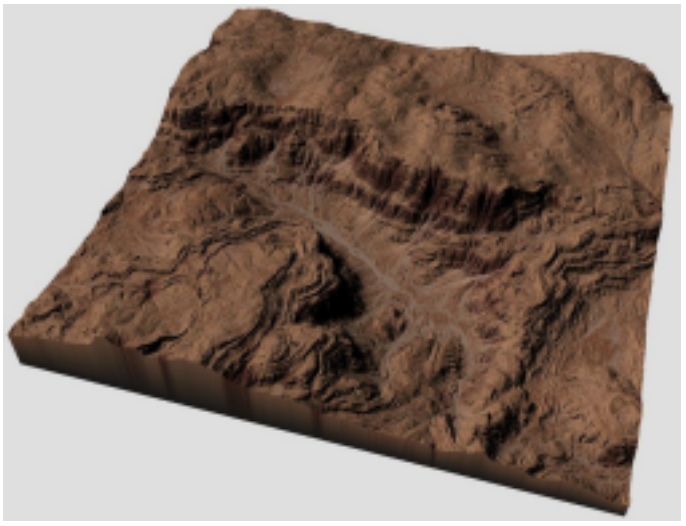
# ConjureScape terrain user manual

## Introduction

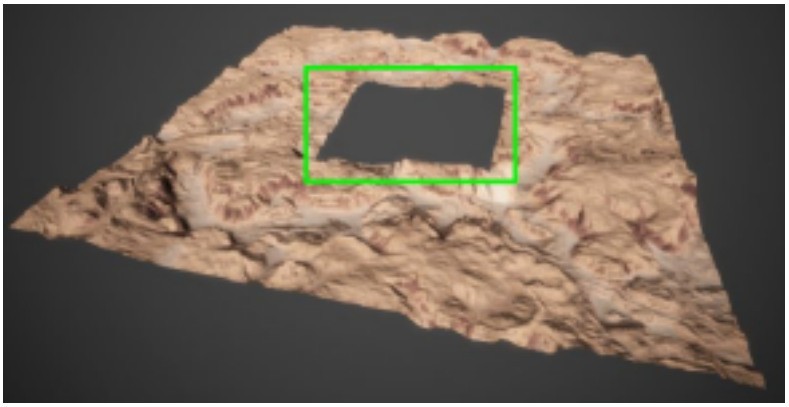
ConjureScape's terrains are designed for a seamless experience by simulating a massive environment. ConjureScape terrains consist of a terrain, surrounding terrain mesh and baked skybox to enhance the seamlessness of ConjureScape terrains. All ConjureScape terrains make use of a custom LOD system for trees and plants, please refer to the "ConjureScape vegetation user manual" for more details regarding the custom LOD system.

## Terrain



ConjureScape makes use of custom heightmaps and splatmaps for the best visual representation of a terrain. Our terrains are converted into Unity terrains and make use of the same components standard Unity terrains use.

## Surrounding Terrain Mesh

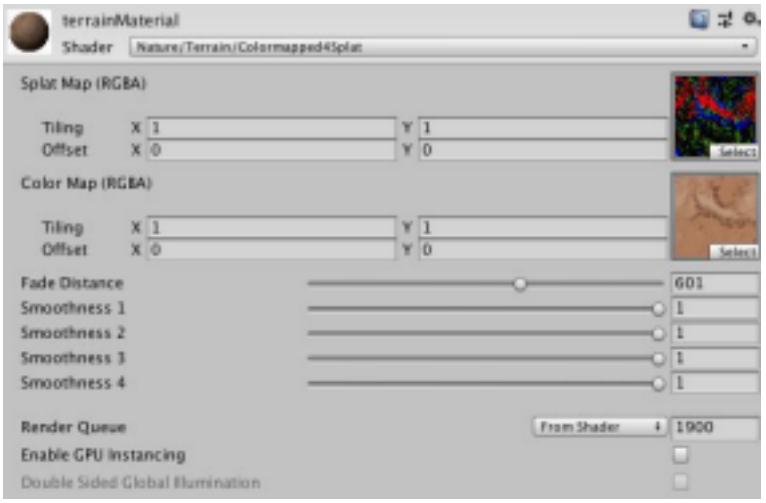


ConjureScape terrains includes a surrounding terrain mesh. The Unity Terrain is placed inside the empty section indicated with the green square in the image above. The surrounding mesh's purpose is to simulate a massive endless environment. The surrounding mesh is a standard optimized mesh object and does not consist of a Unity terrain.

## Baked Skybox

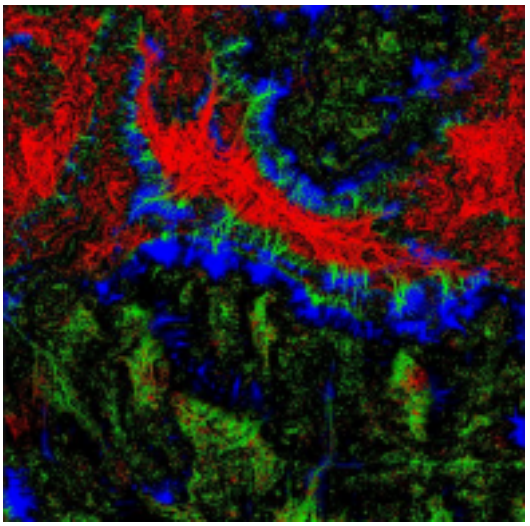

ConjureScape makes use of a custom system for baking an environment into the Skybox. Each terrain is accompanied with a pre baked skybox designed for a specific terrain. This adds to the illusion of an endless terrain. The surrounding mesh and skybox compliments each other to enhance the seamlessness of ConjureScape terrains.

## Terrain Shader




ConjureScape terrains make use of a custom terrain shader developed specifically for ConjureScape terrains. Let's look into each parameter in more detail.

**Splatmap:**

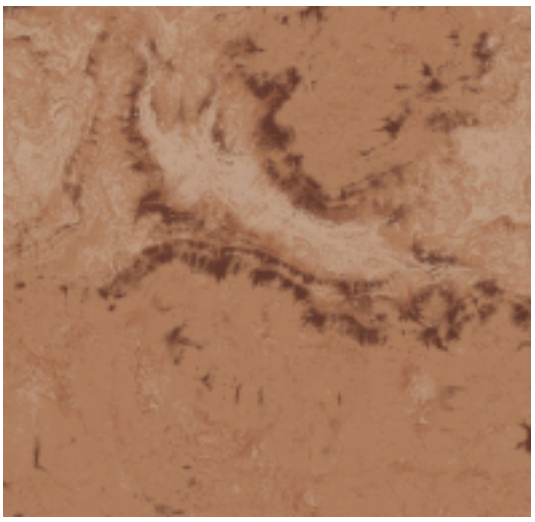


The Splatmap parameter keeps a reference to the splat map texture used for assigning terrain textures to the terrain. ConjureScape terrains consist of 4 terrain textures that are assigned inside Unity's terrain component under the "Terrain Texture" section. Each terrain texture is projected on the terrain using a color ID assignment from the splat map texture.

The texture Color ID assignments are as follow:

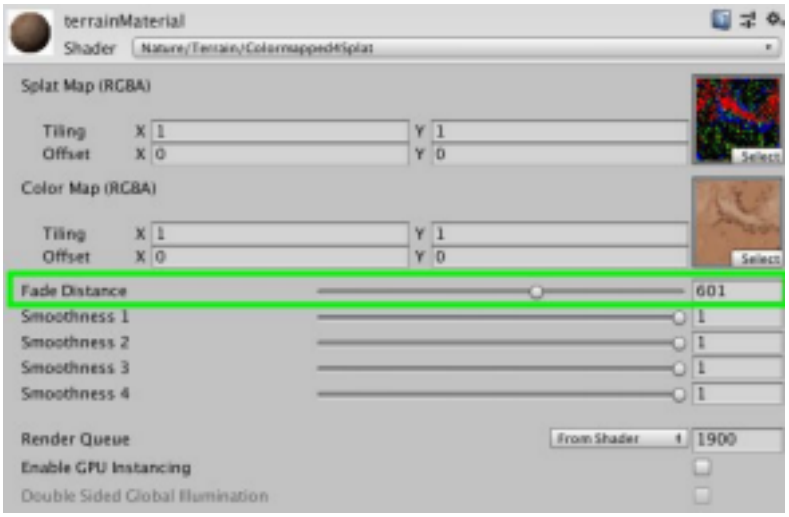
- Terrain texture 0 = Red
- Terrain texture 1 = Green
- Terrain texture 2 = Blue
- Terrain texture 3 = Black (Alpha)

**Colormap:**



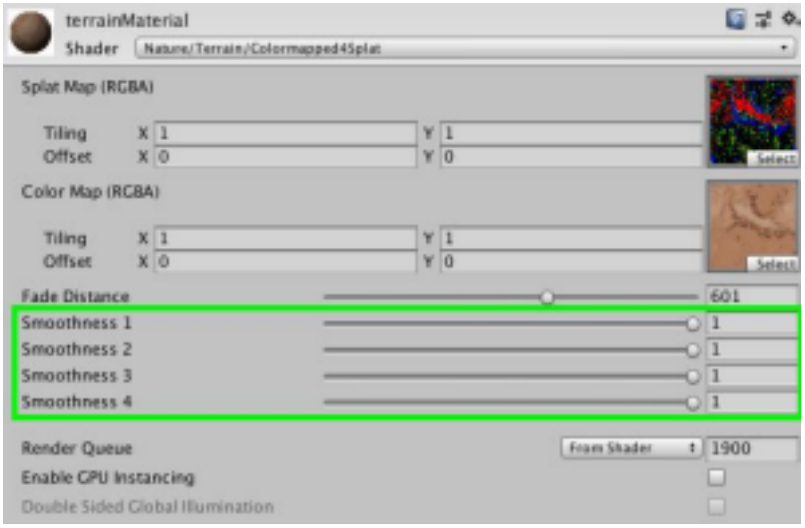
The Colormap parameter keeps a reference to the color map texture. The purpose for the color map is to have a high detail representation of the terrain texture when the camera is zoomed out far displaying the entire terrain. The color map is also used to multiply color values with the tileable textures that are assigned to the terrain. This results in a more realistic look and feel when a character walks on the terrain and the camera is close to the surface of the terrain. This technology gives ConjureScape terrains much more detail in comparison to what a standard Unity terrain can provide. Color maps have been specifically designed for each ConjureScape terrain.

**Fade Distance:**




The fade distance value is used to indicate when the color map should be activated. The value is based on the distance from the camera (meters). The “Fade Distance” value can be tweaked to your liking.

**Smoothness values:**




These smoothness values are similar to the smoothness value of the standard shader. You can set the metallic smoothness of a texture ID by adjusting these sliders. The smoothness IDs are assigned as follow:

- Smoothness 1 = Texture ID 0 in the terrain component
- Smoothness 2 = Texture ID 1 in the terrain component
- Smoothness 3 = Texture ID 2 in the terrain component
- Smoothness 4 = Texture ID 3 in the terrain component

## Boulders and Rocks

ConjureScape terrains comes with a set of boulders and rocks specifically developed for each terrain. Each boulder and rock has been set up with Unity's LOD system and contains all the meshes needed for high performance and includes colliders. Each rock and boulder containing a collider has a "Collider Manager" component attached to it. The purpose of the "Collider Manager" component is to manage performance of collisions.

The “Collider Manager” deactivates colliders attached to the boulder or rock when the camera is far away from the object, the colliders gets activated when the camera is close enough for a collision to happen. Feel free to remove this component if you would not require it for your project needs.

## Trees and Plants

ConjureScape’s vegetation system is a custom system built for high performance and great visual representation. It consists of a custom LOD system that allows for complex billboarding for vegetation that are far away from the camera and fades between different LOD levels. The system was designed for the best visual switching between LOD levels and makes used of the Unity tree system for terrains. For a more detailed explanation of how ConjureScape vegetation works, please refer to the “ConjureScape vegetation user manual”.

## Scene Requirements

When creating a new scene and you would like to use ConjureScape terrains and vegetation, the following needs to be present in a scene.

- Default_Include Prefab
- Terrain with painted ConjureScape trees and plants
- Camera (Any type of Camera, First person or Third person). The camera must be tagged as “MainCamera”.

The Default_Include prefab is located here:
Assets/Egamea/ConjureScape/Utilities/Prefabs/Default_Include.prefab

# Install Notes:

1. Create a new project.

2. Import the package

4. Open the sample scene.

## *Setting up terrains for the different Unity Render Pipelines.*

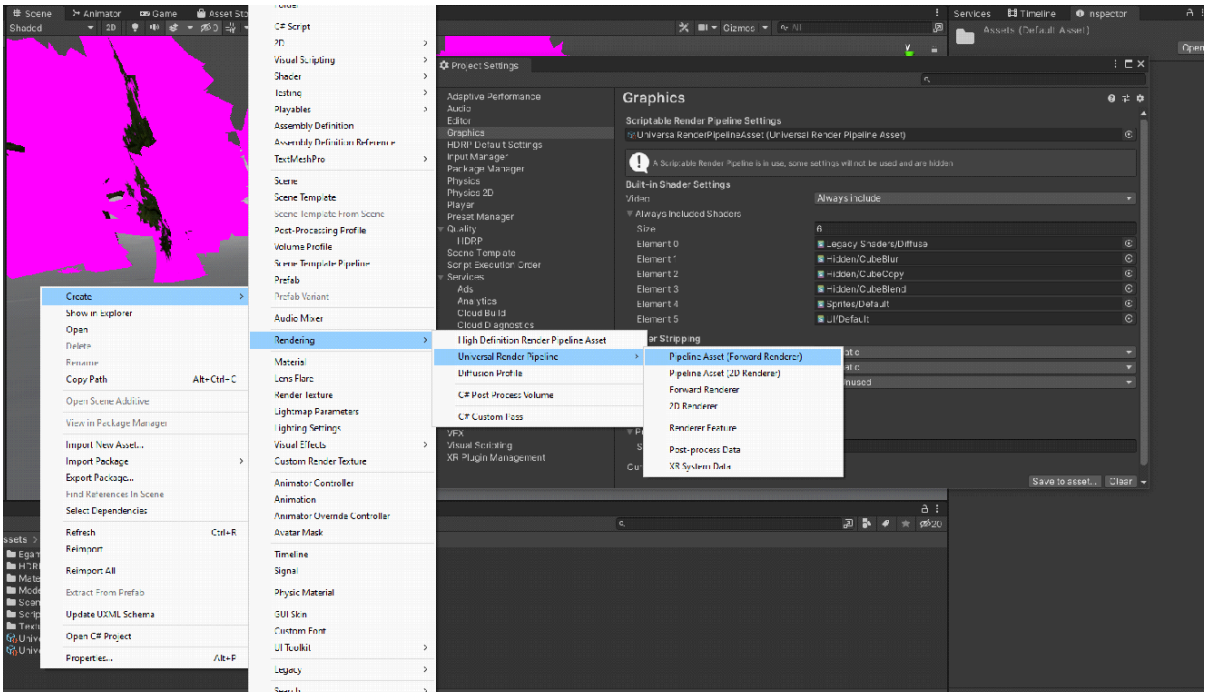
- Install “Editor Coroutines” from windows> Package manager. Change “Packages: In Project” to “Packages: Unity Registry” to find the “Editor Coroutines” Package.

## Universal Render Pipeline (URP):

To create a Universal Render Pipeline Asset:

- In the Editor, go to the Project window.
- Right-click in the Project window, and select **Create** > **Rendering**> **Universal Render Pipeline**> **Pipeline Asset**. Alternatively, navigate to the menu bar at the top, and select **Assets**> **Create**> **Rendering**> **Universal Render Pipeline**> **Pipeline Asset**.

You can either leave the default name for the new Universal Render Pipeline Asset, or type a new one.



# High Definition Render Pipeline:

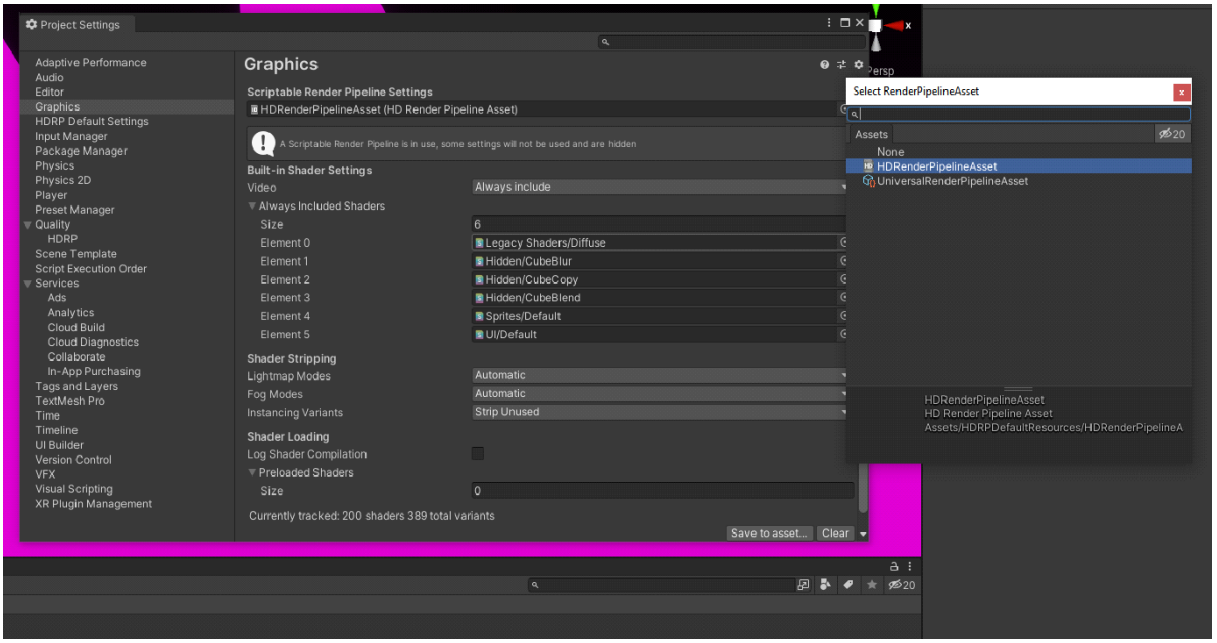
- First, add the HDRP package to your existing Project:
- In the Unity Editor, open the Package Manager window (menu: **Window > Package Manager**).

Next, create and set up a High Definition Render Pipeline Asset.

Create an HDRP Asset by selecting **Assets > Create > Rendering > High Definition Render Pipeline Asset**.

Open the **Graphics Settings** window (menu: **Edit > Project Settings > Graphics)**.

Assign the High Definition Render Pipeline Asset to the **Scriptable Render Pipelines Settings** field, at the top of the window. To do this, click the radio button and select the Asset from the list, or drag the Asset into the field.

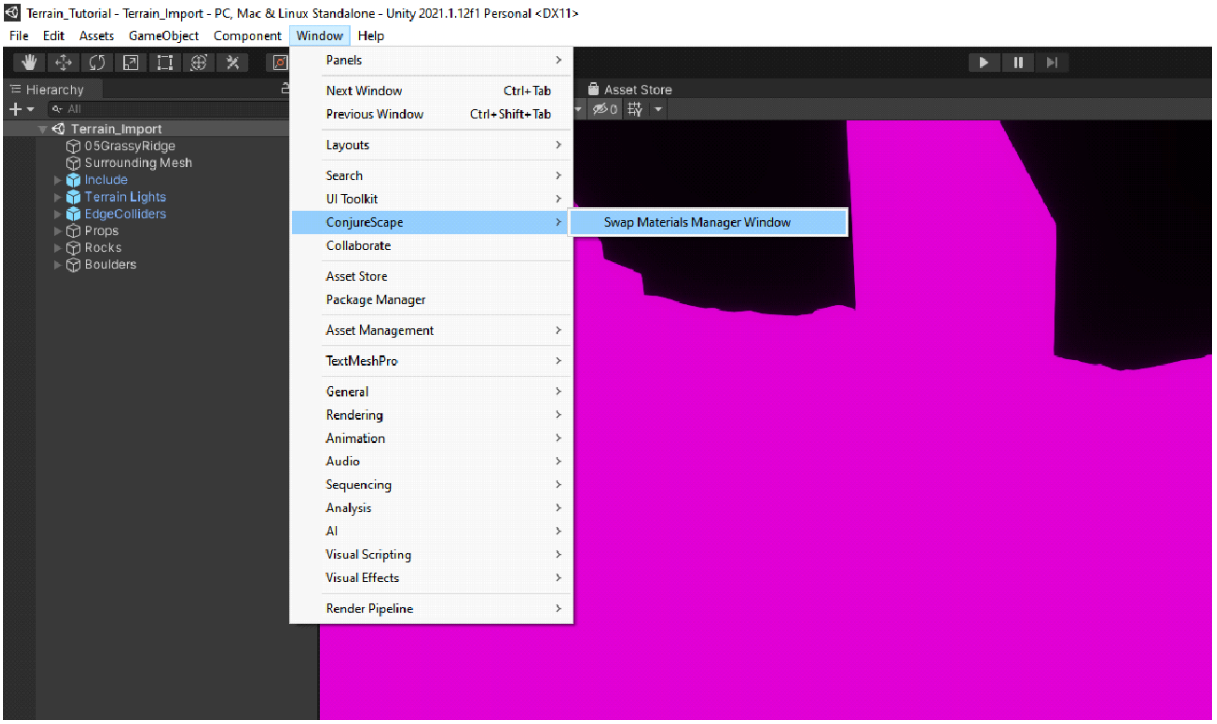


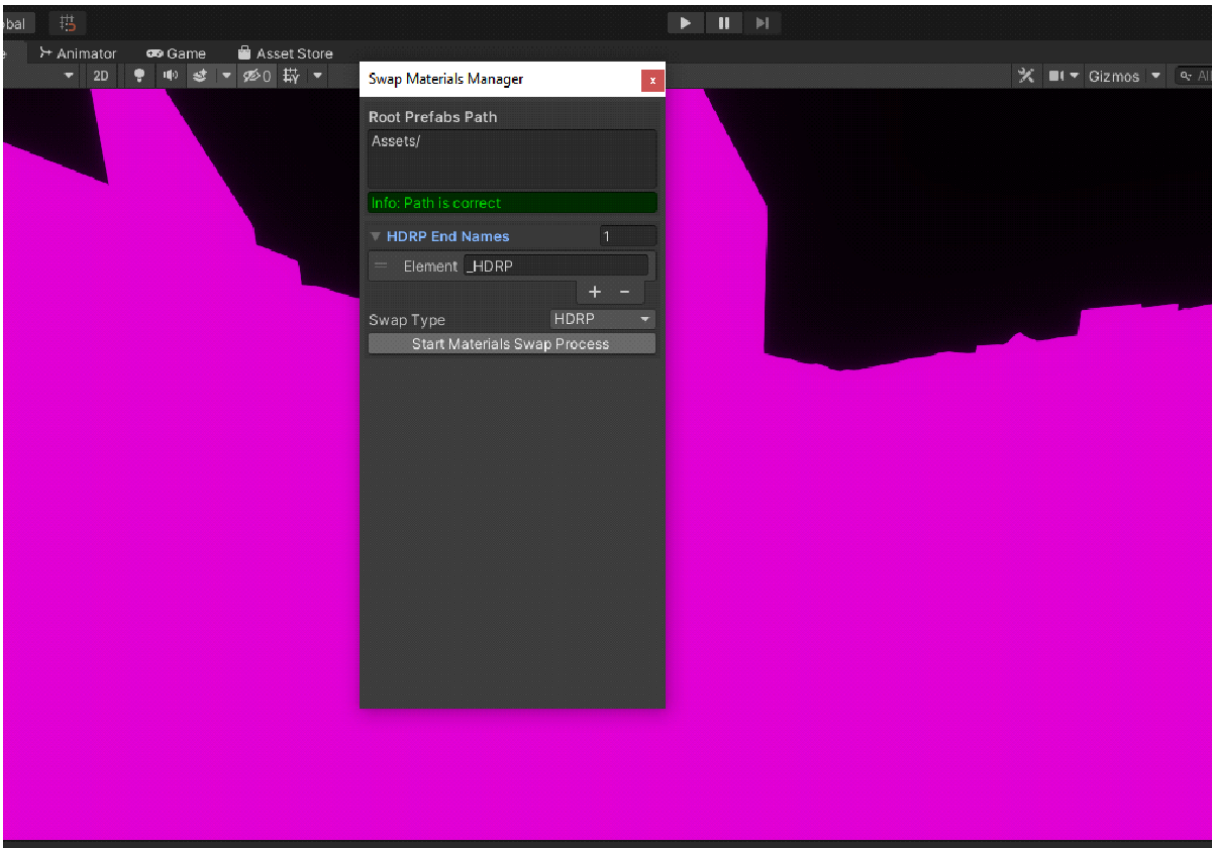
# Different Render Pipeline Scenes:

Select the scene of the pipeline you want to run the project in


# To swap the materials from any Render pipeline to HDRP/URP/Standard:

- In the windows panel navigate to “Conjure Scrape”> “Swap Materials Manager Window”.
- In the “Swap Materials Manager Window” make sure you have the following options:
- =Element:_HDRP
- =Element:_URP
- Swap Type: HDRP/URP/Standard
- Then click on Start Materials Swap Process.


Terrain_Tutorial - Terrain_Import - PC, Mac & Linux Standalone - Unity 2021.1.12f1 Personal <DX11>
File Edit Assets GameObject Component Window Help
Hierarchy
Terrain_Import
05GrassyRidge
Surrounding Mesh
Include
Terrain Lights
EdgeColliders
Props
Rocks
Boulders
Asset Store
Panels
Next Window Ctrl+Tab
Previous Window Ctrl+Shift+Tab
Layouts
Search
UI Toolkit
ConjureScape
Collaborate
Asset Store
Package Manager
Asset Management
TextMeshPro
General
Rendering
Animation
Audio
Sequencing
Analysis
AI
Visual Scripting
Visual Effects
Render Pipeline
Swap Materials Manager Window


Animator
Game
Asset Store
Gizmos
Swap Materials Manager
Root Prefabs Path
Assets/
Info: Path is correct
HDRP End Names
1
Element _HDRP
Swap Type
HDRP
Start Materials Swap Process