

# LOOTMAKER

RANDOM LOOT GENERATOR

v2.0.0

Created by [Ripcord Development](#)

[info@ripcorddev.com](mailto:info@ripcorddev.com)



This project has been thoroughly tested for bugs before being sent to the Unity Asset store. In the event that you do find an issue with this package, please contact us before posting negative feedback on the Unity Asset Store. We are more than willing to help solve any issues you may encounter.

**LootMaker** is a collection of common collectable objects and a system for randomly generating them in your game. The bulk of the work is contained in one function so it's easy to integrate into any project. The loot prefabs have their own scripts attached which will handle the rest of the work.

## PREFABS

The **LootMaker** package includes a large number of models and textures for objects representing common types of loot found in games. These prefabs are sorted into two different categories, 2D and 3D. The 2D icons are sprites (a single polygon that always faces the camera and a texture). The 3D prefabs are full 3D models that can be viewed from any direction.

- **3D** - 3D models representing many common categories of loot objects found in games today.
- **2D** - 2D sprite versions of all the 3D models.
- **Particles** - Particle effects designed for each type of loot. These are generated when the loot object is collected.
- **Text Meshes** - Text meshes that display the value of the collected loot object.

## SCRIPTS

**LootMaker** consists of a couple main scripts and a few support scripts that help enhance the look of the loot objects in the world.

- **LootManager** – This script contains the variables that store the total amount of each type of loot collected. If you want to add more types of loot, or change the types that can be collected simply add or remove options from this script. There should only be one instance of this script in your scene.
- **LootGenerator** – This script when triggered randomly selects loot objects and spawns them at the specified mount points.
  - **MountPoints** – The locations that new loot objects will spawn at. This can either be a single transform, or a transform with multiple child transforms. In the case of multiple child transforms, the parent will not be used as a spawn point.
  - **LootTypes** – A list of all the loot prefab objects that can be spawned. Each type can be given a specific name to help keep things organized in the inspector. Each loot type can have multiple prefabs.
- **LootObject** – This script must be placed on any collectable loot object. It contains the numerical value of the object as well as generates any effects that may be attached to it.
  - **LootType** – Select the type of loot you want the object to modify. The LootTypes are stored in the LootManager.
  - **LootValueMin** – This is the minimum value the loot object can have.
  - **LootValueMax** – This is the maximum value the loot object can have.

**Tip:** The value of the loot object will be randomly selected from anywhere between the min and max values. If you want the loot object to have a specific value just make the MIN and MAX values the same.

- **LootHighlight**
  - **ShowHighlight** – If true, generate an effect to make the loot object more visible in the world.
  - **HighlightEffect** – An effect prefab that spawns when the loot object becomes visible in the scene.  
*Tip: You could skip this feature entirely if you want to build an effect into the loot prefab itself.*
- **LootAudio**
  - **PlayAudio** – If true, audio clips will be played in the instances listed below. If one of the values listed below does not have an audio clip, it will be ignored.
  - **SpawnAudio** – Play an audio clip as soon as the object enters the scene.
  - **IdleAudio** – Play an audio clip while the loot object is in the scene. This clip will stop playing once the loot object is collected.
  - **CollectAudio** – Play an audio clip when the loot object is collected.
- **LootEffects**
  - **Show Effects** – If true, an effect will be generated when the loot object is collected.
  - **LootCollectEffect** – Any prefab (ideally a particle effect) placed here will be instantiated into the scene when the loot object is collected.
  - **DisplayValue** – If true, generate a TextMesh object that displays the value of the loot object collected.
  - **ValueDisplay** – The TextMesh object that will display the value of the loot object collected.
  - **ValuePrefix** – Any string entered will appear in front of the loot valueDisplay.
  - **ValueSuffix** – Any string entered will appear behind the loot valueDisplay.
- **TextBehaviour** – This script is attached to all the text mesh prefabs. It controls the colour and movement of the text mesh.
  - **Text Colour** - The colour of the text mesh when it is generated.
  - **Target Transform** - When the text mesh is generated its position and rotation will update until it matches the position and rotation of the target.  
*Tip: The target object in each of the text mesh prefabs is originally parented to the text mesh but when activated the parent/child link is broken. This allows the target to spawn relative to the text mesh but not move with it once the motion starts.*
  - **Move Speed** - The speed at which the text mesh will move from its start point to the position of the target object
  - **Rotate Speed** - The speed at which the text mesh will rotate from its start point to the rotation of the target object
  - **Fade Out** - If active the text will fade over time until it is invisible.
  - **Time Visible** - How long the text mesh is visible before it disappears.
  - **Fade Speed** - The speed at which the text mesh fades out. The object will be removed from the scene once it fades out completely.

**Common Scripts** – These scripts are used in other asset packages by created by RipcordDevelopment. Their functions aren't essential to this package, but they typically enhance its overall appearance.

- **LookAtCamera** – This script is attached to all the sprite prefabs. Any object it is attached to will face the camera when it is spawned.
  - **AlwaysLookAt** - If you want the object to always face the camera wherever it moves, make sure this option is enabled.  
*Tip: This feature is handy for 2D sprites in a 3D world.*
- **SpinAndBounce** – This simple script modifies the movement of the loot object.
  - **Spin Rate** - The speed at which the object spins. Negative numbers will make the object spin in the reverse direction.
  - **Bounce Amount** - Based on a sine wave, the bounce amount controls how far the object will hover above or below its start point.

- **Bounce Speed** - The speed at which the object travels from the max and min points on the sine wave.
- **DestroyParticleSystem** – This script will remove a particle system from the scene once it has stopped playing.

**Demo Scripts** – These scripts aren't essential to the functionality of this asset package, but are required to properly demonstrate the overall functionality.

- **DEMO\_LootGenerator** – This script is an expanded version of the LootGenerator script. It was created to easily demonstrate the functionality and flexibility of **LootMaker**. The purpose of this script is to allow the user to view various combinations of loot objects and generate them over and over again without having to reload the scene. This script is thoroughly commented.

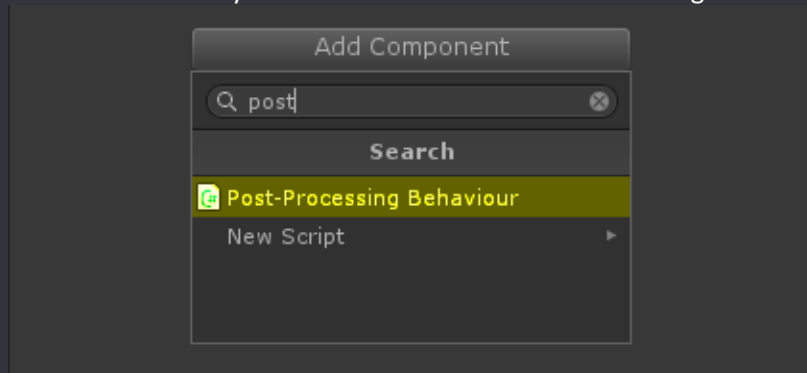
## SCENE SETUP

*The following steps are only required if you want to use the full LootMaker demo as seen in screenshots and the WebGL build. If you want to just get right to using LootMaker for your own project you can skip this section.*

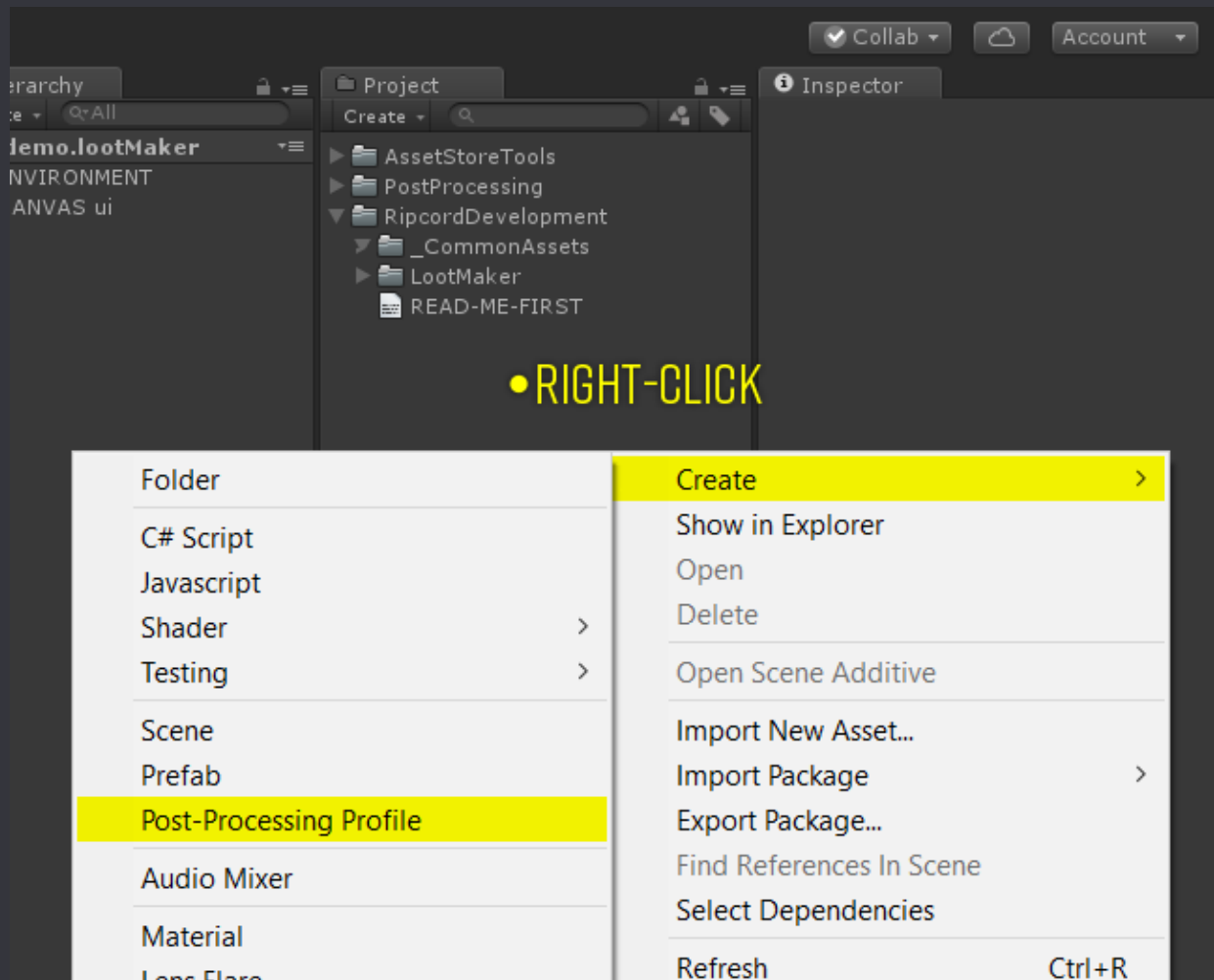
The demo scenes in this package make use of a free asset package provided by Unity. To replicate the exact look and functionality of the demo scenes, you'll need to import this package.

Download this package from the Unity Asset Store and import it into your project - [Post Processing Stack](#)

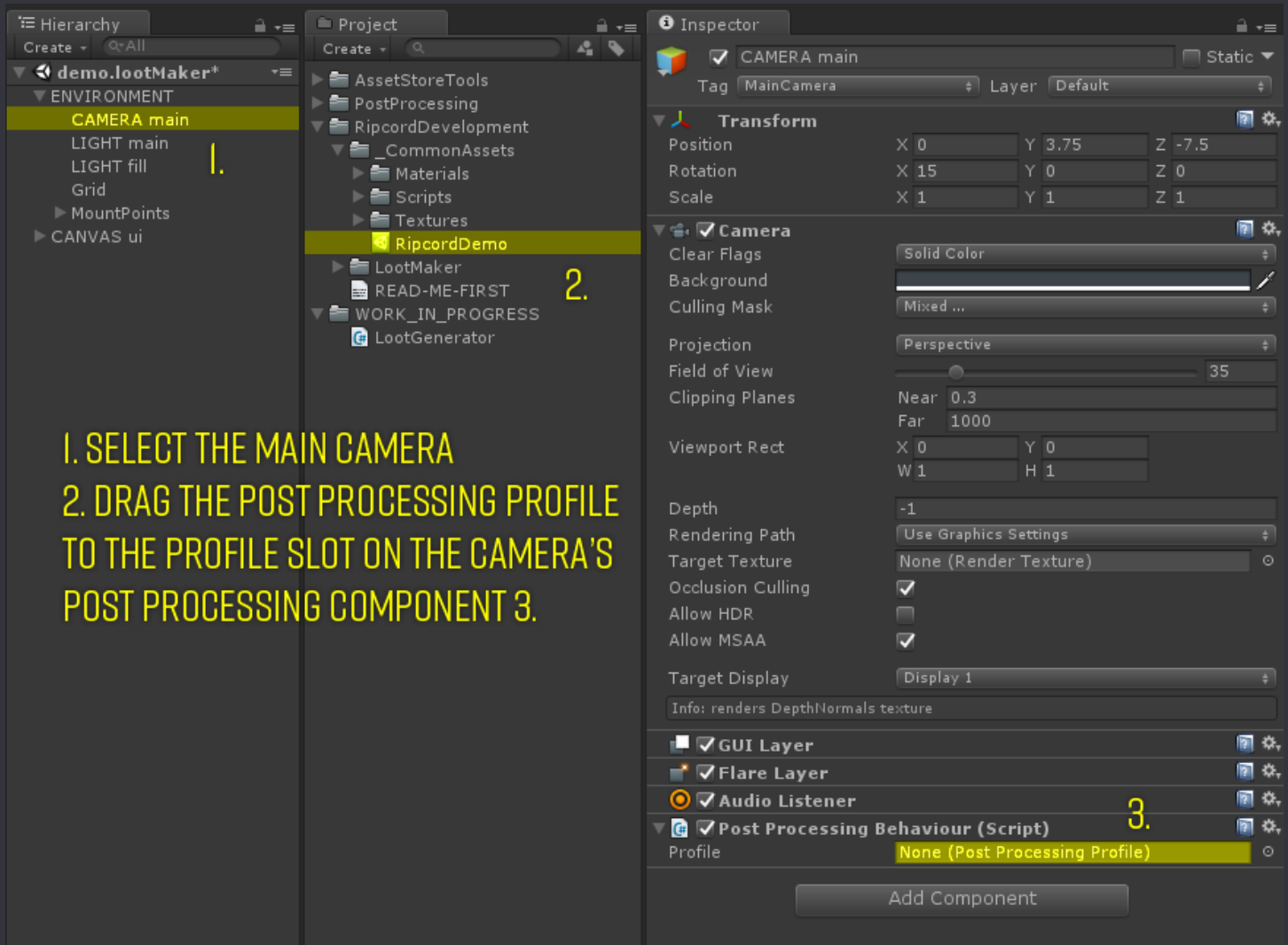
Select the camera in your scene and add the Post-Processing Behaviour component:



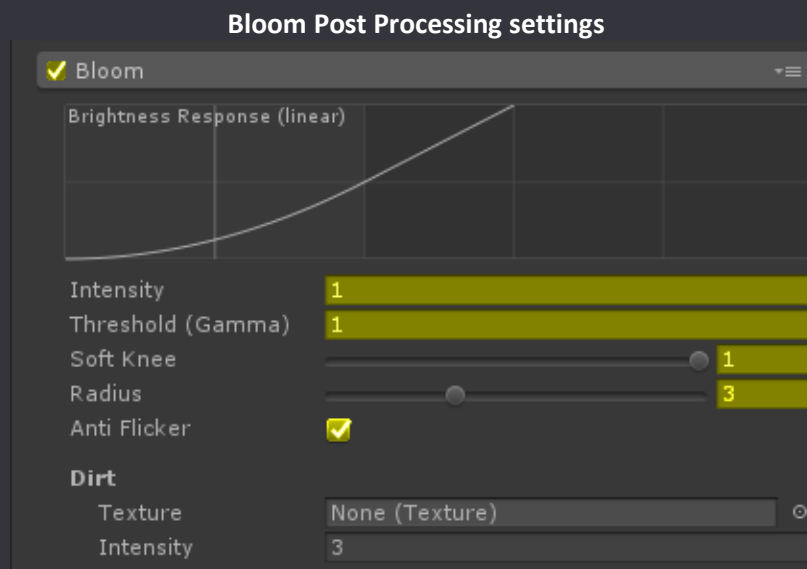
Right-click in your Project panel and select Create > Post Processing Profile. Name the profile whatever you like:



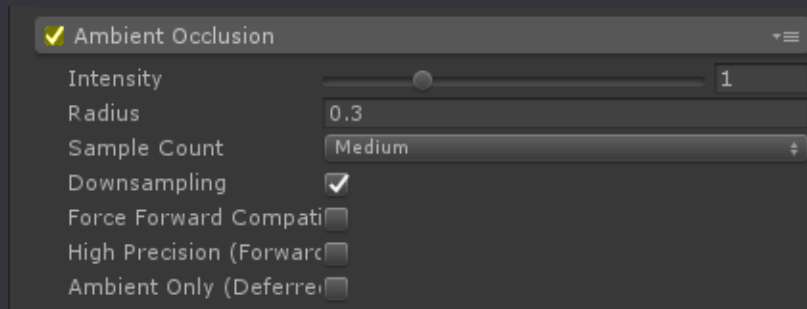
With your Main Camera selected, click and drag the new Post-Processing profile to the Profile slot on the Post Processing Behaviour component on the camera.



Select the Post Processing Profile in the Project panel and apply the following settings:



Ambient Occlusion Post Processing settings



## WRAP UP

**LootMaker** provides a flexible system for easily generating loot objects within your game. Whether you choose to use the LootGenerator, or your own existing system, the loot objects themselves will make a great addition to any project.

The code has been heavily commented and modularized to make this package as easy to understand as possible. If you have any questions or comments, please don't hesitate to reach out.

If you find this package useful, please don't forget to leave positive feedback on the Unity Asset Store. **If you have any issues, please contact me with as much information about the issue as you can, and I will get back to you as soon as possible.**

Thank you!

RIPCORD  
DEVELOPMENT //

[www.ripcorddev.com](http://www.ripcorddev.com)