You can do one or both of the following for this assignment:

1. Analyze the characteristics of RenderMan's one- and two-dimensional noise() functions, and/or
2. Write a shader of your own that seamlessly combines 2D procedural noise with a texture-based noise

OPTION 1: Analyzing RenderMan's noise() functions

Using RenderMan itself as much as possible in your analysis, address the following:

- If you look at a reasonably large domain of SL's one- and two-dimensional noise() calls, what are the average noise values in each case?
- What are the minimum noise values?
- What are the maximum noise values?
- What are the standard deviations[1] of both noise functions?
- Plot a histogram of noise values for each of the noise functions (it would be extra cool if you did this with a RenderMan shader)

OPTION 2: Blending procedural with texture-based patterns

Write a shader that creates a properly antialised noise pattern on a receding plane. Adhere to the following guidelines:

- The procedural component of your shader should use frequency clamping. The red channel of your final image should show the frequency claming taking effect (i.e., the noise values fading out as the sampling frequency gets too high).
- The texture-based component of your shader should use a noise texture created however you see fit. Don't forget that the texture-based noise needs to blend as seamlessly as possible with the procedural noise. This texture should be applied where your procedural texture cannot work due to the Nyquist limit. The green channel of your final image should show the texture-based noise being faded on.
- The blue channel of your final image should show the patterns blended together over the transition region. The region should be impossible to identify!

Some things you might find useful for both of these options:

1. The "near clipping plane" RIB file. This RIB sticks a polygon right at the clipping plane that completely fills the viewport. It is desribed in the "RenderMan Tricks Everyone Should Know" section of the SIGGRAPH 99 course notes (in the handouts folder).

2. Part II of the RenderMan Spec: The RenderMan Shading Language (in its entirety, honestly)

3. The txmake program, which turns TIFF images into RenderMan-readable texture maps. Use it like this to turn inputfile.tif into outputfile_t with "periodic" mode enabled (it means if you access it beyond 0-1 it will cycle):

   ```
   txmake -mode periodic inputfile.tif outputfile_t
   ```

4. The texture() SL call. Read about it in the Spec (see #3, above)

5. The printf() SL call.

6. The mix() SL call.

Assignment 3 - Due Tuesday March 14

[1] To compute the standard deviation, subtract each value from the mean value, square the result, and sum up all the squared values. Then, divide that total by the total number of samples taken.