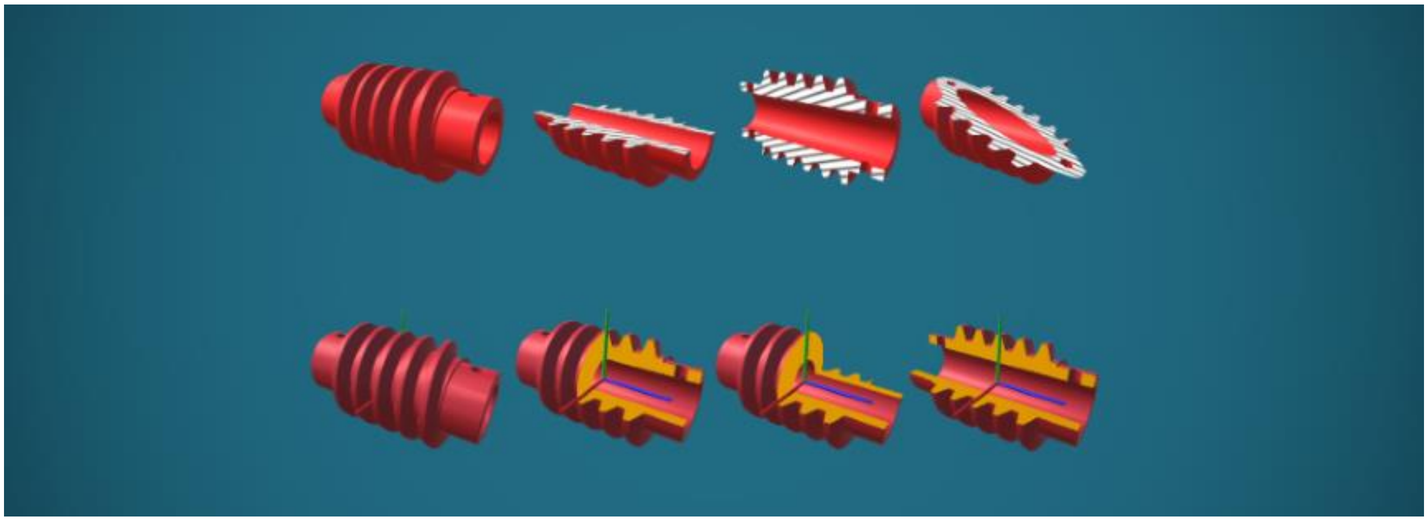
Welcome to the Unity3DCrossSectionShader wiki!

This project include different CG shaders that can be used to generate a cross section through the mesh

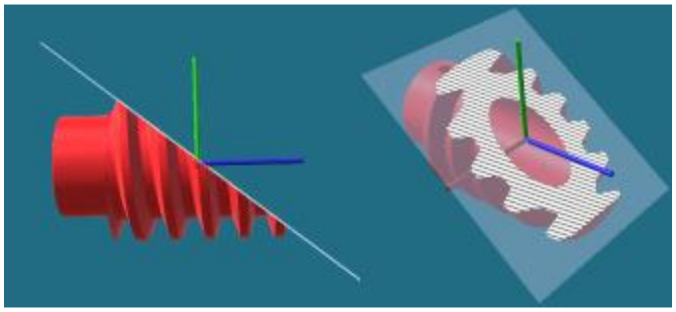


Getting Started

Download the unity package from unity Asset Store (the link will be available soon).

How to Use:

- ***OnePlaneBSP Shader:***
  This shader divide the space into two partitions, the one under the plane will be rendered, and the partition above the plane will be discarded.

  The plane represented in the shader using:

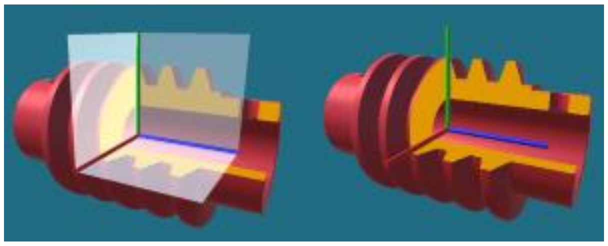
  i. **Plane Position** in World Space
  ii. **Plane Normal** which control the orientation of the plane

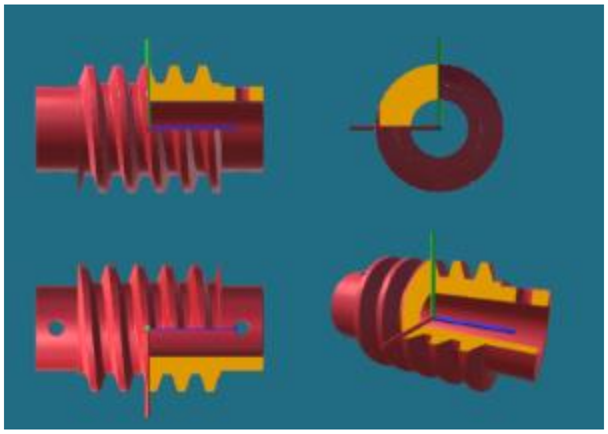
  To add a ***Hatching*** effect:

- Create a plane/Quad with the StencilledUnlitTexture material and assign the hatching pattern texture to it.
- Make sure the Stencil mask value is the same value that is in the OnePlaneBSP (by default it is 255).

- Translate the plane to the same position that is assigned to the OnePlaneBSP shader material, and make its orientation consistent with the normal of the OnePlaneBSP shader material.

  You can easily connect the Plane/Quad transform with the OnePlaneBSP material by adding an `OnePlaneCuttingController` script to the object that will be cut out, and assign the plane/quad object to it, the script will extract the position and normal from the plane and send them to the shader.





- ***ThreeAAPlanesBSP Shader***
  This shader uses three planes to divide the space. All fragments above the three planes will be discarded other fragments will be rendered. The three planes are oriented with the Y-Z, X-Z and X-Y planes; you can control their positions only.
  i. Plane #1 represent Y-Z plane.

  ii. Plane #2 represent X-Z plane.
  iii. Plane #2 represent X-Y plane.

  This shader doesn't support hatching.

- ***GenericThreePlanesBSP Shader:***
  This is the most generic shader in which you can control the planes position and orientation. It does not support hatching too.
  You can use `GenericThreePlanesCuttingController`script with the same way that I mentioned before for the `OnePlaneCuttingController` to extract the positions and normals from existing Quad/Plane