Two Sculpted Barrel Designs, 100% Procedural, 100% PBR

∞ INFINITY SERIES PROCEDURAL PHYSICALLY BASED RENDERING

Barrels

VISIT WWW.INFINITYPBR.COM FOR THE LATEST UPDATES, FORUMS & MORE ASSETS.

** If you have any problems getting set up or using our models, please use the Unity forums to contact us and we'll get it sorted.*

** If you enjoy using our models, please write a review on the Unity Asset Store so other developers know how cool our stuff is :D*

1. Introduction

"Barrels" is a collection of procedural PBR barrel designs for video games developers. The procedural aspect means there are virtually unlimited looks you can give to the barrels, creating unique looks that no one else has. Physically Based Rendering means the looks can appear hyper realistic.

Due to all of this, there is a little setup involved. It shouldn't take long and maybe it'll be quite fun, as you'll get to fine-tune the look of your models.

In most cases the Quick Set Up section will be all that you need. If you're interested in knowing more about each of the values you're able to tweak, check out the Procedural Values section.

For advanced users, if you're interested in scripting run time changes in the texture of the model, refer to the Scripting section.

Finally we include a brief list of the Animations currently included.

We plan on updating our assets periodically, so please check the Asset Store for available updates.

2. Quick Set Up

This quick guide will work for most users, and does not allow for run time changes in the look of the textures. *For videos, please visit our website at www.InfinityPBR.com where you will find much more detailed examples.* ***We highly suggest you create your maps in a new, empty project.***

1. Bring the model prefab into the Scene view. Use the full resolution model for now.
2. Find the latest procedural material: `Assets/SFB_Dungeon/SFB_Barrel/Procedural Material/SFB_DungeonBarrel_v##`
3. Drag that material onto each mesh of the model. It may take a bit of time during any of these steps for Unity to pre-build the material. After dragging, the model may appear black or another solid color until the process is complete. Please be patient. (We hear this speed is based mostly on your GPU)
4. Rotate the camera in the Game and/or Scene view to something that you like, and select the Procedural Material in the Project view to load it in the Inspector.
5. Adjust the various aspects of the material to obtain the look you like. Each time you change something Unity may take a moment to rebuild the material. This is not a run-time optimized material, and with so many options, it may take a few moments to complete.
6. When you are satisfied, make sure you have “Compute all Outputs” selected in the Inspector.
7. Choose “Save Bitmaps” from the Gear menu in the top right corner of the Inspector window. This will save the computed outputs to your disk. It’s best to keep each output group you plan on using in it’s own subfolder, for organizational purposes.
8. Create a new material with the Standard Shader & populate the inputs with the maps you saved in the last step. **Use “metallicForExport” in the metallic field.**
9. Drag the material onto your model.
10. Don’t forget to choose the correct LOD for your game, and play with the size settings of the textures to optimize their system resource usage.

3. Procedural Values

Here you'll find greater details on what each value does and how it may be used. There could be a great many ways to use the value options, often with each other, that we don't know about or don't talk about here. Take a moment and play around with it and see what you can do! The ID is used for scripting run time changes.

SFB_Dungeon_Barrels_1 | SFB_Dungeon_Barrels_2

Category	Name	ID | Type Min,Max	Description
Main	Wood Material	`WoodMaterialNumber` `int (1,10)`	Which material to use
	Metal Material	`MetalMaterialNumber` `int (1,10)`	Which material to use
	Rotate Wood?	`WoodRotate` `Boolean`	Non-default woods require rotation. Click this.
	Metal Roughness	`DefaultMetalRoughness` `float (0.0,1.0)`	Roughness of the default metal.
	Wood Roughness	`DefaultWoodRoughness` `float (0.0,1.0)`	Roughness of the default wood.
Wood Weathering	Dust	`WoodWeatheringDust` `float (0.0,1.0)`	Dust, starts on top, very top-heavy
	Dirtiness	`WoodWeatheringDirtiness` `float (0.0,1.0)`	Dirtiness, starting from seams
	Edge Wearing	`WoodWeatheringEdgeWearing` `float (0.0,1.0)`	Edge Wearing
	Varnish Peeling	`WoodWeatheringVarnishPeeling` `float (0.0,1.0)`	Peels varnish starting from middle areas
	Age	`WoodWeatheringAge` `float (0.0,1.0)`	General Aging of the wood
	Desaturation	`WoodWeatheringDesaturation` `float (0.0,1.0)`	Desaturation of the parts missing varnish
	Brightness	`WoodWeatheringBrightness` `float (0.0,1.0)`	Brightness of the parts missing varnish
Metal Wear	Dust	`MetalWearDust` `float (0.0,1.0)`	Dust, starting on the top
	Dirtiness	`MetalWearDirtiness` `float (0.0,1.0)`	Dirtiness of the metal
	Edge Wearing	`MetalWearEdgeWearing` `float (0.0,1.0)`	Edge Wearing
	Rust	`MetalWearRust` `float (0.0,1.0)`	Adds rust to the metal
	Height	`GroundDirtHeight` `float (0.0,1.0)`	Height (bottom up) of the dirt

Category	Name	ID | Type Min,Max	Description
Ground Dirt	Level	`GroundDirtLevel` `float (0.0,1.0)`	Overall thickness
	Contrast	`GroundDirtContrast` `float (0.0,1.0)`	Contrast of the pattern
	Color	`GroundDirtColor` `Color`	Color of the dirt
	Roughness	`GroundDirtRoughness` `float (0.0,1.0)`	Reflectiveness of the dirt.
Ceiling Dirt	Height	`CeilingDirtHeight` `float (0.0,1.0)`	Height (top down) of the dirt
	Level	`CeilingDirtLevel` `float (0.0,1.0)`	Overall thickness
	Contrast	`CeilingDirtContrast` `float (0.0,1.0)`	Contrast of the pattern
	Color	`CeilingDirtColor` `Color`	Color of the dirt
	Roughness	`CeilingDirtRoughness` `float (0.0,1.0)`	Reflectiveness of the dirt.
Dirt	Level	`DirtLevel` `float (0.0,1.0)`	Thickness of the dirt - starts in seams.
	Contrast	`DirtContrast` `float (0.0,1.0)`	Contrast
	Grunge Amount	`DirtGrungeAmount` `float (0.0,1.0)`	Grunge on the surface
	Color	`DirtColor` `Color`	Color of the dirt
	Roughness	`DirtRoughness` `float (0.0,1.0)`	Reflectiveness of the dirt.

4. Scripting

It's possible to change values during run time. We include a few versions of the material, some of which are optimized for common run-time options. In those cases, you'll likely want to bake maps for the base materials you plan on using (which do not change at run time), and use the optimized versions. This will speed up the changes in game.

Please Note: We are not the best coders. There may be more ways of doing what we're doing, perhaps better ways. Please use the forums on our site and the Unity forums if you'd like to discuss or ask the community about various ways of doing this. ***We are also using Unity Script because, simply, it's what we currently understand.*** *Check out our demo scripts for more extensive examples.*

```
var substance                : ProceduralMaterial;

// Set an Int or a Float value
substance.SetProceduralFloat("Grunge2Volume", 0.5);

// Set a Color value
substance.SetProceduralColor("Grunge2Color", Color(1,1,1,1));

// Get a Vector2 value
var currentOffset   : Vector2 = substance.GetProceduralVector("Grunge2Offset");

// Set a Vector2 value
substance.SetProceduralVector("Grunge2Offset", Vector2(currentOffset[0],currentOffset[1]));
```

5. Animations

No animations for the Barrels. :(

6. Level Of Details

There are multiple level of details available. The full resolution is already very mobile friendly, and the other LODs are a bit distorted so should only be used far away. However, the lowest LOD is only 98 Tris, which is pretty low.

#	Name	Full	LOD1	LOD2	LOD3
1	Barrel	464	278	166	98

7. Change Log

v7.0 - Initial Version.