

SubstanceTween v2.3

1/9/2017

Chris Dougherty

chris.ll.dougherty@gmail.com

chris.dougherty@students.aie.edu.au

<https://www.linkedin.com/in/archarchaic>

What does this tool do?

This tool takes exposed parameters from substance designer files(SBAR) and allows you to create up to two key frames by manipulating the exposed Variables, creating transitions and animating them.

You can Write variables to XML files and read from them as well.

When you are done creating your animated object you can save the object as a Prefab for future use.

Notes:

- This tool is meant for simple/optimized materials. if you have a giant Substance Material that is 2048-2048 or has many exposed parameters near the beginning of the input graph in Substance Designer the tool will lag. There are some materials that I have made and included in this project that are not meant to be used by this tool(Sand,Rock) unless you change them to 1024x1024 . I have included them as examples.
- Allegorithmic has a video about optimizing substances for unity:
<https://www.youtube.com/watch?v=VnhIDRcfAWA>
- The Rock material has a Boolean (true/false) called Surface Mode in the inspector. you will not be able to see the difference in some variables depending on which mode it is set to.
- Expect read XML files to break when using different versions of this tool
- Try not to delete objects or rename materials while in play mode (if you do, exit playmode and restart).
- Try not to use the tool on animated prefabs already created with this tool

- If you update your material by adding parameters or changing the order in substance designer, you will still be able to read parameters from XML files as long as the names are the same (ex: Wall_Color == Wall_Color).

- You can play animations on different versions of your material. Example:

Select Crystal1-5, create 2 transitions and press the 'Toggle Animation' button. After that select Crystal1-6 and press the 'Toggle Animation' button again. It will ask you if you want to name Crystal1-6 to 1-5, press yes and it will try to match all the parameters that have matching names (It renames the material name it creates in the script and not the actual material name that is found in the editor.)

- If your material name in the XML file is different than the name of the material in the inspector (Crystal 1.5 != Crystal 1-6) a dialogue prompt will ask you if you want to change the name of the material in the XML file when you press the 'Toggle Animation' button on a different object.

I have included multiple types of crystal materials as an example for changing and reordering parameters.

Crystal 1-6 has the most variables and is optimized, Crystal 1-5randomOrder has variables in a different order, and Crystal 1-5 missingGuass is missing all of the parameters that control the Guass spots.

Animating any materials that are 2048x2048 or larger can get choppy

Supported exposed parameters

- Float
- Vector2
- Vector3
- Vector4
- Color (RGB / RGBA)
- Integers

Most reliable method without bugs:

- Select object in Hierarchy
- Press play
- Open tool (Window>SubstanceTween)

Possible future features:

- Add object Rotation/Scale instead of only position
- Lerp between Position/Rotation/Scale
- Ability to cause the emission to flicker.
- Adding booleans(True/False): Sometimes in substance designer you will only have a parameter work if another one is turned on or at a specific value.
- Audio: instead of lerping based on time it will also be able to lerp based on audio frequency's.
- XML preview: when you save a XML file it will possibly save a prefab/preview image (like material icons in your project view)
- Mobile tester: small mobile version where you can edit materials on the go and send XML files/prefabs to a server
- Save material without creating a prefab: at the moment if all you want to do is change a couple variables without animating them or creating a prefab I would use the default sliders in the inspector view and not use this tool.
- Create other ways of animating objects other than Mathf.PingPong such as looping or play once.

Bugs/Possible Bugs:

- If you use this tool on an animated prefab in the asset folder and overwrite the prefab you are working on unity can crash.
- If you use this tool on an animated prefab in the asset folder and create a new prefab the animation can glitch.
- My material is Pink after I create Prefabs!? - manually reapply your proceduralMaterial on your object and everything should work again.
- My material is completely black/white or looks wrong - Reset Material (Right click material in inspector view> reset button).

- The tool is blank! - Select a Procedural material and then select the tool window. If that does not work restart unity or the current scene.
- Reading XML files with the sand Material can sometimes have unexpected results.
- In rare cases the Reset button on the tool will not work - try selecting a second object, then the first and try again. if that doesn't work restart the tool or exit/enter playmode again.
- If your XML file is somehow corrupt (0kb file) and you try to read it unity will give you an error saying something like "XmlException: Document element did not appear. Line 1, position 1". If you read a Xml file that is not corrupt after that without restarting the tool your material might look blurry - To fix this, as soon as you read a corrupt xml file restart the tool and read a file that is not corrupt.
- Sliders do not work with Output Parameters (parameters that start with a \$) you have to type in the number manually with the field on the right of the parameter.
- When opening the tool, the emission color gets reverted to (0,0,0)
- Sometimes The material does not recognize any parameters after a certain number when trying to reset- press the 'Save Output Parameters' checkbox and try again
- Sometimes get bugs when I resize the tool. "SUCCEDED(hr)" and
`"rc.right == m_GfxWindow->GetWidth() && rc.bottom == m_GfxWindow->GetHeight()"`
 This error does not affect the tool in any way.

Changelog:

2.3 – 1/5/17

- Removed old/useless code
- Made new functions called AddProceduralVariablesToList and SetProceduralVariablesFromList so I don't have to keep writing the same code every time I want to add or set Procedural variables to or from a list.
- Changed "Lerp Time" label/variable to "Animation time" to make it less confusing to users.
- Made it so creating a second transition would not animate automatically, you now have to create 2 transitions and then click on the "Toggle Animation on/off" button.

- Added colors to the transition and animation buttons. When you create a first or second transition the font color will turn green to indicate you have created a keyframe. When you create two transitions the "Toggle Animation on/off" button will change from red to green. When the button is green it means that you can create an animation from the two keyframes you made.
- Fixed bug where setting Animation Time to 0 or making it have no value would crash unity (It would try to divide by 0) The animation time field also supports basic operations($5*2$, $10/2$, $10\%4$) .
- When creating a debug log if two lines right next to each other are the same it will only write one of them.
- Fixed bug where creating or reading XML file it would tell you the amount of float properties is the same as the amount of total properties.
- Added a Header for the tool
- When you Create a transition or a XML file it will now add the name of the material as "PropertyMaterialName".
- Fixed bug when creating an animation and selecting another object, the new object would try to animate based on values of the last selected object. Now when you select "Toggle Animation" it will compare the name of the substance material and PropertyMaterialName(you save this when you create a transition) if the values are different it will give you a chance to rename it
- Added "Rebuild Textures Immediately" Button to Tool and Created Prefab. Having this on will Trigger an immediate rebuild of this ProceduralMaterial and will Cause FPS to drop.
- Fixed bugs when pausing the game and selecting other objects.
- Fixed bug where you couldn't create a second transition before the first.
- I created myKeys and myValues list so that I can set the parameters based on the parameter name. This allows the user to switch the order of the parameters in substance designer and bring it back in Unity to make changes without messing up the values, The PropertyName, PropertyFloat, PropertyColor, PropertyVector lists will now only be used for keeping track of the different types and amounts of variables. At the moment this does not work with variables that have been renamed, you will have to manually rename the variables in the XML file.
- Added fps counter near sun controller (worldGUI script)
(<http://wiki.unity3d.com/index.php?title=FramesPerSecond>)
- Fixed bug where I was accidently updating Procedural Material twice per frame.
- Added buttons to select the transitions you have created as well as a button to delete them.

- When you create a prefab it saves the animation time.
- Added Feature/ Fixed bug: When Writing xml files it would add parameter types with no range (\$outputSize, \$pixelSize, \$RandomSeed) this could slow down your animation and make it choppy (loading a 4096x4096 animation on a material that is meant to be 1024x1024) Saving these parameters could be useful so I made a toggle button called 'Save Output Parameters'

2.2

- Added a button for debug information (lists changes for materials, saves information about written/read Xml files and created prefabs).
- Fixed bug where the created prefab would not have any saved values unless you selected it before exiting play mode.
- Fixed bug when resetting vectors or trying to set vectors to the minimum/maximum would do nothing.
- Fixed bug where if you select objects with no renderer or procedural material in a specific order it would crash the tool.
- Fixed bug where if you selected objects, restarted the tool and created a prefab it would make the material pink (Unity material error).
- Made it so you cannot open the tool from Window>SubstanceTween if you have nothing selected or if the object does not have the required component(Renderer/ProceduralMaterial).

2.1

- Remade XML read/write (Now using save box instead of manually typing in file path)
- Remade transition/reset buttons
- Remade Prefab button (Now with save box and you are able to name prefabs)
- Remade Sun controls from the old version of the old tool to a separate script on the directional light (Not part of tool yet)
- Recreated _EmissionColor and _MainTex

2.0

- Added Position slider
- Remade tool using editor scripting instead of using old OnGUI code (Lacks any buttons including XML read/write, resetting to default or saving object as prefab)
- cant edit `_EmissionColor` or `_MainTex`
- Color picker instead of sliders
- Converted all old Arraylists into Lists

1.0 - 1.9

- Old version of tool that was written with Unity 4 OnGUI code.
- In these versions you had to manually type the path for XML files instead of using a save/open box.
- Used old Arraylists