

TOOLS DESCRIPTION

Open "NY_Tools_UnityProj" in Unity to test Unity tools

1. GLSL Wind Flailing shader [Unity]

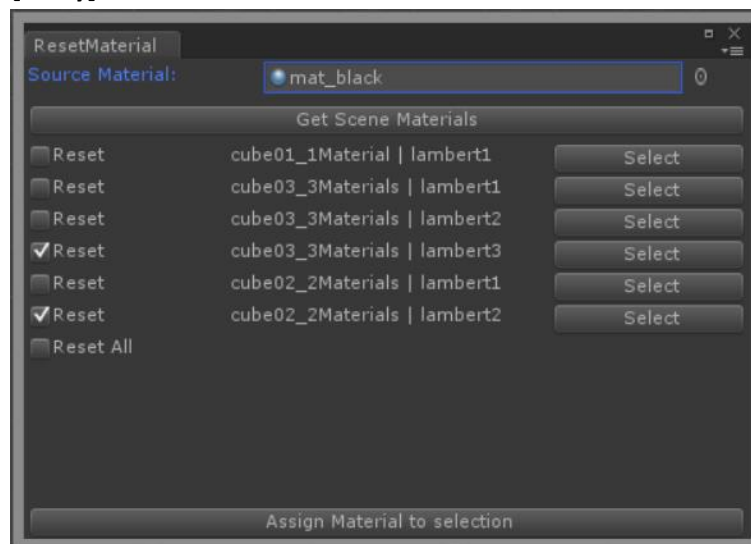


Unity shader that emulates flailing wind.

Location: \NY_Tools_UnityProj\Assets\Shaders\GLSL_WindFlailing.shader

Open **GLSL_wind scene** in the Unity Project. Play the scene to check the simulation. The shader was in GLSL for Unity as per request. If you are working on Windows platform, open Unity in OpenGL mode.

2. Reset Materials [Unity]



A Unity Editor tool that helps to change material based on user selection

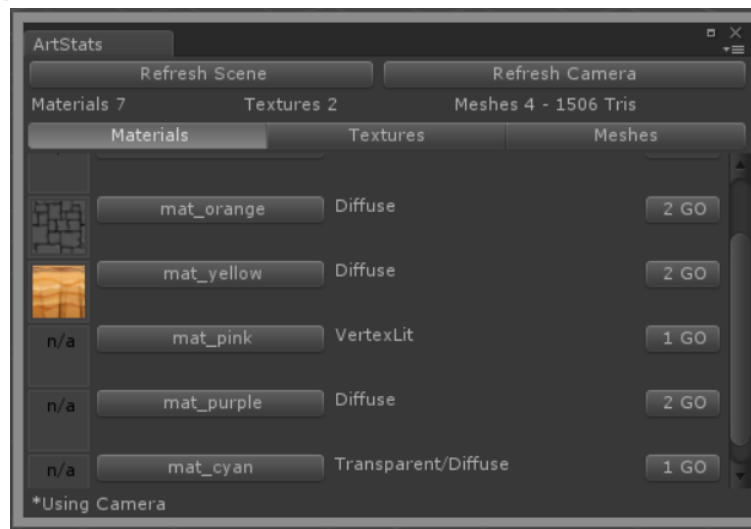
Location: \NY_Tools_UnityProj\Assets\Editor\ResetMaterial.cs

This tool can be accessed through **Menu bar** under **NY_Tools > Reset Materials**. Use Source Material to assign you own Material. Material list is generated as soon as you open the tool. Press

"Get Scene Materials" to refresh the scene. Pressing "select" will focus on the game-object that the material is associated with. You can select individual materials or "Reset All" to perform the events.

Open **ResetMaterials scene** to test this tool. The tool registers Undo state before assigning materials. So, you don't need to worry about assigning them back in-case you need. The tool handles most of the exceptions that occur during interaction. Even if you switch in-between scenes the tool might not crash.

3. Art Stats [Unity]



A Unity Editor tool that shows list of materials, textures, meshes in scene and camera's view
Location: \NY_Tools_UnityProj\Assets\Editor\ArtStats.cs

This tool can be accessed through **Menu bar** under **NY_Tools > Art Stats**. Asset lists are generated based on the total game-objects that exist in the scene and game-objects that appear in the Camera's view. A label in the bottom left "***Using Camera**" helps you to know which process you are using.

Open **Art Stats scene** to test this tool. Cycle through Materials, Textures and Meshes to view their respective lists. The tool handles most of the exceptions that occur during interaction. Even if you switch in-between scenes the tool might not crash.

There is also one more script "RenderExtensions.cs" in Scripts folder. This script is made for this tool to tell whether the game-object is in the camera's view or not.