PxrSeExpr

Because it is sometimes useful for an artist to be able to experiment with a procedural materials and patterns, rather than relying on painting textures or built-in nodes, PRMan includes this SeExpr node. Based on Disney Animations's open source SeExpr system, this node is a scriptable pattern generator and combiner. It is unique among the factory plugins in its "dynamic binding" capabilities. Pattern network authors can wire upstream nodes into arbitrary (named) connection points and these can be referred to in the SeExpr script.

See the Quick Reference for more details.

Note that the data in the `RixShadingContext` are available directly in scripts. Currently that includes: P, Pobj, Vn, VLen, Nn, Ngn, Tn, s, t, u, v, w, curvature, curvature_u, curvature_v, objectId; if an input is given one of these names, it will be ignored in favor of the `RixShadingContext` data.

PxrSeExpr does not support rand()

Input Parameters

Expression

The SeExpr script used to generate the pattern. Additional input connections are "late-bound" to SeExpr expression variables.

Output Parameters

resultRGB

The three channel output if the result of the expression script is a vector.

resultR

The first channel from the resultRGB output.

resultG

The second channel from the resultRGB output.

resultB

The third channel from the resultRGB output.

resultF

If the result of the expression script is a scalar, it will be returned in this output parameter.

Examples

An SeExpr script that mixes two color textures might be written in a RIB file as:

```
Pattern "PxrTexture" "tex1" "string filename" ["checker.tx"]
Pattern "PxrTexture" "tex2" "string filename" ["ratGrid.tex"]
Pattern "PxrSeExpr" "mixer"
        "float mix" [0.5]
        "reference color c1" ["tex1:result"]
        "reference color c2" ["tex2:result"]
        "string expression"
["\
c = c1*mix + c2*(1-mix); \
c\
"]
```

Note that the last line of the script contains just the name of the variable that is the desired return value and does not end in a semi-colon. Also note that the name of the returned variable is not used in the connections to other nodes in the pattern network, and instead, the name used for the output connection is always one of *Output Parameters* mentioned above which are chosen depending on the type required.

There is only one formal input to the PxrSeExpr node, the string for the text of the expression. All other inputs are dynamic: the renderer assumes these inputs are to be used as variables in the script. Non-referenced parameters such as "mix" above will resolve to the value provided in the RI binding (here 0.5). Referenced parameters will trigger an evaluation of the pattern graph (here triggering the execution of two upstream texture nodes).

SeExpr could also be used to generate a fleck-like normal perturbation, which could be attached to the *Bump Normal* parameter of PxrSurface to easily create a metallic fleck paint.

```
res = cvoronoi(P*floatInput1) * 2 - 1;
res *= floatInput2;
res += Nn;
res = norm(res);
res
```

Note that the above example was for the RenderMan for Maya node and there the inputs have already been named *floatInput1* and so on which would appear as *Float Input 1* in the Node Editor window. In this example the output name to used in the Node Editor would be *resultRGB*. In this case the output is not a color but the RGB output is used for any connection that needs three floats.

Debugging

To debug an expression, we can use printf to print the value to stdout.

```
res = cellnoise(P);
msg = printf("My result = %f %f %f",res[0],res[1],res[2]);
res
```

See Also

- SeExpr QuickRef (Mirrored from the WDAS Language Documentation)
- Source code at GitHub
- SeExpr API Documentation