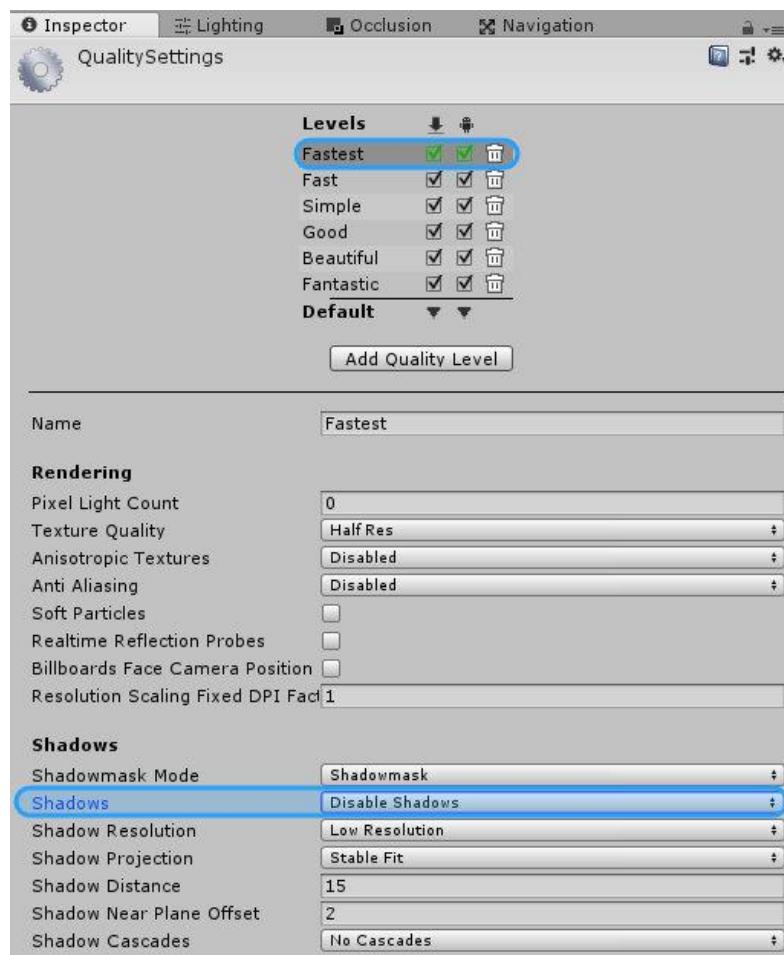


*If you set **Rendering Path** to **Deferred**, the game might slow down a lot on mobile!

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.

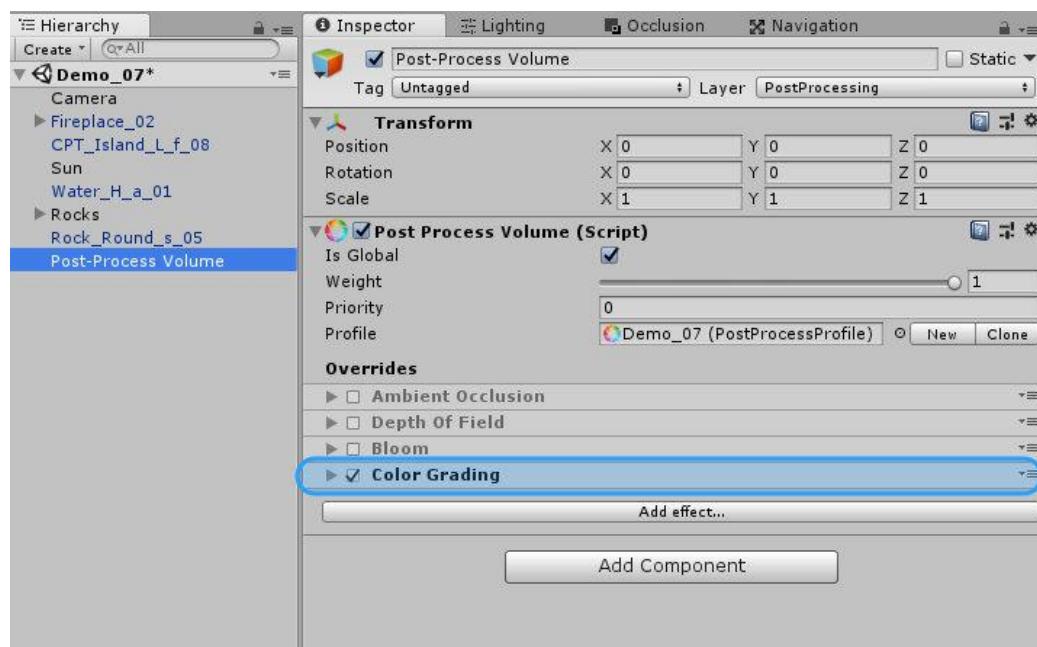
7. Import and enable **Post Processing** image effects (Optional – **Big performance hit for mobile devices!**).

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

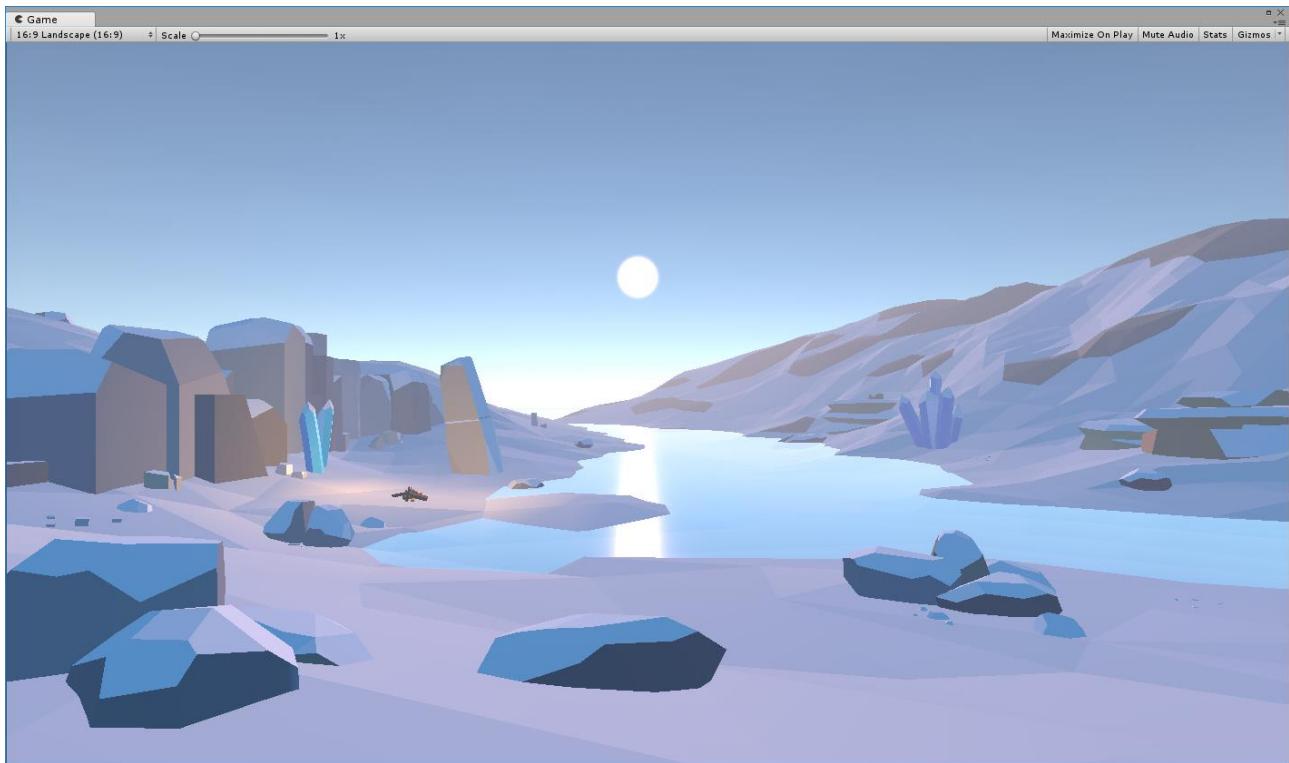
And follow those steps.

*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_07** scene should look like this (if you skipped all **Optional** steps, and with Realtime Shadows **Disabled**):

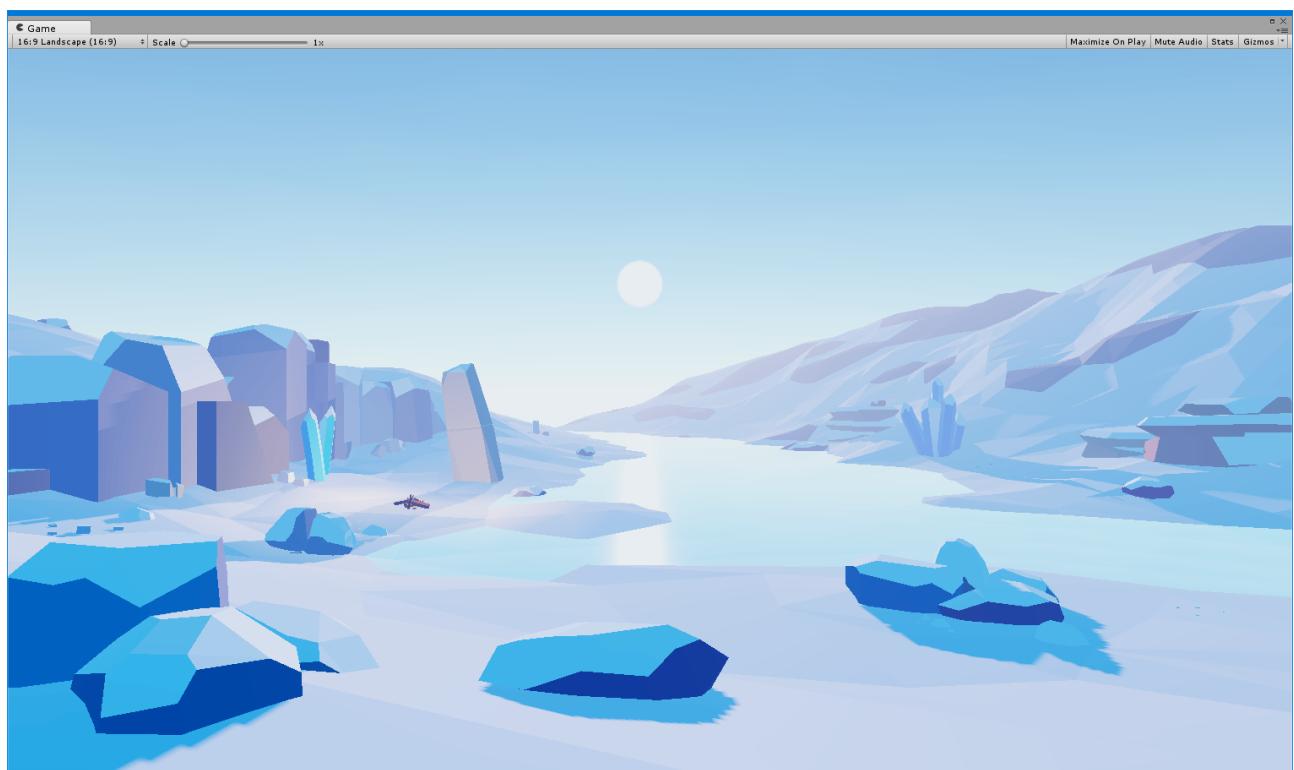
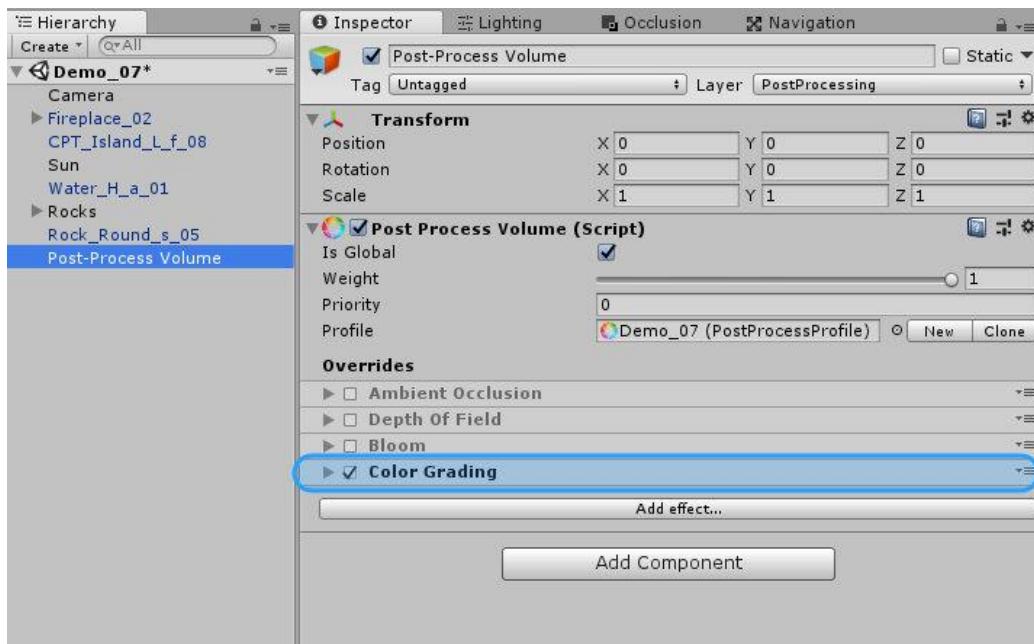


By using **Unity 5.5** and up + **Linear** lighting feature for **Android** and **iOS**, you can achieve much better results than using **Gamma** lighting!

All demo scenes including **Demo_07** 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. (Strangely by using Unity 5.6 I had Realtime Shadows: enabled on Medium resolution and it runs at solid 60fps. And on Unity 2018.1 it was 45fps using Realtime shadows with the same settings).

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

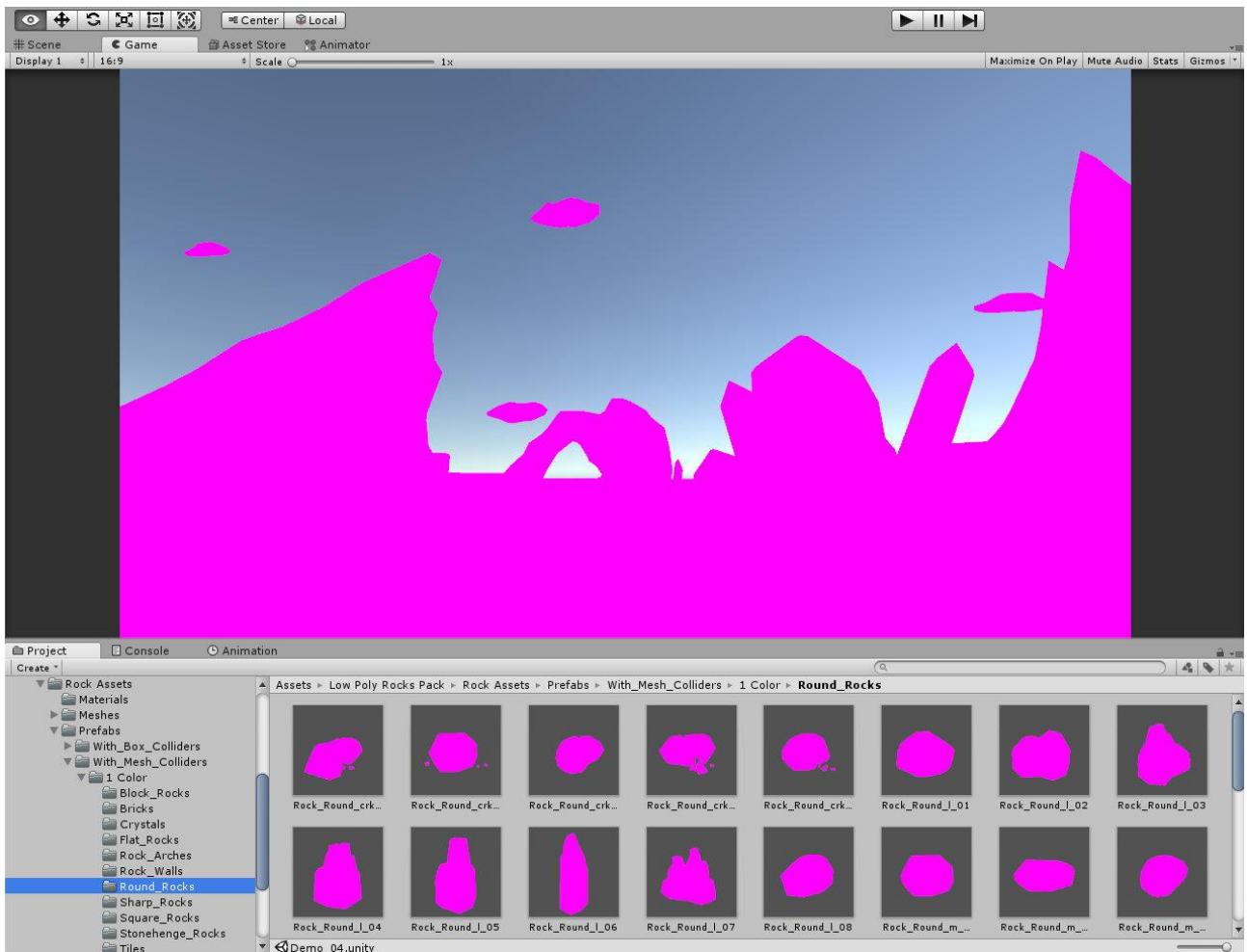
Demo_07 scene with the same settings + Post Processing (**Color Grading** enabled only) +
Realtime Shadows: medium resolution enabled:



Tested on Google Pixel 2 XL – runs at solid 60fps. Xperia Z Ultra drops to 26fps for using
Realtime shadows and Color Grading.

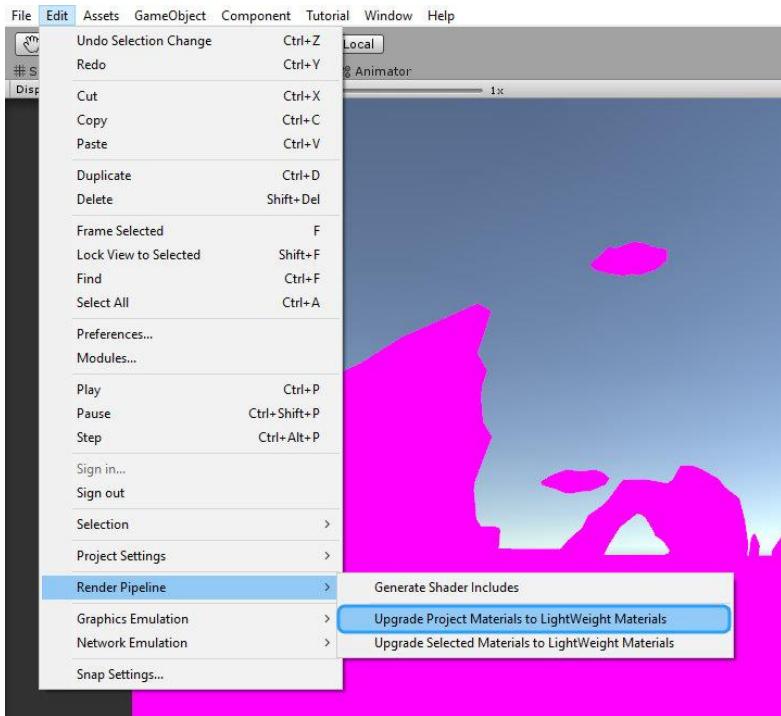
Unity 2018.1 and up Lightweight Render Pipeline (LWRP)

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

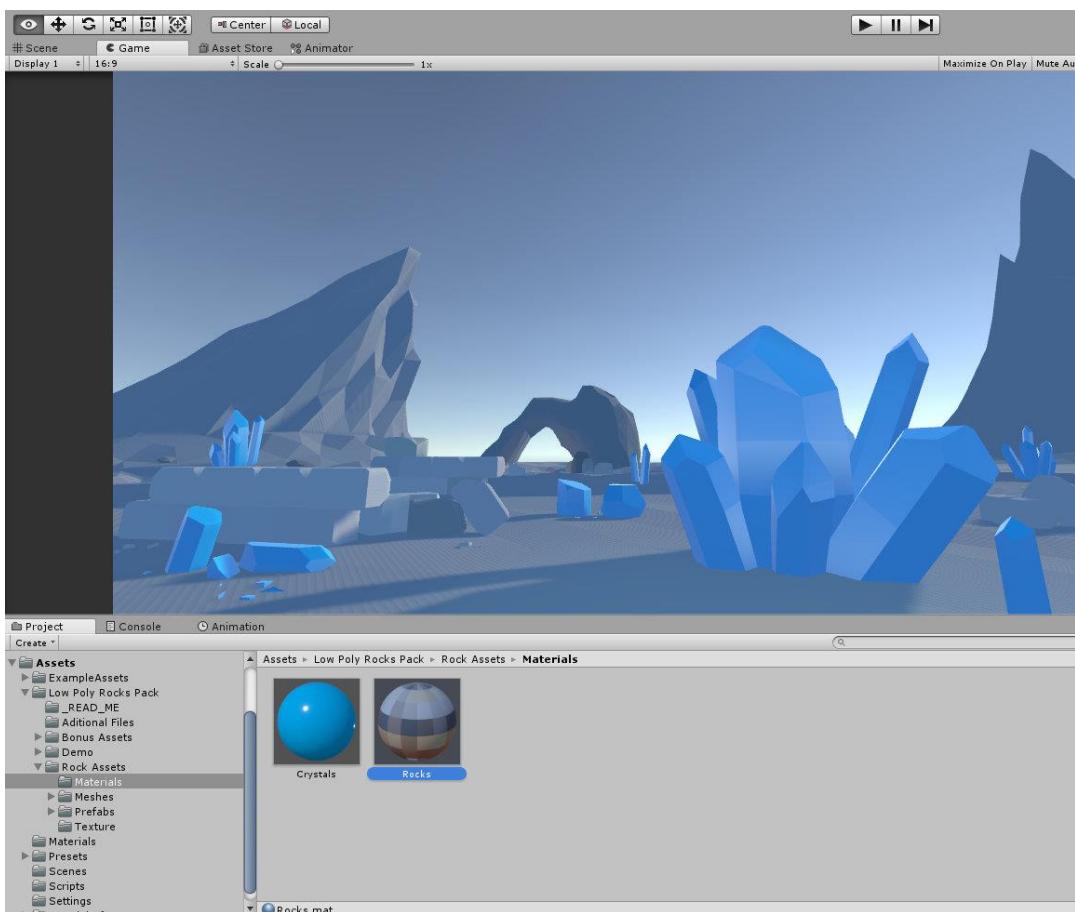


It's because all of **Low Poly Rocks Pack** assets use material with a default **Standard Unity shader**. **LWRP** use different materials and shaders. So we need to change all materials from **Standard shader** to **LightweightPipeline shader**.

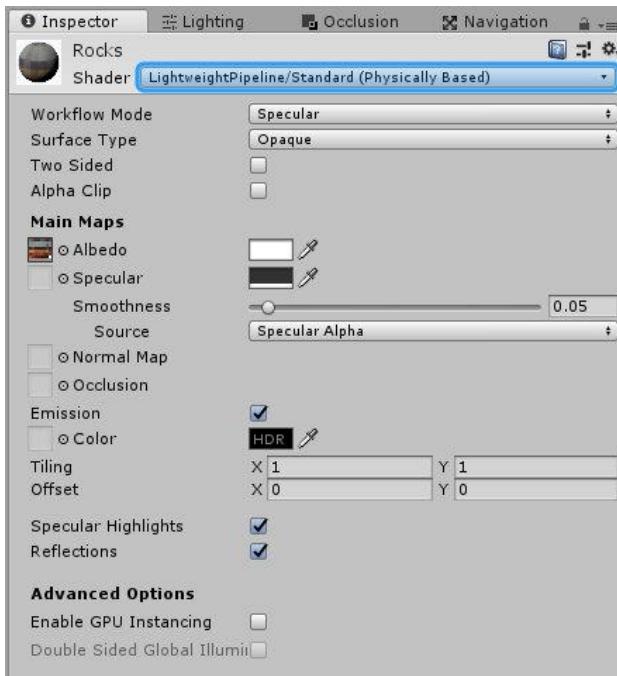
Go to *Edit > Render Pipeline > Upgrade Project Materials to LightWeight Materials*



And it's done!

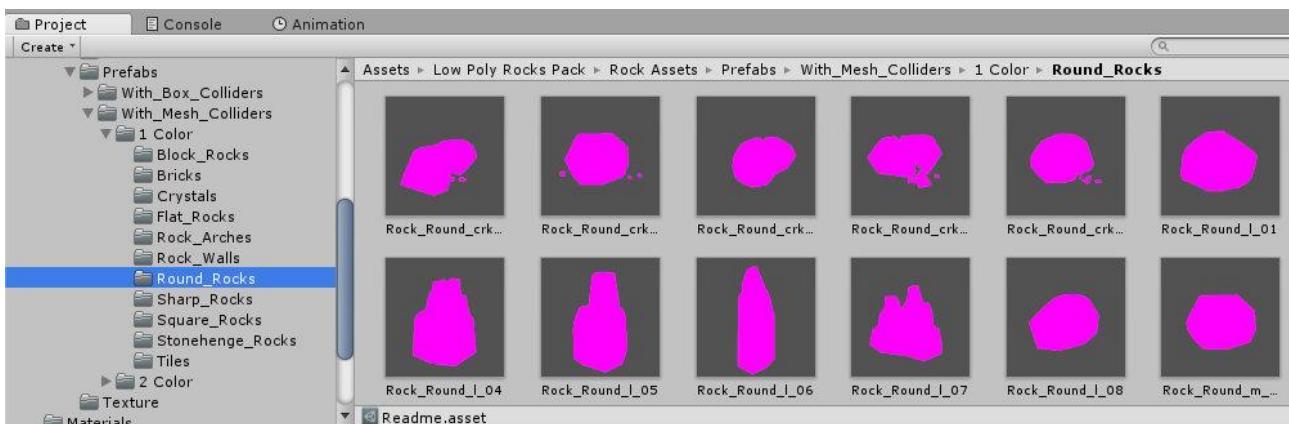


All project **Material** shaders were changed to **LightweightPipeline/Standard (Physically Based)**.

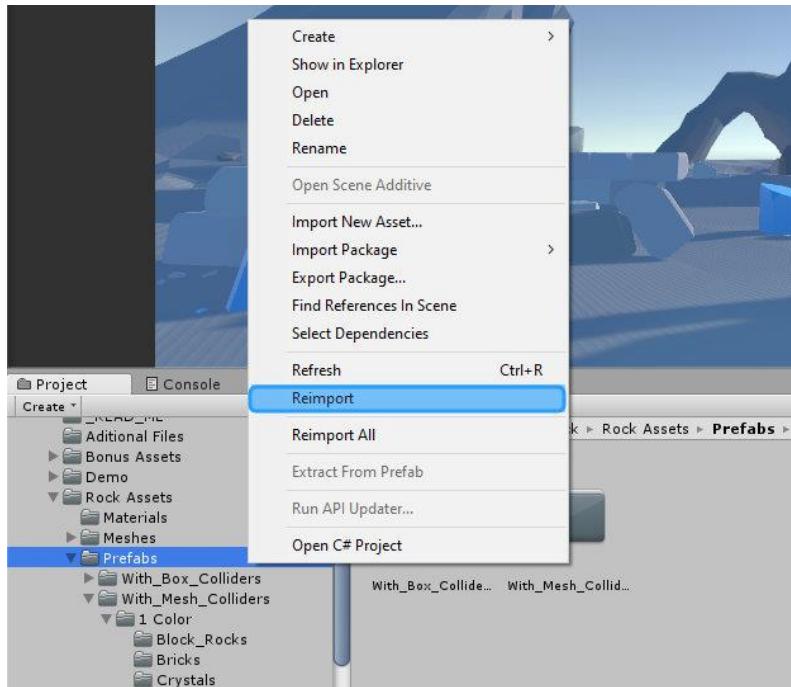


*You can do it manually by selecting **Material** and changing the **Shader** but it's much slower.

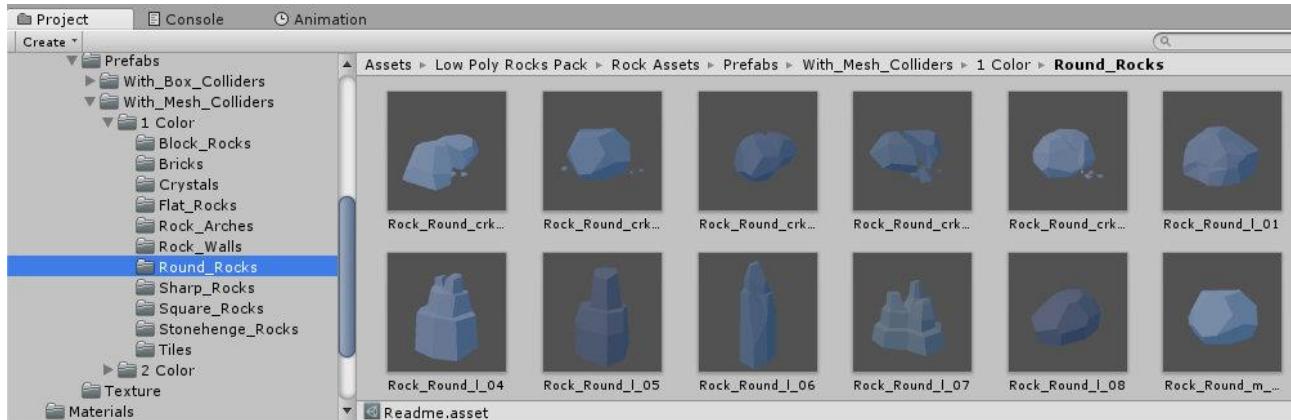
Now if you go to *Low Poly Rocks Pack > Rocks Assets > Prefabs > With_Mesh_Colliders > 1 Color > Round_Rocks* or inside any other rocks folder. You might see all of the prefabs in **Pink** color.



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

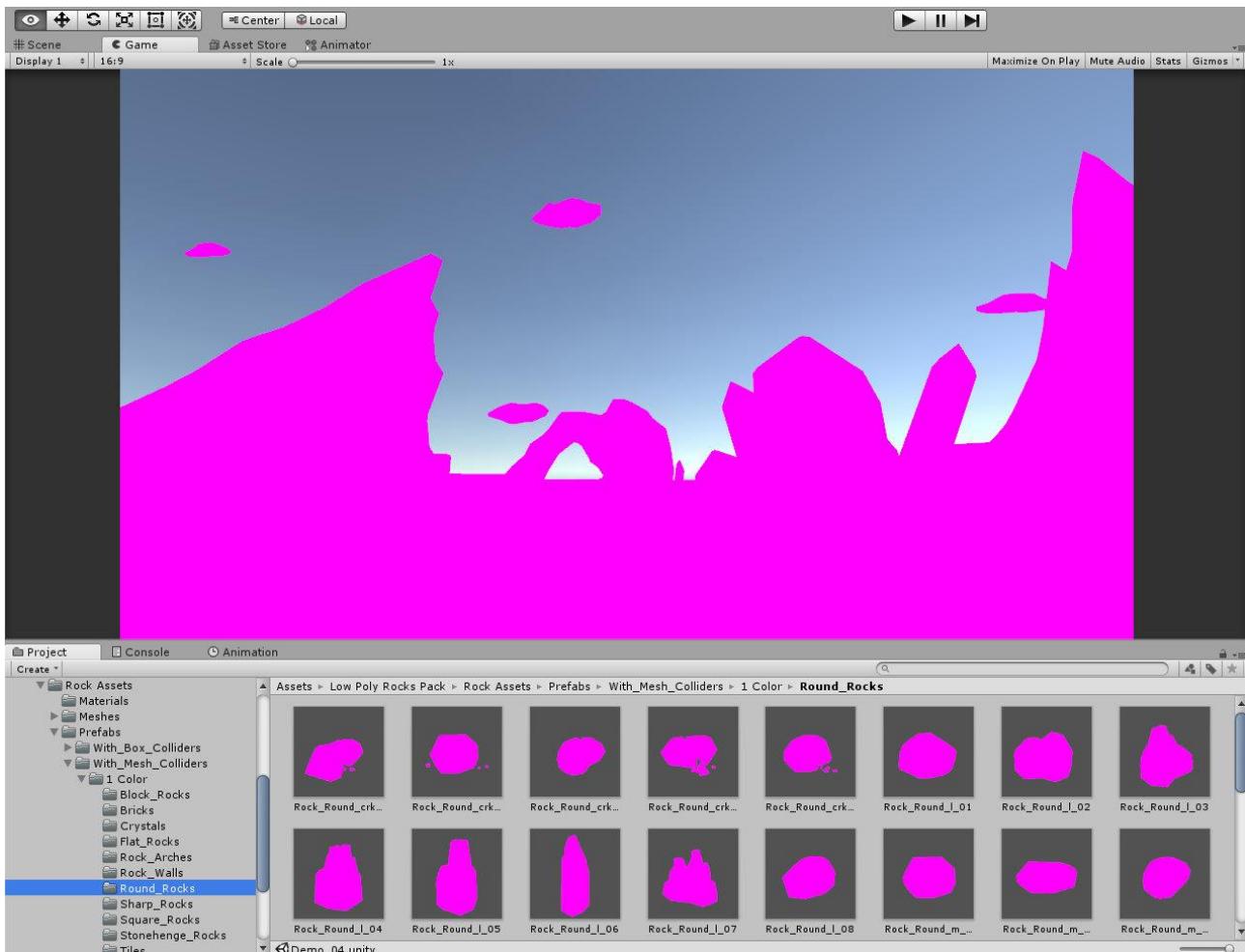


And it's fixed!



Unity 2018.1 and up High Definition Render Pipeline (HDRP)

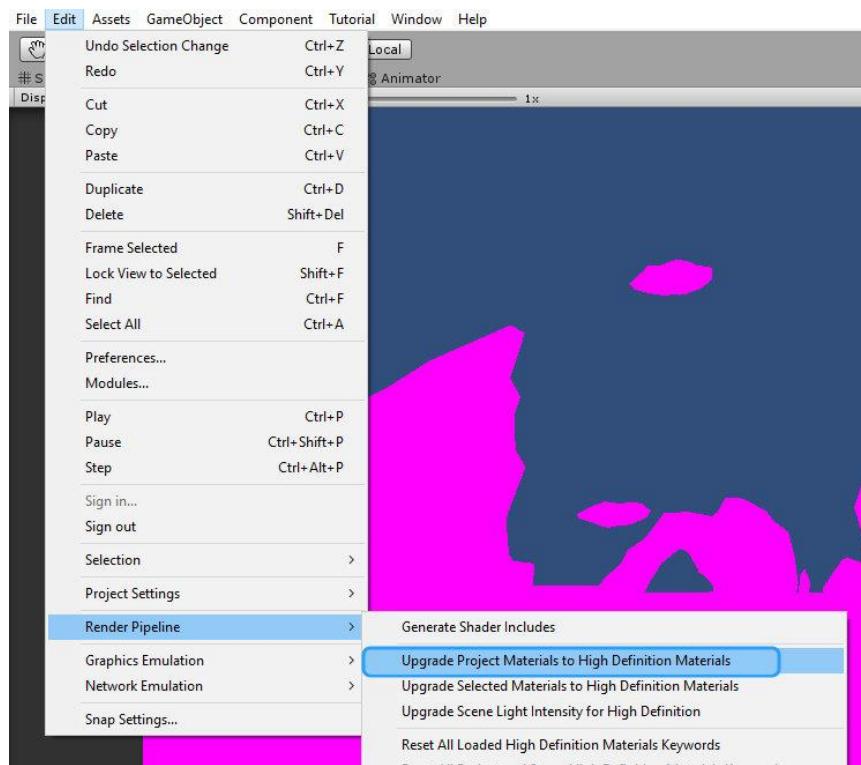
You might encounter pink textures after importing **Low Poly Rocks Pack** to your Unity project, which is using **High Definition Render Pipeline (HDRP)**.



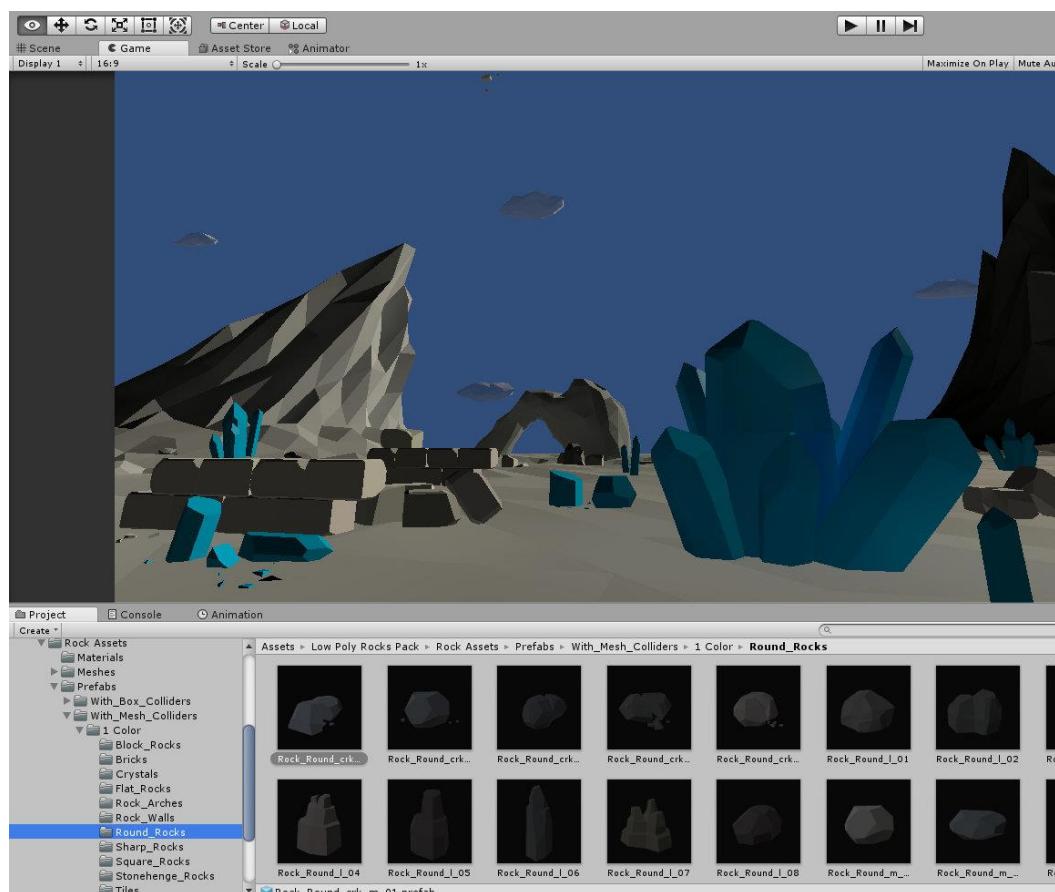
It's because all of **Low Poly Rocks Pack** assets use material with a default **Standard Unity shader**. **HDRP** use different materials and shaders. So we need to change all materials from Standard shader to HDRP pipeline shader.

1. Fix Purple Materials

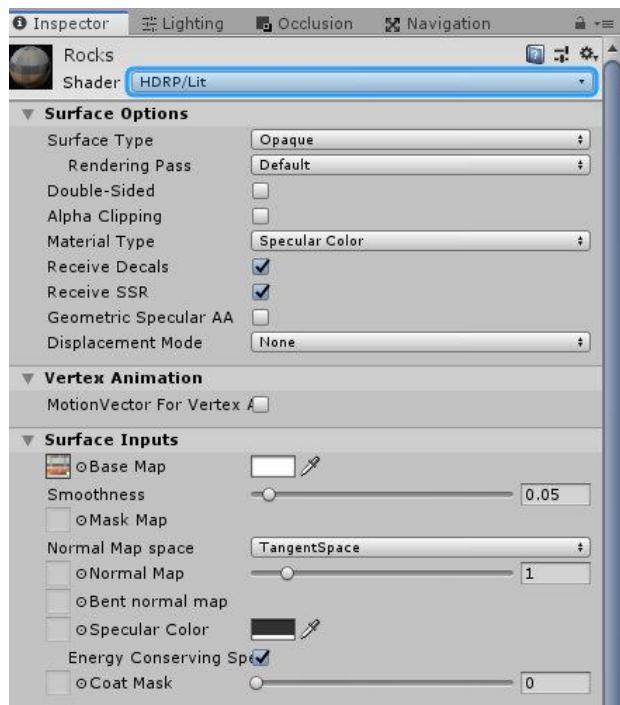
Go to *Edit > Render Pipeline > Upgrade Project Materials to High Definition Materials*



And it's done!

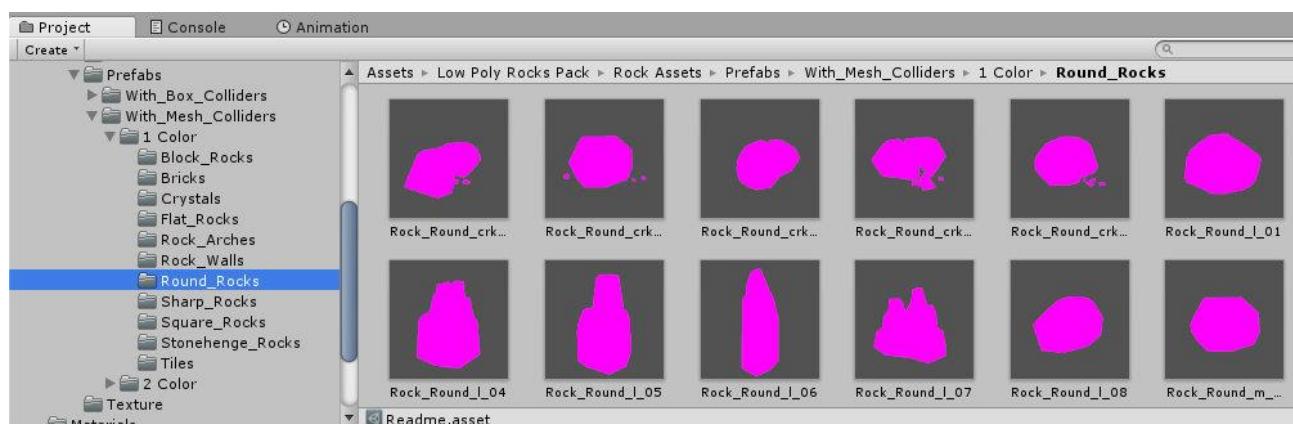


All project **Material** shaders were changed to **HDRP/Lit**.

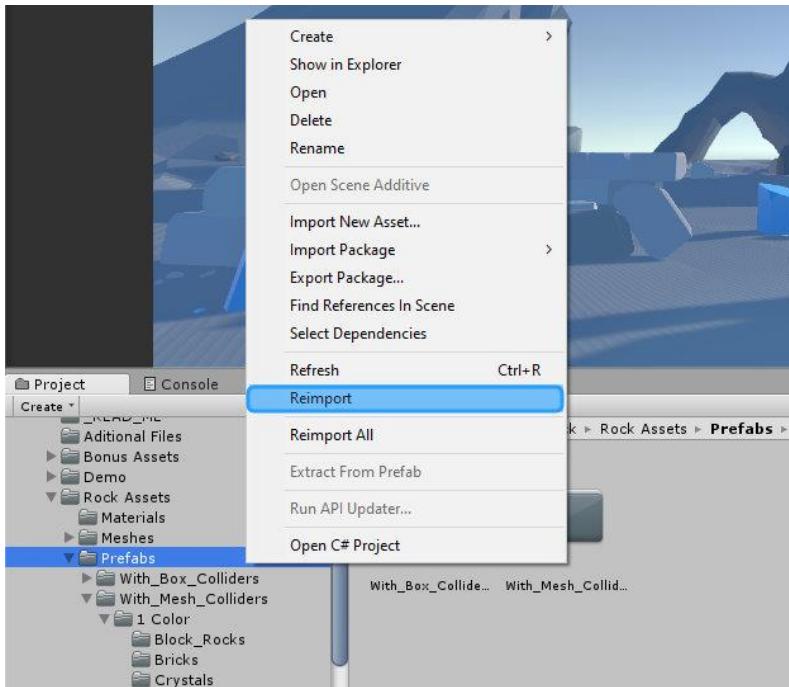


*You can do it manually by selecting **Material** and changing the **Shader** but it's much slower.

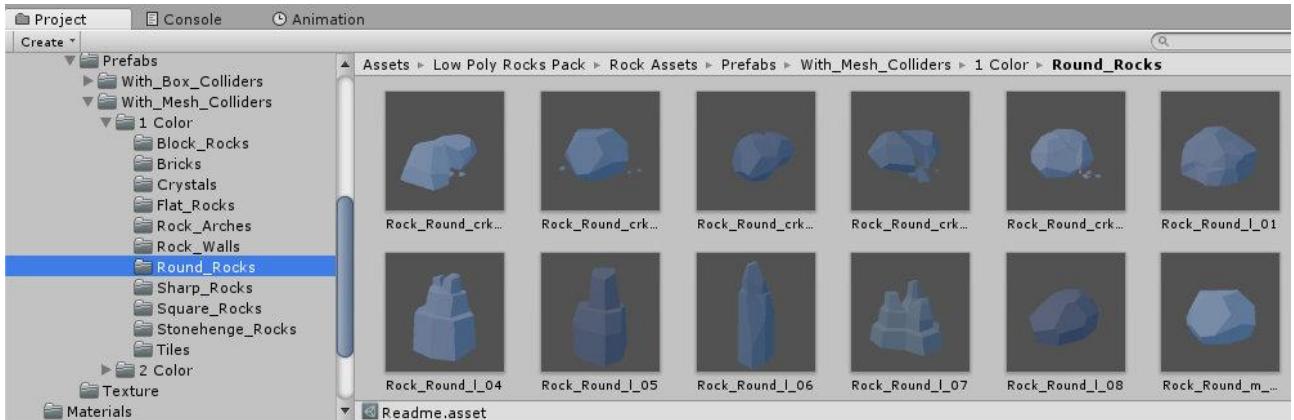
Now if you go to *Low Poly Rocks Pack > Rocks Assets > Prefabs > With_Mesh_Colliders > 1 Color > Round_Rocks* or inside any other rocks folder. You might see all of the prefabs in **Pink** color.



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



And it's fixed!

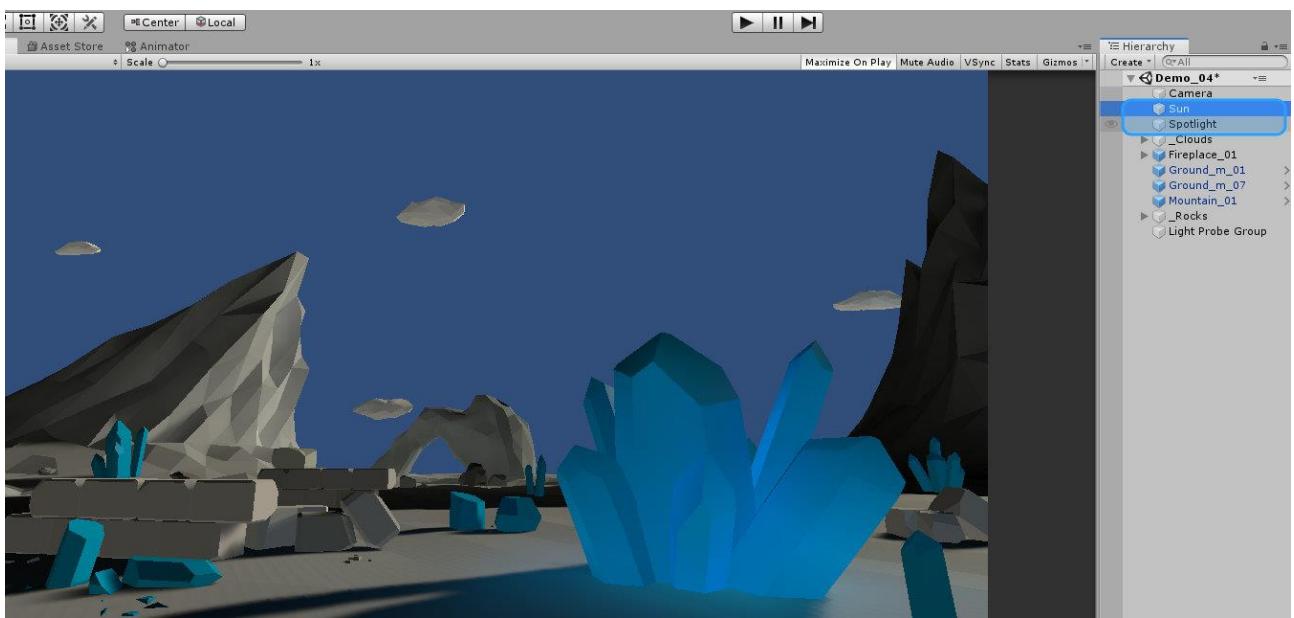


As you can see, the scene looks very dull. It has no skybox and proper lighting.



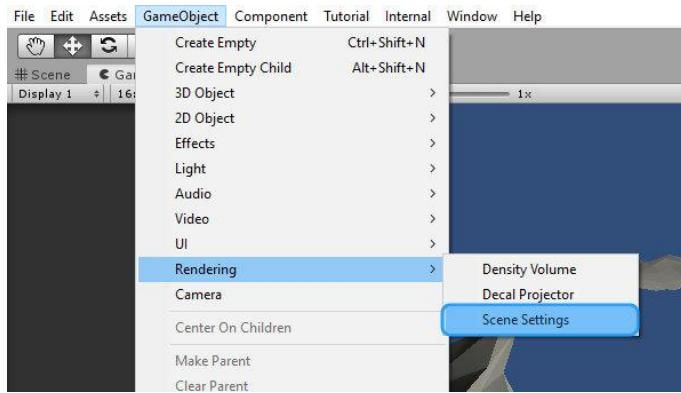
2. Fix Shadows and Lighting

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

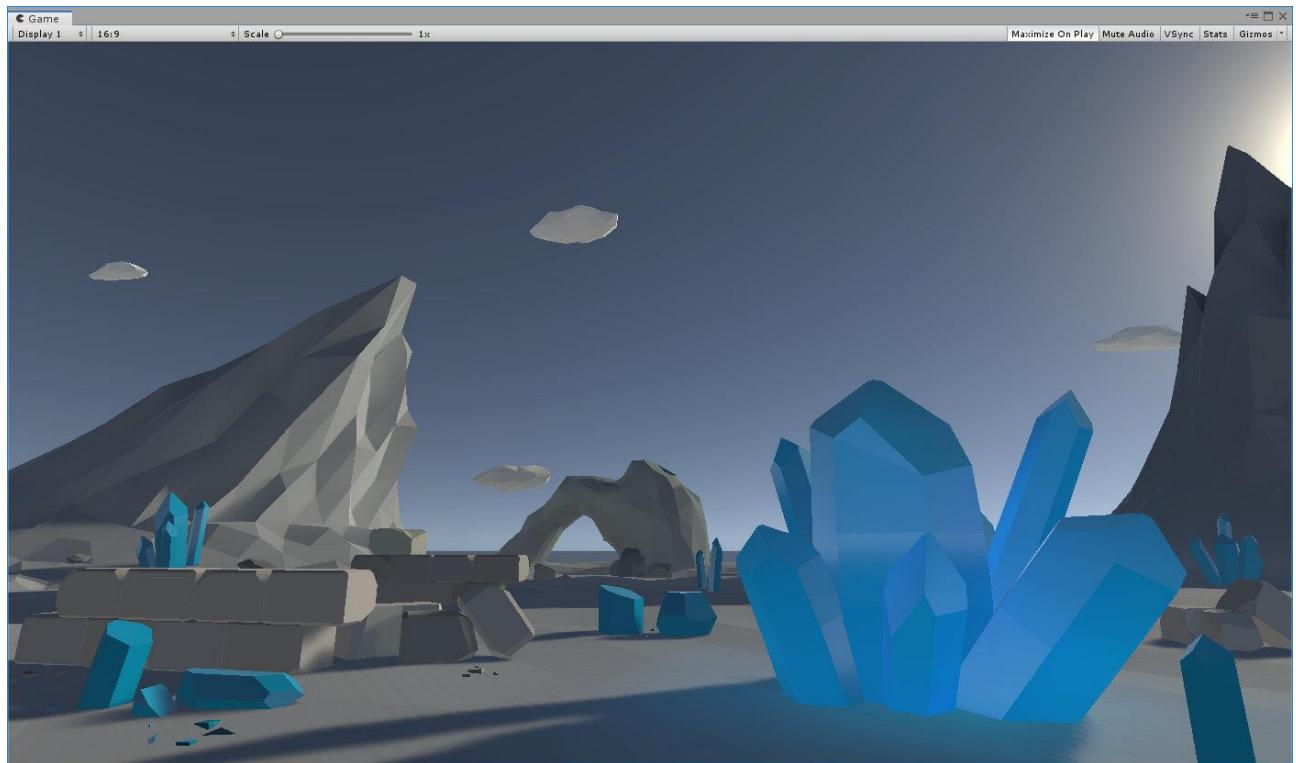


3. Fix the **Skybox**

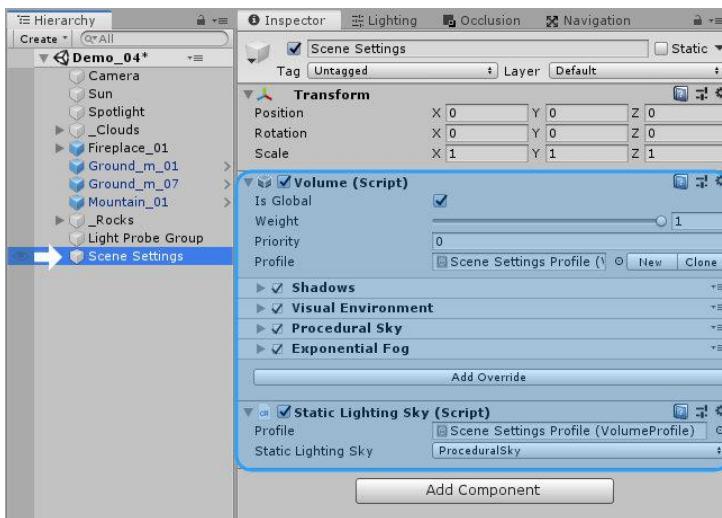
Using HDRP, you need to use **Scene Settings** - to change the **Skybox** and other scene settings. Go to *GameObject > Rendering > Scene Settings*



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



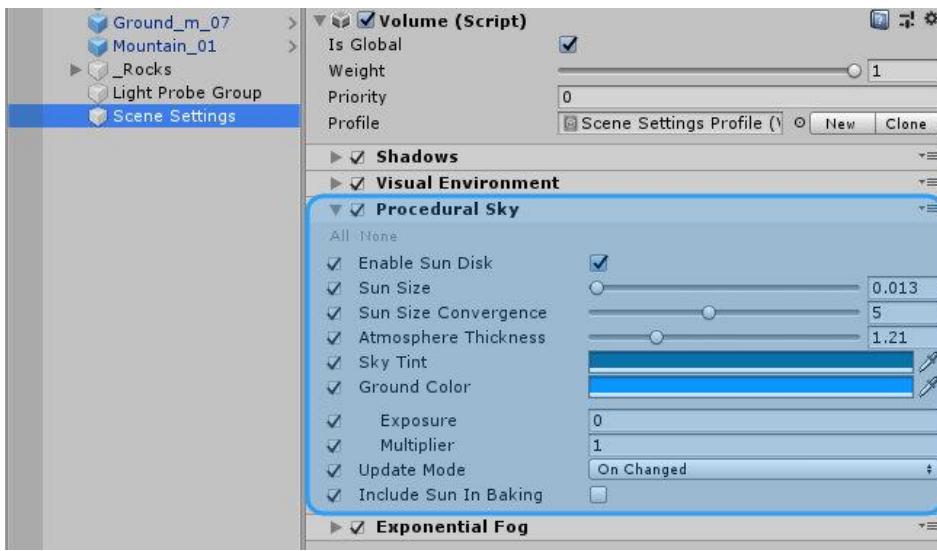
With a **Scene Settings** selected, you can change a bunch of scene settings like (Shadows, Skybox, Fog, and much more).



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)

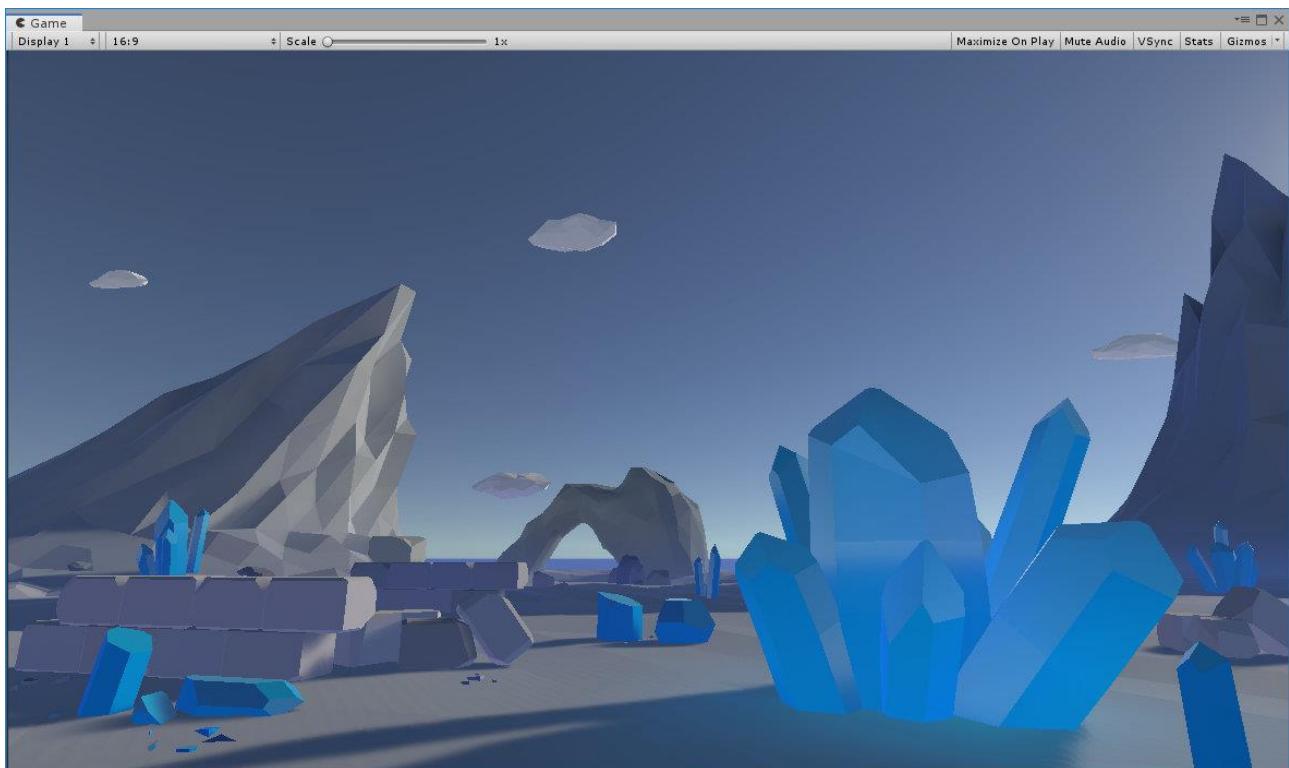
Use my **Procedural Skybox** settings:



Sky Tint (Color code): 006290

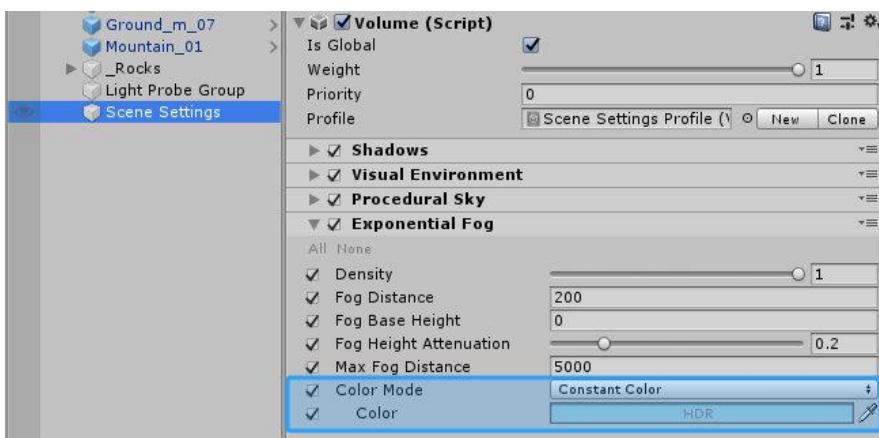
Ground Color (Color code): 0092FF

to achieve this:

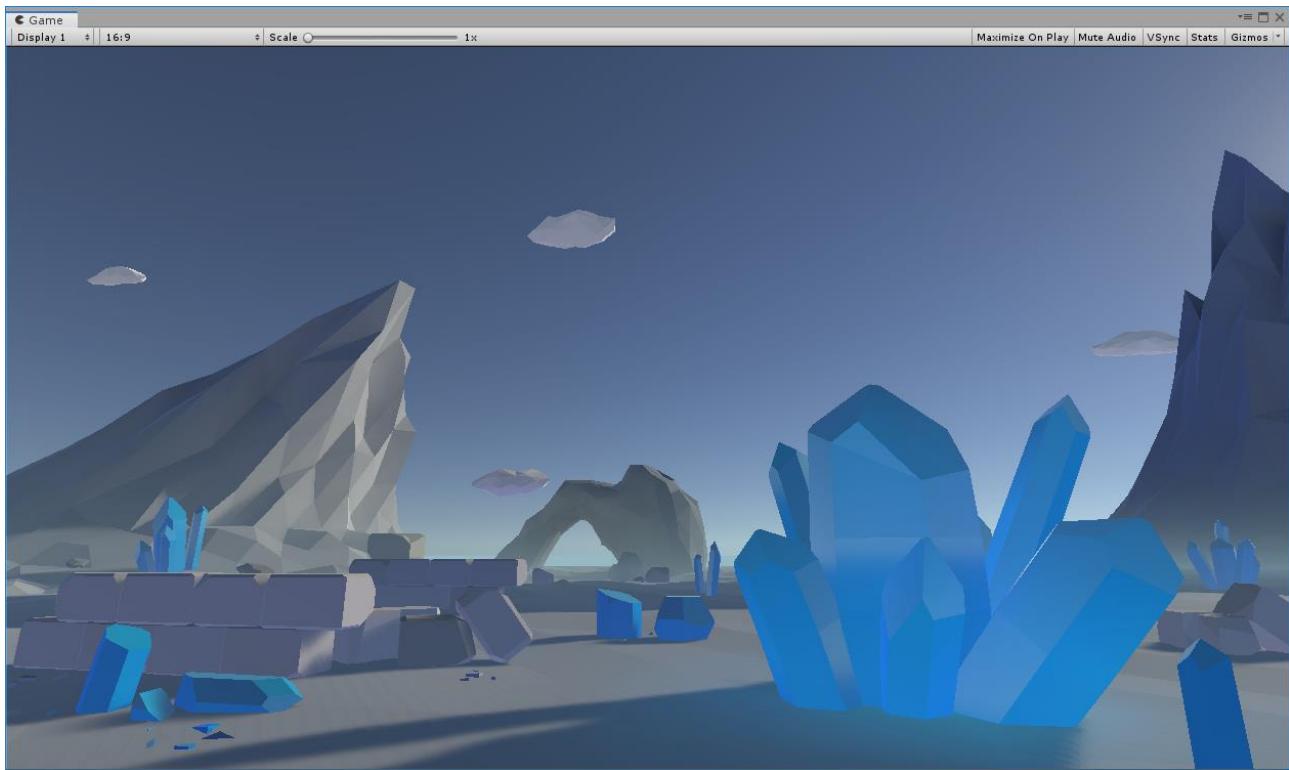


5. Edit the Exponential Fog

Set **Color Mode** to **Constant Color** and use this **Color** (R: 93; G:147; B:168)



Final result



How to use “Low Poly Rocks Pack”

Go to *Assets > Low Poly Rocks Pack > Rock Assets > Prefabs*

Select the Collider type:

- **With_Box_Colliders** – Prefabs use Box Colliders for Unity Terrain support (you can paint Prefabs on the Terrain). Also, Box Colliders are better for performance than Mesh Colliders!
- **With_Mesh_Colliders** – Prefabs use accurate Mesh Colliders (you can use Prefabs by placing with your mouse manually or using “Polybrush” from the Asset Store - recommended).

So, let's say that you want to place Prefabs by hand and use more accurate Colliders - open folder **With_Mesh_Colliders**. Select Color type:

- **1 Color** – One Rock Prefab uses 1 color.
- **2 Color** – One Rock Prefab uses 2 colors (by default, the second color is white for snow. Also, it can be used for (sand, dirt, moss, etc.).

Select which rock type you want to import to your scene. For example, open folder **Round_Rocks** - select and drag **Prefab** to your scene. That's it.

Same for **Bonus Assets**.

Go to *Assets > Low Poly Rocks Pack > Bonus Assets > Prefabs*

Select what you want and drag it to the scene.

Every model pivot is at the center bottom of the model, so you can quickly drop it on the ground, scale and rotate.

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

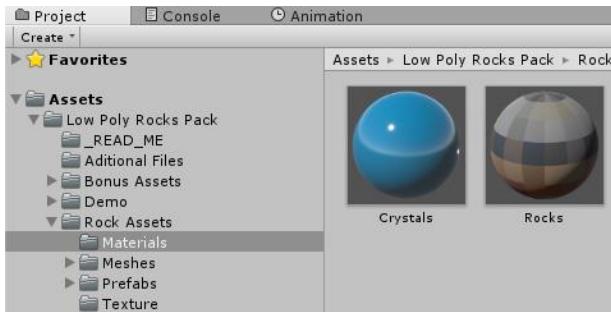
You can change it by tapping on the **buttons**, which are near Move, Scale tools.



How to Change Prefabs Color / Texture

Rocks / Crystals

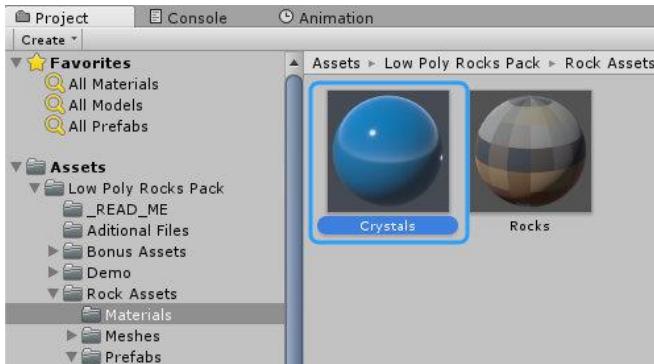
Go to *Low Poly Rocks Pack > Rock Assets > Materials* - here you will find 2 materials.



- Material **Rocks** used for all **Rock Prefabs**
- Material **Crystals** used for all **Crystal Prefabs**

Change Crystals Color

Select **Crystals** Material



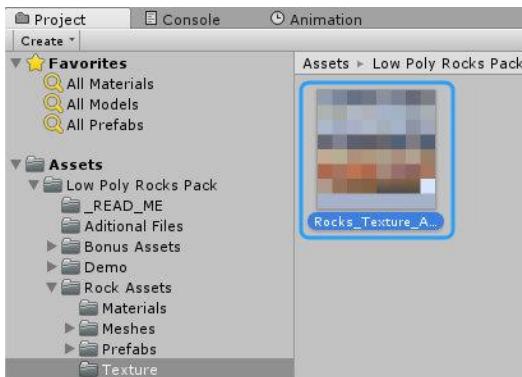
and change **Albedo** Color



Change Rocks Color

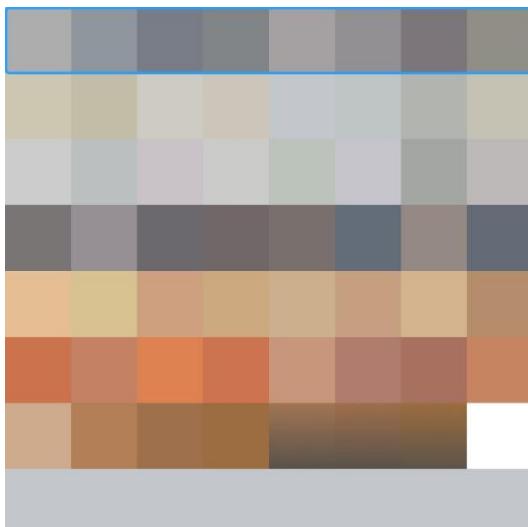
Rocks Material use **1 Texture Atlas**. So, we need to change colors for that texture to change Rock Prefab colors.

Go to *Low Poly Rocks Pack > Rock Assets > Texture*

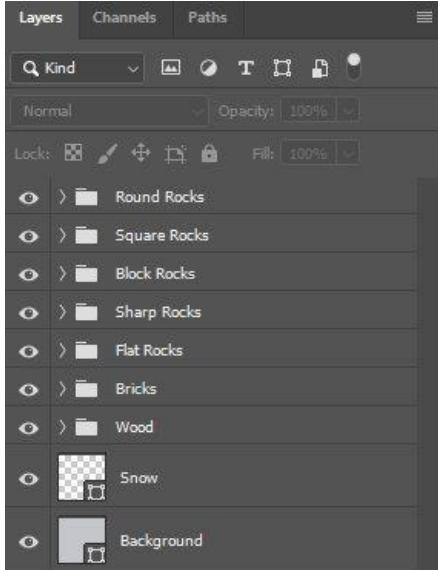


And open **Rocks_Texture_Atlas.png** inside Photoshop, Gimp, Affinity or any other image editing software. Every color square used for one random rock.

The first line of colors used for **Round Rocks**. That means every Round Rock is using one of these colors.



I also included **.psd** file of this texture inside *Low Poly Rocks Pack > Additional Files* folder. Extract **Rocks_Texture_Atlas_PSD.rar** file and open **.psd** inside **Photoshop, Gimp, Affinity Designer, etc.** This way, you can see which colors are for which rock type by looking into **Layer Names**, and edit those colors more easily.

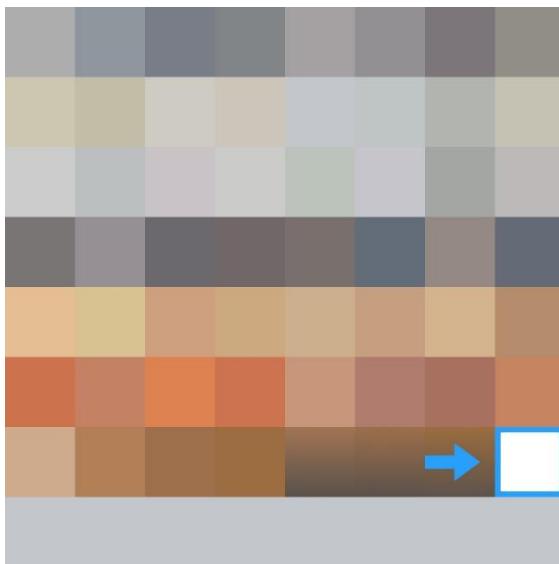


Change The Second Color For (2 Color - Rock Prefabs)

Go to *Low Poly Rocks Pack > Rock Assets > Texture*

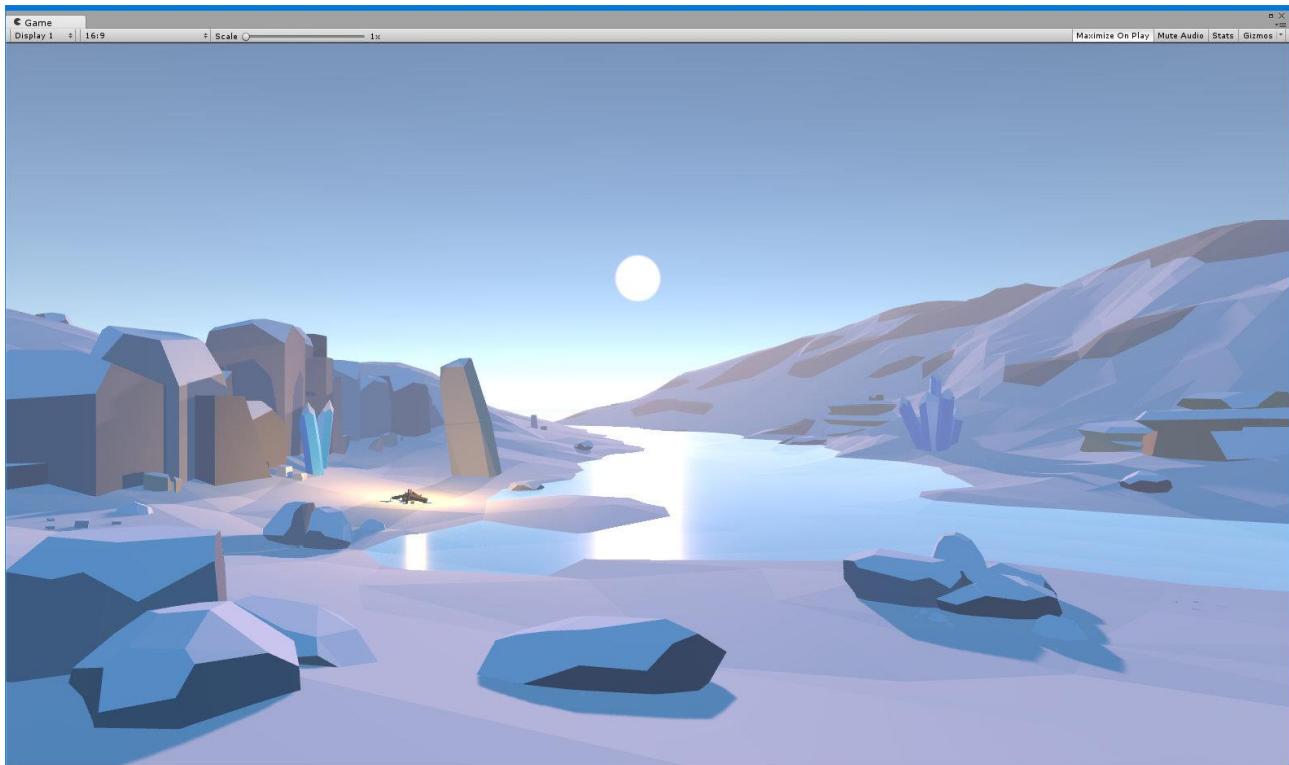
And open **Rocks_Texture_Atlas.png** inside Photoshop, Gimp, Affinity or any other image editing software. Every color square used for one random rock.

The **White** color square at the bottom right used as a second color for 2 Color prefabs (Snow color by default).

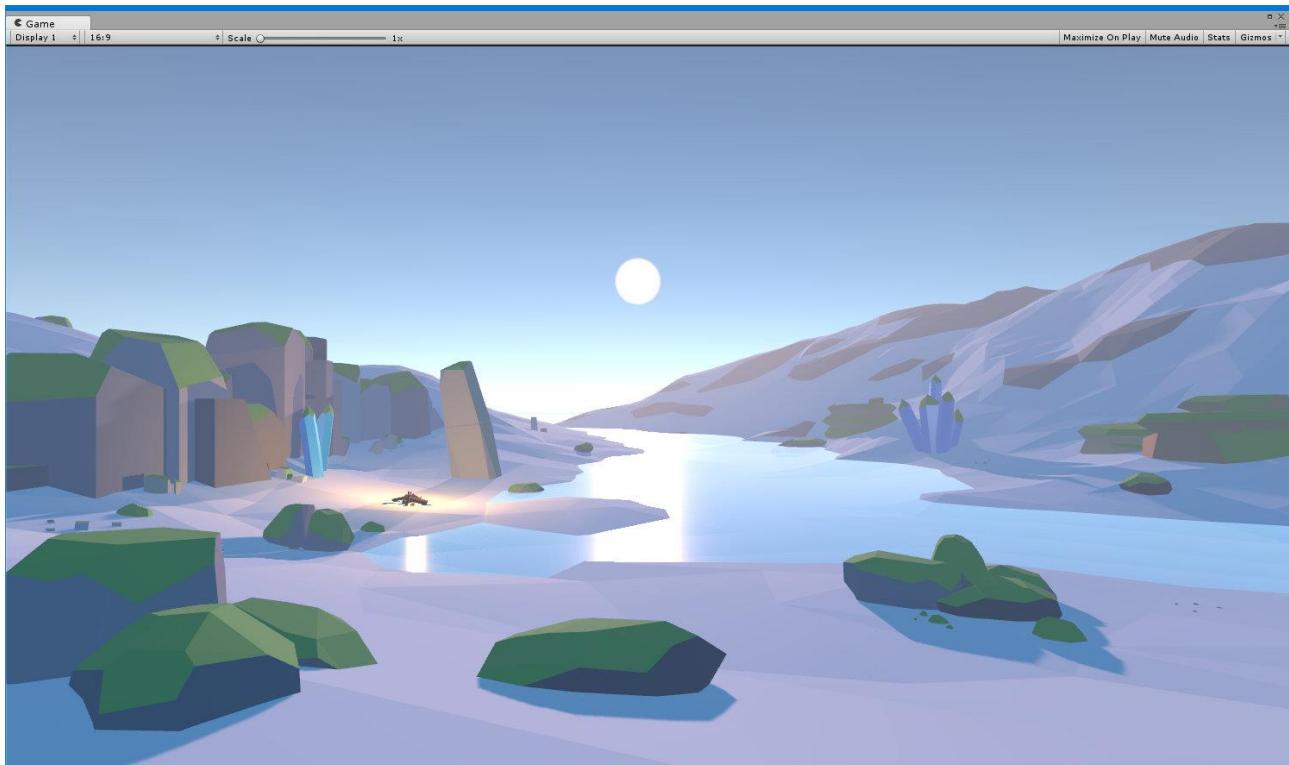


You can easily change it to any other color - like yellow for sand, green for moss, etc.

Here is an example of the same scene with a **White** color



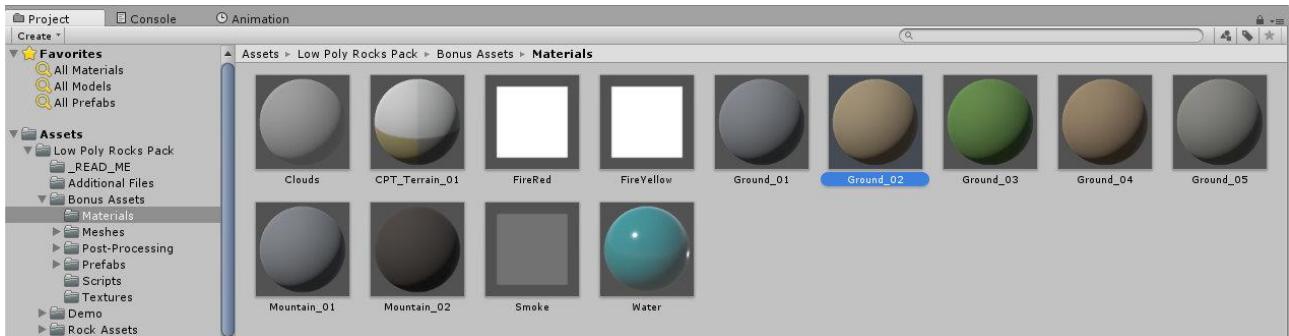
And with a **Green** color



*To change the terrain color, follow these steps!

Change Bonus Assets Color

To change colors for Bonus Assets (Clouds, Fireplace, Ground, Mountains, Pyramid, and Water), go to *Low Poly Rocks Pack > Bonus Assets > Materials*.

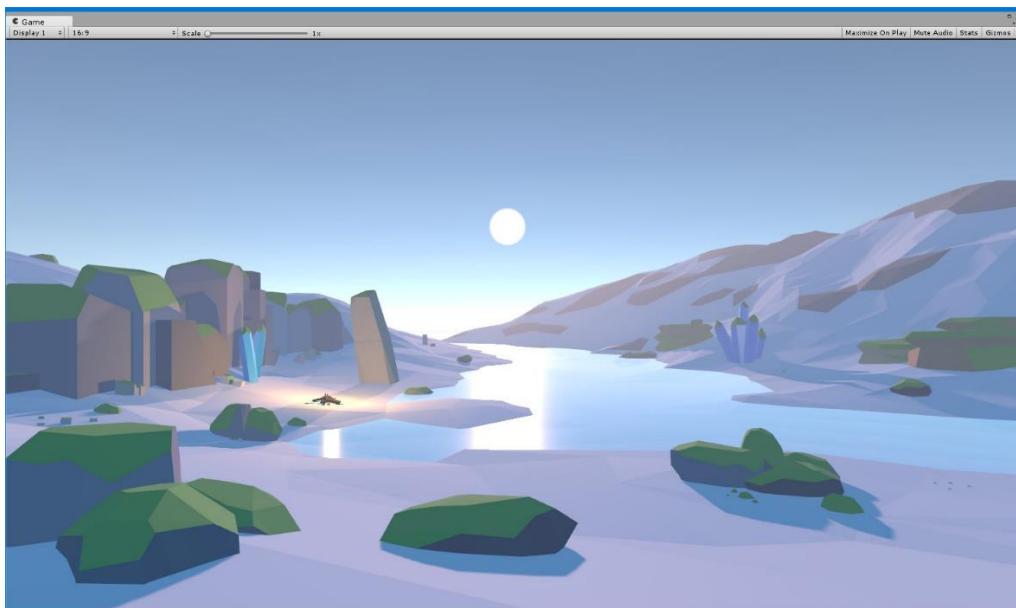


Select the Material you want to edit and in the **Inspector** change **Albedo Color**.

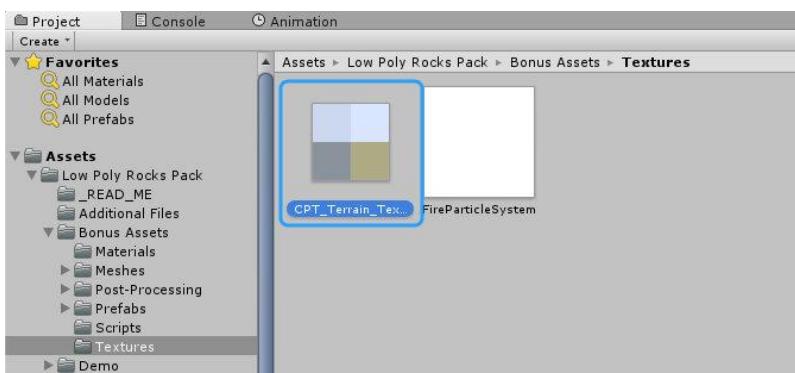


There is a **CPT_Terrain_01** material which is applied to the Terrain prefab -

CPT_Island_L_f_08. This Terrain prefab is used in **Demo_07** scene. Snow terrain:

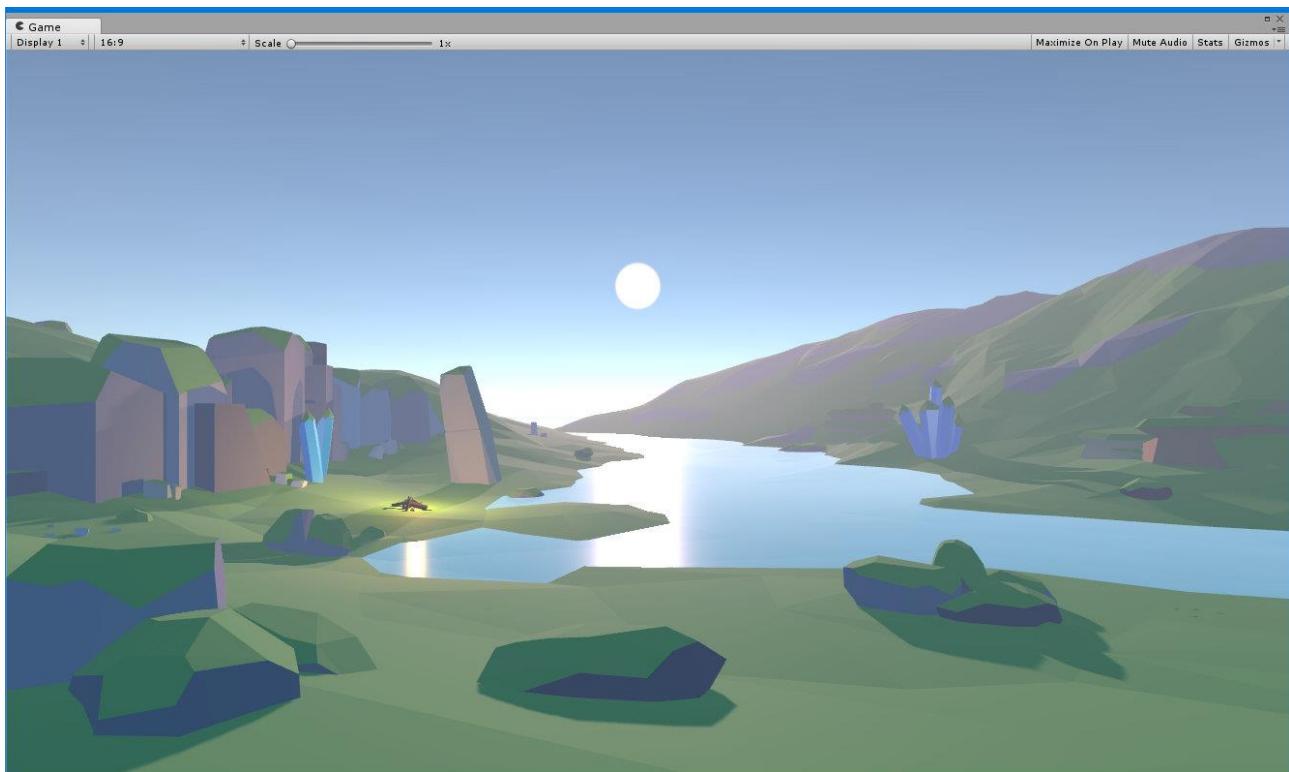


It uses two colors from the texture atlas. You can change those colors by editing the **CPT_Terrain_Texture_Atlas.png**. Texture located in *Low Poly Rocks Pack > Bonus Assets > Textures*.



Open **CPT_Terrain_Texture_Atlas.png** inside Photoshop, Gimp, Affinity, or any other image editing software.

The 1st **White** color square at the top left is used as a Snow. 2nd Dark Grey color at the bottom left is used for terrain rocks.

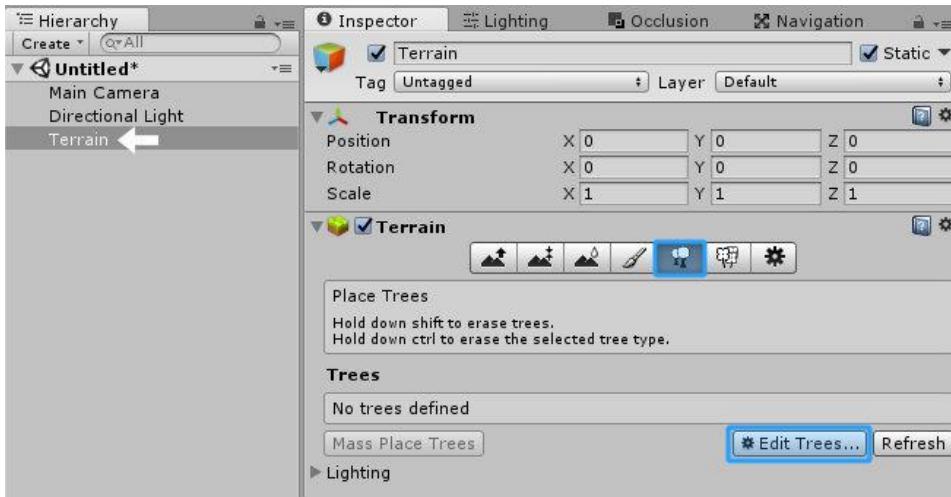


At this point, I just changed the 1st **White** color to **Green**.

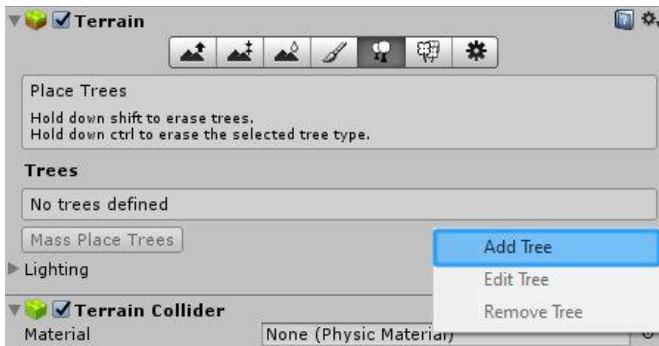


How to Draw Rock Prefabs on Unity Terrain

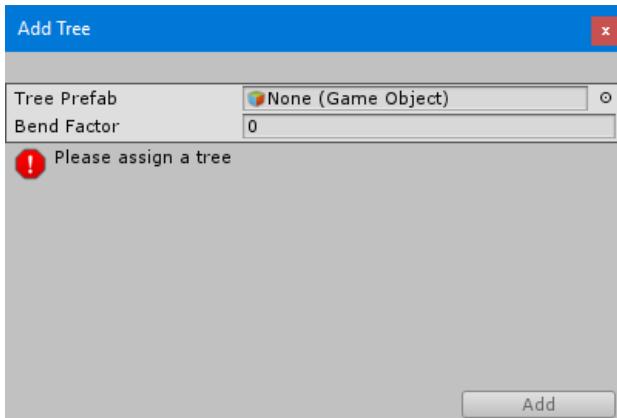
Select your Unity Terrain and go to **Place Trees** tab. Click on **Edit Trees...**



...and press on **Add Tree**

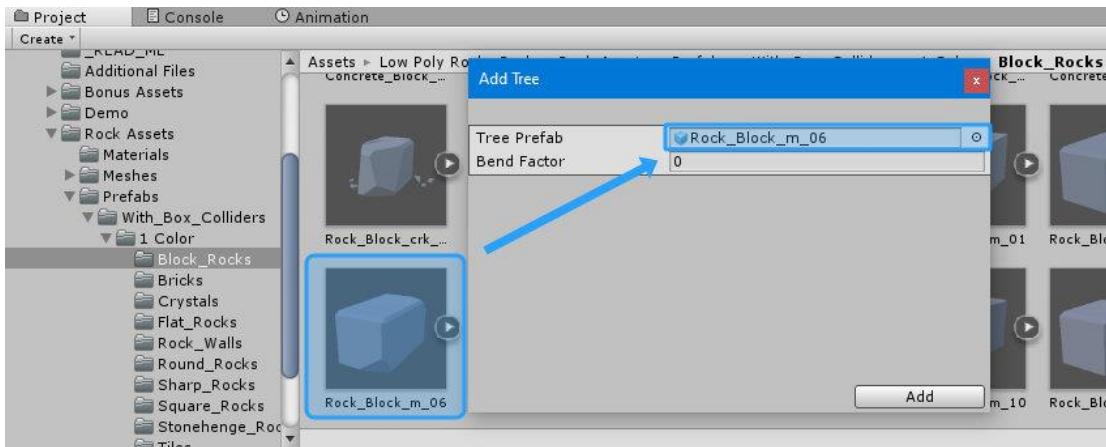


You should see a popup window **Add Tree**

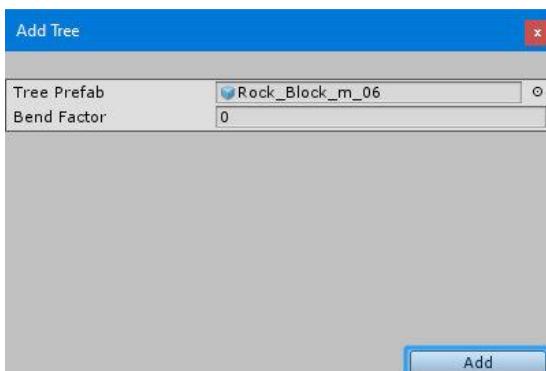


Go to *Low Poly Rocks Pack > Rock Assets > Prefabs > With_Box_Colliders* > and select any Rock Type you want to use (I used **1 Color > Block_Rocks**), drag and drop Prefab to **Tree**

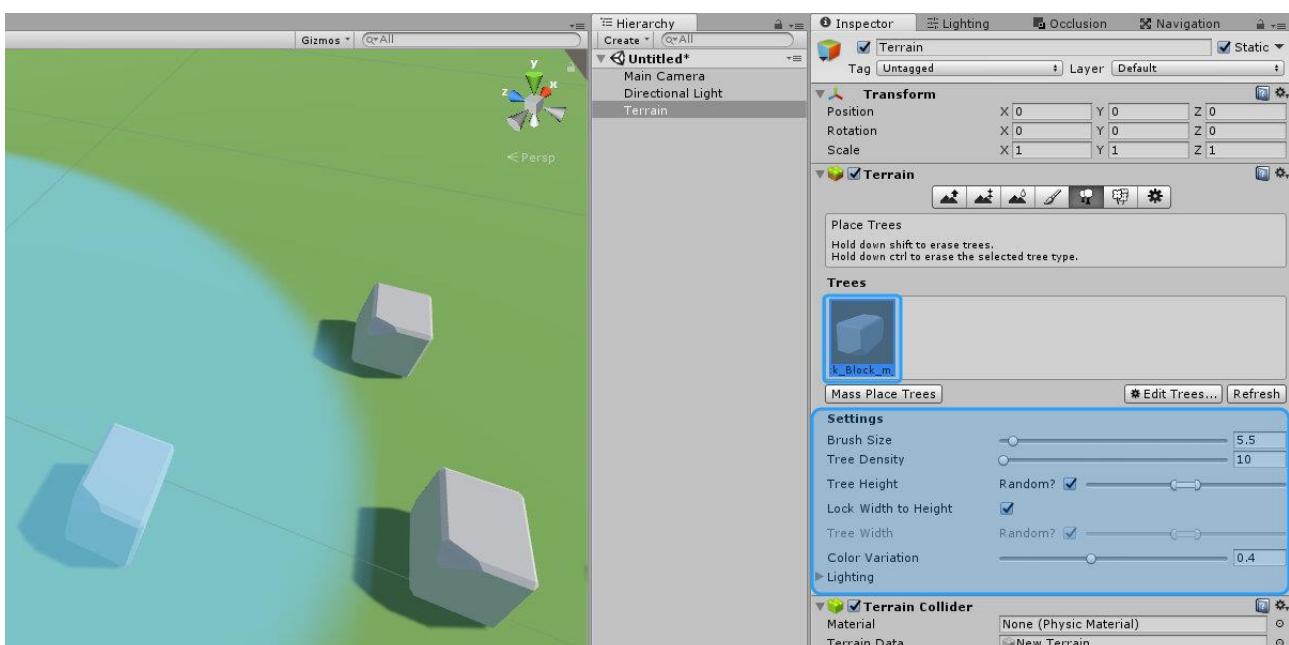
Prefab tab:



Press **Add**



That's it! Select **Rock Prefab**, change **Settings**, and paint.



How to Draw Rock Prefabs on Mesh Terrain Using Polybrush

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

1. Import Polybrush.

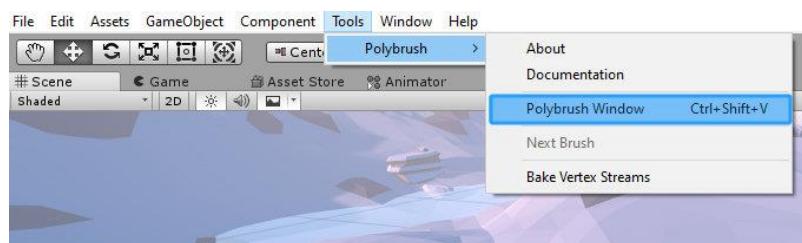
Go to *Window > Asset Store* and search for Polybrush:



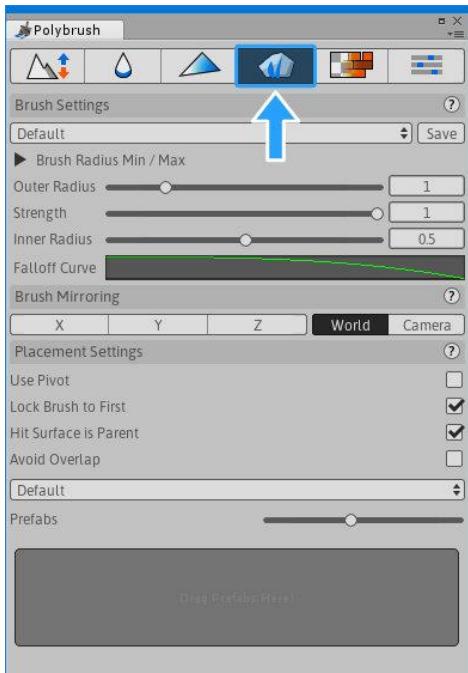
Import it to your project.

2. Setup and use Polybrush to paint Rock Prefabs on any Mesh.

Go to *Tools > Polybrush > Polybrush Window*

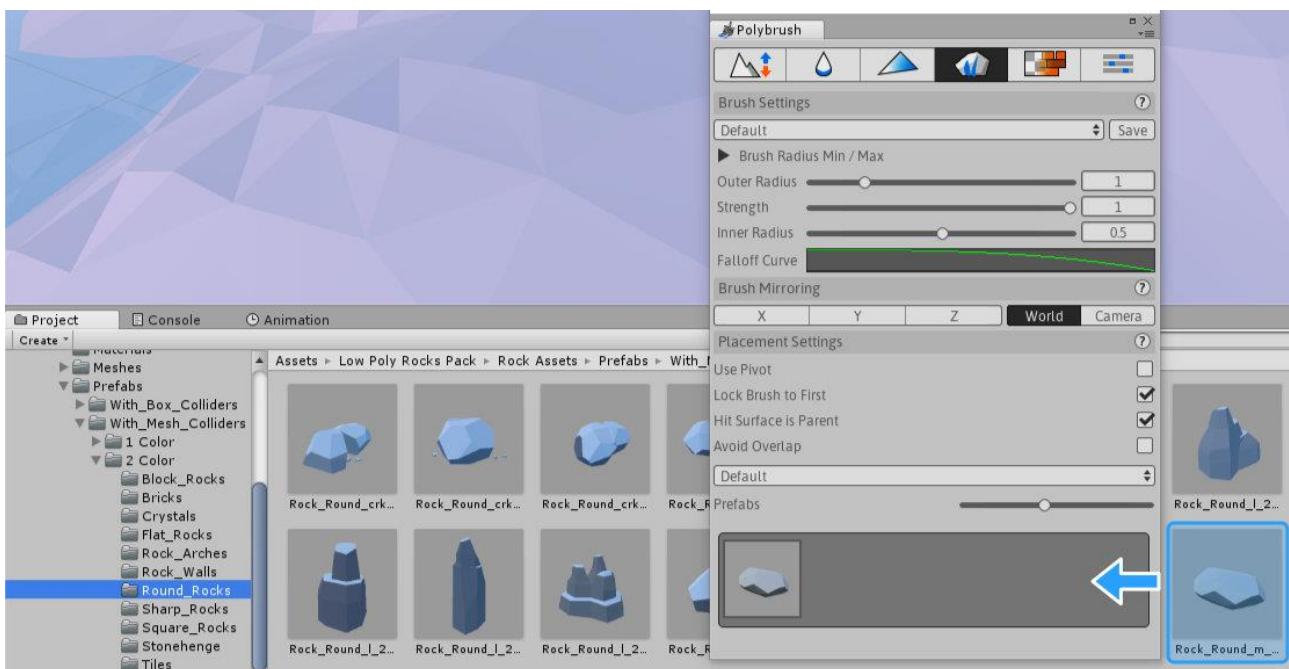


You should see a Polybrush window. Open the Paint Prefabs tab:



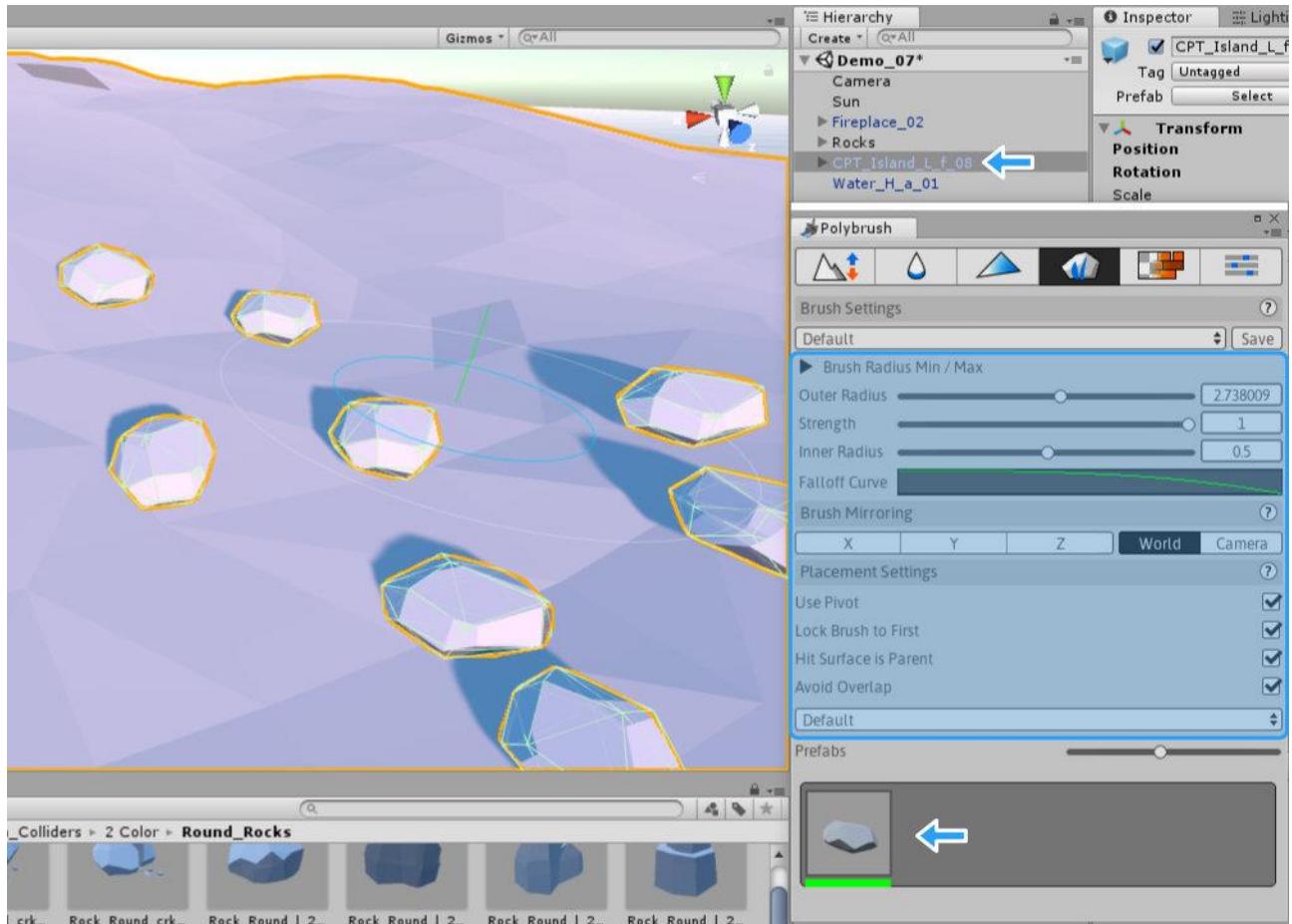
Now add any Rock Prefab you want to paint. I will choose Rock_Round_m_2C_04 from *Low Poly Rocks Pack > Rock Assets > Prefabs > With_Mesh_Colliders > 2 Color > Round_Rocks*.

Drag and drop any Rock Prefab to Prefabs window:



*You can add as many Prefabs as you want.

Select the mesh to paint on. I will select **CPT_Island_L_f_08** from the **Demo_07** scene. Then decide which Prefab to paint, in this case, press on the **Rock_Round_m_2C_04** prefab we just added. And paint prefab on the mesh in the **Scene** view.



*I recommend enabling **Use Pivot** in the **Placement Settings** if you are painting my Rock Prefabs!

Additional Info

Naming Conventions

Prefab name example 1: **Rock_Round_crk_m_01**

- **Rock_Round** – Rock Type
- **crk** – means the rock is cracked
- **m** – medium size
- **01** – Prefab number

Prefab name example 2: **Rock_Round_crk_m_2C_01**

- **2C** – 2 Color (rock uses 2 colors. By default, the second color is white - snow)

You can find these letters:

s – small size

m – medium size

l – large size

crk – means the rock is cracked.

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

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



Same with **Direction Lights** 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