Mushroom Monster

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

1. Introduction

"Mushroom Monster" is a procedural PBR character designed for video games developers. The procedural aspect means there are virtually unlimited looks you can give to the character, creating a unique look that no one else has. Physically Based Rendering means the look 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 character.

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. Load the Assets/SFBayStudios/SFB Demo Scenes/SFB Mushroom Monster Texture Creation.unity scene
2. Select the main Procedural Material in the inspector. Assets/SFBayStudios/SFB Mushroom Monster/Procedural Materials/SFBv#_MushroomMonster_v#
3. Make any adjustments you'd like to the "Head" section. Be sure to check the "Enable Texture Modification" box under the top "Main" section. Also make any adjustments to the Environment options, such as Ground Dirt, Damage etc.
4. In the scene view, select "Copy This To That Control". This editor script (see Inspector) will copy the settings from "Head" to all the listed sections in the target materials. You can add or remove them as you'd like, but if you plan on making a custom Copy This To That controller, I suggest you create a new object, and keep the demo defaults as is. ***Visit www.infinityPBR.com/CopyThisToThat/ for a "cheat sheet" of section names in each Procedural Material.***
5. Click the "Copy Settings" button, and wait until all materials are finished updating. You'll see them update in the scene view, but it may take a few moments.
6. Do the same for the 2nd Copy This To That Controller.
7. If you want to make any more modifications to individual parts or materials, do so now. If you click "Copy Settings" again, any changes you make will be lost.
8. When you are satisfied with the look of your model, select the "Mass Exporter" object. In the inspector, confirm that all of the Procedural Materials are seen in the "Substances" array. Give your new character a descriptive name, such as "Blue Mushroom #1", and choose whether to remove Emissive & Height maps. If you are going to share maps with a previous export, add the "Previous Group Name" and select which maps to share. If this is your first export, or you've changed so much of the look that you can't share maps, uncheck the boxes. Finally, click "Export Materials." It will take a few minutes for Unity to import all the .tga files that are created.
9. When complete, you'll find your game-ready materials in Assets/SFBayStudios/Exported Materials/[Group Name]
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.

NOTE: Follow the same steps above with SFB_MushroomBase to create stone & metal textures for use with the Skin. Each base texture has options of its own. Once you're set, export the textures to your project just like in the above instructions. Bring those finished textures into the Input fields in the Skin procedural material, and you can use the non-skin textures on the head, stem or both parts of the mushroom.

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_RT_Mushroom_Color

Category	Name	ID | Type Min,Max	Description
Stem	Hue	StemHue float (0.0,1.0)	Hue
	Saturation	StemSaturation float (0.0,1.0)	Saturation
	Lightness	StemLightness float (0.0,1.0)	Lightness
	Contrast	StemContrast float (-1.0,1.0)	Contrast
Bumps	Hue	BumpsHue float (0.0,1.0)	Hue
	Saturation	BumpsSaturation float (0.0,1.0)	Saturation
	Lightness	BumpsLightness float (0.0,1.0)	Lightness
	Contrast	BumpsContrast float (-1.0,1.0)	Contrast
Head	Hue	HeadHue float (0.0,1.0)	Hue
	Saturation	HeadSaturation float (0.0,1.0)	Saturation
	Lightness	HeadLightness float (0.0,1.0)	Lightness
	Contrast	HeadContrast float (-1.0,1.0)	Contrast

SFB_RT_SFX

Category	Name	ID | Type Min,Max	Description
	Water Level	SFXWaterLevel float (0.0,1.0)	How much water
	Water Details	SFXWaterDetails float (0.0,1.0)	Details of the water

Category	Name	ID | Type Min,Max	Description
SFX	Refraction	SFXRefraction float (0.0,1.0)	Refraction of the water. (Try 0 for an "ooze" look)
	Reflection	SFXReflection float (0.0,1.0)	Reflection amount
	Reflection Distance	SFXReflectionDistance float (0.0,1.0)	Reflection distance
	Flow Direction	SFXFlowDirection float (0.0,1.0)	Changes the direction the water appears to be flowing
	Water Color	SFXWaterColor Color	Color of the water (clear = default)
	Ice	SFXIce float (0.0,1.0)	How much of the water has turned to ice? Water is required for this to work.
	Ice Details	SFXIceDetails float (0.0,1.0)	Details of the ice
	Snow	SFXSnow float (0.0,1.0)	Amount of snow
	Moss	SFXMoss float (0.0,1.0)	Amount of moss
	Moss Scale	SFXMossScale int (1,4)	Scale of the moss texture
	Moss Color	SFXMossColor Color	Color of the moss

SFB_RT_DirectionalBlood

Category	Name	ID | Type Min,Max	Description
N/A	Axis	axis int (1,6)	Which direction should be blood go? X, X-inverted, Y, Y-inverted, Z, Z-inverted
Blood	Height	bloodHeight float (0.0,1.0)	How high does the blood extend
	Level	bloodLevel float (0.0,1.0)	How "thick" is the blood, works in conjunction with height
	Contrast	bloodContrast float (0.0,1.0)	The contrast of the pattern
	Color	bloodColor Color()	Color of the blood
	Roughness	bloodRoughness float (0.0,1.0)	How reflective the blood is, suggested range 0.25-0.8

SFB_RT_TextureBlend

Category	Name	ID | Type Min,Max	Description
N/A	Blend Amount	`blendAmount` `float (0.0,1.0)`	Blend amount between Texture group 1 and Texture group 2.

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]));
```

Note: Visit http://www.InfinityPBR.com for tutorials and videos showing more things you can do with our work.

5. Animations

We've included a sizable selection of animations designed specifically for this character. Read about them here. Many, if not all, can be looped in various ways inside Unity to create animation combinations.

Animation	Looped?	Description
MushAttack01	No	A "Head Smash" attack
MushAttack02	No	A Spinning head attack, could also be used for casting
MushAttack03	No	A "Jump-Forward Head Smash" attack
MushAttack04start	No	Mushroom lowers head and begins to run
MushAttack04loop	Yes	Mushroom runs forward, ready to bash into anything
MushAttack04end	No	Mushroom raises head & recovers
MushBlock01	No	Mushroom lowers head to block attacks
MushBlock02	Yes	Looped animation with the head lowered for blocking
MushBlock03	No	End of the block animation cycle
MushDeath	No	A death animation, mushroom spins to the ground, with only his head remaining
MushHit	No	Got hit reaction
MushIdle	Yes	Idle loop
MushIdleBreak	No	Break from the idle loop
MushStatic01	Yes	Static pose, as if he's not alive
MushStatic02	Yes	Static pose, as if he's not alive
MushStatic03	Yes	Static pose, as if he's not alive
MushWalk	Yes	Forward walking motion.
MushWalkBack	Yes	A slower backward walking, head slightly down with his guard up
MushSquash	No	A "bounce" as if he was jumped on or is casting a spell of sorts.

6. Level Of Details

There are multiple level of details available. Each LOD has value depending on your project and needs. Generally it's best to figure out what's the lowest resolution you can use for your project, and go with that.

LOD	Tris
SFB_Mushroom_LOD0	9.8k
SFB_Mushroom_LOD1	5.9k
SFB_Mushroom_LOD2	3.5k
SFB_Mushroom_LOD3	2.1k
SFB_Mushroom_LOD4	1.3k
SFB_Mushroom_LOD5	758

7. Change Log

v22	Added “Copy This To That” editor script along with a “Texture Customization” scene to make editing the model a little bit easier.
	Added a “Mass Exporter” script to make exporting game-ready textures and materials into a single-click operation
v21	Added downloads available for registered users at http://www.InfinityPBR.com
	Updated default Unity version to 5.3.4
	Updated main graph to include more options and load changes much faster
v19.0.1	Fixed prefab issue
	Included Position Map for use with Directional Blood and potential future Runtime Materials
	Added Run Time Materials Directional Blood & Texture Blending
	Updated folder structure in project.
v19.0	Added “Contrast” for Mushroom Stem, Head & Bumps
	Added new “Color” RunTime material for in-game changes to Hue | Saturation | Lightness | Contrast for the Head, Bumps & Stem
v18.0	Created a separate object for Runtime materials, so that in your game you can remove the base and skin materials to keep things speedy. Remember we suggest you create your textures in a new project.
	Separated Base & Skin textures into their own Materials
	Sped up texture creation time.
v15.0	Initial Version.