

Low Poly Vegetation 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

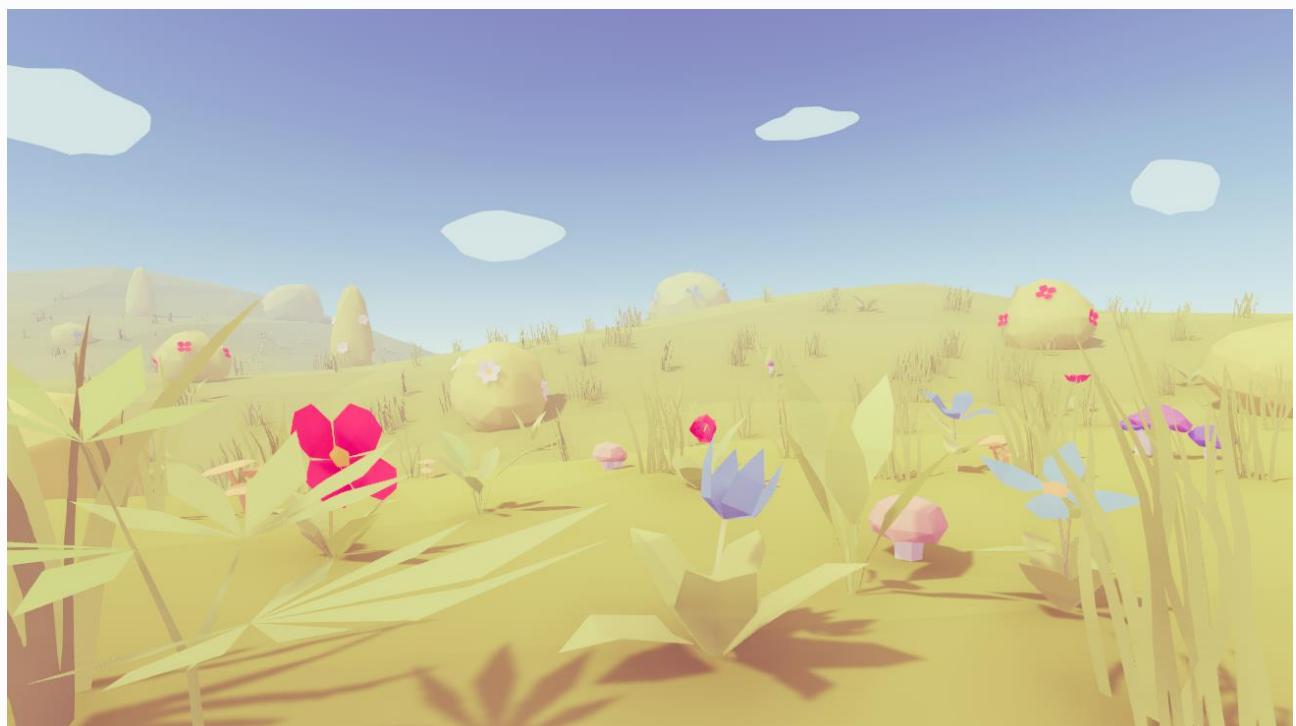
HOW TO SETUP DEMO SCENES (POST-PROCESSING) IN UNITY 2018.4 LTS AND UP (FOR PC).....	5
HOW TO SETUP DEMO SCENES IN UNITY 2018.4 LTS AND UP (FOR ANDROID).....	16
UNITY 2019.3 AND UP - UNIVERSAL RENDER PIPELINE (URP)	25
UNITY 2018.4 LTS AND UP - LIGHTWEIGHT RENDER PIPELINE (LWRP).....	31
UNITY 2018.4 LTS AND UP - HIGH DEFINITION RENDER PIPELINE (HDRP)	35
HOW TO USE "LOW POLY VEGETATION PACK".....	47
HOW TO CHANGE VEGETATION PREFABS COLOR / TEXTURE.....	49
CHANGE VEGETATION PREFAB COLOR	49
CHANGE GRASSPLANE PREFAB COLOR.....	51
CHANGE BONUS ASSETS COLOR	52
HOW TO PAINT VEGETATION PREFABS ON UNITY TERRAIN.....	53
HOW TO PAINT VEGETATION PREFABS ON MESH TERRAIN USING POLYBRUSH.....	55
HOW TO PAINT GRASS TEXTURES ON UNITY TERRAIN.....	58
ADDITIONAL INFO.....	62
NAMING CONVENTIONS	62
SCRIPTS	63
CONTACTS	64

Demo scenes

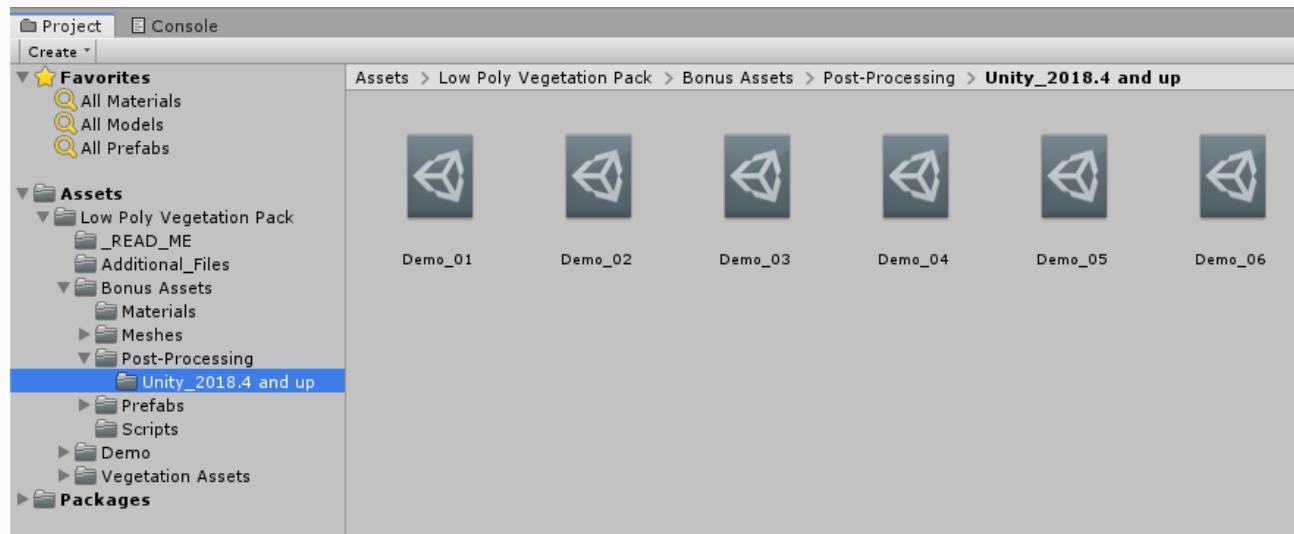
Now as you have imported the whole “**Low Poly Vegetation Pack**” to your Unity project, go to *Low Poly Vegetation Pack > Demo > Demo_Scenes* - and open any Demo Scene (here is a **Demo_01** example). By default, the scene should look like this inside the **Game** view without any image effects applied. 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!

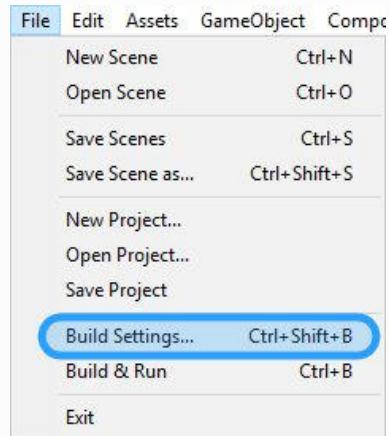
Post-Processing in Unity 2018.4 LTS and up

*You need at least Unity 2018.4 LTS to setup Post-Processing by following my tutorial!

How to Setup Demo Scenes (Post-Processing) in Unity 2018.4 LTS and up (For PC)

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

Go to *File > Build Settings*



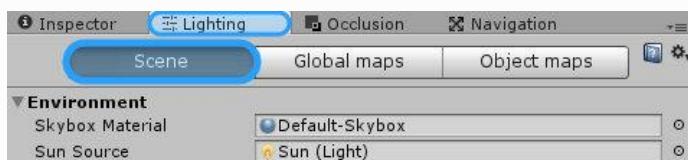
Select **PC, Mac & Linux Standalone** and hit **Switch Platform** button.



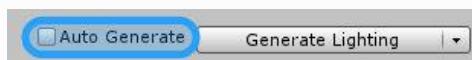
2. Clean GI Cache (Optional) – Skip this if you don't have any light baking errors!

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*).

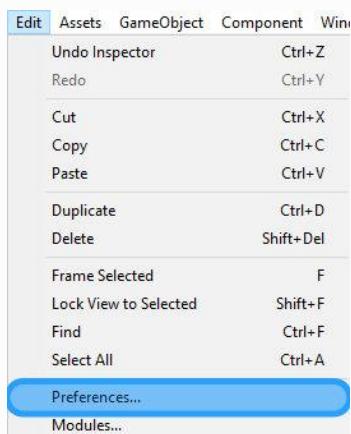


At the bottom you will see this:

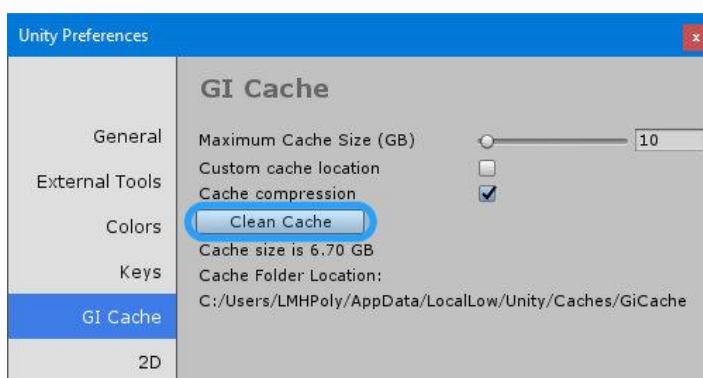


unchecked **Auto Generate**.

Go to *Edit > Preferences*



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



Enable **Auto Generate** / bake feature

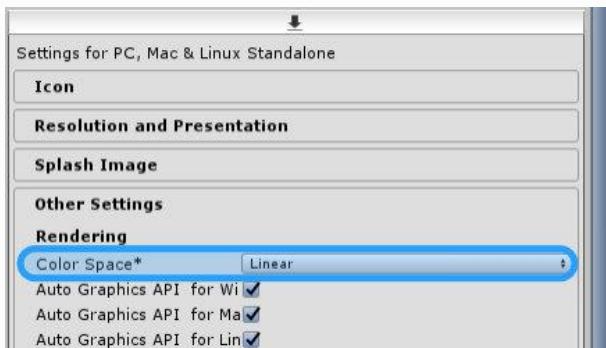


and wait until the generation is done (blue loading bar at the right bottom corner).

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

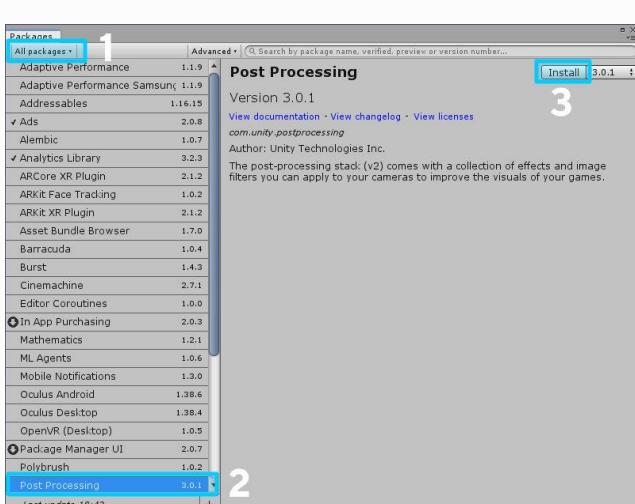
Go to *Edit > Project Settings > Player*

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



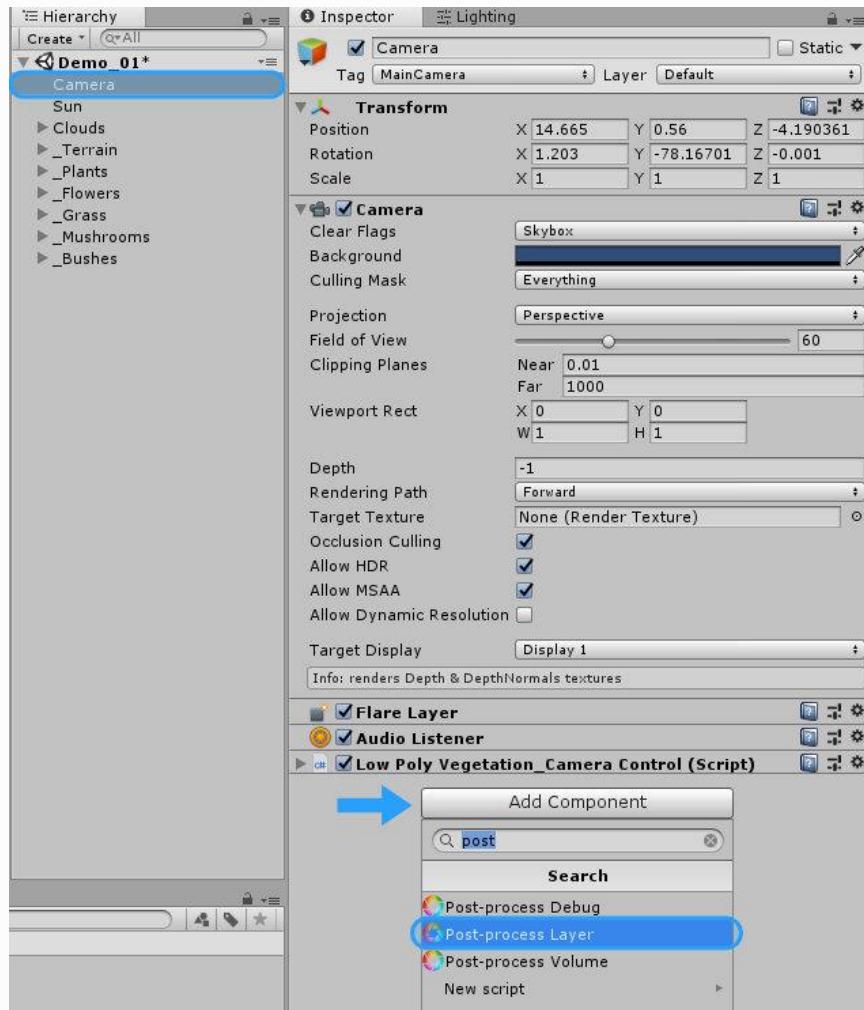
4. Install the **Post Processing** package from the **Package Manager**.

Go to *Window > Package Manager* - open the tab **All packages** and search for **Post Processing** (if you don't see it wait for packages to load!). Select it and hit the **Install** button:



5. Apply Post-process Layer to the Camera.

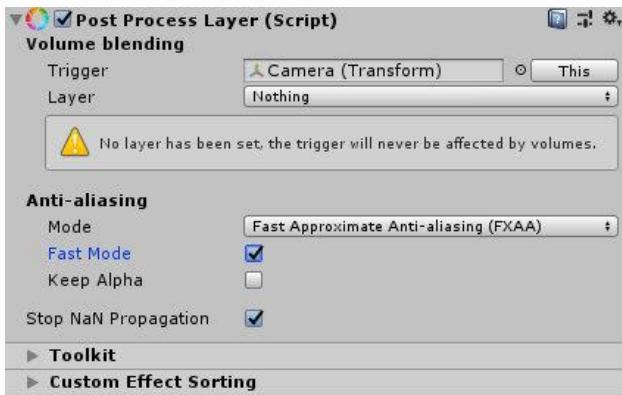
Select the **Camera** in the Hierarchy, press on **Add Component**, type **post** in the search window, and press on **Post-process Layer** to apply.



*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:

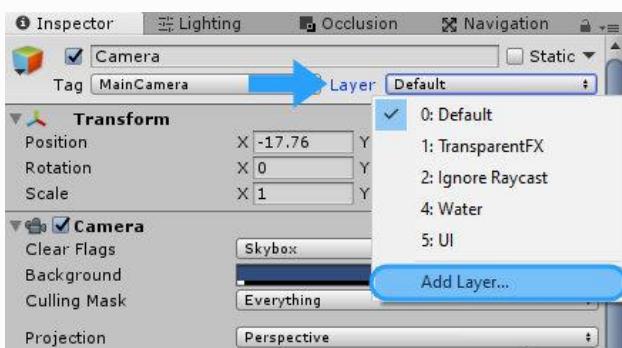
- Restart Unity.
 - If it still doesn't work, go to *Window > Package Manager*, remove **Post Processing** package.
 - Restart Unity
 - Install the **Post Processing** package again. Now it should work.

Post Process Layer settings:

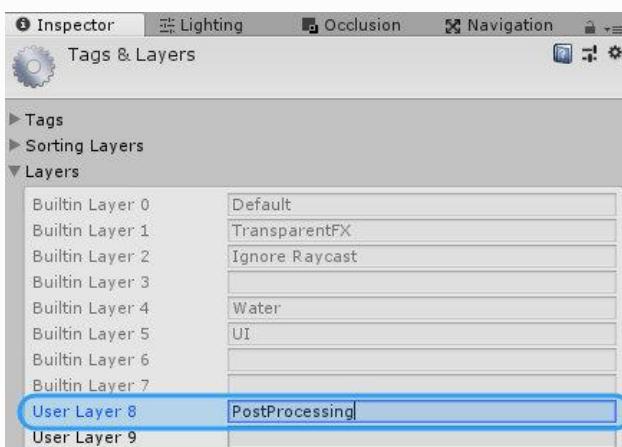


6. Create **PostProcessing** layer and apply it to the **Camera**.

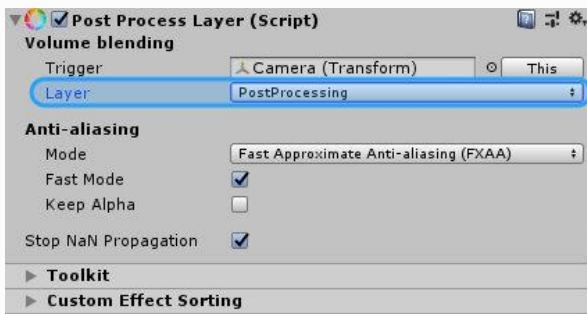
Press on **Layer Default > Add Layer...**



Create a new layer called **PostProcessing**:

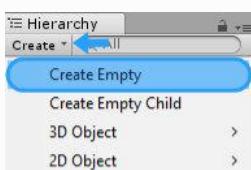


Select the **Camera** and inside the **Post Process Layer – Volume blending – Layer** apply **PostProcessing** layer we just created:

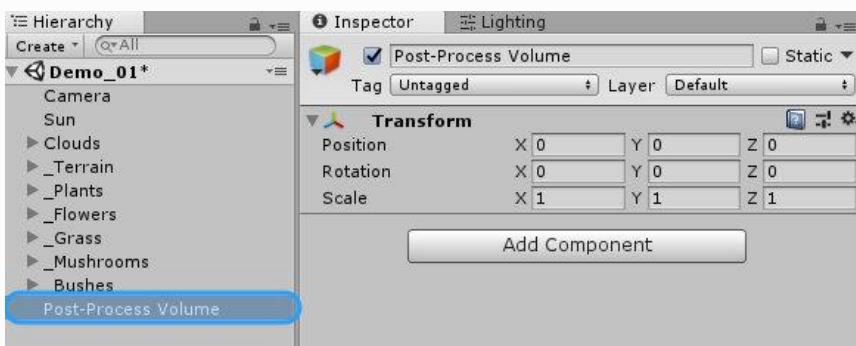


7. Create a Post-Process Volume.

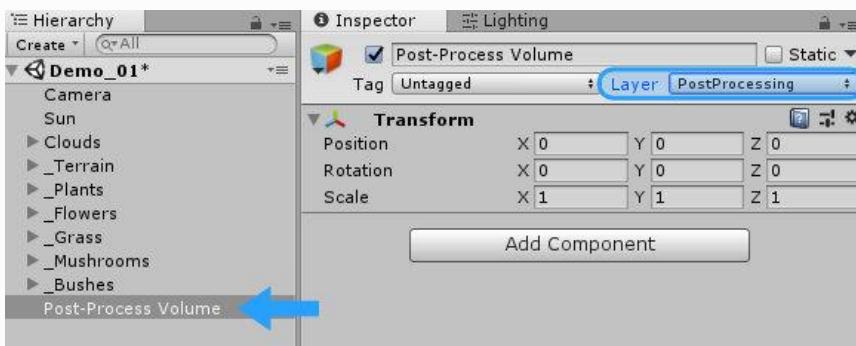
Create Empty game object:



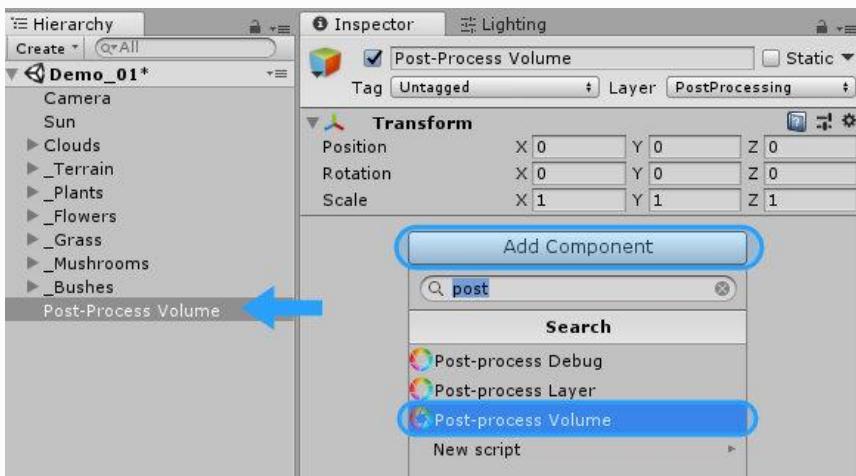
Rename it to something like **Post-Process Volume**:



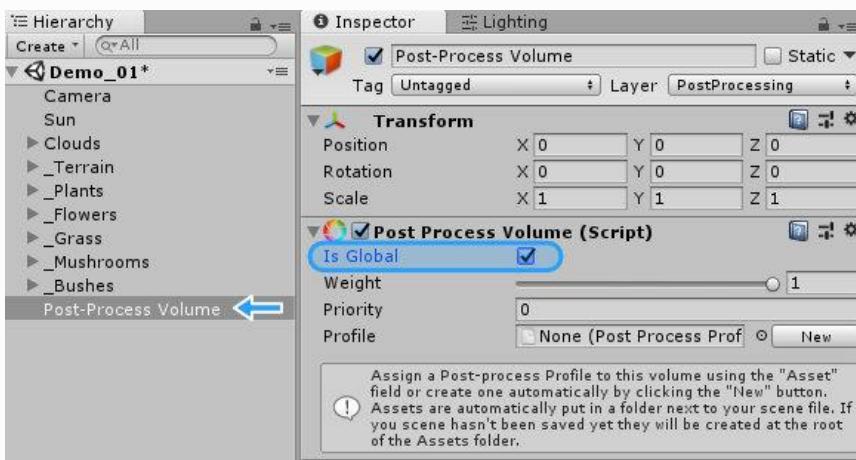
Set **Layer** to **PostProcessing**, the layer we just created before:



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



Enable **is Global**:

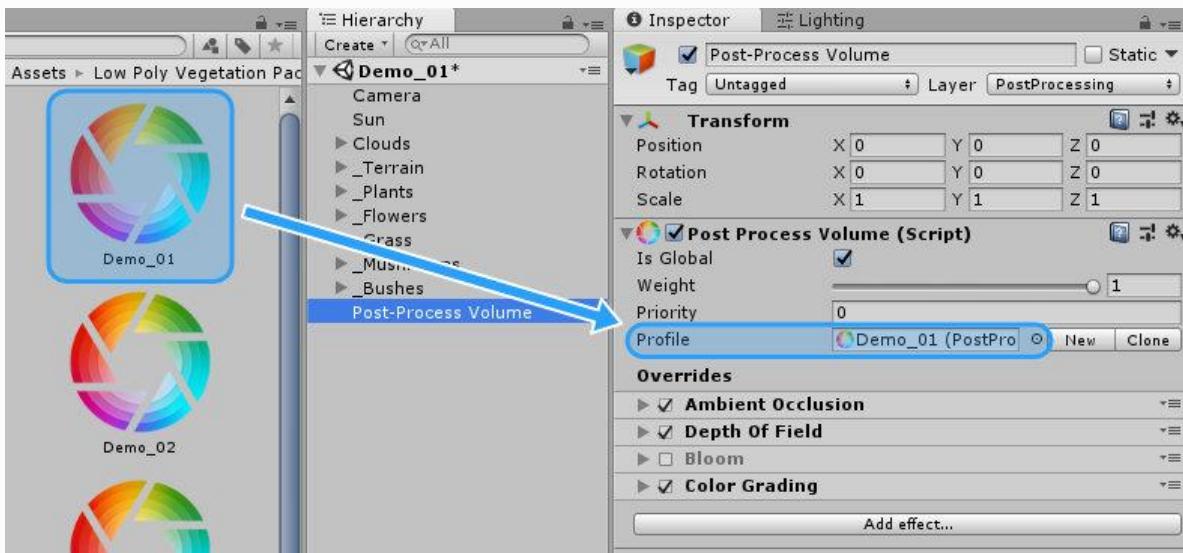


8. Apply Post-Processing Profile.

I've created 7 Post-Processing Profiles for 7 Demo scenes with different settings. Go to *Low Poly Vegetation Pack > Bonus Assets > Post_Processing > Unity_2018.4 and up*.



We have opened **Demo_01** scene - so we will use **Demo_01** Post-Processing Profile. Drag and drop **Demo_01** Post-Processing Profile to **Profile** slot inside the **Post Process Volume**:



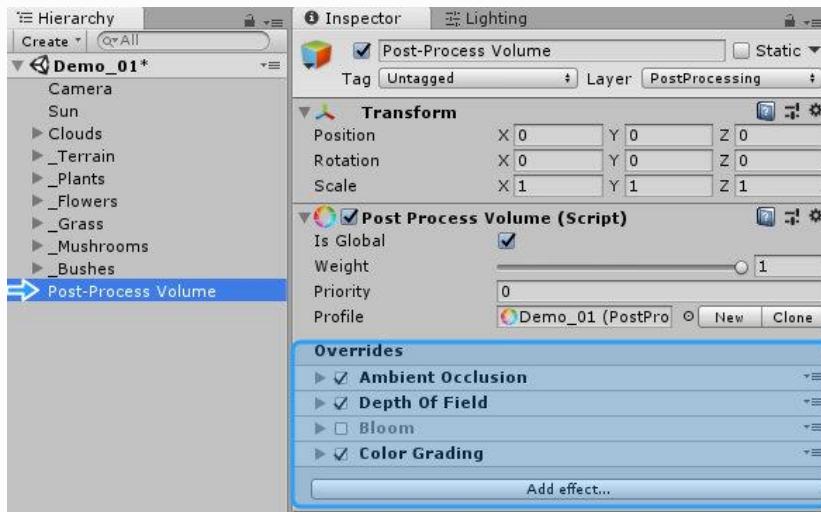
Now your scene should look like this (Demo_01):



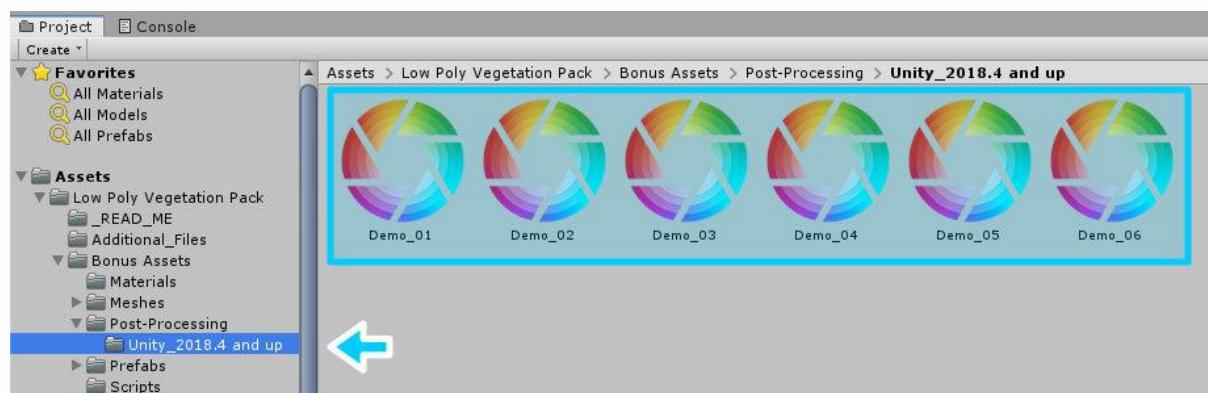
***For Low-End PC's** - if you hit play and it lags, try disabling Post-Processing effects one by one on the Post-Processing Profile settings!

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.



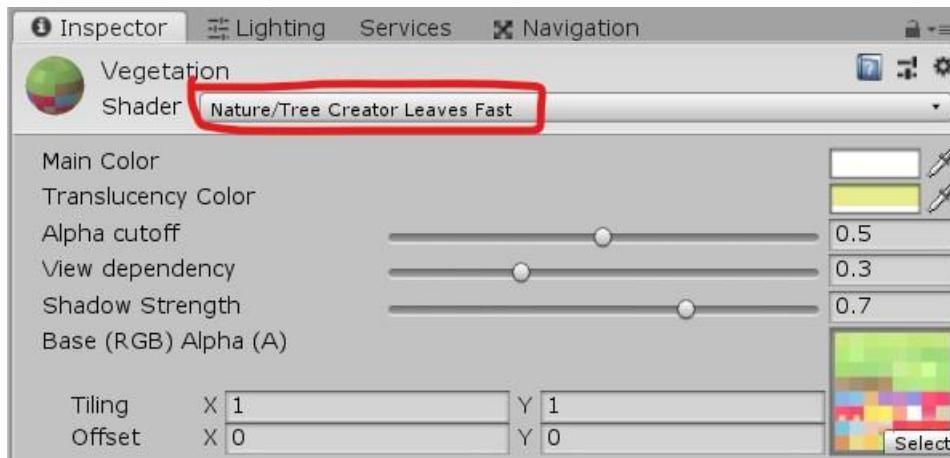
To make everything look a bit lighter and more pleasant to the eyes. From this:



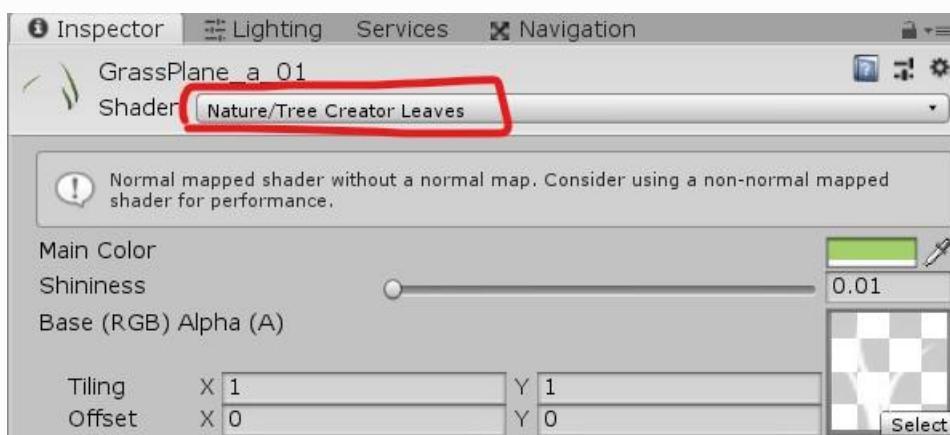
To this:



You can change **Vegetation** material to ***Nature/Tree Creator Leaves Fast***



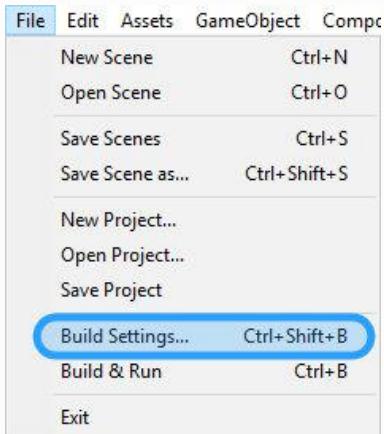
And all of the **GrassPlane** materials to ***Nature/Tree Creator Leaves***



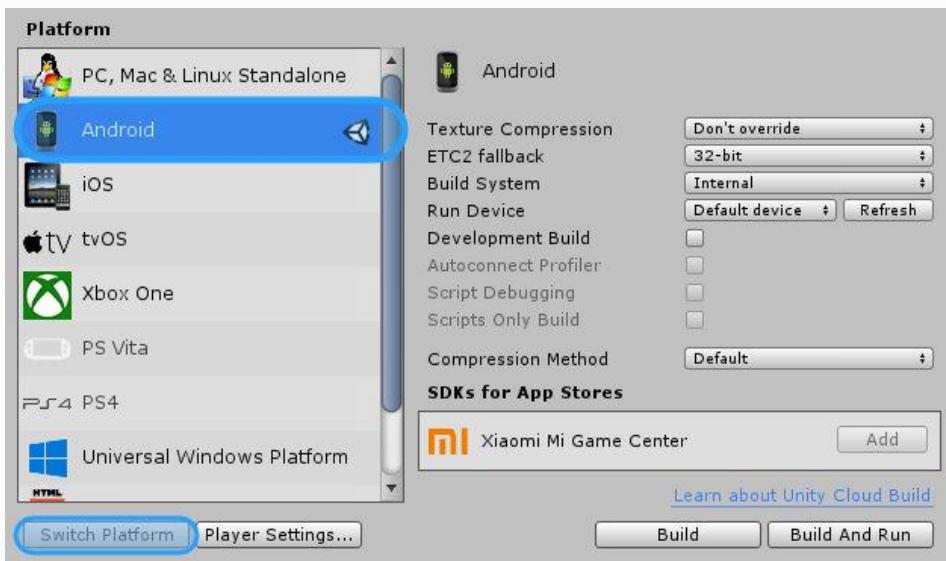
How to Setup Demo Scenes in Unity 2018.4 LTS and up (For Android)

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

Go to *File > Build Settings*



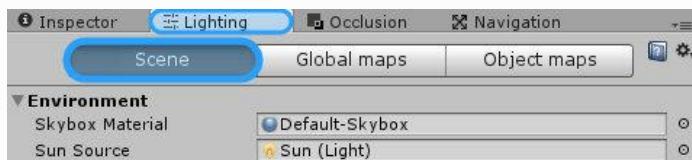
Select **Android** and hit the **Switch Platform** button.



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:

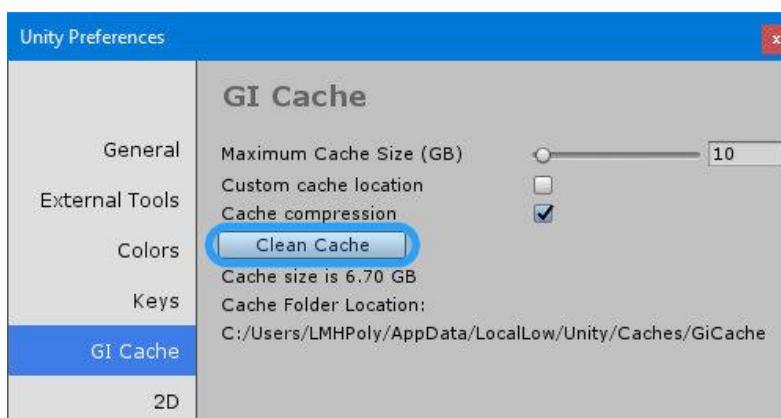


unchecked **Auto Generate**.

Go to *Edit > Preferences*



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



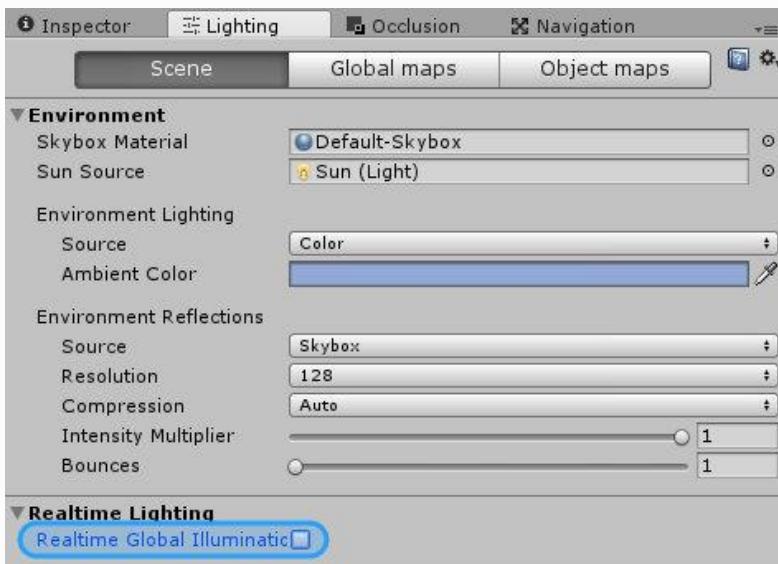
Enable **Auto Generate** / bake feature



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*).

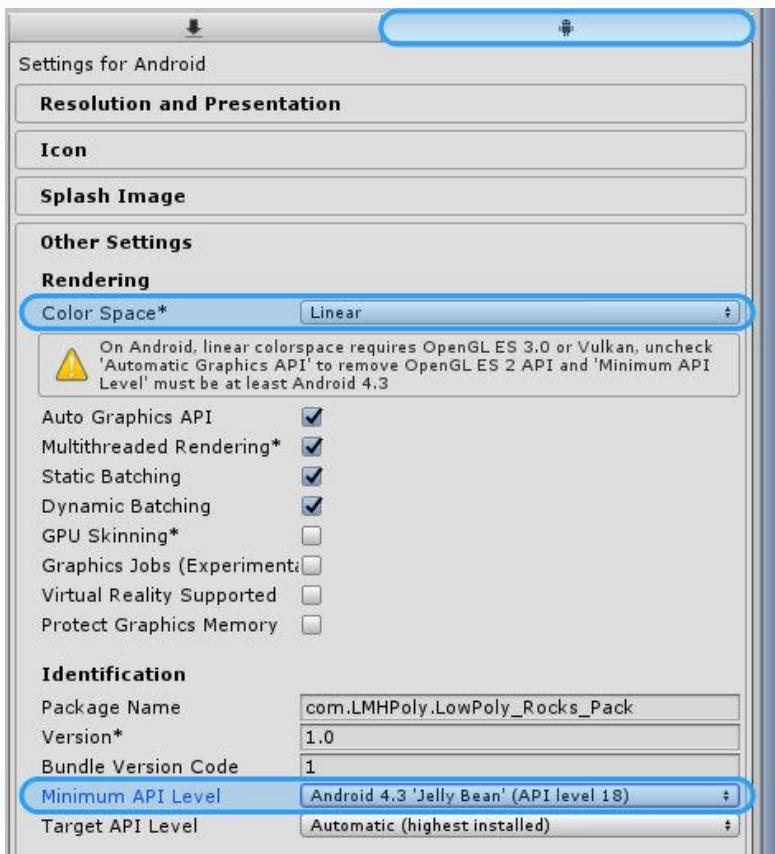


4. Make sure that **Color Space** is set to **Linear** (not all devices support it).

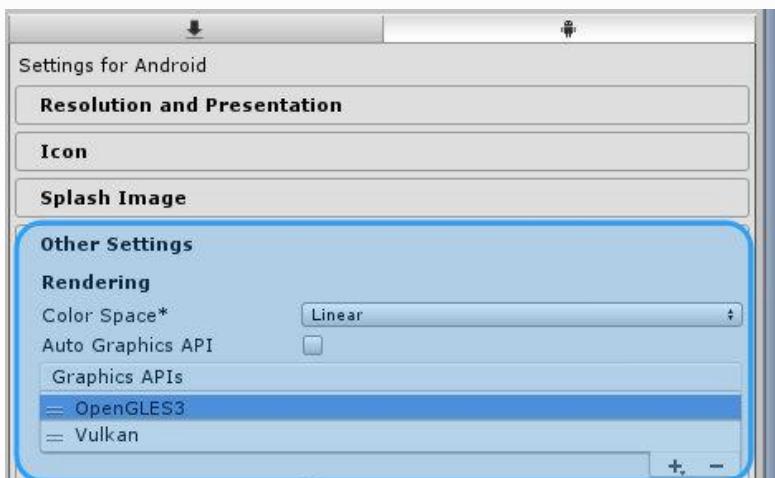
Go to *Edit > Project Settings > Player*

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!

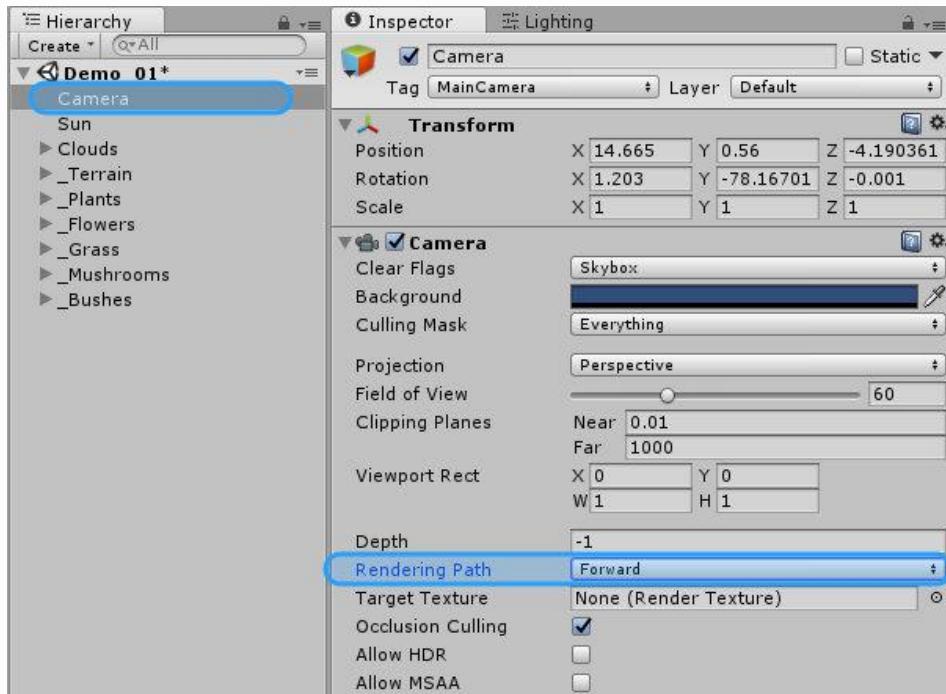


Also, uncheck **Auto Graphics API** and remove all Graphic APIs from the list, leave only **OpenGL ES3** and **Vulkan** (if you can't see it, press on **+** to add it). Make sure your Android device supports one of those graphic APIs!



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**.

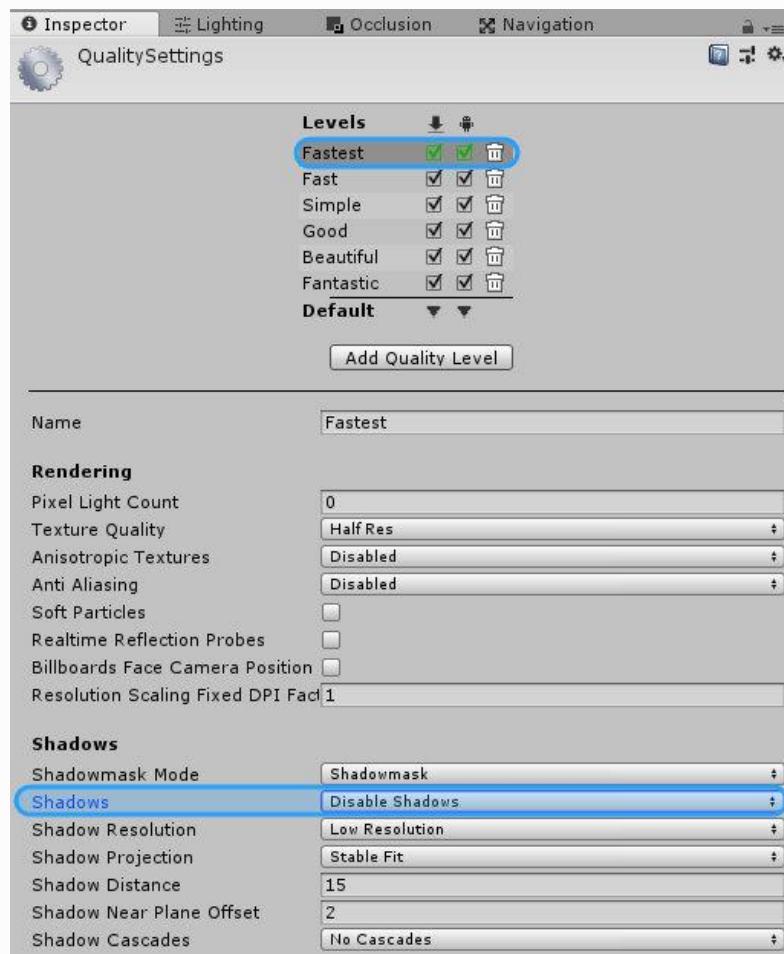


*If you set **Rendering Path** to **Deferred**, the game might slow down a lot on mobile!

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.

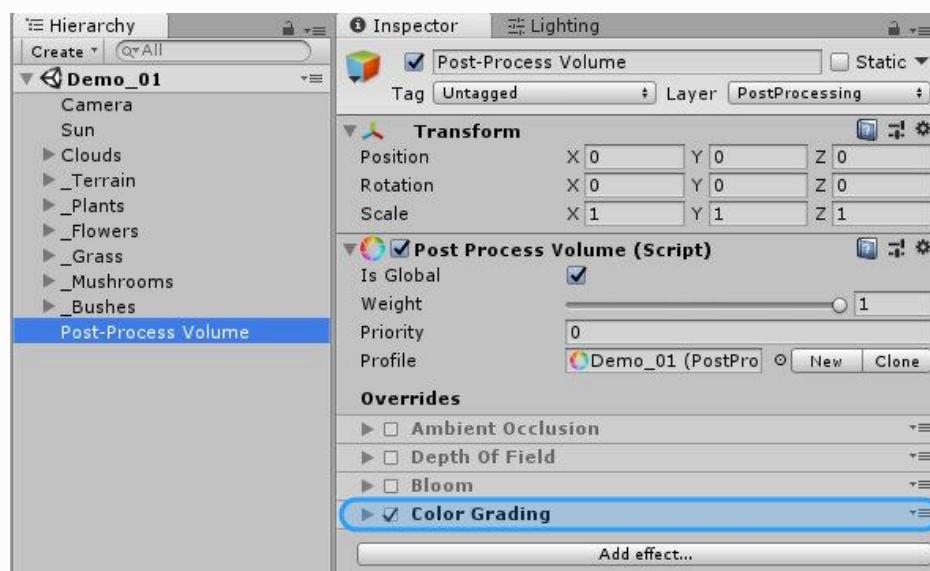
7. Import and enable **Post Processing** image effects (Optional – Big performance hit for mobile devices!).

Go to the part of the documentation: [Post-Processing in Unity 2018.4 LTS and up](#)

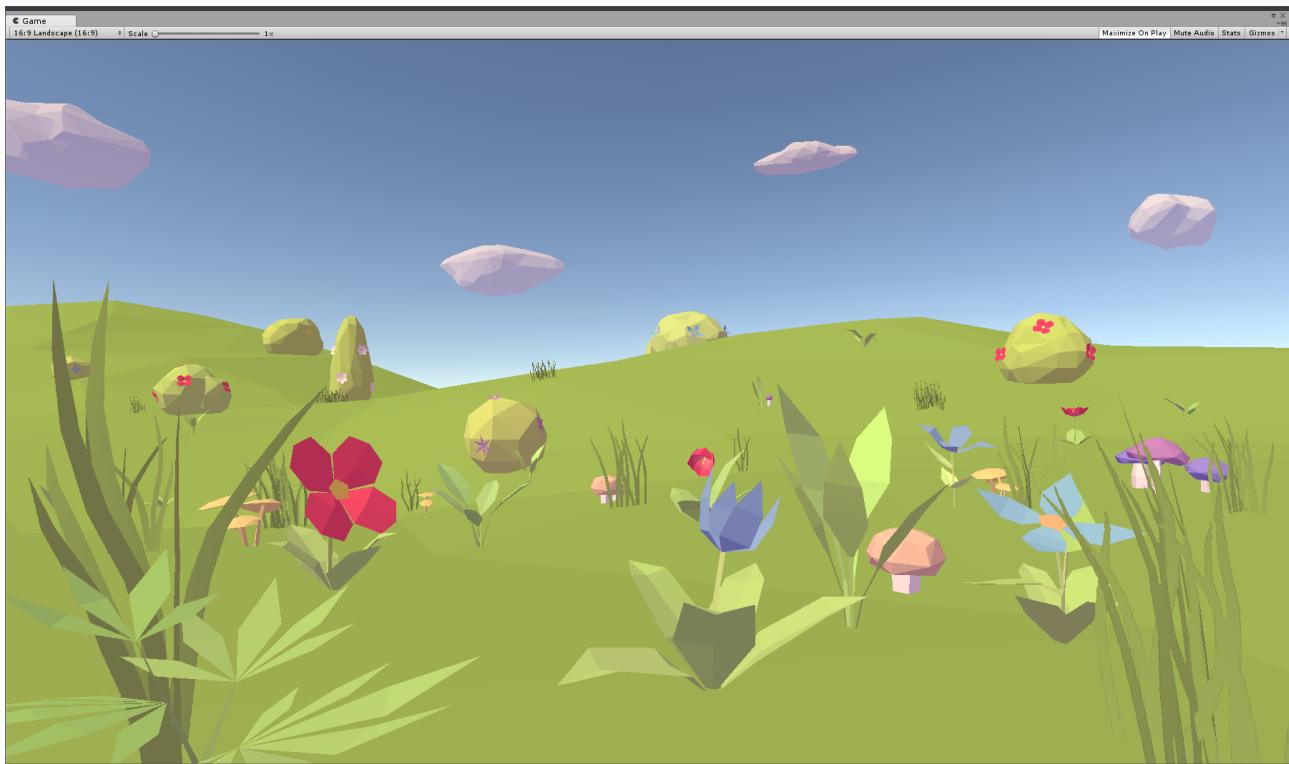
And follow those steps.

*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_01** scene should look like this (if you skipped all **Optional** steps, and with Realtime Shadows **Disabled**):



Old Demo_01 picture, now it looks slightly different!

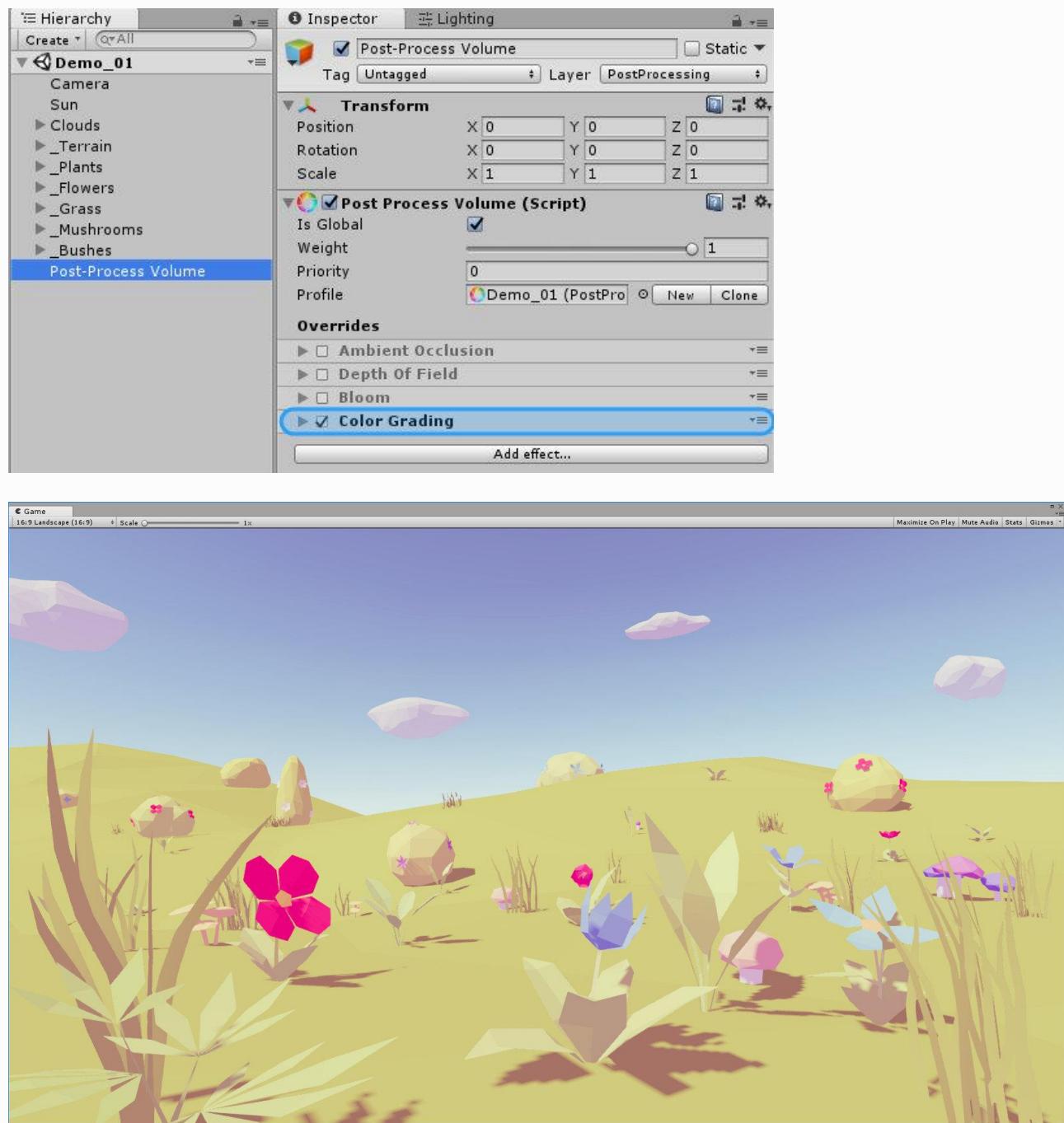
By using **Unity 5.5** and up + **Linear** lighting feature for **Android** and **iOS**, you can achieve much better results than using **Gamma** lighting!

All demo scenes including **Demo_01** 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.

Demo_06 scene with a lot of animated grass made for PC runs at 38FPS+ on Xperia Z Ultra. Also, tested on Google Pixel 2 XL - runs at solid 60FPS with realtime Shadows enabled.

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

Demo_01 scene with the same settings + Post Processing (**Color Grading** enabled only) +
Realtime Shadows: medium resolution enabled:

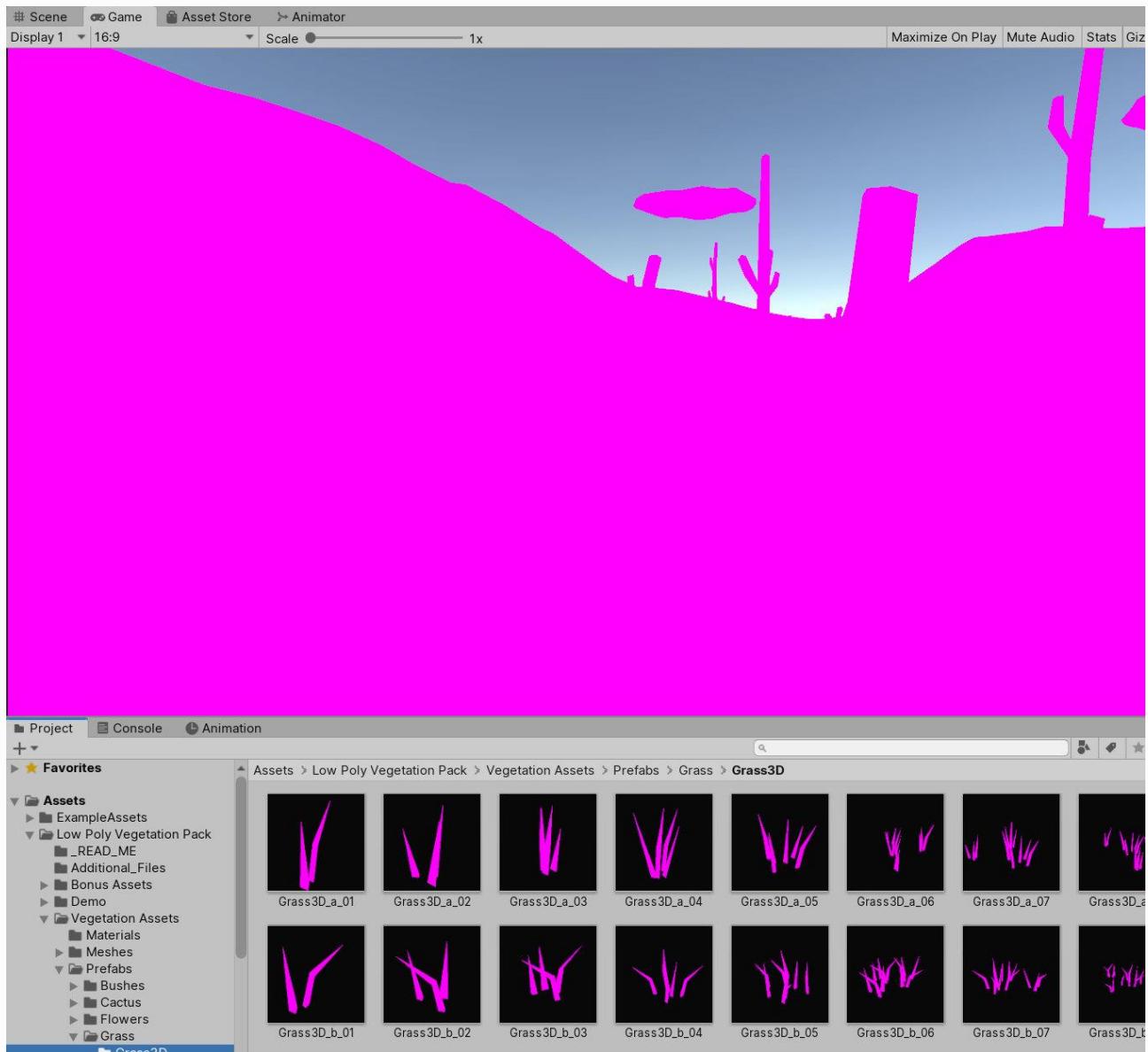


Old Demo_01 picture, now it looks slightly different!

Tested on Google Pixel 2 XL – runs at solid 60fps. Xperia Z Ultra drops to ~20fps for using Realtime shadows and Color Grading.

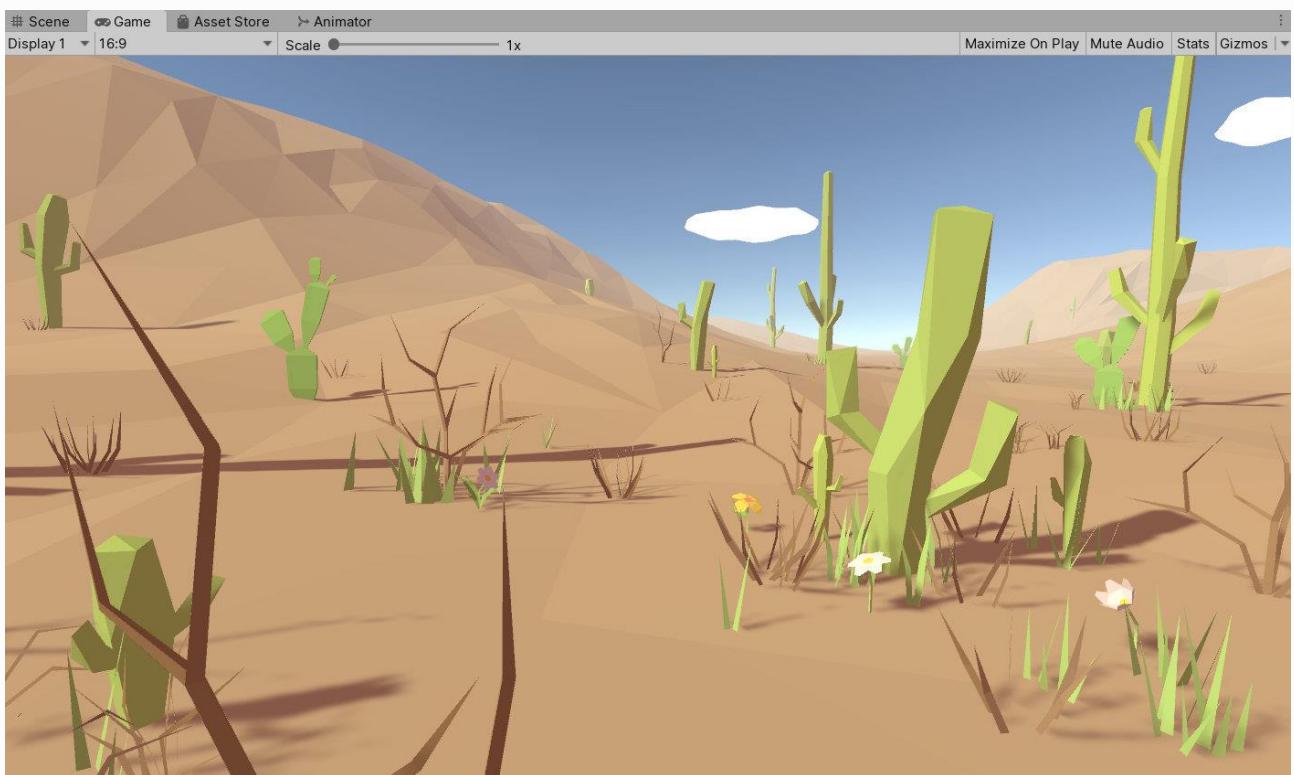
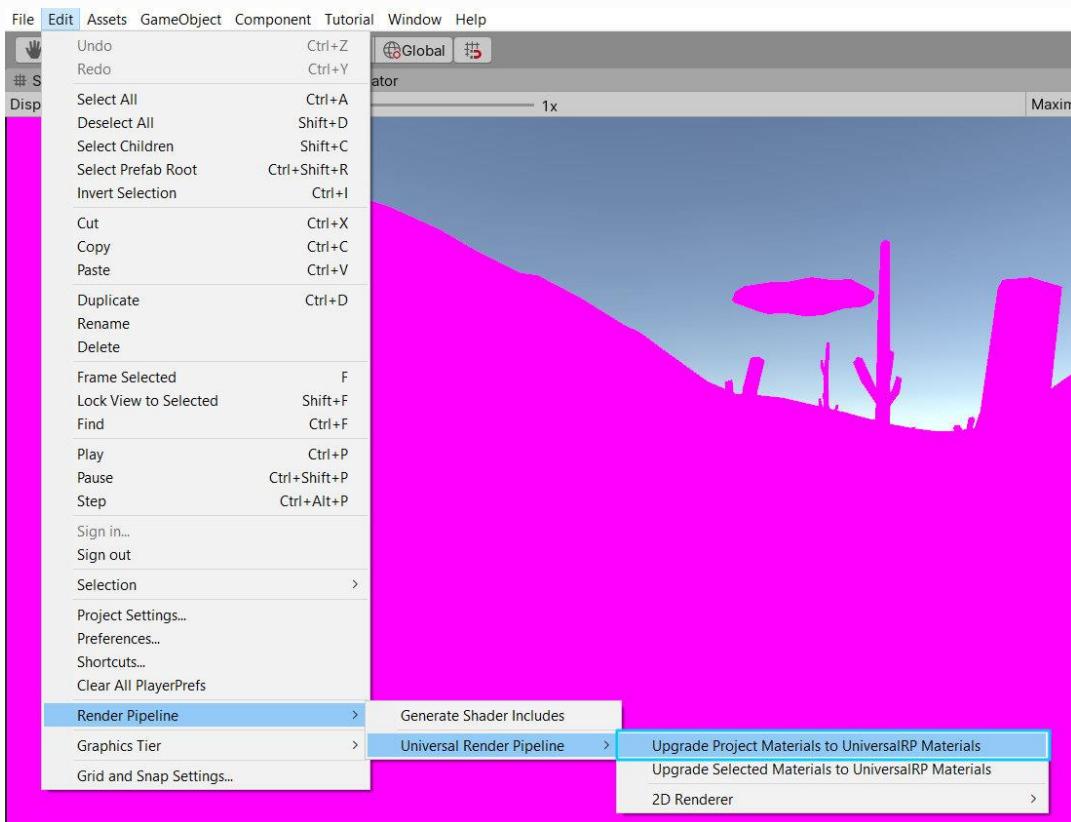
Unity 2019.3 and up - Universal Render Pipeline (URP)

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

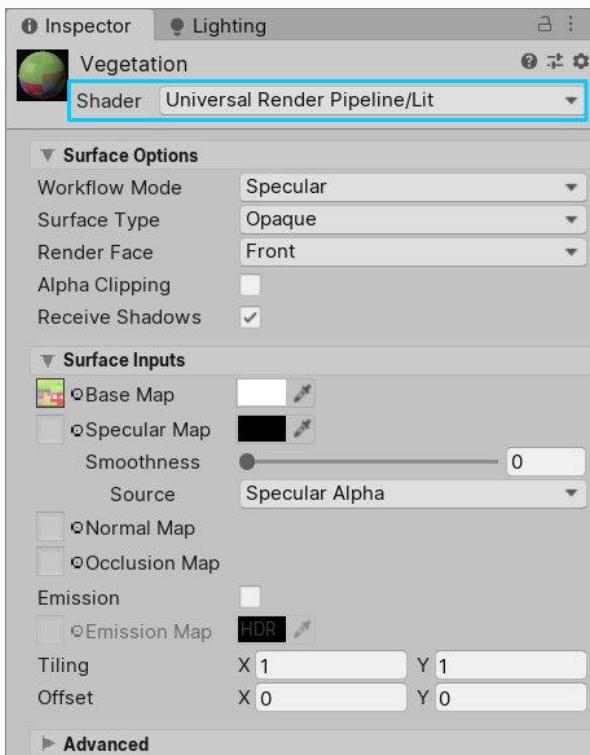


It's because all of **Low Poly Vegetation Pack** assets use material with a default **Standard Unity shader**. **URP** use different materials and shaders. So we need to change all materials from **Standard shader** to **Universal Render Pipeline/Lit shader**.

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



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



If your grass plane prefabs look like this (Demo_01 example):



Go to *Low Poly Vegetation Pack > Vegetation Assets > Materials*. Select all **GrassPlane** materials and enable **Alpha Clipping**.

The image consists of two screenshots from the Unity Editor.

Project Panel: Shows the file structure. Under the "Assets" folder, there is a "Low Poly Vegetation Pack" folder containing "Materials" (with subfolders "GrassPlane_a" and "GrassPlane_b"), "Textures", and "Prefabs".

Inspector Panel: Shows the "9 Materials" selected in the list. The "Shader" dropdown is set to "Universal Render Pipeline/Lit". In the "Surface Options" section, the "Alpha Clipping" checkbox is checked (highlighted with a red box) and the "Threshold" slider is set to 0.5 (also highlighted with a red circle).

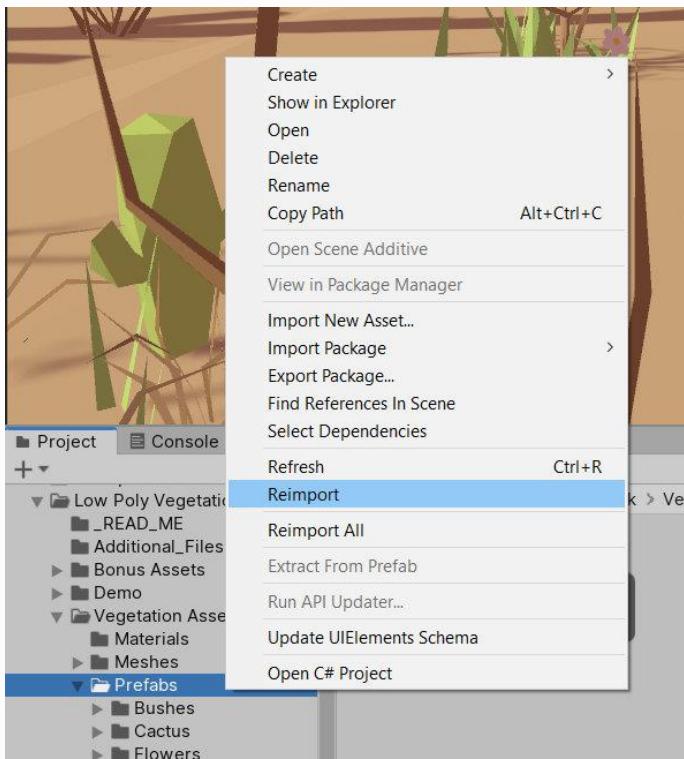
Now all of the prefabs should have correct materials with correct settings applied.



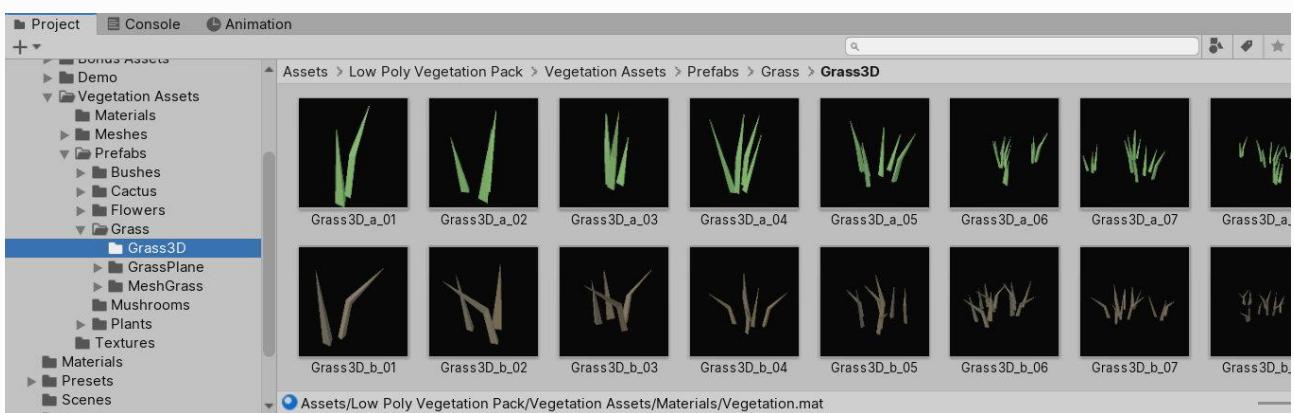
Also, if you go to *Low Poly Vegetation Pack > Vegetation Assets > Prefabs > Grass > Grass3D* - or inside any other prefab folder. You might see all of the prefabs in **Pink** color.



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

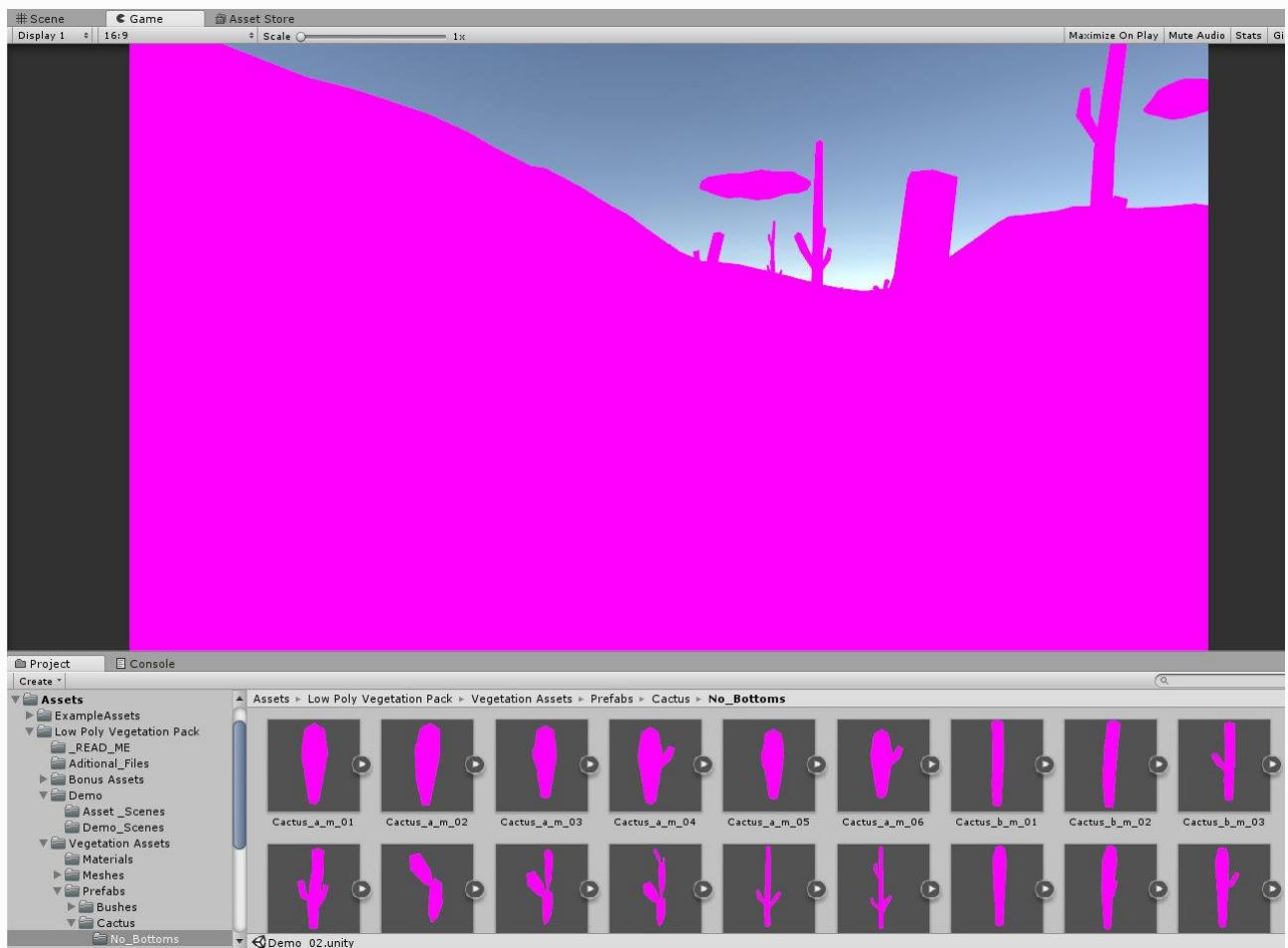


And it's fixed!



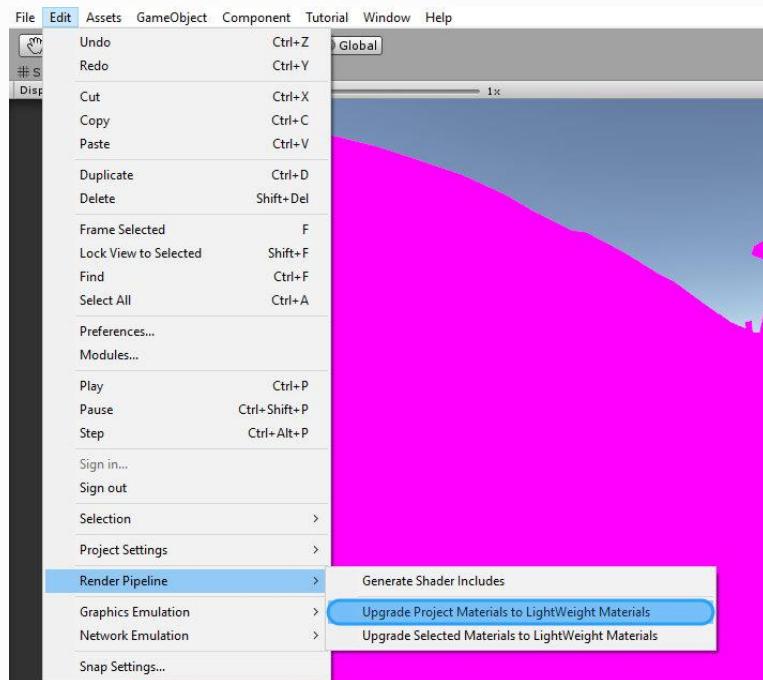
Unity 2018.4 LTS and up - Lightweight Render Pipeline (LWRP)

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

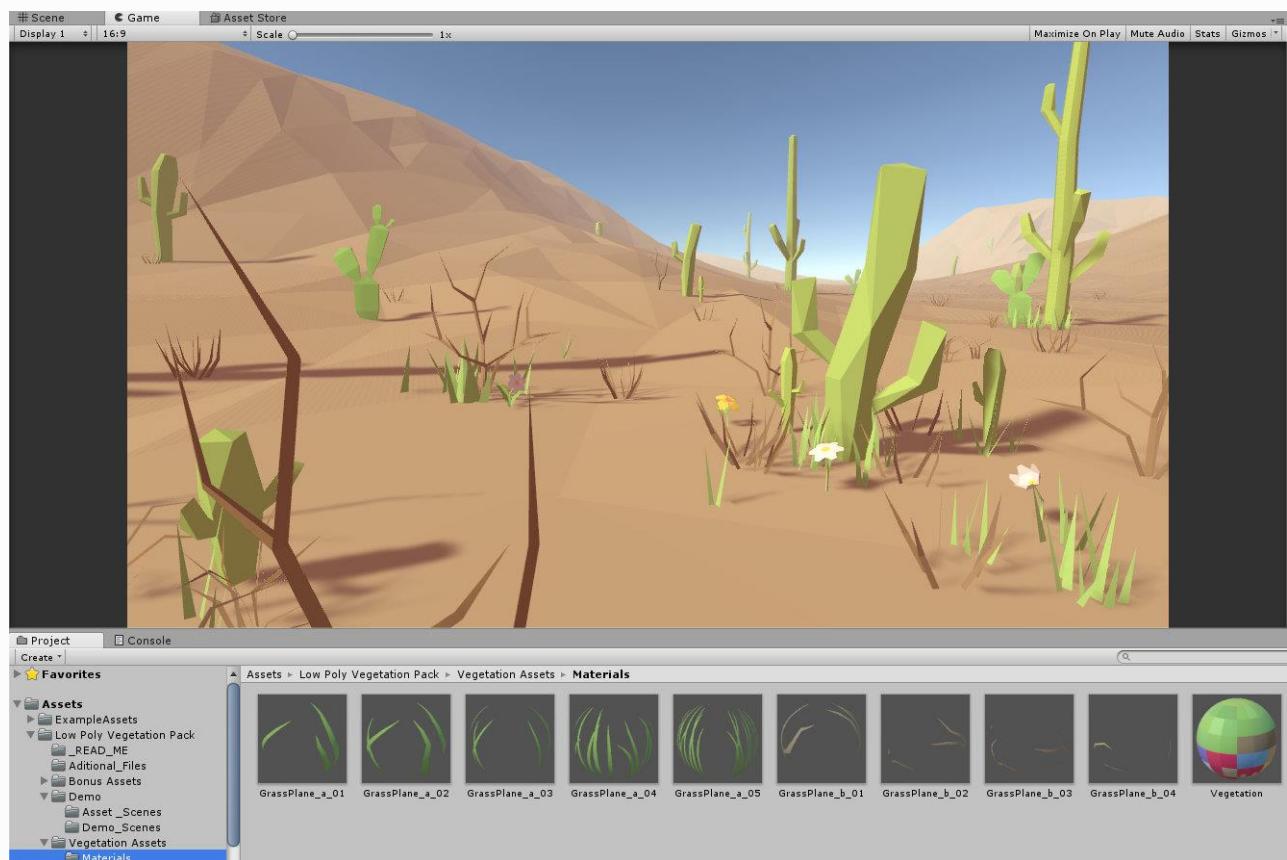


It's because all of **Low Poly Vegetation Pack** assets use materials with a default **Standard Unity shader**. LWRP use different materials and shaders. So we need to change all materials from **Standard shader** to **LightweightPipeline shader**.

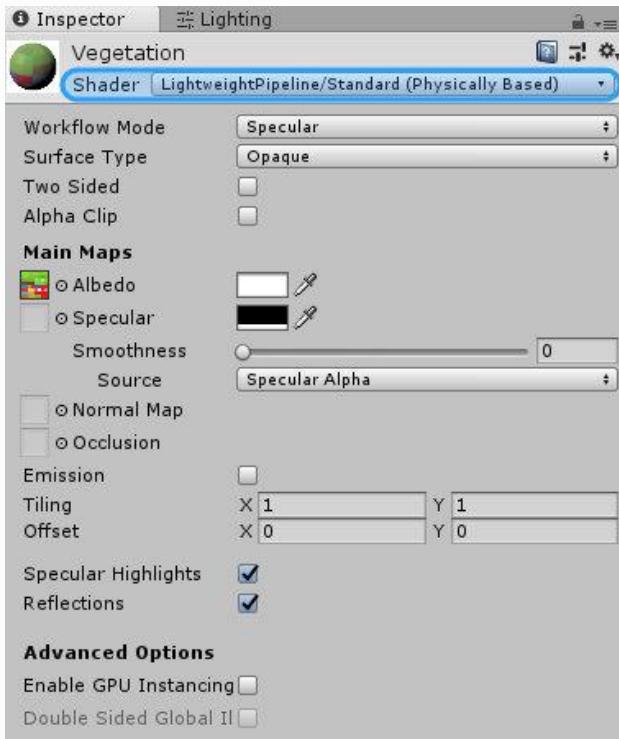
Go to *Edit > Render Pipeline > Upgrade Project Materials to LightWeight Materials*



And it's done!

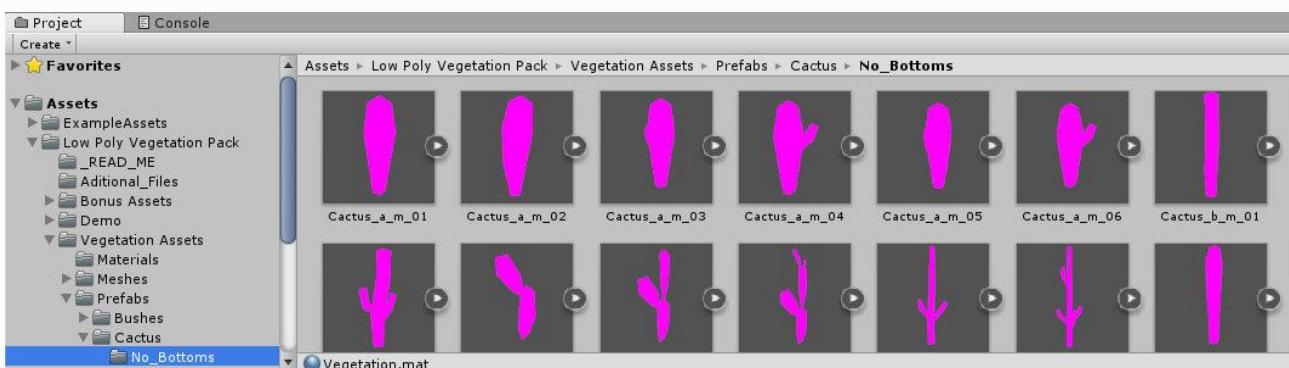


All project **Material** shaders were changed to **LightweightPipeline/Standard (Physically Based)**.

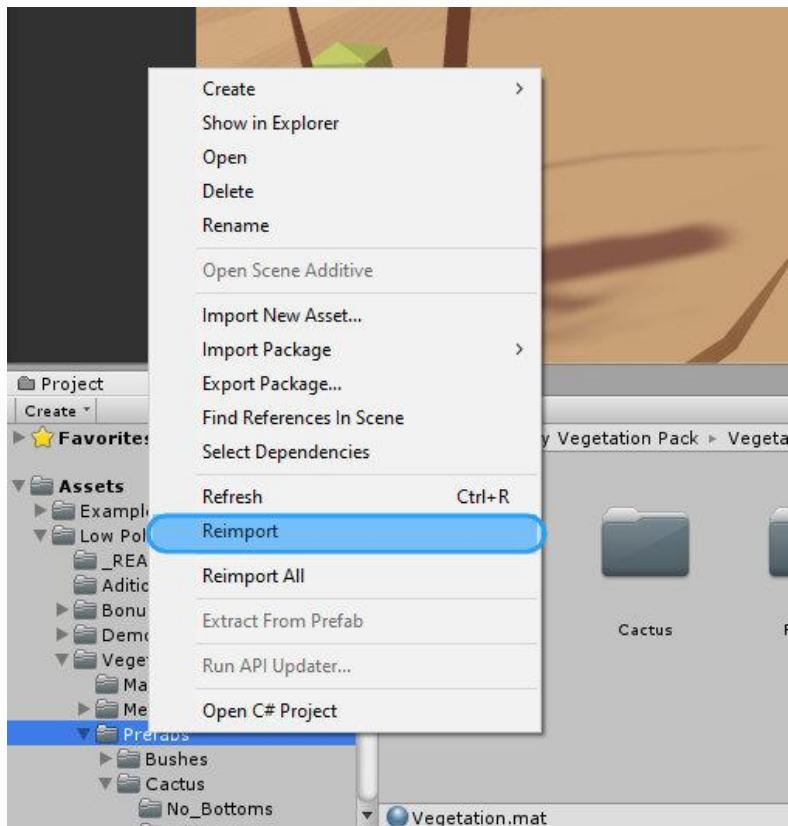


*You can do it manually by selecting **Material** and changing the **Shader** but it's much slower.

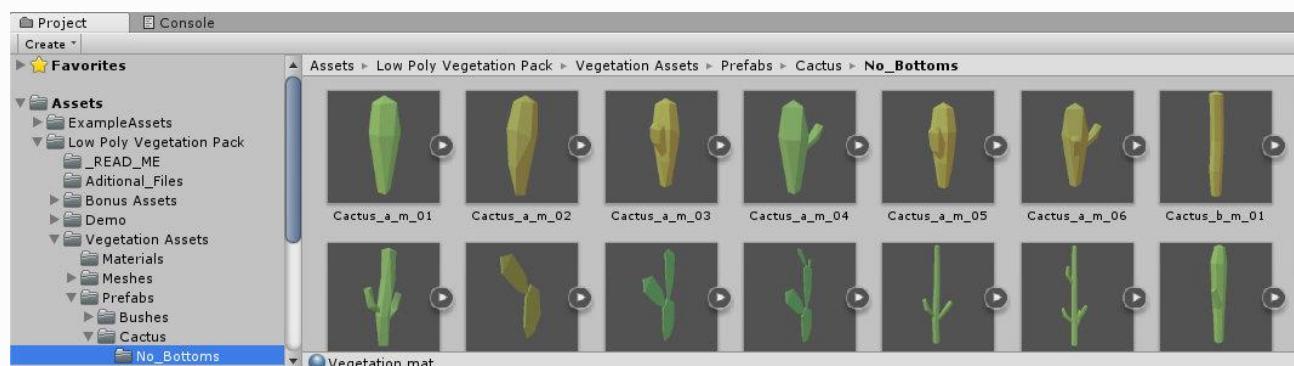
Now if you go to *Low Poly Vegetation Pack > Vegetation Assets > Prefabs > Cactus > No_Bottoms* - or inside any other vegetation folder. You might see all of the prefabs in **Pink** color.



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

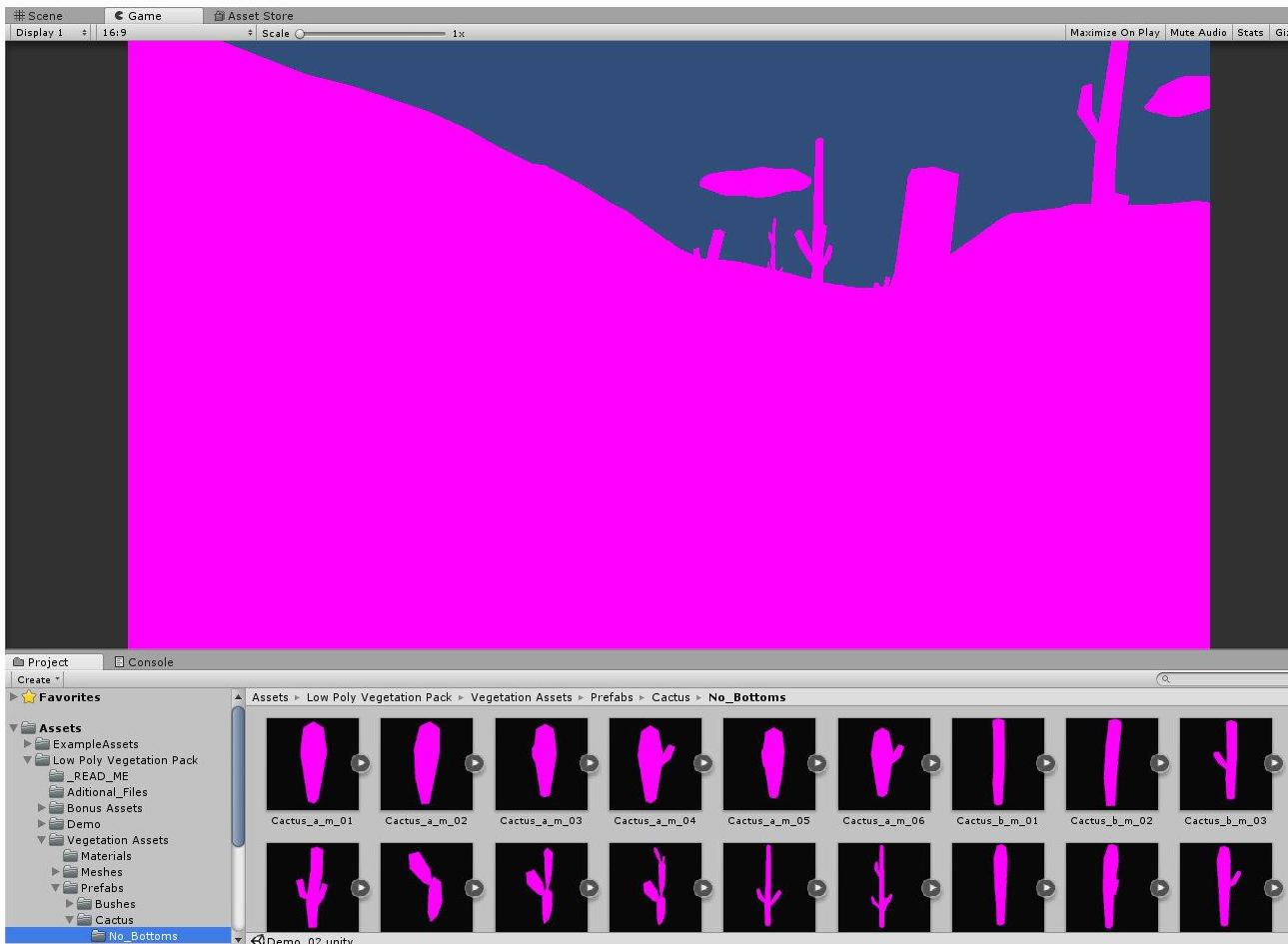


And it's fixed!



Unity 2018.4 LTS and up - High Definition Render Pipeline (HDRP)

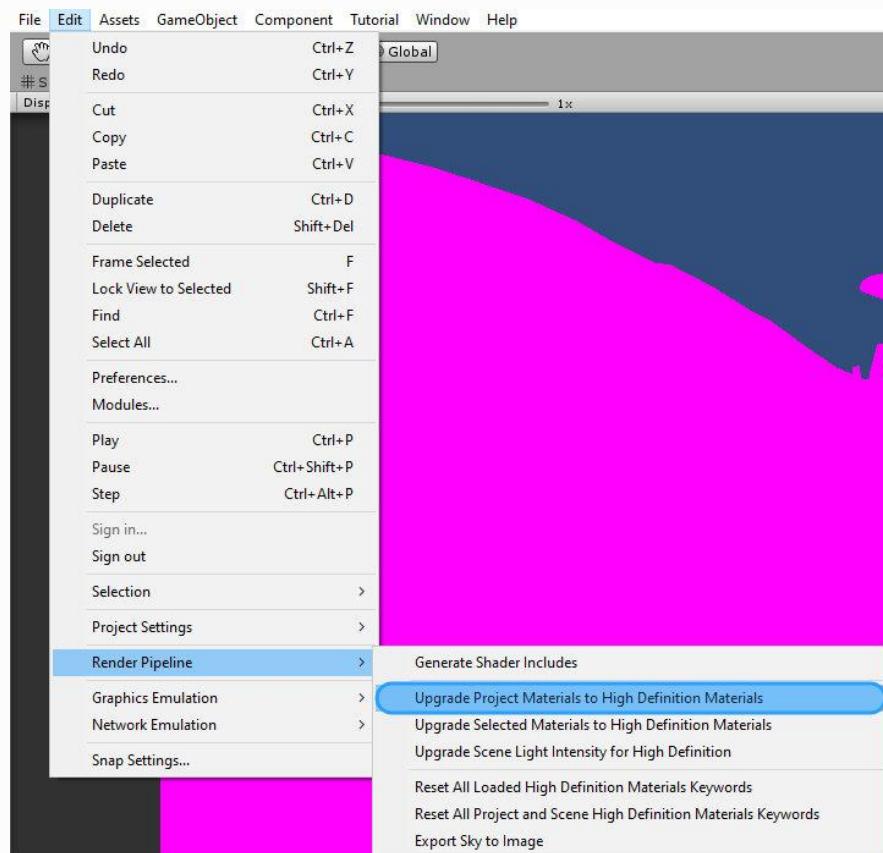
You might encounter pink textures after importing **Low Poly Vegetation Pack** to your Unity project, which is using **High Definition Render Pipeline (HDRP)**.



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

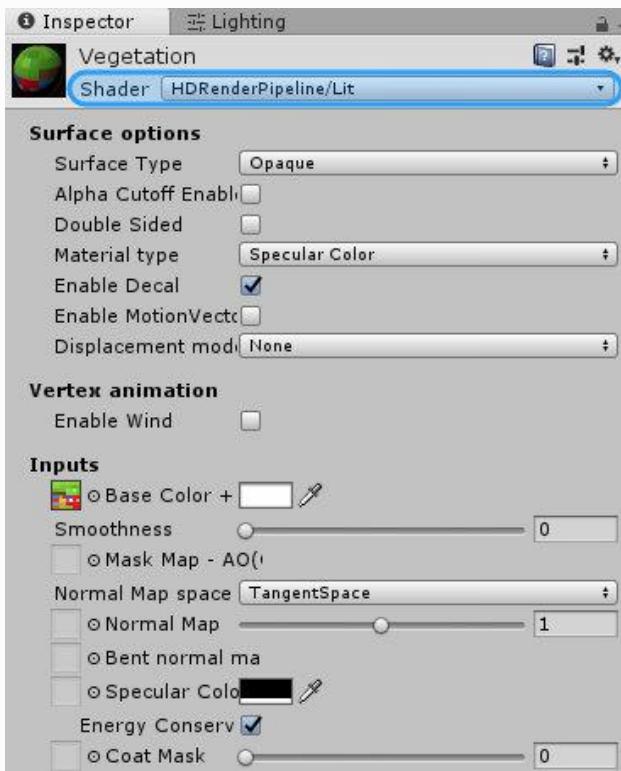
1. Fix Purple Materials

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

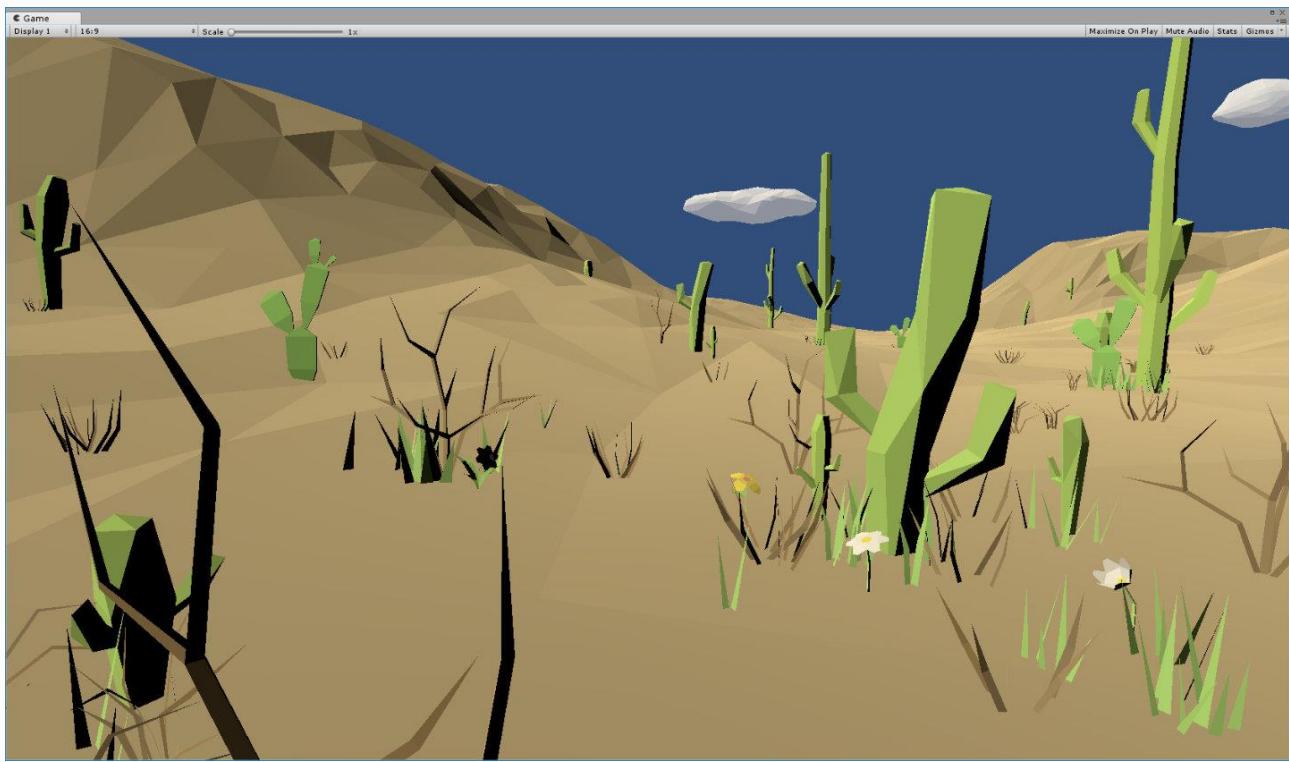


And it's done! Almost.

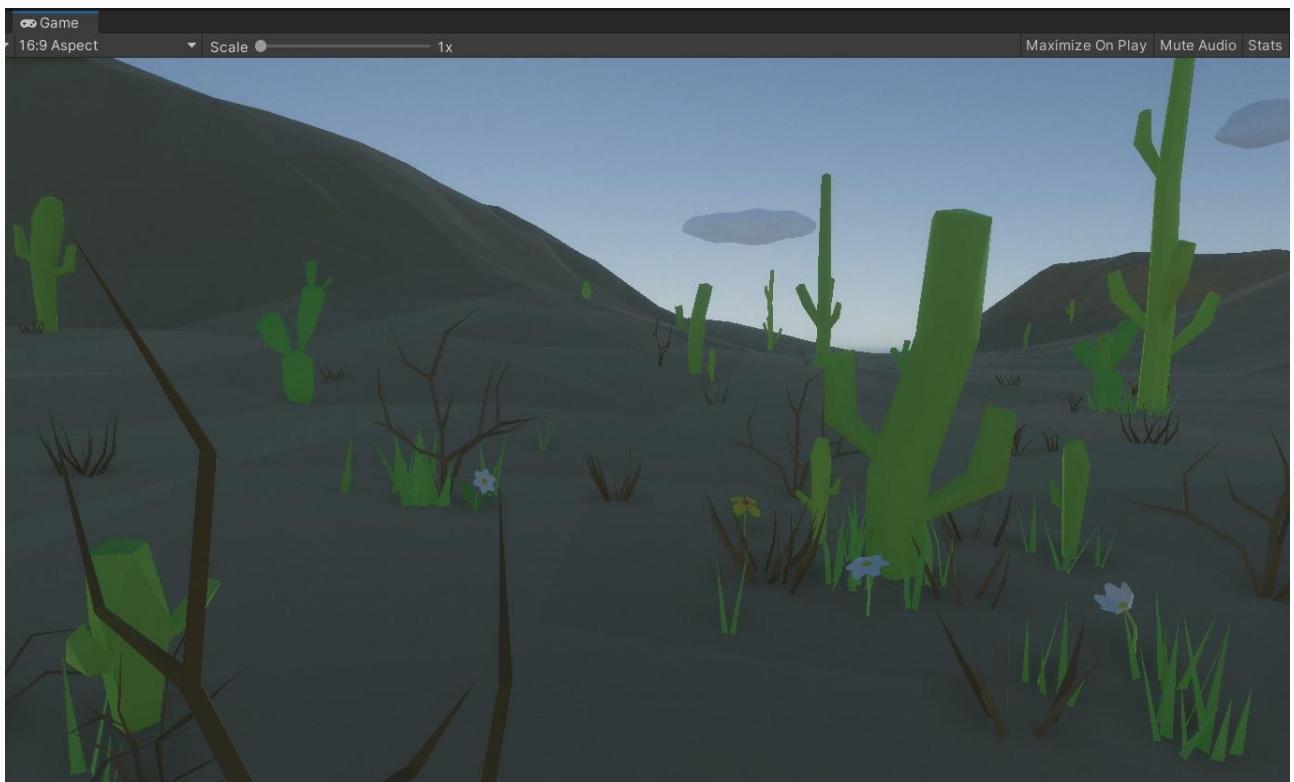
All project **Material** shaders were changed to **HDRP/Lit**.



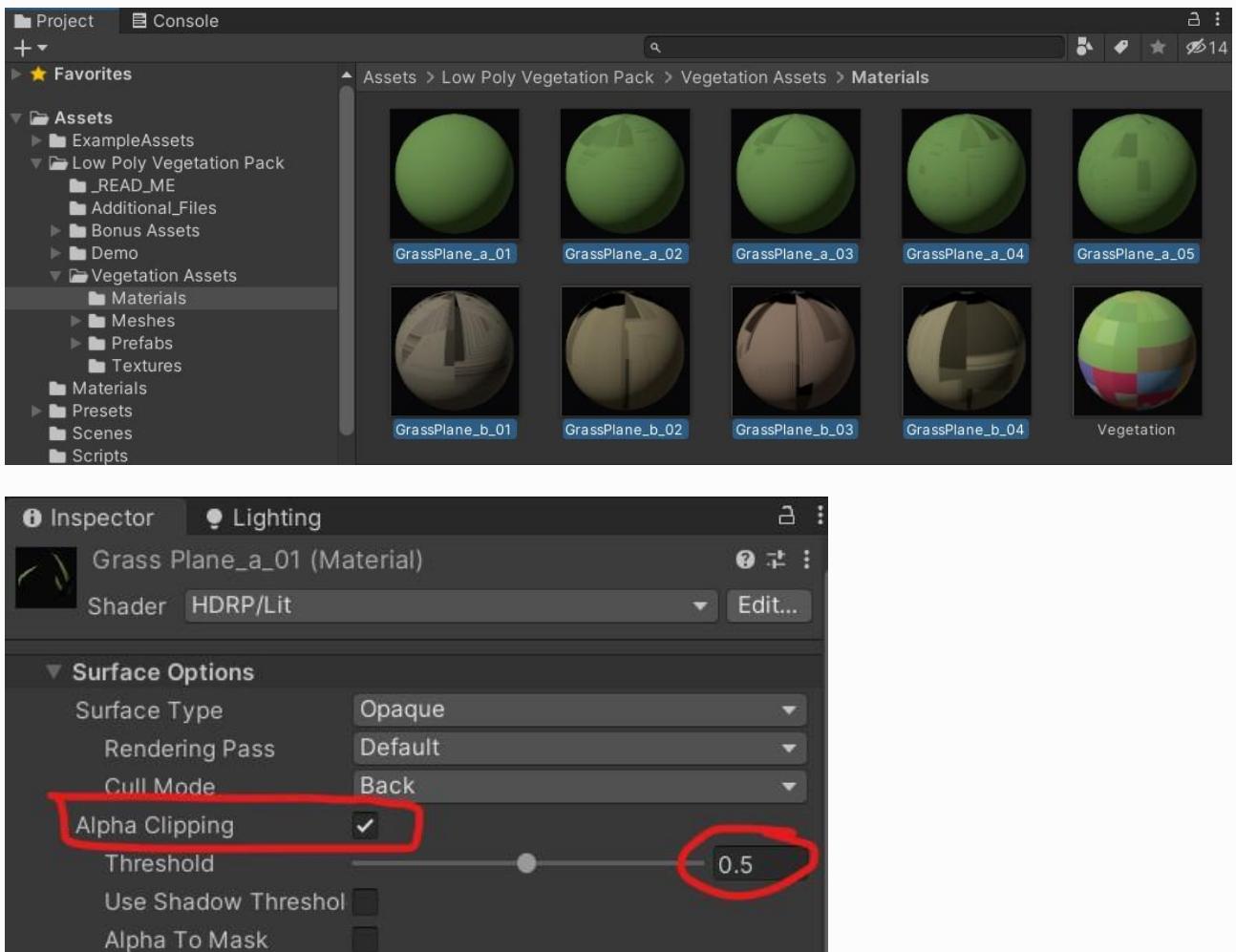
Your scene should look something like this:



Or like this in a newer version of Unity:

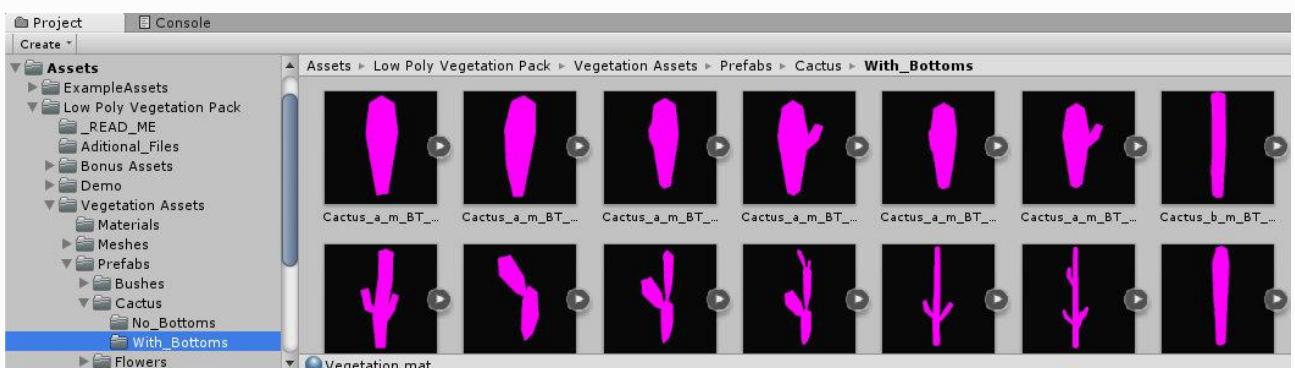


Go to *Low Poly Vegetation Pack > Vegetation Assets > Materials*. Select all **GrassPlane** materials and make sure that **Alpha Clipping** is enabled!

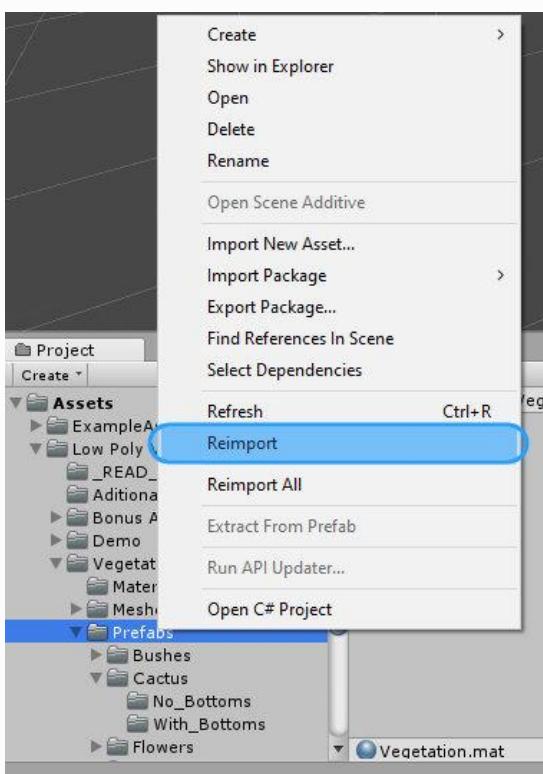


Now all of the prefabs should have correct materials with correct settings applied.

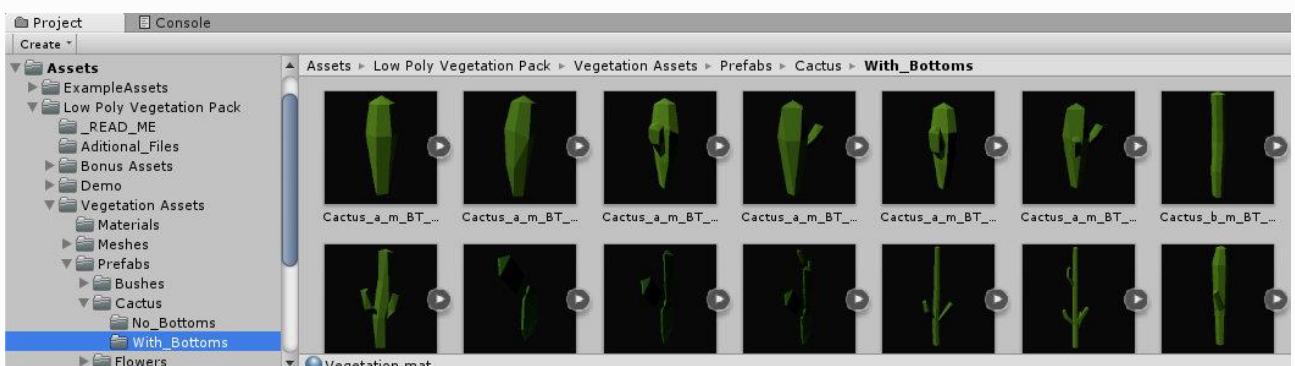
Now if you go to *Low Poly Vegetation Pack > Vegetation Assets > Prefabs > Cactus > With_Bottoms* - or inside any other vegetation folder. You might see all of the prefabs in **Pink** color.



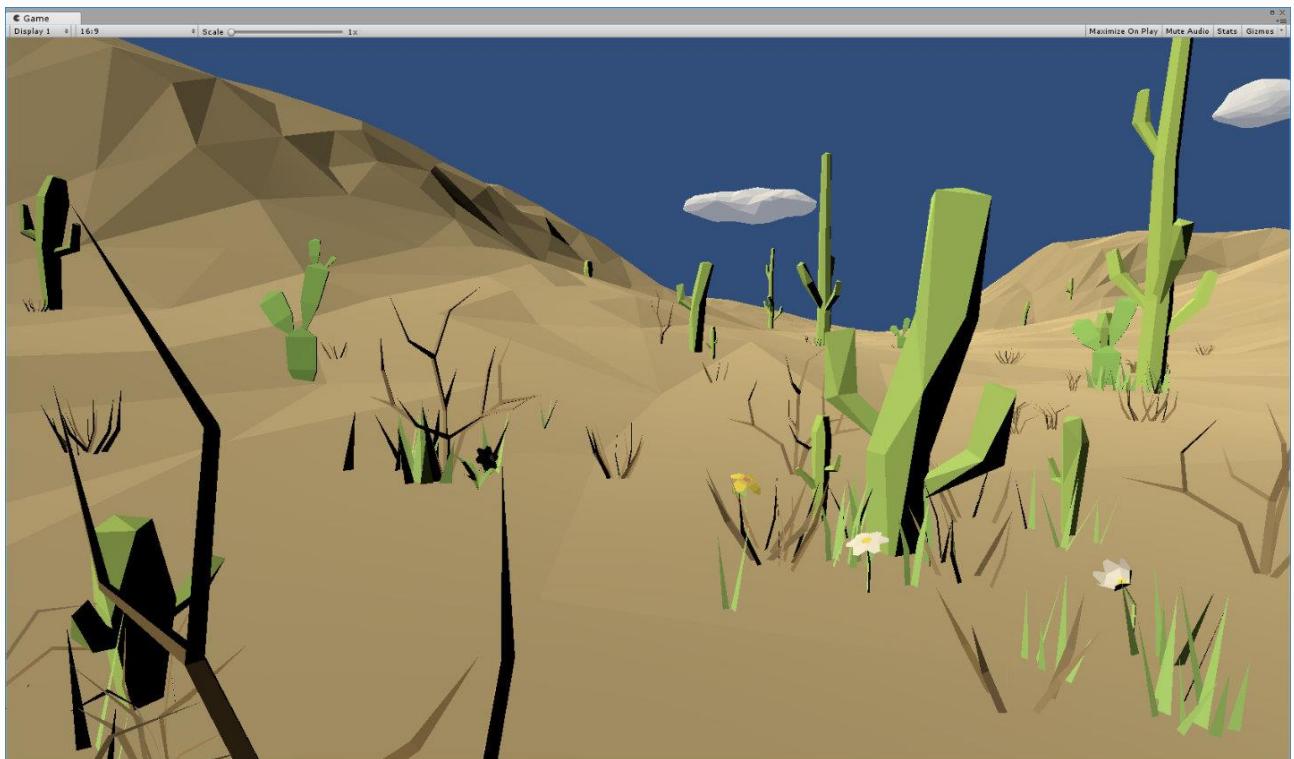
To fix that - press **Right Mouse Button** on **Prefabs** folder and select **Reimport**.



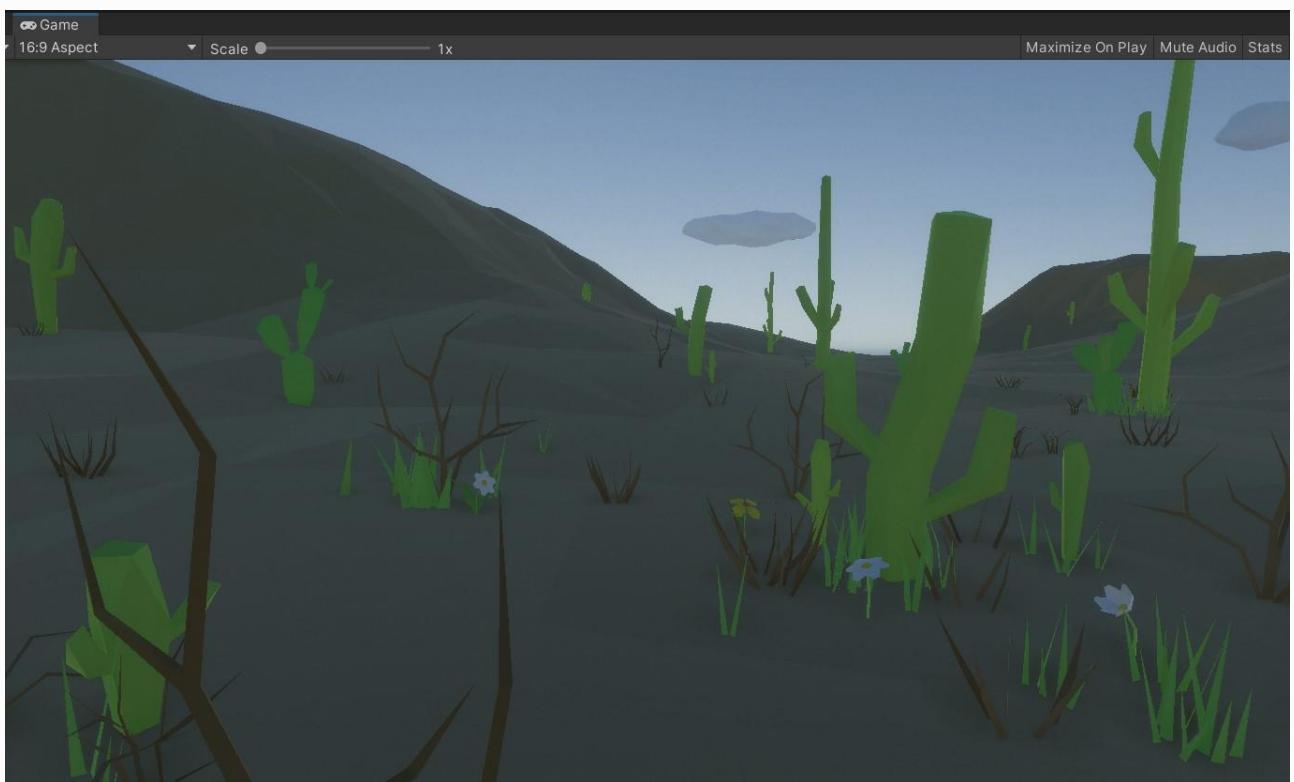
And it's fixed!



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



Newer Unity version example:



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.

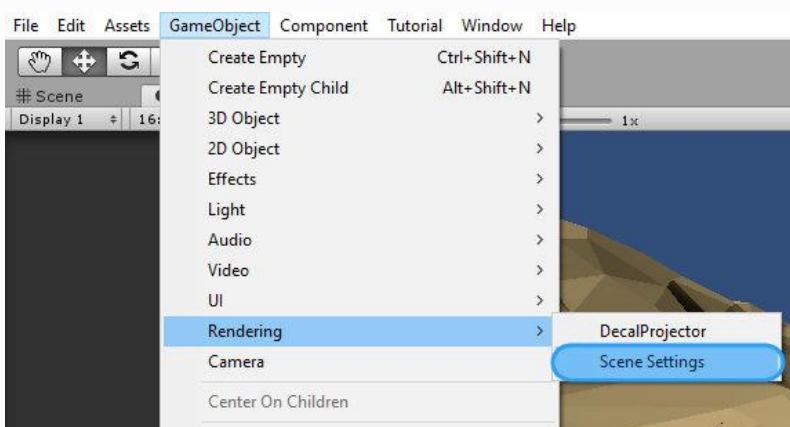
Newer Unity version example:



Next steps are for Unity 2018.4 LTS+ (can be different on newer versions of Unity)!

3. Fix the **Skybox**

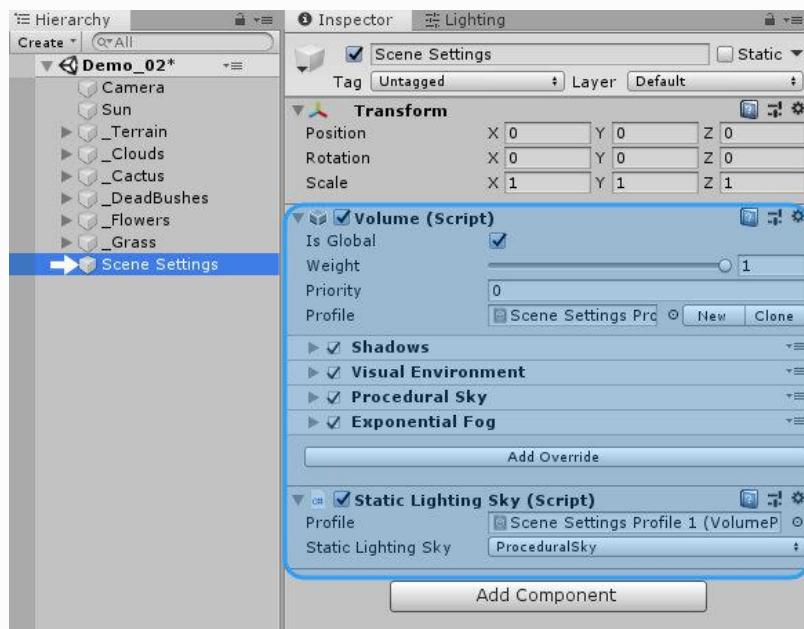
Using HDRP, you need to use **Scene Settings** - to change the **Skybox** and other scene settings. Go to *GameObject > Rendering > Scene Settings*



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)

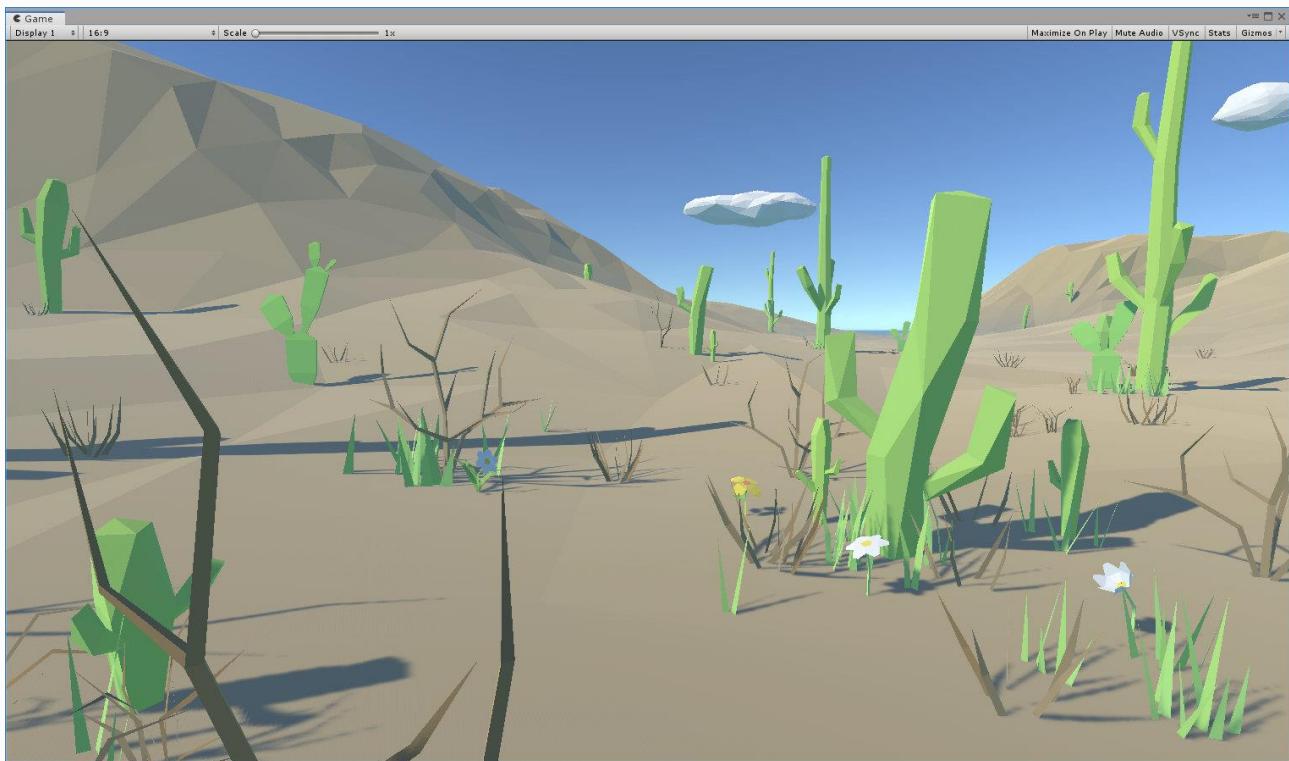
Use my **Procedural Skybox** settings:



Sky Tint (Color code): 68A4C3

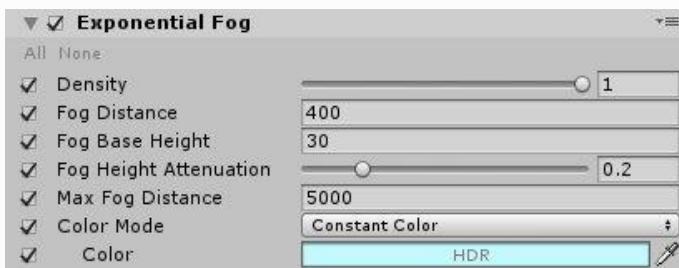
Ground Color (Color code): 96B2C0

to achieve this:



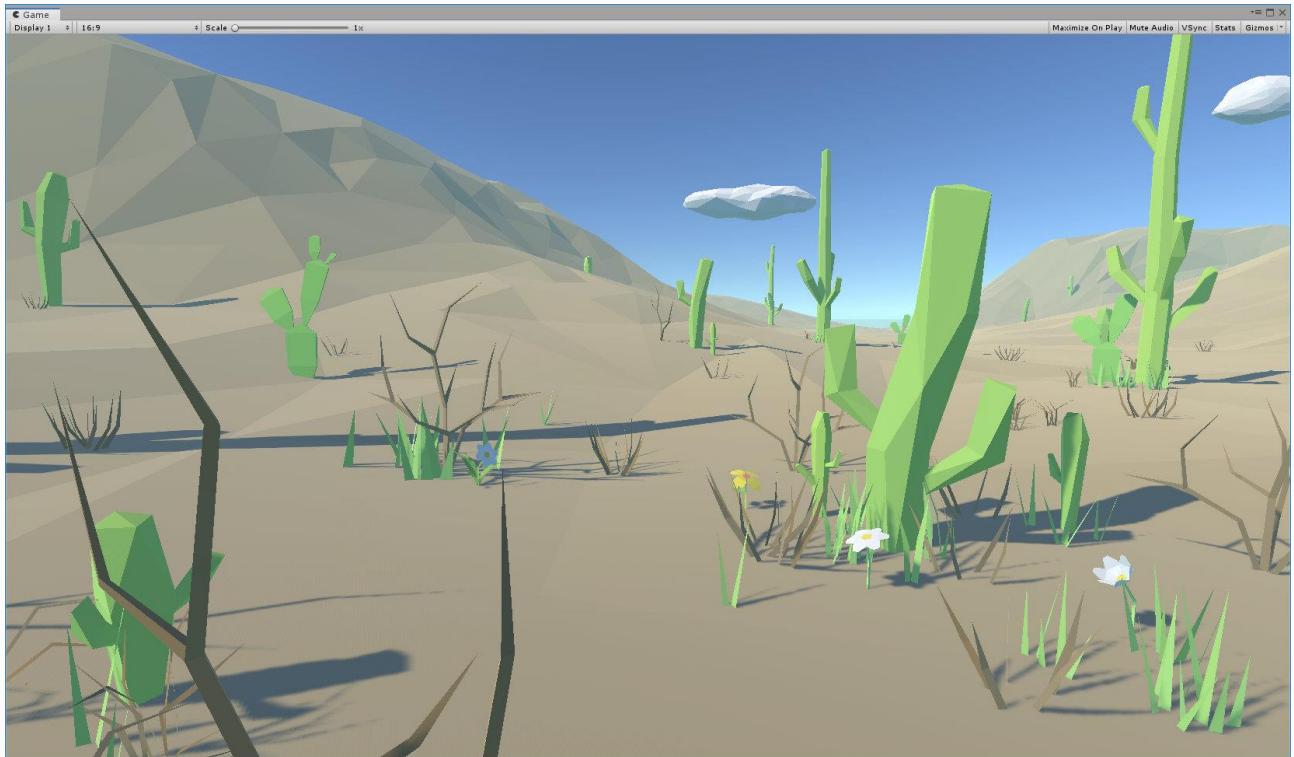
5. Edit the Exponential Fog

Use my **Exponential Fog** settings:



Set **Color Mode** to **Constant Color** and use this **Color** (R: 141; G:244; B:255)

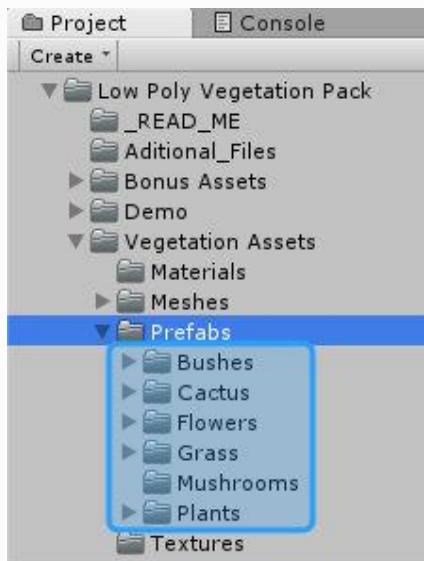
Final result



How to use “Low Poly Vegetation Pack”

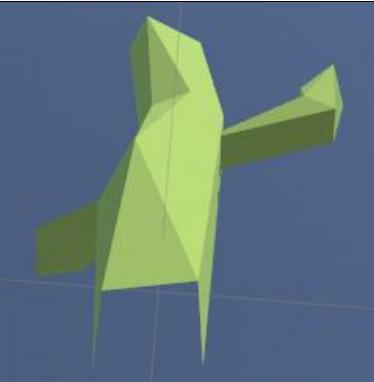
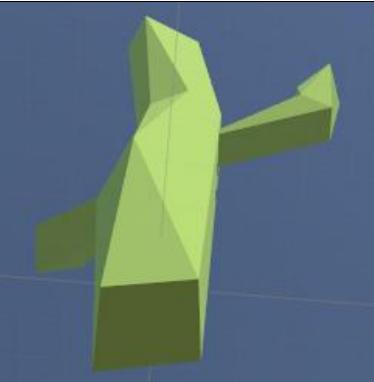
Go to *Assets > Low Poly Vegetation Pack > Vegetation Assets > Prefabs*

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



For example, open folder **Cactus**. You will see that you can choose between **2** types of Cactus:



<p>No_Bottoms – Cactus meshes don't have faces at the bottom.</p> 	<p>With_Bottoms - Cactus meshes have faces at the bottom.</p> 
--	---

For example, open folder **No_Bottoms**, select and drag **Prefab** to your scene. That's it.

Same for **Bonus Assets**.

Go to *Assets > Low Poly Vegetation 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.



I recommend using **Polybrush** for painting the grass or plants on any mesh terrain! Check out my tutorial on [How To use Polybrush in Unity!](#)

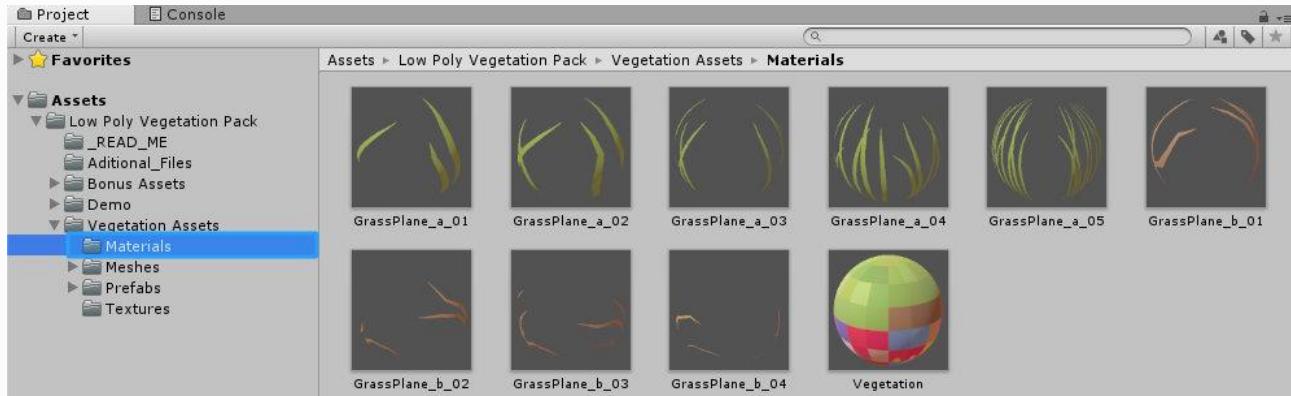
Some Prefabs like **Flowers**, **Plants**, and **Grass** comes in **2** types:

OneSided – Mesh can be seen from one side (Good for games from a top-down view).

TwoSided - Mesh can be seen from both sides (Good for any game where you can see asset from the bottom).

How to Change Vegetation Prefabs Color / Texture

Go to *Low Poly Vegetation Pack > Vegetation Assets > Materials* - here, you will find 10 materials.

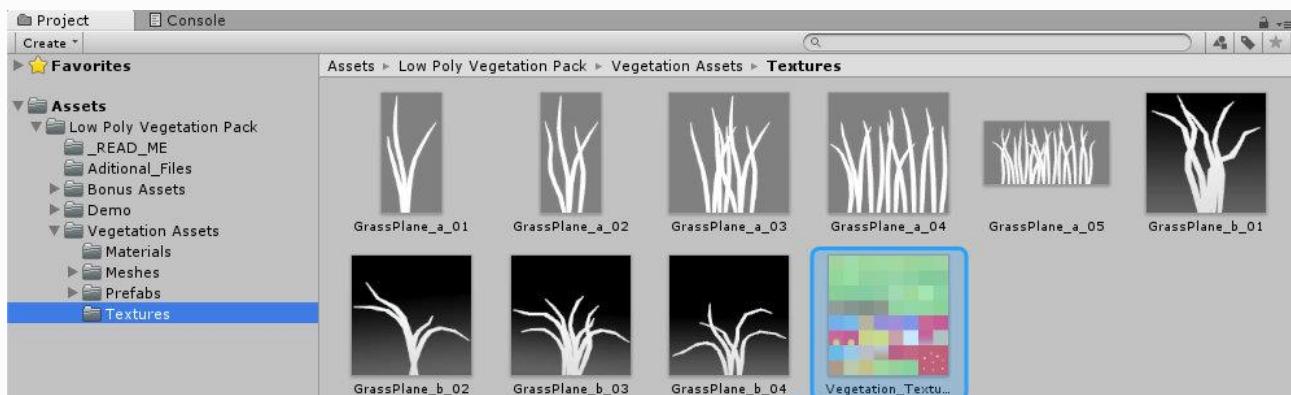


- Material **Vegetation** is used for all **Vegetation Asset**: (Bushes, Cactuses, Flowers, Grass3D, MeshGrass, Mushrooms, and Plants).
- **GrassPlane** Prefabs use Other **9 Materials**: (GrassPlane_a_01, GrassPlane_a_02..., GrassPlane_b_01, GrassPlane_b_02..., etc).

Change Vegetation Prefab Color

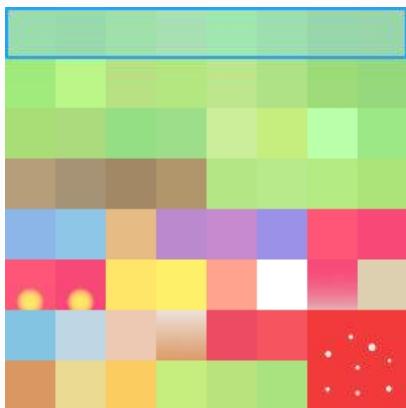
Vegetation Material use **1 Texture Atlas**. So, we need to change colors for that texture to change Vegetation Prefab colors.

Go to *Low Poly Vegetation Pack > Vegetation Assets > Textures*

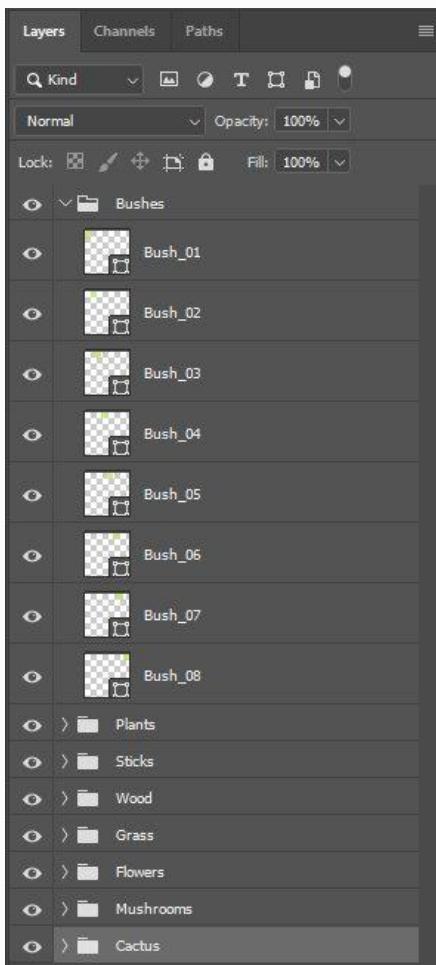


And open **Vegetation_Texture_Atlas.png** inside Photoshop, Gimp, Affinity or any other image editing software. Every color square used for one random Vegetation asset.

For example, the first line of squares is used for **Bushes**.



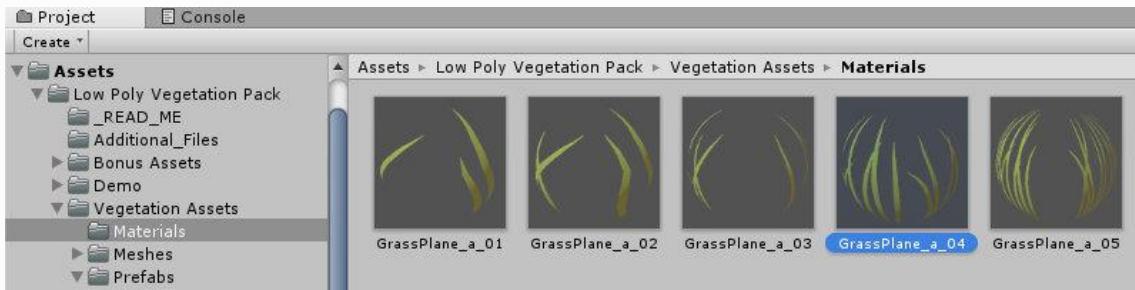
I also included **.psd** file of this texture inside *Low Poly Vegetation Pack > Additional_Files* folder. Extract **Vegetation_Texture_Atlas_PSD.rar** file and open **.psd** inside **Photoshop**, **Gimp**, **Affinity Designer**, etc. This way, you can see which colors are for which Vegetation assets by looking into **Layer Names** and edit those colors easily.



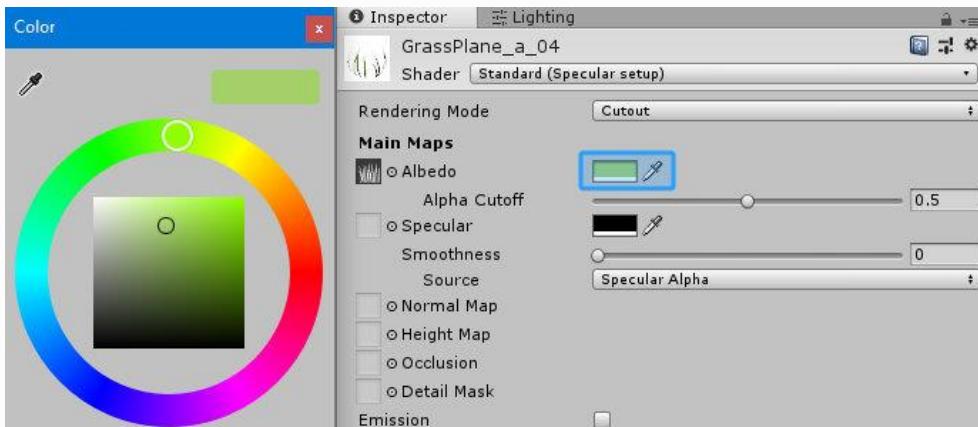
Change GrassPlane Prefab Color



Go to *Low Poly Vegetation Pack > Vegetation Assets > Materials* - Select any **GrassPlane** Material.

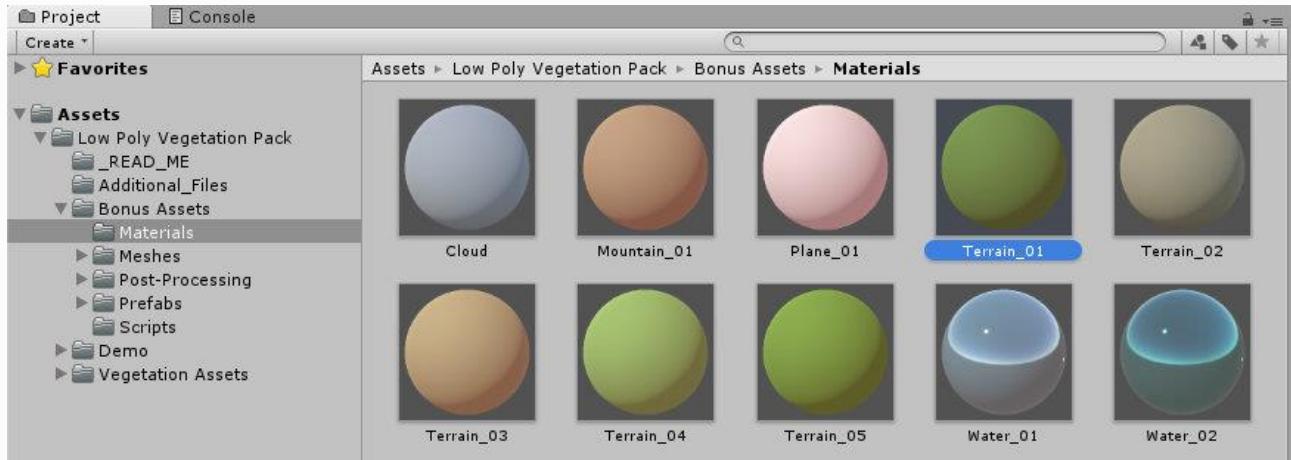


In the **Inspector**, change **Albedo Color** to any color you want to paint the grass.

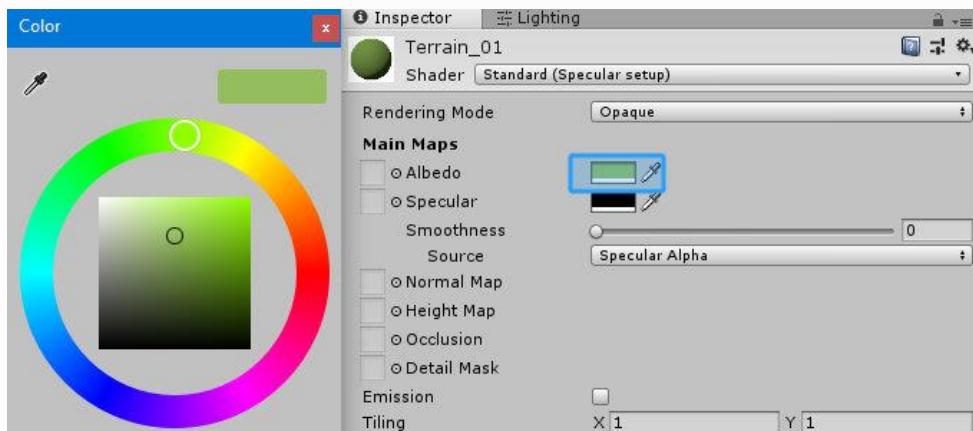


Change Bonus Assets Color

To change colors for **Bonus Assets** (Cloud, Hills, Terrain, and Water), go to *Low Poly Vegetation Pack > Bonus Assets > Materials*.

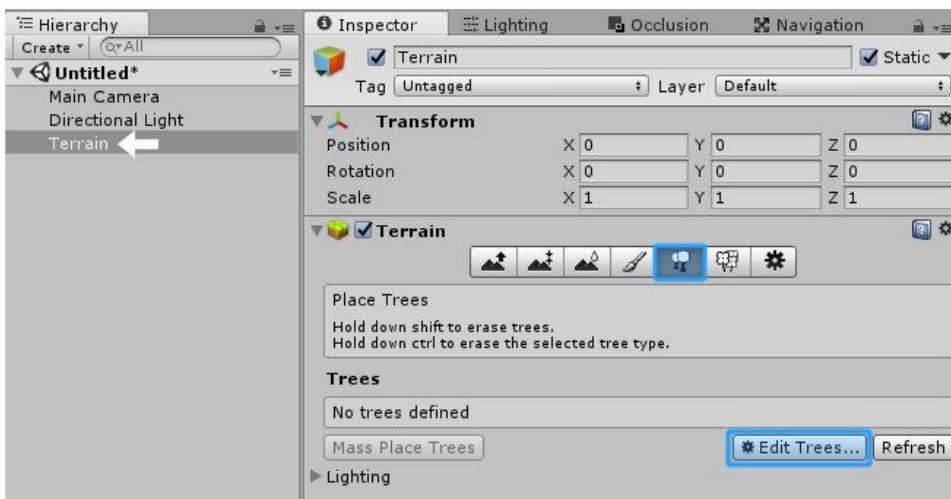


Select the **Material** you want to edit and change **Albedo Color** in the **Inspector**.

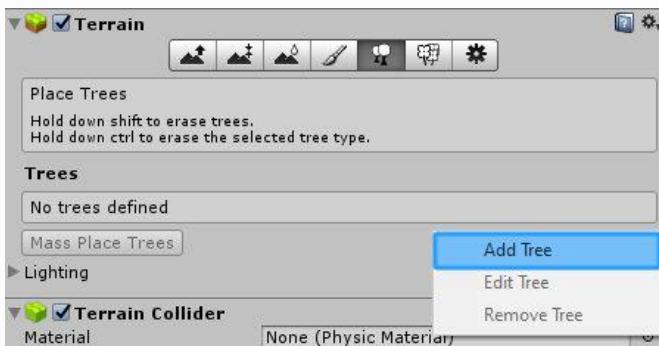


How to Paint Vegetation Prefabs on Unity Terrain

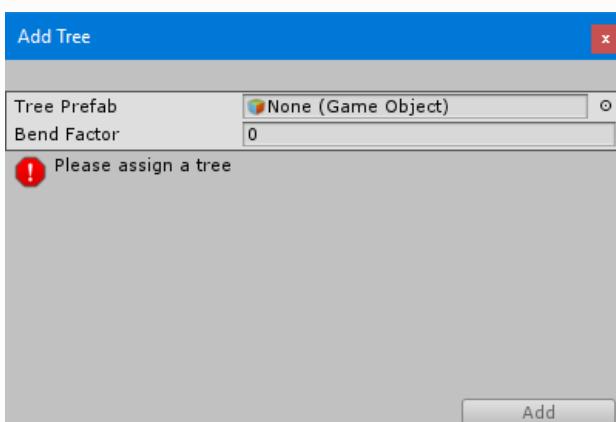
Select your **Unity Terrain** and go to **Place Trees** tab. Click on **Edit**



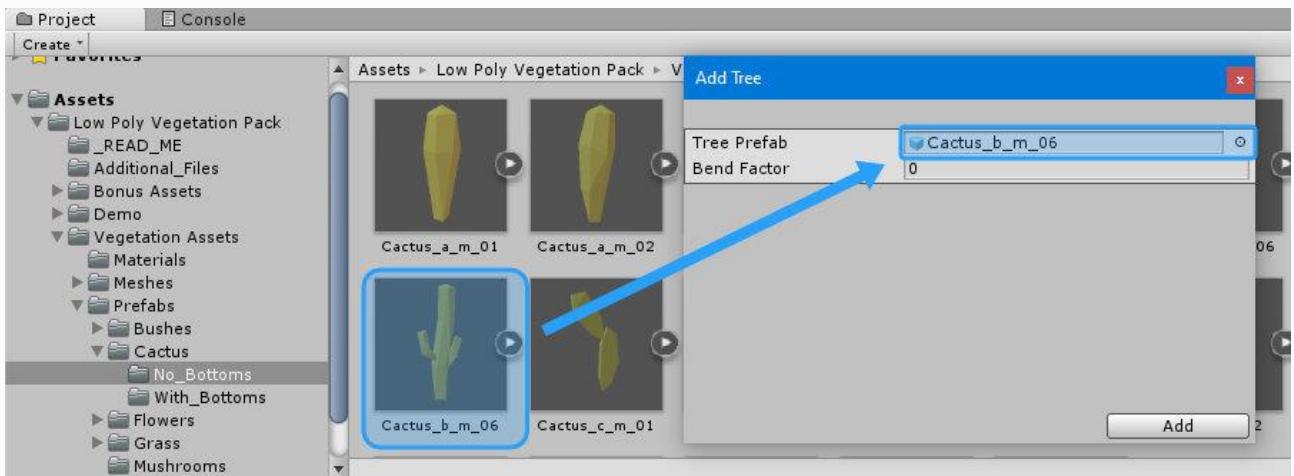
...and press on **Add Tree**



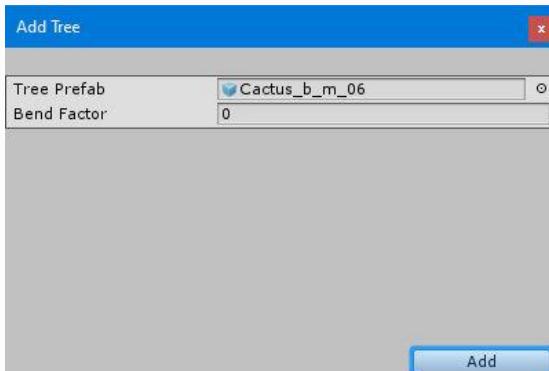
You should see a popup window **Add Tree**



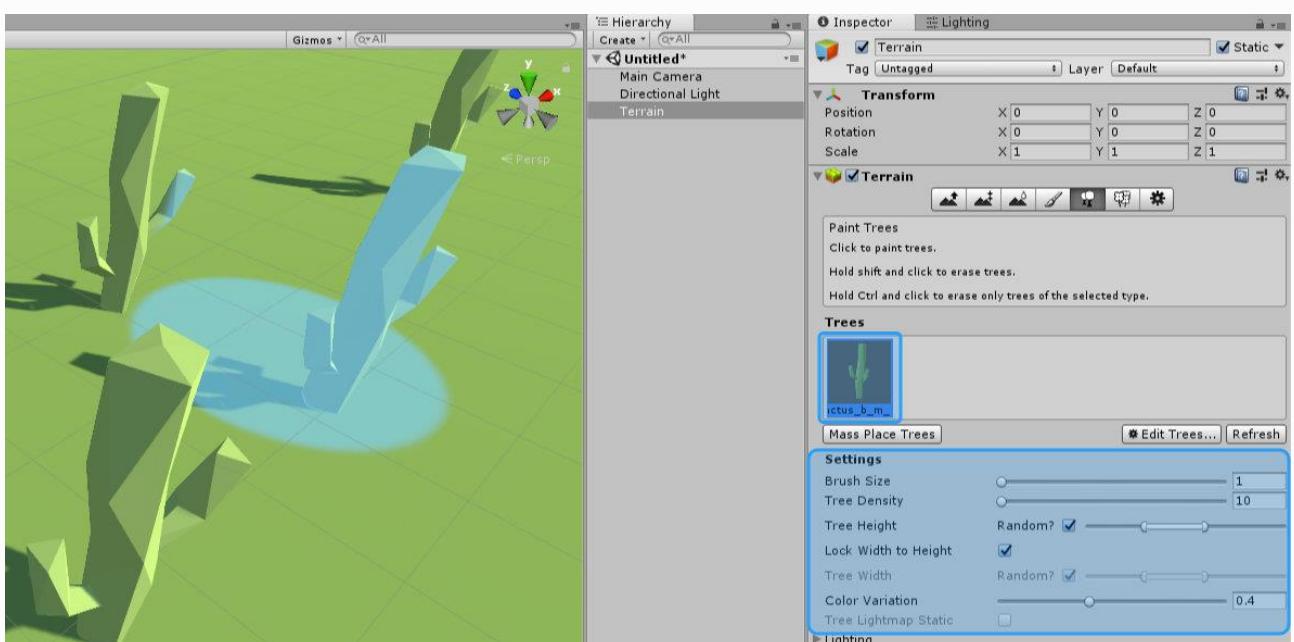
Go to *Low Poly Vegetation Pack > Vegetation Assets > Prefabs* - select any Vegetation type you want to use (I used **Cactus**). Drag and drop Prefab to **Tree Prefab** tab:



Press **Add**



That's it! Select **Vegetation Prefab**, change **Settings**, and paint on the **Terrain**.



How to Paint Vegetation Prefabs on Mesh Terrain Using Polybrush

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

UPDATE! Watch my Video Tutorial on [How to use Polybrush!](#)

1. Import Polybrush.

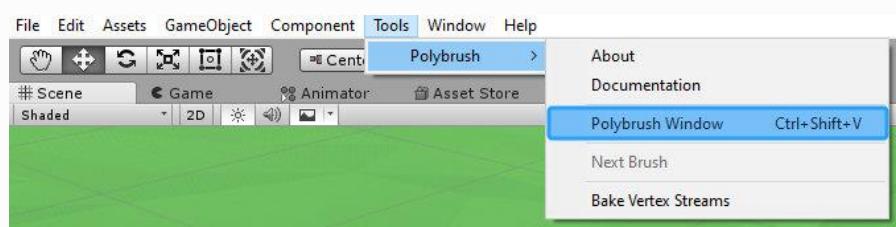
Go to *Window > Asset Store* and search for **Polybrush**:



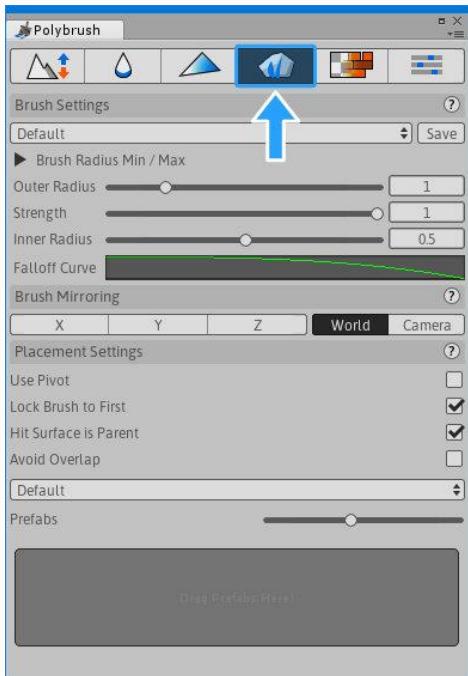
Import it to your project.

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

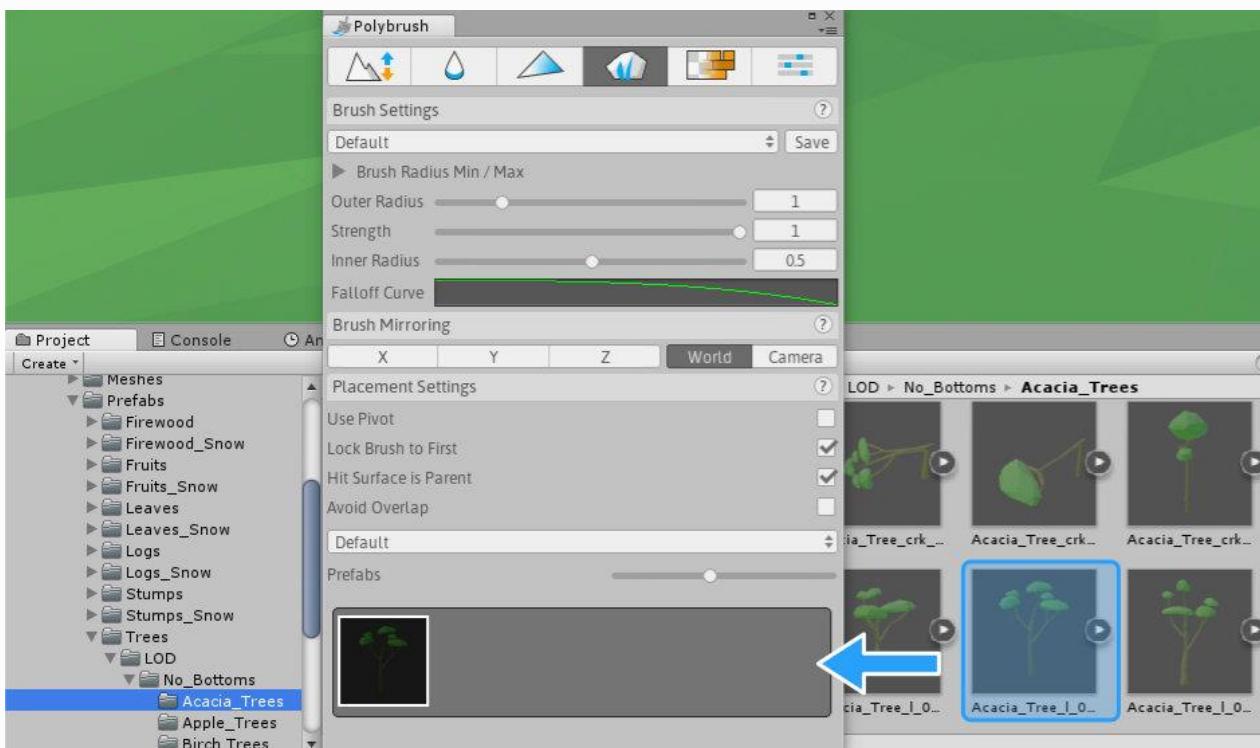
Go to *Tools > Polybrush > Polybrush Window*



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

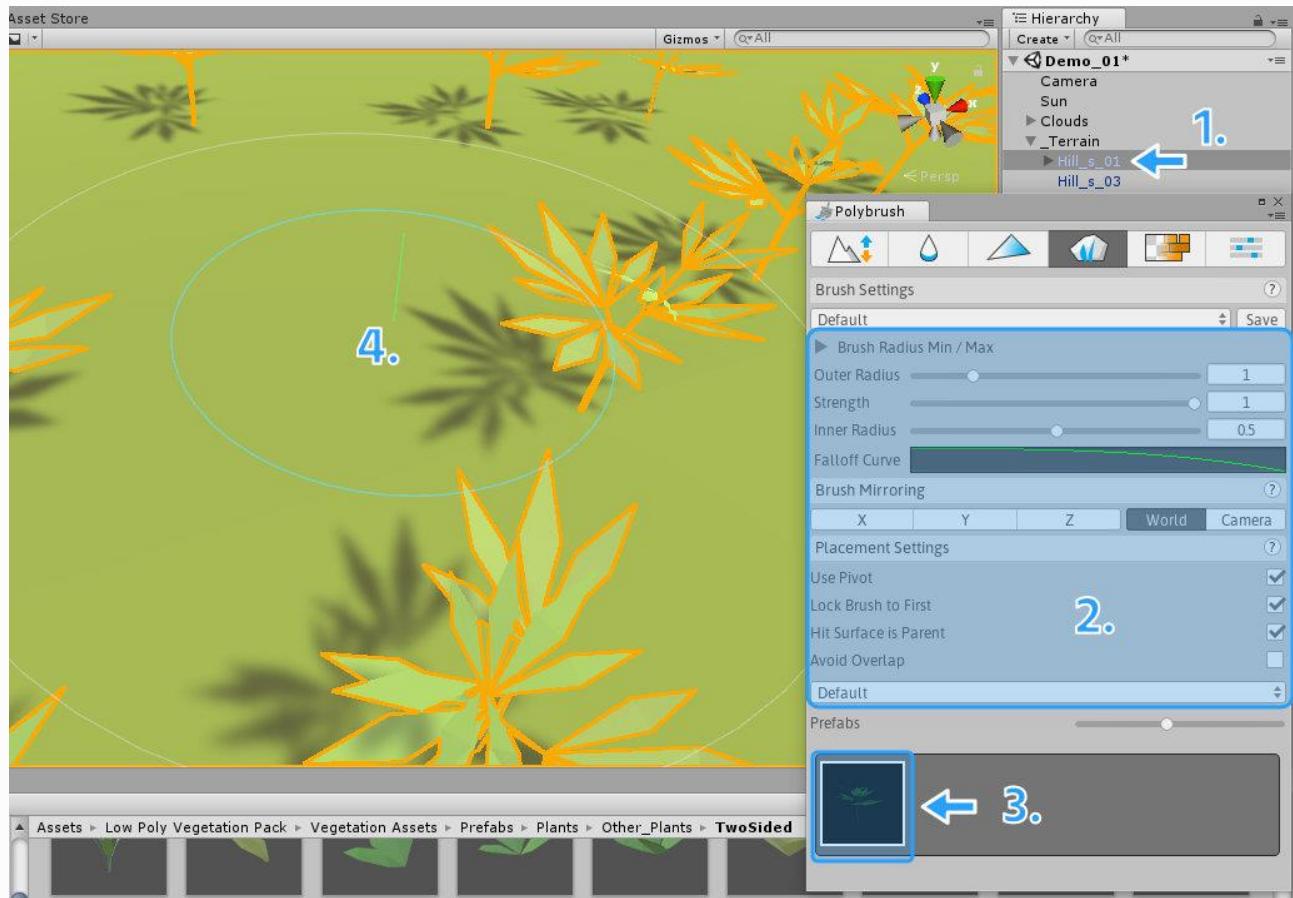


Now add any **Vegetation Prefab** you want to paint on the **Mesh Terrain**. I will choose **Plant_i_TwoS_09** from *Low Poly Vegetation Pack > Vegetation Assets > Prefabs > Plants > Other_Plants > TwoSided*. Drag and drop any **Vegetation Prefab** to **Prefabs** window:



*You can add as many Prefabs as you want.

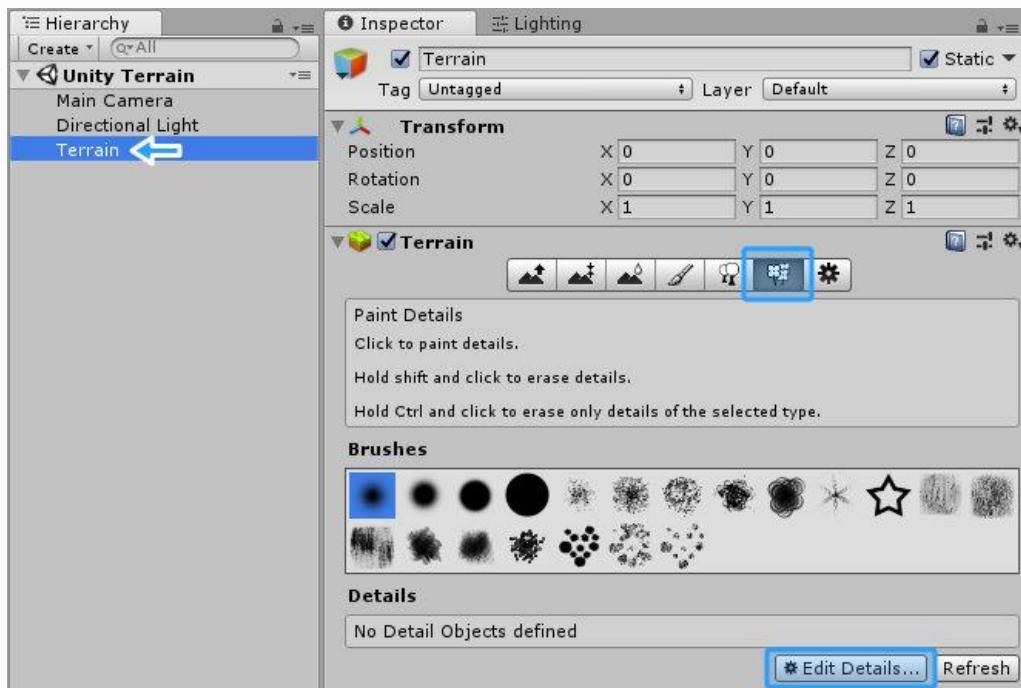
Select the mesh to paint on. I will select **Hill_s_01**. Then decide which Prefab to paint, in this case, press on the **Plant_i_TwoS_09** prefab we just added. And paint prefab on the mesh in the **Scene** view.



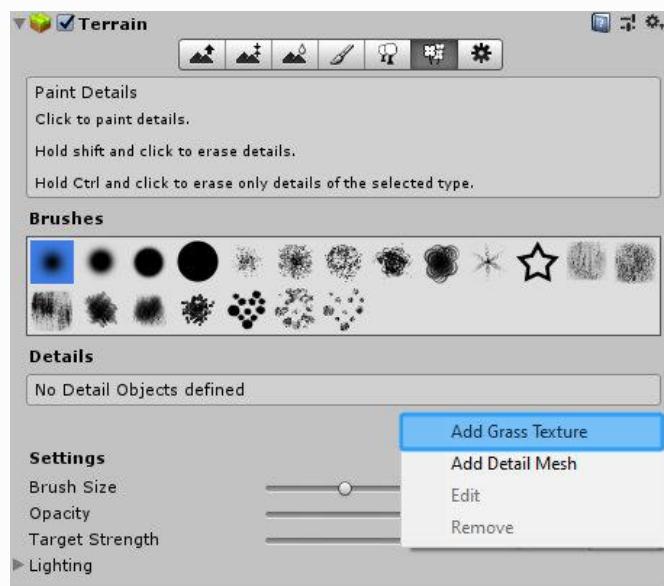
*I recommend enabling **Use Pivot** in the **Placement Settings** if you are painting my Vegetation Prefabs!

How to Paint Grass Textures on Unity Terrain

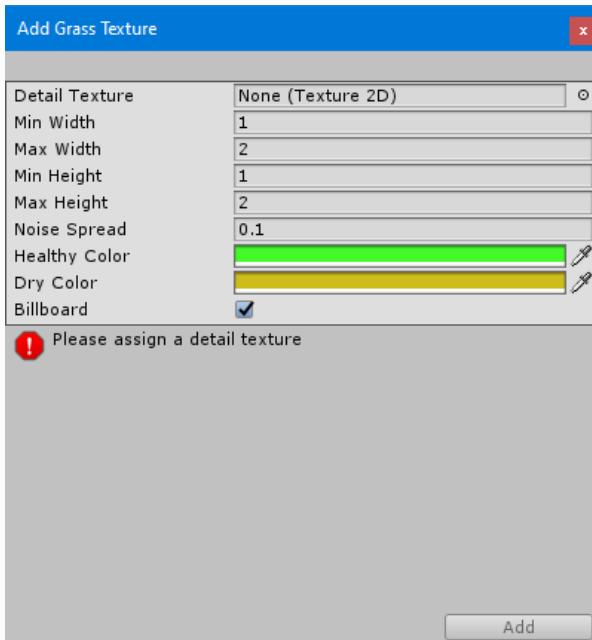
Select your **Unity Terrain** and go to the **Paint Details** tab. Click on **Edit Details...**



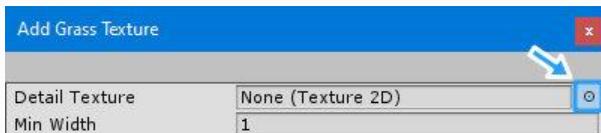
...and press on **Add Grass Texture**



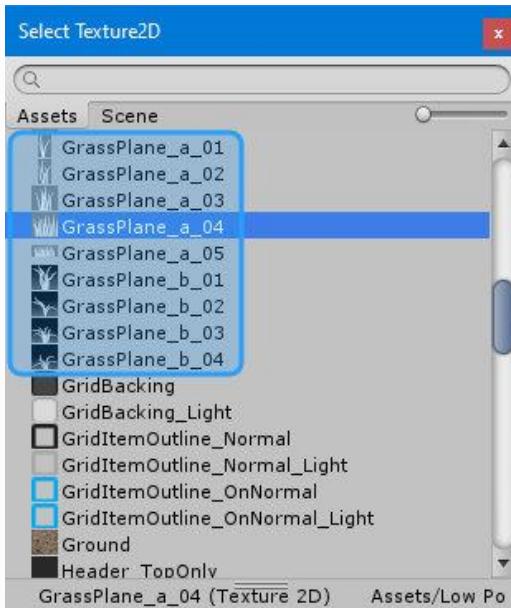
You should see a popup window **Add Grass Texture**.



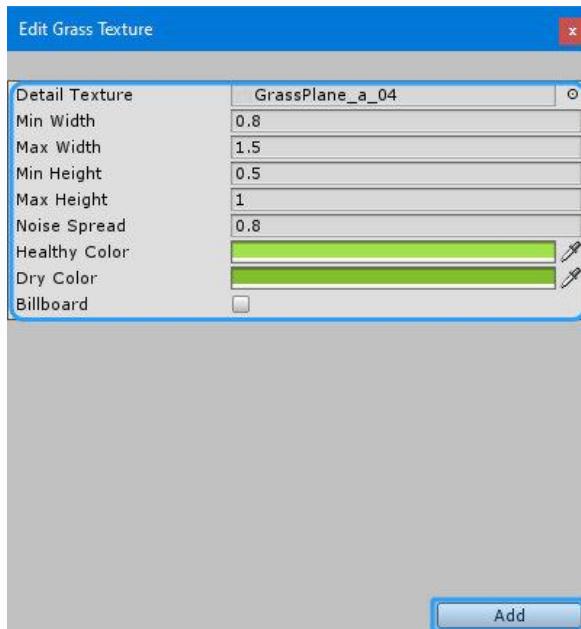
Press on a little **Circle Icon** to select a texture



And select any of **GrassPlane...** Textures you want



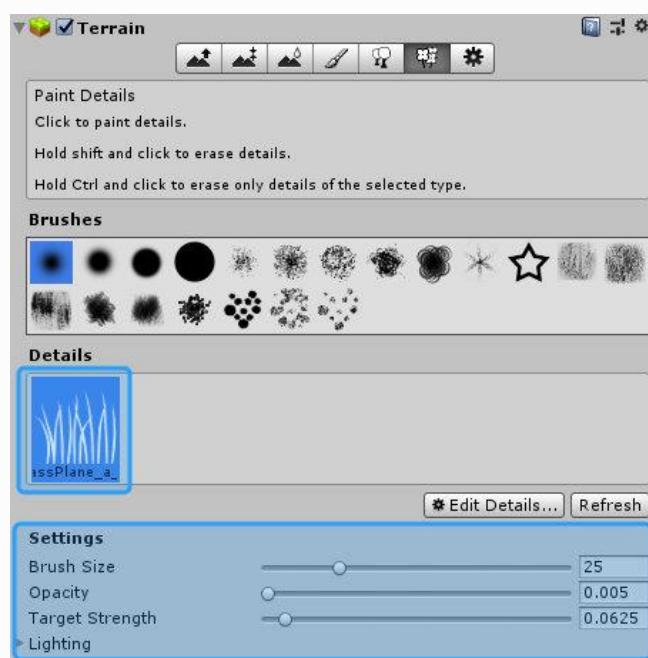
You can change the settings like grass size and color. After you are done, press **Apply**. My settings:



Healthy Color: A2E24C

Dry Color: 81C02C

That's it! Select **Grass Texture**, change **Settings**, and paint.



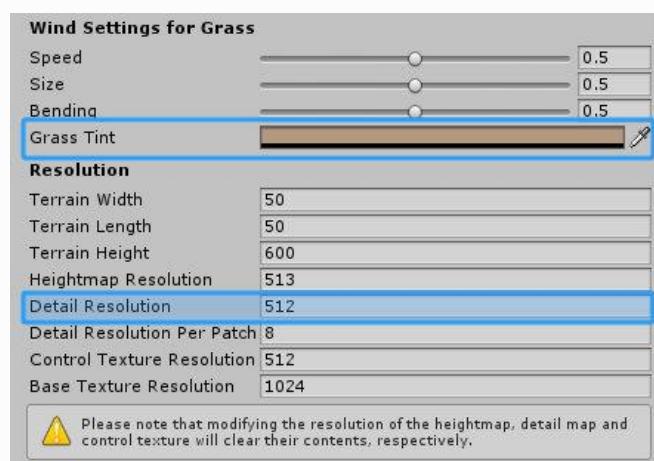


TIP! Open the Terrain Settings



Scroll down, and you will find **Detail Resolution**. I used **512** – so when I paint the grass it's not so dense.

Also, when you press **Play**, you will see that the grass moves and changes its color. You can change that color by changing **Grass Tint** color.



Additional Info

Naming Conventions

Prefab name example 1: **Cactus_a_m_01**

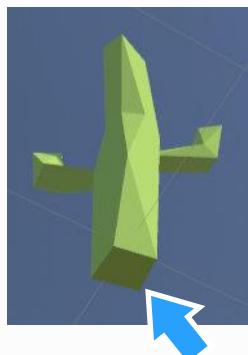
- **a** – Prefab type (there are types like a, b, c, d...)
- **m** – Medium size
- **01** – Prefab number

Prefab name example 2: **Plant_e_TwoS_04**

- **e** - Prefab type (there are types like a, b, c, d...)
- **TwoS** – Two-sided, mesh (can be seen from both sides)
- **04** - Prefab number

Prefab name example 3: **Cactus_b_m_BT_05**

- **BT** – Cactus **With_Bottom** – Cactus meshes have faces at the bottom



You can find these letters:

- **s** – small size
- **m** – medium size
- **l** – large size
- **OneS** – one-sided, mesh can be seen from only one side
- **TwoS** – two-sided, mesh can be seen from both sides
- **BT** - meshes have faces at the bottom

*Keep in mind that every vegetation 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 **Camera** and on **Inspector** scroll down to the bottom, you will see **LowPolyVegetation_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