Download tiff file here: <https://github.com/kayikunmi/COSC-321/blob/main/Final/file.tiff>

Follow the order of the instructions to compile correctly. Compile Instructions:

* export RMANTREE=/Applications/Pixar/RenderManProServer-24.4
* export PATH=$RMANTREE/bin:$PATH
* javac -cp .:jama-1.0.3.jar \*.java
* oslc KNoise.osl
* oslc Ksea.osl
* oslc KFish.osl
* javac Final.java
* java Final
* prman java.rib

*Final.java* uses the RenderMan Interface to create a 3D underwater scene with multiple objects such as rocks, fish, sea plants, and stars with various objects and lighting. The scene consists of several parts, each creating a different object with unique properties, surfaces, textures and volumes. First, the program sets the output file format and projection using the *RiFormat* and *RiProjection* functions, respectively. The camera view is translated and set to look at the center of the scene using the *RiTranslate* function. The scene is then wrapped in the RiWorldBegin and RiWorldEnd functions.

The moon is created using a *BSpline surfac*e *(KBSpline().KPatch(testBB))* with moonlight added using a distant light source *(PxrDistantLight)*. This is accomplished by setting the color of the light and assigning it to a shader using the *RiLight, RiPattern, RiBxdf, and RiAttributeBegin* functions. The light from the moon also adds shadows to the objects in the scene. The moon also has noise *(KNoise.osl)* on it that is randomly generated.

The sea is created using a blue color with a bump map *(KBump.osl)* using the RiPattern, RiBxdf, and RiAttributeBegin functions. The bump map gives the surface the appearance of waves. The rocks *(KObject().rock())* are created using a gray color with a clear coat roughness *(float clearcoatRoughness)* effect added to them, and they are also transparent *(float presence)*. The *RiBxdf* and *RiTranslate* functions are used to create and position the rocks.

Next, three instances of sea plants are created using an *L-System algorithm (KLsysExample().plant())*. The plants are created using the *RiTransformBegin, RiTranslate, RiScale,* and *RiTransformEnd* functions. Several fish are also created with stripes using a stripe pattern *(KFish.osl)*. This is accomplished using the *RiPattern, RiBxdf,* and *RiTransformBegin* functions. The fish *(KObject().fish())* are created using *RiSphere* and *RiCone*.

Finally, stars and clouds are added to the scene using the *RiAttributeBegin* function. The stars are created using a point cloud *(KObject().stars())* and are positioned randomly in the sky. The clouds *(KObject().clouds())* are created using a white color and are positioned randomly over the sea.